**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**--------------**



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

**TÊN ĐỀ TÀI:**

**QUẢN LÍ CỬA HÀNG BÁN SÁCH**

**GVHD:** TS. Nguyễn Thành Sơn

**Lớp HP:** DBMS330284\_22\_2\_01CLC

**Nhóm thực hiện:** 05

**Học kỳ:** 2

**Năm học:** 2022-2023

*Tp Thủ Đức, tháng 2 năm 2023*

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 2 NĂM HỌC 2022-2023**

**Nhóm 05**

***Đề tài: Quản lí cửa hàng bán sách***

|  |  |
| --- | --- |
| **MSSV** | **Họ và tên** |
| 21110821 | Lê Đức Duy |
| 21110840 | Phạm Tuấn Minh |
| 21110270 | Đào Duy Phát |
| 21110349 | Lê Nguyễn Thiên Tứ |

**NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………….………………………………………………………………………………………………………………………………………………………………………….………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Tp Thủ Đức, tháng 5 năm 2023

GV chấm điểm

**MỤC LỤC**

[**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG** 7](#_Toc134646405)

[**1.** **Đặc tả đề tài** 7](#_Toc134646406)

[**1.1** **Mô tả bài toán** 7](#_Toc134646407)

[**1.2** **Nghiệp vụ của bài toán** 8](#_Toc134646408)

[**1.3** **Phương pháp tính chi phí** 10](#_Toc134646409)

[**1.4** **Các chức năng của bài toán** 11](#_Toc134646410)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 14](#_Toc134646411)

[**1.** **Thiết kế cơ sở dữ liệu mức quan niệm** 14](#_Toc134646412)

[**2.** **Thiết kế cơ sở dữ liệu mức logic** 14](#_Toc134646413)

[**3.** **Các ràng buộc cần có** 15](#_Toc134646414)

[**4.** **Cài đặt CSDL và các ràng buộc** 17](#_Toc134646415)

[**5.** **Các Trigger** 22](#_Toc134646416)

[**6.** **Các View** 26](#_Toc134646417)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG** 35](#_Toc134646418)

[**1.** **Kết nối cơ sở dữ liệu** 35](#_Toc134646419)

[**2.** **Danh mục xem thông tin** 35](#_Toc134646420)

[**3.** **Danh mục kiểm tra** 45](#_Toc134646421)

[**4.** **Danh mục cập nhật** 47](#_Toc134646422)

[**5.** **Danh mục hiển thị** 58](#_Toc134646423)

[**6.** **Danh mục tìm kiếm** 65](#_Toc134646424)

[**7.** **Danh mục phân loại** 70](#_Toc134646425)

[**8.** **Danh mục thêm** 75](#_Toc134646426)

[**CHƯƠNG 4: THIẾT KẾ GIAO DIỆN** 85](#_Toc134646427)

[**Môi trường cài đặt và công nghệ sử dụng** 85](#_Toc134646428)

[**Giao diện và chức năng của chương trình:** 85](#_Toc134646429)

[**1.** **Form đăng nhập** 85](#_Toc134646430)

[**2.** **Form quên mật khẩu** 86](#_Toc134646431)

[**3.** **Form quản lí chung** 89](#_Toc134646432)

[**4.** **Form quản lí nhân viên** 108](#_Toc134646433)

[**5.** **Form báo cáo thống kê** 115](#_Toc134646434)

[**6.** **Form thông tin** 118](#_Toc134646435)

[**7.** **Form trợ giúp** 120](#_Toc134646436)

[**CHƯƠNG 5: PHÂN QUYỀN** 121](#_Toc134646437)

[**1.** **Nhân viên tư vấn** 121](#_Toc134646438)

[**2.** **Nhân viên thu ngân** 122](#_Toc134646439)

[**3.** **Nhân viên thủ kho** 123](#_Toc134646440)

[**4.** **Tạo login** 125](#_Toc134646441)

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin chân thành cảm ơn sự hướng dẫn tận tình của thầy ***Nguyễn Thành Sơn*** đã giúp chúng em hoàn thành bài báo cáo cuối kì bộ môn Hệ quản trị cơ sở dữ liệu này. Nhờ những kiến thức thầy truyền đạt mà chúng em đã nắm chắc hơn được những kiến thức căn bản, nền tảng cho việc khi xây dựng một hệ thống quản trị cơ sở dữ liệu. Thông qua đồ án này, nhóm em xin trình bày lại quá trình phát triển của một hệ thống quản trị cơ sở dữ liệu, cũng như lập trình một đồ án liên quan.

Trong quá trình thực hiện đồ án môn học, khó tránh khỏi những sai sót. Do đó nhóm chúng em rất mong sự góp ý đến từ thầy để bài làm của nhóm chúng em được hoàn thiện hơn, phát triển và đầy đủ hơn. Kính chúc thầy sức khỏe, hạnh phúc thành công trên con đường sự nghiệp giảng dạy.

Cuối cùng chúng em cũng chân thành cảm ơn các thầy cô và các bạn cùng tham gia học bộ môn Hệ quản trị sơ sở dữ liệu đã nhiệt tình giúp đỡ động viên chúng em trong trình thực hiện đồ án cuối kì.

**LỜI NÓI ĐẦU**

Trong nhiều năm trở lại đây, ngành công nghệ thông tin thực sự đã len lỏi vào từng ngóc ngách của đời sống xã hội. Bất kể một lĩnh vực hay ngành nghề nào cũng đều ứng dụng các công nghệ thông tin. CNTT đã và đang đóng vai trò quan trọng trong đời sống kinh tế, xã hội của nhiều quốc gia trên thế giới và Việt Nam cũng vậy, ứng dụng của ngành này có rất nhiều trong đời sống hằng ngày.

Không thể phủ nhận rằng các hệ thống quản lý dữ liệu, thông tin dần trở thành công cụ đắc lực của nhiều doanh nghiệp. Không chỉ ở các công ty lớn, các những nơi có quy mô lớn nhưng hiện nay trong công tác quản lý dữ liệu, công nghệ thông tin giúp cho quá trình quản lý dữ liệu đạt hiệu quả cao trong việc giảm tối thiểu về nhân lực, thời gian cũng như tránh thất lạc, sai sót dữ liệu thay vì việc quản lý bằng giấy tờ bằng phương pháp cổ truyền như trước đây. Chính vì những lý do đó, ứng dụng của Công nghệ thông tin đối với công tác quản lý dữ liệu trong những nơi có kinh doanh hay cần quản lí những dữ liệu hằng ngày. Và để làm rõ hơn về ứng dụng của Công nghệ thông tin đối với công tác quản lý trong thực tế nhóm chúng em lựa chọn đề tài ***“Quản lý cửa hàng bán sách”*** để nghiên cứu và phát triển ứng dụng của việc quản lý hệ thống dữ liệu trong môn học Hệ quản trị cơ sở dữ liệu.

# **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

1. **Đặc tả đề tài**
   1. **Mô tả bài toán**

Cửa hàng bán sách mà em đã tìm hiểu đến là một cửa hàng bán sách có quy mô tương đối lớn với nhiều loại sách khác nhau. Hàng ngày cửa hàng đón nhận nhiều khách hàng đến mua sách.

Thông qua quá trình tìm hiểu các khâu quản lý mua bán sách, thống kê, quản lý khách hàng, nhà cung cấp, nhân viên… Em đã nhận thấy tại cửa hàng còn nhiều tồn đọng cần được giải quyết.

Em đã nhận thấy sau mỗi lần giao dịch (mua bán sách) thì công tác quản lý, kinh doanh tốn rất nhiều thời gian, công sức, giấy tờ, sổ sách.

• Là một cửa hàng với nhiều loại sách khác nhau vì vậy việc quản lý nhân viên, khách hàng mua bán, thanh toán không tránh khỏi những sai sót, nhầm lẫn.

• Mọi quá trình mua bán sách, nhập sách từ nhà cung cấp đều bằng giấy viết tay và lưu trữ bằng giấy mất nhiều thời gian và công sức rất dễ xảy ra nhầm lẫn, mất mát.

• Công việc quản lý nhân viên từ lý lịch, công việc, đặc biệt là mối quan hệ với nhà cung cấp, với khách hàng 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 mình các nhân viên lại phải tìm tòi, lùng sục từng hoá đơn, sổ sách mất nhiều thời gian, vì vậy các nhà quản lý khó nắm bắt được hoạt động kinh doanh thường xuyên của cửa hàng mình dẫn đến sự chậm trễ trong việc đưa ra chiến lược kinh doanh.

Xuất phát từ thực tế đó, yêu cầu đặt ra đối với hệ thống quản lý cửa hàng bán sách là giải quyết và khắc phục những khó khăn trong nghiệp vụ với mục đích tiết kiệm được nhân lực, thời gian, tài chính cho cửa hàng để cho cửa hàng hoạt động ngày càng hiệu quả hơn.

Một cửa hàng có hệ thống quản lý mua bán sách thì cần lưu trữ những dữ liệu như sau:

Trong cửa hàng sẽ có kho sách. Kho sách thì có nhiều mã sách, tên sách, số lượng, giá của cuốn sách và một vài mô tả về cuốn sách đó.

Thông tin sản phẩm sách cần có như Tên sách, tác giả, NXB, số lượng, giá bán, danh mục thể loại, vị trí trong cửa hàng.

Khách hàng thì sẽ có mã của vị khách mua bán sách, họ tên của khách hàng, địa chỉ của khách hàng và số điện thoại của họ.

Chúng ta cũng cần có nhà cung cấp gồm có mã nhà của nhà cung cấp, tên của nhà cung cấp đó, địa chỉ nhà cung cấp đó là ở đâu cũng như là số điện thoại.

Trong cửa hàng bán sách thì sẽ có nhiều nhân viên. Mỗi nhân viên thì gồm có họ tên nhân viên, mã số của nhân viên, ngày sinh, giới tính của nhân viên đó, địa chỉ nơi ở cũng như là số điện thoại.

Để dễ dàng thì họ cũng cần có tài khoản để nhân viên đăng nhập vào hệ thống thông tin của cửa hàng.

Chúng ta cũng cần thêm hóa đơn. Chúng ta cần có số hóa đơn, ngày, mã sách của quyển sách đó. Đối với hóa đơn nhập chúng ta cần có ngày nhập, mã của nhà cung cấp tên của nhà cung cấp cũng như địa chỉ và số điện thoại của họ. Còn về hóa đơn bán thì cần có họ tên, số điện thoại, địa chỉ và mã của khách hàng đó.

Dữ liệu thống kê: Số lượng sách đã bán, doanh thu theo ngày, tháng, năm, số lượng khách hàng, đánh giá của khách hàng về sản phẩm và dịch vụ.

Ngoài ra cần có thêm phiếu thu như mã của phiếu thu, mã của khách hàng, mã của nhà cung cấp và lý do của phiếu thu.

* 1. **Nghiệp vụ của bài toán**

Cửa hàng bán sách em khảo sát là một cửa hàng tương đối lớn với nhiều thể loại sách khác nhau, quá trình hoạt động của cửa hàng luôn phải tuân theo một nghiệp vụ rõ ràng.

Quản lí nhập xuất sách:

* Quản lí thông tin về các đầu sách trong kho, bao gồm tên sách, tác giả, nhà xuất bản, giá bán và số lượng tồn kho.
* Cập nhật thông tin sách khi nhập hàng mới hoặc bán hàng.
* Tính toán số lượng sách trong kho và đưa ra quyết định về việc đặt hàng mới để đáp ứng nhu cầu của khách hàng.
* Quản lí việc nhập và xuất sách, bao gồm nhận và kiểm tra hàng hóa, lưu trữ thông tin về hóa đơn nhập và hóa đơn xuất, và cập nhật thông tin về số lượng sách trong kho.

Quản lí khách hàng:

* Lưu trữ thông tin khách hàng, bao gồm tên, địa chỉ, số điện thoại và lịch sử mua hàng.
* Xác định các khách hàng thân thiết để có thể tặng cho họ các ưu đãi đặc biệt hoặc giảm giá.
* Tạo ra các chương trình khuyến mãi và quảng cáo để thu hút khách hàng mới và duy trì khách hàng hiện tại.

Quản lí hóa đơn:

* Tạo và lưu trữ thông tin về hóa đơn bán hàng, bao gồm tên khách hàng, ngày bán, danh sách sách và tổng số tiền thanh toán.
* Tạo ra các báo cáo thống kê về doanh thu, lợi nhuận và các mặt hàng bán chạy.

Quản lí nhân viên:

* Bao gồm có 5 loại nhân viên theo hệ số lương giảm dần: Quản lí, Thu ngân, Thủ kho, Tư vấn viên, Bảo vệ và lao công.
* Lưu trữ thông tin nhân viên, bao gồm tên, địa chỉ, số điện thoại và lịch sử công việc.
* Quản lí lịch làm việc, tính lương và các khoản phụ cấp cho nhân viên.
* Phân quyền truy cập cho các nhân viên để đảm bảo an toàn dữ liệu.

Quản lí kho:

* Giám sát số lượng sách trong kho và thông báo khi cần đặt hàng mới.
* Cập nhật thông tin tồn kho sau mỗi lần nhập hoặc bán hàng.
* Đảm bảo an toàn và bảo quản sách trong kho.

Ngoài ra, các nghiệp vụ khác trong việc quản lí cửa hàng bán sách cũng có thể bao gồm quản lí chi phí, quản lí doanh số, quản lí thuế, quản lí tài chính, quản lí trang thiết bị và vật tư. Tùy thuộc vào yêu cầu cụ thể của cửa hàng, các nghiệp vụ này có.

* 1. **Phương pháp tính chi phí**
* Phương pháp chi phí thực tế: Phương pháp này dựa trên việc tính toán chi phí thực tế mà cửa hàng phải chi trả để sản xuất và kinh doanh các sản phẩm sách. Chi phí thực tế bao gồm chi phí hàng hóa (mua sách, bìa sách, giấy in,…) và chi phí hoạt động (tiền thuê cửa hàng, tiền điện, tiền nước, tiền lương, tiền quảng cáo, …).

Tổng chi phí = Chi phí hàng hóa + Chi phí hoạt động

* Phương pháp chi phí trực tiếp: Phương pháp này tính toán các chi phí trực tiếp được chi tiêu cho từng sản phẩm sách, bao gồm chi phí hàng hóa, chi phí lao động, chi phí vận chuyển, … Phương pháp này thường được sử dụng để tính toán chi phí sản xuất các sản phẩm sách riêng lẻ.

Chi phí trực tiếp = Chi phí hàng hóa + Chi phí lao động + Chi phí vận chuyển + ...

* Phương pháp chi phí gián tiếp: Phương pháp này tính toán các chi phí gián tiếp không thể liên kết trực tiếp với sản phẩm sách, nhưng vẫn ảnh hưởng đến hoạt động kinh doanh của cửa hàng, bao gồm chi phí nhân viên quản lý, chi phí tiền thuê cửa hàng, chi phí văn phòng phẩm, chi phí quản lý tài chính, …

Chi phí gián tiếp = Tổng chi phí - Chi phí trực tiếp

Lương tháng nhân viên= Lương theo chức vụ + Phụ cấp ( nếu có ) / ngày công chuẩn của tháng X số ngày làm việc thực tế

* Phương pháp chi phí trung bình: Phương pháp này tính toán chi phí trung bình cho mỗi sản phẩm sách dựa trên tổng chi phí của cửa hàng chia cho số lượng sản phẩm sách đã sản xuất hoặc bán được trong một khoảng thời gian nhất định.

Chi phí trung bình = Tổng chi phí / Số lượng sản phẩm sách đã sản xuất hoặc bán được

Các phương pháp tính toán chi phí trên đây có thể được áp dụng tùy thuộc vào quy mô và mô hình kinh doanh của cửa hàng bán sách. Việc tính toán chi phí chính xác và kịp thời giúp cửa hàng có được cái nhìn toàn diện về tình hình tài chính của mình, từ đó có các quyết định kinh doanh phù hợp để tối ưu hóa lợi nhuận và phát triển bền vững.

* 1. **Các chức năng của bài toán**

*Quản lí hàng hóa:*

* Lưu trữ thông tin về sách trong kho, bao gồm tên sách, tác giả, nhà xuất bản, giá bán và số lượng tồn kho.
* Cập nhật thông tin sách khi nhập hàng mới hoặc bán hàng.
* Giám sát số lượng sách trong kho và thông báo khi cần đặt hàng mới.

*Quản lí khách hàng:*

* Lưu trữ thông tin khách hàng, bao gồm tên, địa chỉ, số điện thoại và lịch sử mua hàng.
* Xác định các khách hàng thân thiết để có thể tặng cho họ các ưu đãi đặc biệt hoặc giảm giá.

*Quản lí hóa đơn:*

* Tạo và lưu trữ thông tin về hóa đơn bán hàng, bao gồm tên khách hàng, ngày bán, danh sách sách và tổng số tiền thanh toán.
* Tạo ra các báo cáo thống kê về doanh thu, lợi nhuận và các mặt hàng bán chạy.

*Quản lí nhân viên:*

* Lưu trữ thông tin nhân viên, bao gồm tên, địa chỉ, số điện thoại và lịch sử công việc.
* Phân quyền truy cập cho các nhân viên để đảm bảo an toàn dữ liệu.

*Quản lí kho:*

* Giám sát số lượng sách trong kho và thông báo khi cần đặt hàng mới.
* Cập nhật thông tin tồn kho sau mỗi lần nhập hoặc bán hàng.

*Đối với người quản trị:*

Thêm quyền, xóa quyền, phân quyền cho nhóm người dùng, xóa quyền khỏi nhóm người dùng.

Thêm phân nhóm, xóa phân nhóm, Thêm nhân viên vào phân nhóm, xóa nhân viên khỏi phân nhóm.

Quản lý đặt hàng, mua hàng, trả hàng, chi trả tiền mua hàng, tiền cọc, công nợ từ nhà cung cấp.

*Đối với nhân viên:*

Thêm khách hàng, xóa khách hàng, chỉnh sửa thông tin khách hàng và tra cứu thông tin của khách hàng.

Tra cứu thông tin của sách, số lượng của sách còn trong kho.

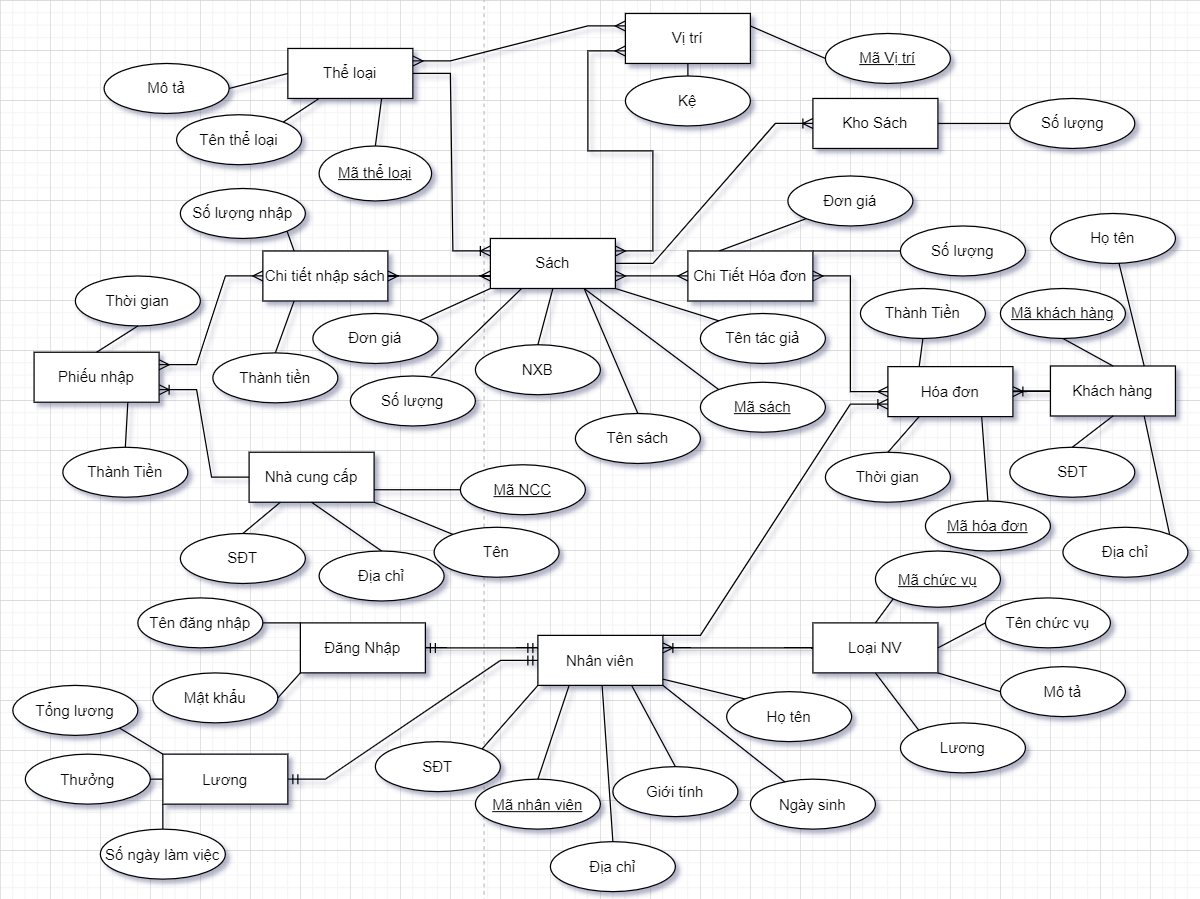
Phân quyền:

Có khả năng phân quyền đến từng người sử dụng theo chức năng. Mỗi vị trí được phân công có nhiệm vụ và quyền hạn được phân định rõ ràng, đảm bảo người không có trách nhiệm không thể truy cập vào công việc của người khác. Có thể thêm hoặc xóa tài khoản mới mà không làm ảnh hưởng đến hệ thống.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. **Thiết kế cơ sở dữ liệu mức quan niệm**

Từ mô tả về dữ liệu cần có ở phần mô tả bài toán, hình thành được sơ đồ thực thể kết hợp (ERD).



1. **Thiết kế cơ sở dữ liệu mức logic**

**a) Các quan hệ cơ sở**

● TheLoai (MaTheLoai, TenTheLoai, MoTa)

● Sach (MaSach, TenSach, TacGia, NXB, MaTheLoai, SoLuongTrenKe, DonGia)

● KhoSach (MaSach, SoLuong)

● ViTri (MaViTri, MaTheLoai, Ke)

● KhachHang (MaKhachHang, HoTen, DiaChi, SoDienThoai)● PhanLoaiNhanVien (MaChucVu, TenChucVu, MoTaCongViec, Luong)

● NhanVien (MaNhanVien, HoTen, NgaySinh, GioiTinh, MaChucVu, DiaChi, SoDienThoai)

● DangNhap (MaNhanVien, TenDangNhap, MatKhau)

● Luong (MaNhanVien, SoNgayLamViec, Thuong)● HoaDon (MaHoaDon, MaNhanVien, MaKhachHang, ThoiGian)● NhaCungCap (MaNCC, TenNCC, DiaChi, SoDienThoai)● PhieuNhap (MaPhieuNhap, MaNCC, NgayNhap)

**b) Các quan hệ liên kết**

● ViTriTheLoai (MaViTri, MaTheLoai)

● ViTriSach (MaViTri, MaSach, Tang, Ngan)

● ChiTietHoaDon (MaHoaDon, MaSach, SoLuong, DonGia)● ChiTietNhapSach (MaPhieuNhap, MaSach, SoLuongNhap, DonGia)

1. **Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **TABLE** | **RÀNG BUỘC** |
| **1** | SACH | Ràng buộc khóa chính *MaSach*  Ràng buộc khóa ngoại *MaTheLoai* tham chiếu đến *THELOAI* |
| **2** | THELOAI | Ràng buộc khóa chính *MaTheLoai* |
| **3** | KHOSACH | Ràng buộc khóa ngoại *MaSach* tham chiếu đến **SACH** |
| **4** | VITRI | Ràng buộc khóa chính *MaViTri* |
| **5** | VITRI\_THELOAI | Ràng buộc khóa ngoại *MaViTri*  tham chiếu đến VITRI  Ràng buộc khóa ngoại *MaTheLoai* tham chiếu đến **THELOAI** |
| **6** | VITRI\_SACH | Ràng buộc khóa ngoại *MaViTri* tham chiếu đến **VITRI**  Ràng buộc khóa ngoại *MaSach* tham chiếu đến **SACH** |
| **7** | KHACHHANG | Ràng buộc khóa chính *MaKhachHang* |
| **8** | NHANVIEN | Ràng buộc khóa chính *MaNhanVien*  Ràng buộc khóa ngoại *MaChucVu* tham chiếu đến **PHANLOAINHANVIEN** |
| **9** | PHANLOAINHANVIEN | Ràng buộc khóa chính *MaChucVu* |
| **10** | LUONG | Ràng buộc khóa ngoại *MaNhanVien* tham chiếu đến **NHANVIEN** |
| **11** | HOADON | Ràng buộc khóa chính *MaHoaDon*  Ràng buộc khóa ngoại *MaNhanVien* tham chiếu đến **NHANVIEN**  Ràng buộc khóa ngoại *MaKhachHang* tham chiếu đến **KHACHHANG** |
| **12** | CHITIETHOADON | Ràng buộc khóa ngoại *MaHoaDon* tham chiếu đến **HOADON**  Ràng buộc khóa ngoại *MaSach* tham chiếu đến **SACH** |
| **13** | NHACUNGCAP | Ràng buộc khóa chính *MaNCC* |
| **14** | PHIEUNHAP | Ràng buộc khóa chính *MaPhieuNhap*  Ràng buộc khóa ngoại *MaNCC* tham chiếu đến **NHACUNGCAP** |
| **15** | CHITIETNHAPSACH | Ràng buộc khóa ngoại *MaPhieuNhap* tham chiếu đến **PHIEUNHAP**  Ràng buộc khóa ngoại *MaSach* tham chiếu đến **SACH** |
| **16** | DANG\_NHAP | Ràng buộc khoá ngoại *MaNhanVien* tham chiếu đến **NHANVIEN** |

1. **Cài đặt CSDL và các ràng buộc**

**Bảng thể loại**

-- Bảng Thể loại

CREATE TABLE TheLoai (

MaTheLoai NVARCHAR(10) PRIMARY KEY,

TenTheLoai NVARCHAR(100) NOT NULL,

MoTa NVARCHAR(255) NOT NULL

)

go

**Bảng nhà cung cấp**

-- Bảng Nhà cung cấp

CREATE TABLE NhaCungCap (

MaNCC NVARCHAR(10) PRIMARY KEY,

TenNCC NVARCHAR(100) NOT NULL,

DiaChi NVARCHAR(200) NOT NULL,

SoDienThoai NVARCHAR(20) NOT NULL

)

go

**Bảng sách**

-- Bảng Sách

CREATE TABLE Sach (

MaSach NVARCHAR(10) PRIMARY KEY,

TenSach NVARCHAR(100) NOT NULL,

TacGia NVARCHAR(100) NOT NULL,

NXB NVARCHAR(100) NOT NULL,

MaTheLoai NVARCHAR(10),

MaNCC NVARCHAR(10),

SoLuongTrenKe INT check (SoLuongTrenKe <= 50),

DonGia DECIMAL(18,2) NOT NULL,

CONSTRAINT FK\_TheLoai\_Sach FOREIGN KEY (MaTheLoai) REFERENCES TheLoai(MaTheLoai),

CONSTRAINT FK\_NCC\_Sach FOREIGN KEY (MaNCC) REFERENCES NhaCungCap(MaNCC)

)

go

**Bảng kho sách**

-- Bảng Kho sách

CREATE TABLE KhoSach (

MaSach NVARCHAR(10),

SoLuong INT,

CONSTRAINT FK\_Sach\_KhoSach FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)

)

go

**Bảng vị trí**

-- Bảng Vị trí

CREATE TABLE ViTriSach (

MaViTri NVARCHAR(10) PRIMARY KEY,

MaSach NVARCHAR(10),

Ke INT,

Tang INT check (Tang <= 3),

Ngan INT check (Ngan <= 4),

CONSTRAINT FK\_ViTri\_Sach FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)

)

go

**Bảng khách hàng**

-- Bảng Khách hàng

CREATE TABLE KhachHang (

MaKhachHang NVARCHAR(10) PRIMARY KEY,

HoTen NVARCHAR(100) NOT NULL,

DiaChi NVARCHAR(200) NOT NULL,

SoDienThoai NVARCHAR(20) NOT NULL

)

go

**Bảng phân loại nhân viên**

-- Bảng Phân loại nhân viên

CREATE TABLE PhanLoaiNhanVien (

MaChucVu NVARCHAR(10) PRIMARY KEY,

TenChucVu NVARCHAR(50) NOT NULL,

MoTaCongViec NVARCHAR(200) NOT NULL,

Luong DECIMAL(18,2) NOT NULL

)

go

**Bảng nhân viên**

-- Bảng Nhân viên

CREATE TABLE NhanVien (

MaNhanVien NVARCHAR(10) PRIMARY KEY,

HoTen NVARCHAR(100) NOT NULL,

NgaySinh DATE NOT NULL,

GioiTinh NVARCHAR(10) NOT NULL,

MaChucVu NVARCHAR(10),

DiaChi NVARCHAR(200) NOT NULL,

SoDienThoai NVARCHAR(20) NOT NULL,

HoatDong NVARCHAR(10),

CONSTRAINT FK\_ChucVu\_NhanVien FOREIGN KEY (MaChucVu) REFERENCES PhanLoaiNhanVien(MaChucVu)

)

go

**Bảng đăng nhập**

-- Bảng Đăng nhập

CREATE TABLE DangNhap (

MaNhanVien NVARCHAR(10) PRIMARY KEY,

TenDangNhap NVARCHAR(30) UNIQUE NOT NULL,

MatKhau NVARCHAR(30) NOT NULL check(len(MatKhau)>=5),

CONSTRAINT FK\_DangNhap\_NhanVien FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien)

)

go

**Bảng  lương**

-- Bảng Lương

CREATE TABLE Luong (

MaNhanVien NVARCHAR(10) NOT NULL,

SoNgayLamViec INT NOT NULL check (SoNgayLamViec <= 26),

Thuong DECIMAL(18,2),

TongLuong DECIMAL(18,2),

CONSTRAINT FK\_NhanVien\_Luong FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien)

)

go

**Bảng hóa đơn**

-- Bảng Hóa đơn

CREATE TABLE HoaDon (

MaHoaDon NVARCHAR(10) PRIMARY KEY,

MaNhanVien NVARCHAR(10),

MaKhachHang NVARCHAR(10) NOT NULL,

ThoiGian DATETIME NOT NULL,

ThanhTien DECIMAL(18,2),

CONSTRAINT FK\_NhanVien\_HoaDon FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien),

CONSTRAINT FK\_KhachHang\_HoaDon FOREIGN KEY (MaKhachHang) REFERENCES KhachHang(MaKhachHang)

)

go

**Bảng chi tiết hóa đơn**

-- Bảng Chi tiết hóa đơn

CREATE TABLE ChiTietHoaDon (

MaHoaDon NVARCHAR(10) NOT NULL,

MaSach NVARCHAR(10),

SoLuong INT NOT NULL,

DonGia INT NOT NULL,

FOREIGN KEY (MaHoaDon) REFERENCES HoaDon(MaHoaDon),

FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)

)

go

**Bảng phiếu nhập**

-- Bảng Phiếu nhập

CREATE TABLE PhieuNhap (

MaPhieuNhap NVARCHAR(10) PRIMARY KEY,

MaNCC NVARCHAR(10),

NgayNhap DATETIME NOT NULL,

ThanhTien DECIMAL(18,2),

FOREIGN KEY (MaNCC) REFERENCES NhaCungCap(MaNCC)

)

go

**Bảng chi tiết nhập sách**

-- Bảng Chi tiết nhập sách

CREATE TABLE ChiTietNhapSach (

MaPhieuNhap NVARCHAR(10),

MaSach NVARCHAR(10),

SoLuongNhap INT NOT NULL,

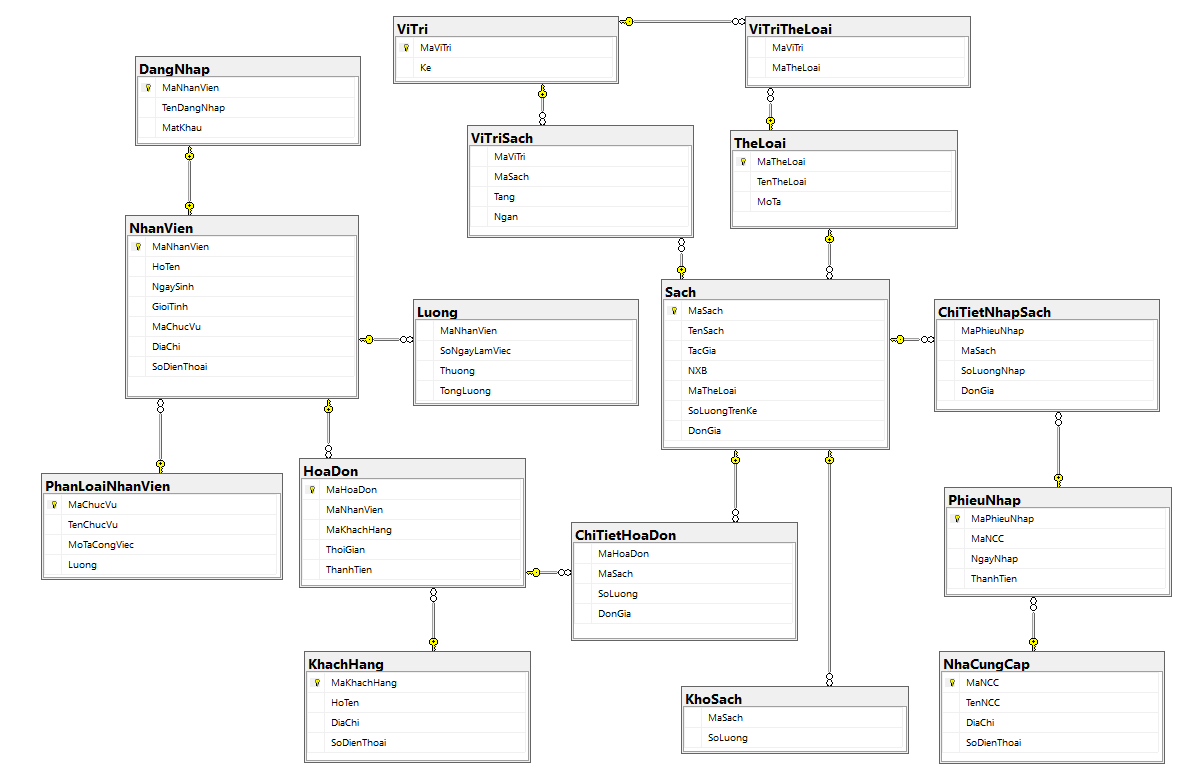
DonGia INT NOT NULL,

FOREIGN KEY (MaPhieuNhap) REFERENCES PhieuNhap(MaPhieuNhap),

FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)

)

go



1. **Các Trigger**

**Trigger để cập nhật số lượng trên kệ của sách**

-- Trigger để cập nhật số lượng trên kệ của Sách

CREATE TRIGGER CapNhatSoLuongKe\_Kho ON ChiTietHoaDon

INSTEAD OF INSERT

AS

BEGIN

-- Kiểm tra số lượng sách

DECLARE @MaSach NVARCHAR(10)

DECLARE @SoLuong INT

DECLARE @SoLuongKho INT

DECLARE @SoLuongTrenKe INT

-- Lấy giá trị từ bảng inserted

SELECT @MaSach = MaSach, @SoLuong = SoLuong

FROM inserted

-- Lấy giá trị từ bảng Kho

SELECT @SoLuongKho = SoLuong

FROM KhoSach

WHERE MaSach = @MaSach

-- Lấy giá trị từ bảng Sach

SELECT @SoLuongTrenKe = SoLuongTrenKe

FROM Sach

WHERE MaSach = @MaSach

IF (SELECT COUNT(\*) FROM inserted WHERE SoLuong < 0 OR DonGia < 0) > 0

BEGIN

RAISERROR('Dữ liệu không hợp lệ. Vui lòng kiểm tra lại.', 16, 1)

END

-- Điều kiện nếu số lượng sách mua lớn hơn tổng số lượng trên kệ và trong kho

ELSE IF @SoLuong > (@SoLuongKho + @SoLuongTrenKe)

BEGIN

RAISERROR ('Số lượng sách trong hóa đơn vượt quá số lượng trong kho và số lượng trên kệ', 16, 1)

END

-- Điều kiện nếu số lượng sách mua lớn hơn số lượng trên kệ nhưng nhỏ hơn tổng số lượng trên kệ và trong kho

ELSE IF (@SoLuong > @SoLuongTrenKe AND @SoLuong <= (@SoLuongKho + @SoLuongTrenKe))

BEGIN

-- Cập nhật số lượng trên kệ và số lượng kho

UPDATE Sach

SET SoLuongTrenKe = 0

WHERE MaSach = @MaSach

UPDATE KhoSach

SET SoLuong = (SELECT SoLuongKhoTrenKe FROM (

SELECT (@SoLuongKho + @SoLuongTrenKe - @SoLuong) AS SoLuongKhoTrenKe

FROM KhoSach ks

INNER JOIN Sach s ON ks.MaSach = s.MaSach

WHERE ks.MaSach = @MaSach

) t)

WHERE MaSach = @MaSach

INSERT INTO ChiTietHoaDon (MaHoaDon, MaSach, SoLuong, DonGia)

SELECT MaHoaDon, MaSach, SoLuong, DonGia

FROM inserted

END

-- Điều kiện nếu số lượng sách mua nhỏ hơn hoặc bằng số lượng trên kệ

ELSE

BEGIN

UPDATE Sach

SET SoLuongTrenKe = SoLuongTrenKe - @SoLuong

WHERE MaSach = @MaSach

INSERT INTO ChiTietHoaDon (MaHoaDon, MaSach, SoLuong, DonGia)

SELECT MaHoaDon, MaSach, SoLuong, DonGia

FROM inserted

END

END

GO

**Trigger tăng số lượng sách khi thêm đơn nhập sách mới**

-- Trigger tăng số lượng sách khi thêm đơn nhập sách mới

CREATE TRIGGER TangSoLuongSachKho ON ChiTietNhapSach

AFTER INSERT AS

BEGIN

DECLARE @MaSach NVARCHAR(10)

DECLARE @SoLuong INT

SELECT @MaSach = MaSach, @SoLuong = SoLuongNhap FROM inserted

UPDATE KhoSach SET SoLuong = SoLuong + @SoLuong WHERE MaSach = @MaSach

END

GO

**Trigger kiểm tra khi dữ liệu nhập vào là âm**

-- Check insert dữ liệu âm

CREATE TRIGGER CheckChiTietPhieuNhap ON ChiTietNhapSach

INSTEAD OF INSERT AS

BEGIN

IF (SELECT COUNT(\*) FROM inserted WHERE SoLuongNhap<0 OR DonGia<0) >0

BEGIN

RAISERROR('Dữ liệu không hợp lệ. Vui lòng kiểm tra lại.', 16,1)

END

ELSE

BEGIN

INSERT INTO ChiTietNhapSach(MaPhieuNhap, MaSach, SoLuongNhap, DonGia)

SELECT MaPhieuNhap, MaSach, SoLuongNhap, DonGia

FROM inserted

END

END

GO

**Trigger hạn chế sự trùng lập dữ liệu của khách hàng**

-- Trigger hạn chế sự trùng lặp dữ liệu của khách hàng

CREATE TRIGGER TrungLapKhachHang ON dbo.KhachHang

INSTEAD OF INSERT

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO KhachHang(MaKhachHang, HoTen, DiaChi, SoDienThoai)

SELECT i.MaKhachHang, i.HoTen, i.DiaChi, i.SoDienThoai

FROM inserted i

LEFT JOIN KhachHang kh ON i.HoTen = kh.HoTen AND i.SoDienThoai = kh.SoDienThoai

WHERE kh.HoTen IS NULL;

IF @@ROWCOUNT = 0

BEGIN

RAISERROR('Khách hàng này đã tồn tại trong bảng Sách!', 16, 1);

END

END

GO

**Trigger hạn chế sự trùng lặp của dữ liệu sách**

-- Trigger hạn chế sự trùng lặp dữ liệu của Sach

CREATE TRIGGER TrungLapSach ON dbo.Sach

INSTEAD OF INSERT

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO Sach(MaSach, TenSach, TacGia, NXB, MaTheLoai, MaNCC, SoLuongTrenKe, DonGia)

SELECT i.MaSach, i.TenSach, i.TacGia, i.NXB, i.MaTheLoai, i.MaNCC, i.SoLuongTrenKe, i.DonGia

FROM inserted i

LEFT JOIN Sach s ON i.TenSach=s.TenSach AND i.TacGia=s.TacGia

WHERE s.TenSach IS NULL;

IF @@ROWCOUNT = 0

BEGIN

RAISERROR('Sách này đã tồn tại trong bảng Sách!', 16, 1);

END

ELSE IF (SELECT COUNT(\*) FROM inserted WHERE SoLuongTrenKe < 0 OR DonGia < 0) > 0

BEGIN

RAISERROR('Dữ liệu không hợp lệ. Vui lòng kiểm tra lại.', 16,1)

END

END

**Trigger tự động thêm vị trị sách**

--Tự động thêm vị trí sách

CREATE TRIGGER trgThemViTriSach

ON Sach

AFTER INSERT

AS

BEGIN

DECLARE @MaViTri NVARCHAR(10)

DECLARE @MaSach NVARCHAR(10)

DECLARE @Ke INT

DECLARE @Tang INT

DECLARE @Ngan INT

--SELECT TOP 1 @MaViTri = MaViTri FROM ViTriSach ORDER BY MaViTri DESC

-- Tách lấy số cuối cùng trong mã vị trí sách để tăng lên 1

--SET @MaViTri = 'VT' + RIGHT('000' + CAST(RIGHT(@MaViTri, 3) AS INT) + 1, 4)

DECLARE @MaxMaViTri NVARCHAR(10) = (SELECT MAX(MaViTri) FROM ViTriSach)

DECLARE @NextMaViTri NVARCHAR(10) = FORMAT(CAST(RIGHT(@MaxMaViTri, 3) AS INT) + 1, 'VT000')

-- Nếu không có bản ghi nào trong bảng ViTriSach, thì gán mã vị trí đầu tiên là VT001

IF @MaxMaViTri IS NULL SET @NextMaViTri = 'VT001'

SELECT @MaSach = MaSach FROM inserted

-- Giá trị tầng cao nhất là 3 và ngăn cao nhất là 4

IF (@Tang > 3)

BEGIN

SET @Tang = 1

SET @Ngan = 1

SET @Ke = @Ke + 1

END

SET @Ke = (SELECT MAX(Ke) FROM ViTriSach)

SET @Tang = (SELECT MAX(Tang) FROM ViTriSach WHERE Ke = @Ke)

SET @Ngan = (SELECT MAX(Ngan) FROM ViTriSach WHERE Ke = @Ke AND Tang = @Tang)

IF @Ke IS NULL

BEGIN

SET @Ke = 1

SET @Tang = 1

SET @Ngan = 1

END

ELSE IF @Tang < 3 AND @Ngan < 4

BEGIN

SET @Ngan = @Ngan + 1

END

ELSE IF @Tang < 3 AND @Ngan = 4

BEGIN

SET @Tang = @Tang + 1

SET @Ngan = 1

END

ELSE IF @Tang = 3 AND @Ngan = 4

BEGIN

SET @Ke = @Ke + 1

SET @Tang = 1

SET @Ngan = 1

END

INSERT INTO ViTriSach(MaViTri, MaSach, Ke, Tang, Ngan)

VALUES (@NextMaViTri, @MaSach, @Ke, @Tang, @Ngan)

END

GO

1. **Các View**

**View thông tin sách**

-SQL:

-- View thông tin Sách

CREATE VIEW View\_Book AS

SELECT \* FROM Sach

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Book";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvSach.Items.Clear();

while (reader.Read())

{

string MaSach = reader.GetString(0);

string TenSach = reader.GetString(1);

string TacGia = reader.GetString(2);

string NXB = reader.GetString(3);

string MaTheLoai = reader.GetString(4);

string MaNCC = reader.GetString(5);

int SoLuongTrenKe = reader.GetInt32(6);

decimal DonGia = reader.GetDecimal(7);

ListViewItem lvi = new ListViewItem(MaSach);

lvi.SubItems.Add(TenSach);

lvi.SubItems.Add(TacGia);

lvi.SubItems.Add(NXB);

lvi.SubItems.Add(MaTheLoai);

lvi.SubItems.Add(MaNCC);

lvi.SubItems.Add(SoLuongTrenKe.ToString());

lvi.SubItems.Add(DonGia.ToString());

lsvSach.Items.Add(lvi);

}

reader.Close();

**View thông tin kho sách**

-SQL:

-- View thông tin Kho Sách

CREATE VIEW View\_Inventory AS

SELECT KhoSach.MaSach, Sach.TenSach, KhoSach.SoLuong

FROM KhoSach JOIN Sach ON KhoSach.MaSach = Sach.MaSach

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Inventory";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvKhoSach.Items.Clear();

while (reader.Read())

{

string MaSach = reader.GetString(0);

string TenSach = reader.GetString(1);

int SoLuongKho = reader.GetInt32(2);

ListViewItem lvi = new ListViewItem(MaSach);

lvi.SubItems.Add(TenSach);

lvi.SubItems.Add(SoLuongKho.ToString());

lsvKhoSach.Items.Add(lvi);

}

**View thông tin vị trí sách**

-SQL:

-- View thông tin vị trí sách

CREATE VIEW View\_BookLocation AS

SELECT \* FROM ViTriSach

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_BookLocation";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvVTS.Items.Clear();

while (reader.Read())

{

string MaViTri = reader.GetString(0);

string MaSach = reader.GetString(1);

string Ke = reader.GetInt32(2).ToString();

string Tang = reader.GetInt32(3).ToString();

string Ngan = reader.GetInt32(4).ToString();

ListViewItem lvi = new ListViewItem(MaViTri);

lvi.SubItems.Add(MaSach);

lvi.SubItems.Add(Ke);

lvi.SubItems.Add(Tang);

lvi.SubItems.Add(Ngan);

lsvVTS.Items.Add(lvi);

}

**View thông tin thể loại**

-SQL:

-- View thông tin Thể Loại

CREATE VIEW View\_Genre AS

SELECT \* FROM TheLoai

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Genre";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvTheLoai.Items.Clear();

while (reader.Read())

{

string MaTheLoai = reader.GetString(0);

string TenTheLoai = reader.GetString(1);

string MoTa = reader.GetString(2);

ListViewItem lvi = new ListViewItem(MaTheLoai);

lvi.SubItems.Add(TenTheLoai);

lvi.SubItems.Add(MoTa);

lsvTheLoai.Items.Add(lvi);

}

**View thông tin nhân viên**

-SQL:

-- View thông tin Nhân viên

CREATE VIEW View\_Staff AS

SELECT \* FROM NhanVien

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Staff";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvNhanVien.Items.Clear();

while (reader.Read())

{

string MaNhanVien = reader.GetString(0);

string HoTen = reader.GetString(1);

string NgaySinh = reader.GetDateTime(2).ToString("yyyy-MM-dd");

string GioiTinh = reader.GetString(3);

string MaChucVu = reader.GetString(4);

string DiaChi = reader.GetString(5);

string SoDienThoai = reader.GetString(6);

string HoatDong = reader.GetString(7);

ListViewItem lvi = new ListViewItem(MaNhanVien);

lvi.SubItems.Add(HoTen);

lvi.SubItems.Add(NgaySinh);

lvi.SubItems.Add(GioiTinh);

lvi.SubItems.Add(MaChucVu);

lvi.SubItems.Add(DiaChi);

lvi.SubItems.Add(SoDienThoai);

lvi.SubItems.Add(HoatDong);

lsvNhanVien.Items.Add(lvi);

}

**View thông tin phân loại nhân viên**

-SQL:

-- View thông tin phân loại Nhân viên

CREATE VIEW View\_ClassifiedStaff AS

SELECT \* FROM PhanLoaiNhanVien

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_ClassifiedStaff";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvPhanLoaiNhanVien.Items.Clear();

while (reader.Read())

{

string MaChucVu = reader.GetString(0);

string TenChucVu = reader.GetString(1);

string MoTaCongViec = reader.GetString(2);

string Luong = reader.GetDecimal(3).ToString();

ListViewItem lvi = new ListViewItem(MaChucVu);

lvi.SubItems.Add(TenChucVu);

lvi.SubItems.Add(MoTaCongViec);

lvi.SubItems.Add(Luong);

lsvPhanLoaiNhanVien.Items.Add(lvi);

}

**View thông tin lương**

-SQL:

-- View thông tin Lương

CREATE VIEW View\_Salary AS

SELECT \* FROM Luong

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Salary";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvLuong.Items.Clear();

while (reader.Read())

{

string MaNhanVien = reader.GetString(0);

string SoNgayLamViec = reader.GetInt32(1).ToString();

string Thuong = reader.GetDecimal(2).ToString();

string TongLuong = reader.GetDecimal(3).ToString();

ListViewItem lvi = new ListViewItem(MaNhanVien);

lvi.SubItems.Add(SoNgayLamViec);

lvi.SubItems.Add(Thuong);

lvi.SubItems.Add(TongLuong);

lsvLuong.Items.Add(lvi);

}

**View thông tin khách hàng**

-SQL:

-- View thông tin Khách hàng

CREATE VIEW View\_Customer AS

SELECT \* FROM KhachHang

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Customer";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvKhachHang.Items.Clear();

while (reader.Read())

{

string MaKhachHang = reader.GetString(0);

string HoTen = reader.GetString(1);

string DiaChi = reader.GetString(2);

string SoDienThoai = reader.GetString(3);

ListViewItem lvi = new ListViewItem(MaKhachHang);

lvi.SubItems.Add(HoTen);

lvi.SubItems.Add(DiaChi);

lvi.SubItems.Add(SoDienThoai);

lsvKhachHang.Items.Add(lvi);

}

**View thông tin hóa đơn**

-SQL:

--View thông tin Hóa Đơn

CREATE VIEW View\_Bill AS

SELECT \* FROM HoaDon

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Bill";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvHoaDon.Items.Clear();

while (reader.Read())

{

string MaHoaDon = reader.GetString(0);

string MaNhanVien = reader.GetString(1);

string MaKhachHang = reader.GetString(2);

DateTime ThoiGian = reader.GetDateTime(3);

Decimal ThanhTien = reader.GetDecimal(4);

ListViewItem lvi = new ListViewItem(MaHoaDon);

lvi.SubItems.Add(MaNhanVien);

lvi.SubItems.Add(MaKhachHang);

lvi.SubItems.Add(ThoiGian.ToString());

lvi.SubItems.Add(ThanhTien.ToString());

lsvHoaDon.Items.Add(lvi);

}

**View thông tin chi tiết hóa đơn**

-SQL:

--View thông tin chi tiết Hóa Đơn

CREATE VIEW View\_BillDetails AS

SELECT \* FROM ChiTietHoaDon

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_BillDetails";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvChiTietHoaDon.Items.Clear();

while (reader.Read())

{

string MaHoaDon = reader.GetString(0);

string MaSach = reader.GetString(1);

int SoLuong = reader.GetInt32(2);

int DonGia = reader.GetInt32(3);

ListViewItem lvi = new ListViewItem(MaHoaDon);

lvi.SubItems.Add(MaSach);

lvi.SubItems.Add(SoLuong.ToString());

lvi.SubItems.Add(DonGia.ToString());

lsvChiTietHoaDon.Items.Add(lvi);

}

**View thông tin phiếu nhập**

-SQL:

-- View thông tin phiếu nhập

CREATE VIEW View\_Receipt AS

SELECT \* FROM PhieuNhap

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Receipt";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvPhieuNhap.Items.Clear();

while (reader.Read())

{

string MaPhieuNhap = reader.GetString(0);

string MaNCC = reader.GetString(1);

DateTime NgayNhap = reader.GetDateTime(2);

string ThanhTien = reader.GetDecimal(3).ToString();

ListViewItem lvi = new ListViewItem(MaPhieuNhap);

lvi.SubItems.Add(MaNCC);

lvi.SubItems.Add(NgayNhap.ToString("yyyy-MM-dd HH:mm:ss"));

lvi.SubItems.Add(ThanhTien);

lsvPhieuNhap.Items.Add(lvi);

}

**View thông tin chi tiếp phiếu nhập**

-SQL:

-- View thông tin chi tiết phiếu nhập

CREATE VIEW View\_ReceiptDetails AS

SELECT \* FROM ChiTietNhapSach

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_ReceiptDetails";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvChiTietPhieuNhap.Items.Clear();

while (reader.Read())

{

string MaPhieuNhap = reader.GetString(0);

string MaSach = reader.GetString(1);

string SoLuongNhap = reader.GetInt32(2).ToString();

string DonGia = reader.GetInt32(3).ToString();

ListViewItem lvi = new ListViewItem(MaPhieuNhap);

lvi.SubItems.Add(MaSach);

lvi.SubItems.Add(SoLuongNhap);

lvi.SubItems.Add(DonGia);

lsvChiTietPhieuNhap.Items.Add(lvi);

}

**View thông tin nhà cung cấp**

-SQL:

-- View thông tin Nhà Cung Cấp

CREATE VIEW View\_Supplier AS

SELECT \* FROM NhaCungCap

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Supplier";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvNCC.Items.Clear();

while (reader.Read())

{

string MaNCC = reader.GetString(0);

string TenNCC = reader.GetString(1);

string DiaChi = reader.GetString(2);

string SoDienThoai = reader.GetString(3);

ListViewItem lvi = new ListViewItem(MaNCC);

lvi.SubItems.Add(TenNCC);

lvi.SubItems.Add(DiaChi);

lvi.SubItems.Add(SoDienThoai);

lsvNCC.Items.Add(lvi);

}

**View thông tin tài khoản đăng nhập**

-SQL:

-- View thông tin tài khoản đăng nhập

CREATE VIEW View\_Account AS

SELECT \* FROM DangNhap

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Account";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvDangNhap.Items.Clear();

while (reader.Read())

{

string MaNhanVien = reader.GetString(0);

string TaiKhoan = reader.GetString(1);

string MatKhau = reader.GetString(2);

ListViewItem lvi = new ListViewItem(MaNhanVien);

lvi.SubItems.Add(TaiKhoan);

lvi.SubItems.Add(MatKhau);

lsvDangNhap.Items.Add(lvi);

}

reader.Close();

# **CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

1. **Kết nối cơ sở dữ liệu**

SqlConnection sqlCon = null;

string username;

string password;

public fDangNhap()

{

InitializeComponent();

}

private void fDangNhap\_Load(object sender, EventArgs e)

{

//KetNoiVoiCSDL();

}

private void KetNoiVoiCSDL()

{

username = this.txtUserName.Text;

password = this.txtPassWord.Text;

try

{

string strCon = $"Data Source=(local);Initial Catalog=QLCHBS;" +

$"User ID={username};" +

$"Password={password};";

sqlCon = new SqlConnection(strCon);

sqlCon.Open();

fQuanLy f = new fQuanLy(username, password);

this.Hide();

f.ShowDialog();

this.Show();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi rồi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

1. **Danh mục xem thông tin**

**View doanh thu cửa hàng từ việc bán sách tính theo tuần**

-SQL:

--View Doanh thu của cửa hàng từ việc bán sách tính theo tuần

CREATE VIEW View\_RevenueStatistics\_Week AS

SELECT DATEPART(WEEK, Ngay) AS Tuan,

SUM(SoLuongSachDaBan) AS TongSoLuong,

SUM(TongTien) AS TongDoanhThu

FROM View\_RevenueStatistics

GROUP BY DATEPART(WEEK, Ngay)

GO

-C#:

// Thiết lập kích thước và vị trí của biểu đồ cột

chartDoanhThu.Series["Series1"].ChartType = SeriesChartType.Column;

chartDoanhThu.Series["Series1"].IsVisibleInLegend = false;

chartDoanhThu.Series["Series1"].XValueMember = "Tuan";

chartDoanhThu.Series["Series1"].YValueMembers = "TongDoanhThu";

chartDoanhThu.ChartAreas[0].AxisX.Interval = 1;

chartDoanhThu.Titles.Add("Title1").Text = "Biểu Đồ Doanh Thu";

string query = "SELECT \* FROM View\_RevenueStatistics\_Week";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

// Thiết lập dữ liệu cho biểu đồ tròn và cột

chartDoanhThu.DataSource = dataTable;

chartDoanhThu.DataBind();

}

// Thêm biểu đồ cột vào form

pDoanhThu.Controls.Add(chartDoanhThu);

**View thanh toán của cửa hàng từ việc nhập sách tính theo tuần**

-SQL:

-- View Thanh toán của cửa hàng từ việc nhập sách tính theo tuần

CREATE VIEW View\_PaymentStatistics\_Week AS

SELECT DATEPART(MONTH, Ngay) AS Thang,

SUM(SoLuongNhap) AS TongSoLuong,

SUM(TongTien) AS TongTien

FROM View\_PaymentStatistics

GROUP BY DATEPART(MONTH, Ngay)

GO

-C#:

// Thiết lập kích thước và vị trí của biểu đồ cột

chartPhieuNhap.Series["Series1"].ChartType = SeriesChartType.Column;

chartPhieuNhap.Series["Series1"].IsVisibleInLegend = false;

chartPhieuNhap.Series["Series1"].Color = System.Drawing.ColorTranslator.FromHtml("#FFA500");

chartPhieuNhap.Series["Series1"].XValueMember = "Thang";

chartPhieuNhap.Series["Series1"].YValueMembers = "TongTien";

chartPhieuNhap.ChartAreas[0].AxisX.Interval = 1;

chartPhieuNhap.Titles.Add("Title1").Text = "Biểu Đồ Phiếu Nhập";

string query = "SELECT \* FROM View\_PaymentStatistics\_Week";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

// Thiết lập dữ liệu cho biểu đồ tròn và cột

chartPhieuNhap.DataSource = dataTable;

chartPhieuNhap.DataBind();

}

// Thêm biểu đồ cột vào form

pPhieuNhap.Controls.Add(chartPhieuNhap);

**View top 10 sách bán chạy nhất**

-SQL:

-- View top 10 sách bán chạy nhất

CREATE VIEW View\_BookBestSell AS

SELECT TOP 10 Sach.MaSach, Sach.TenSach,

SUM(ChiTietHoaDon.SoLuong) AS TongSoLuongBan

FROM Sach JOIN ChiTietHoaDon ON Sach.MaSach = ChiTietHoaDon.MaSach

GROUP BY Sach.TenSach, Sach.MaSach

ORDER BY TongSoLuongBan DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_BookBestSell";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaSach"].ToString());

item.SubItems.Add(row["TenSach"].ToString());

item.SubItems.Add(row["TongSoLuongBan"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 10 sách chậm nhất**

-SQL:

-- View top 10 sách bán chậm nhất

CREATE VIEW View\_BookWorstSell AS

SELECT TOP 10 Sach.MaSach, Sach.TenSach,

SUM(ChiTietHoaDon.SoLuong) AS TongSoLuongBan

FROM Sach JOIN ChiTietHoaDon ON Sach.MaSach = ChiTietHoaDon.MaSach

GROUP BY Sach.TenSach, Sach.MaSach

ORDER BY TongSoLuongBan ASC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_BookWorstSell";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaSach"].ToString());

item.SubItems.Add(row["TenSach"].ToString());

item.SubItems.Add(row["TongSoLuongBan"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 5 thể loại có số lượng sách trên kệ và trong kho nhiều nhất**

-SQL:

-- View top 5 thể loại có số lượng sách trên kệ và trong kho nhiều nhất

CREATE VIEW View\_GenreHighestNumber AS

SELECT TOP 5

tl.MaTheLoai,

tl.TenTheLoai,

SUM(ks.SoLuong + s.SoLuongTrenKe) AS TongSoLuong

FROM TheLoai tl

JOIN Sach s ON tl.MaTheLoai = s.MaTheLoai

JOIN KhoSach ks ON s.MaSach = ks.MaSach

GROUP BY tl.TenTheLoai, tl.MaTheLoai

ORDER BY TongSoLuong DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_GenreHighestNumber";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaTheLoai"].ToString());

item.SubItems.Add(row["TenTheLoai"].ToString());

item.SubItems.Add(row["TongSoLuong"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 5 thể loại có số lượng sách trên kệ và trong kho ít nhất**

-SQL:

-- View top 5 thể loại có số lượng sách trên kệ và trong kho ít nhất

CREATE VIEW View\_GenreLeastNumber AS

SELECT TOP 5

tl.MaTheLoai,

tl.TenTheLoai,

SUM(ks.SoLuong + s.SoLuongTrenKe) AS TongSoLuong

FROM TheLoai tl

JOIN Sach s ON tl.MaTheLoai = s.MaTheLoai

JOIN KhoSach ks ON s.MaSach = ks.MaSach

GROUP BY tl.TenTheLoai, tl.MaTheLoai

ORDER BY TongSoLuong ASC

GO

-C#:

if (rbLeastNumber.Checked)

{

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_GenreLeastNumber";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaTheLoai"].ToString());

item.SubItems.Add(row["TenTheLoai"].ToString());

item.SubItems.Add(row["TongSoLuong"].ToString());

lsvDanhGia.Items.Add(item);

}

}

**View top 5 thể loại có lượng sách bán nhiều nhất**

-SQL:

-- View top 5 thể loại có lượng sách bán nhiều nhất

CREATE VIEW View\_GenreBestSell AS

SELECT TOP 5

tl.MaTheLoai,

tl.TenTheLoai,

SUM(cthd.SoLuong) AS TongSoLuongBan

FROM TheLoai tl

JOIN Sach s ON tl.MaTheLoai = s.MaTheLoai

JOIN ChiTietHoaDon cthd ON s.MaSach = cthd.MaSach

GROUP BY tl.TenTheLoai, tl.MaTheLoai

ORDER BY TongSoLuongBan DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_GenreBestSell";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaTheLoai"].ToString());

item.SubItems.Add(row["TenTheLoai"].ToString());

item.SubItems.Add("");

item.SubItems.Add(row["TongSoLuongBan"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 5 thể loại có lượng sách bán ít nhất**

-SQL:

-- View top 5 thể loại có lượng sách bán ít nhất

CREATE VIEW View\_GenreWorstSell AS

SELECT TOP 5

tl.MaTheLoai,

tl.TenTheLoai,

SUM(cthd.SoLuong) AS TongSoLuongBan

FROM TheLoai tl

JOIN Sach s ON tl.MaTheLoai = s.MaTheLoai

JOIN ChiTietHoaDon cthd ON s.MaSach = cthd.MaSach

GROUP BY tl.TenTheLoai, tl.MaTheLoai

ORDER BY TongSoLuongBan ASC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_GenreWorstSell";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaTheLoai"].ToString());

item.SubItems.Add(row["TenTheLoai"].ToString());

item.SubItems.Add("");

item.SubItems.Add(row["TongSoLuongBan"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 20 khách hàng mua nhiều nhất**

-SQL:

-- View 20 khách hàng mua nhiều nhất

CREATE VIEW View\_CustomerMostBuy AS

SELECT TOP 20 KhachHang.MaKhachHang, KhachHang.HoTen,

COUNT(\*) AS TongSoHoaDon,

SUM(ChiTietHoaDon.SoLuong) AS TongSoLuongMua

FROM KhachHang

JOIN HoaDon ON KhachHang.MaKhachHang = HoaDon.MaKhachHang

JOIN ChiTietHoaDon ON HoaDon.MaHoaDon = ChiTietHoaDon.MaHoaDon

GROUP BY KhachHang.MaKhachHang, KhachHang.HoTen

ORDER BY TongSoLuongMua DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_CustomerMostBuy";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaKhachHang"].ToString());

item.SubItems.Add(row["HoTen"].ToString());

item.SubItems.Add(row["TongSoHoaDon"].ToString());

item.SubItems.Add(row["TongSoLuongMua"].ToString());

lsvDanhGia.Items.Add(item);

}

**View 20 khách hàng mua ít nhất**

-SQL:

-- View 20 khách hàng mua ít nhất

CREATE VIEW View\_CustomerLeastBuy AS

SELECT TOP 20 KhachHang.MaKhachHang, KhachHang.HoTen,

COUNT(\*) AS TongSoHoaDon,

SUM(ChiTietHoaDon.SoLuong) AS TongSoLuongMua

FROM KhachHang

JOIN HoaDon ON KhachHang.MaKhachHang = HoaDon.MaKhachHang

JOIN ChiTietHoaDon ON HoaDon.MaHoaDon = ChiTietHoaDon.MaHoaDon

GROUP BY KhachHang.MaKhachHang, KhachHang.HoTen

ORDER BY TongSoLuongMua ASC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_CustomerLeastBuy";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaKhachHang"].ToString());

item.SubItems.Add(row["HoTen"].ToString());

item.SubItems.Add(row["TongSoHoaDon"].ToString());

item.SubItems.Add(row["TongSoLuongMua"].ToString());

lsvDanhGia.Items.Add(item);

}

**View 10 sách tồn kho nhiều nhất**

-SQL:

-- View 10 sách tồn kho nhiều nhất

CREATE VIEW View\_MostInStock AS

SELECT TOP 10 Sach.MaSach, TenSach, SUM(SoLuong) AS TongSoLuongTrongKho

FROM Sach

JOIN KhoSach ON Sach.MaSach = KhoSach.MaSach

GROUP BY Sach.MaSach, TenSach

ORDER BY TongSoLuongTrongKho DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_MostInStock";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaSach"].ToString());

item.SubItems.Add(row["TenSach"].ToString());

item.SubItems.Add(row["TongSoLuongTrongKho"].ToString());

lsvDanhGia.Items.Add(item);

}

**View 10 sách tồn kho ít nhất**

-SQL:

-- View 10 sách tồn kho ít nhất

CREATE VIEW View\_LeastInStock AS

SELECT TOP 10 Sach.MaSach, TenSach, SUM(SoLuong) AS TongSoLuongTrongKho

FROM Sach

JOIN KhoSach ON Sach.MaSach = KhoSach.MaSach

GROUP BY Sach.MaSach, TenSach

ORDER BY TongSoLuongTrongKho ASC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_LeastInStock";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaSach"].ToString());

item.SubItems.Add(row["TenSach"].ToString());

item.SubItems.Add(row["TongSoLuongTrongKho"].ToString());

lsvDanhGia.Items.Add(item);

}

**View những nhân viên không vắng làm**

-SQL:

-- view những nhân viên không vắng làm

CREATE VIEW View\_NoAbsence AS

SELECT \* FROM Luong

WHERE SoNgayLamViec = 26

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_NoAbsence";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaNhanVien"].ToString());

item.SubItems.Add(row["SoNgayLamViec"].ToString());

item.SubItems.Add(row["Thuong"].ToString());

item.SubItems.Add(row["TongLuong"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 5 nhân viên có tổng lương cao nhất**

-SQL:

-- view top 5 nhân viên có tổng lương cao nhất

CREATE VIEW View\_Top5HighSalary AS

SELECT TOP 5 \* FROM Luong

ORDER BY TongLuong DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Top5HighSalary";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaNhanVien"].ToString());

item.SubItems.Add(row["SoNgayLamViec"].ToString());

item.SubItems.Add(row["Thuong"].ToString());

item.SubItems.Add(row["TongLuong"].ToString());

lsvDanhGia.Items.Add(item);

}

**View top 3 nhân viên có thưởng cao nhất**

-SQL:

-- view top 3 nhân viên có thưởng cao nhất

CREATE VIEW View\_Top3Bonus AS

SELECT TOP 3 \*

FROM Luong

ORDER BY Thuong DESC

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Top3Bonus";

command.Connection = conn;

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dt = new DataTable();

da.Fill(dt);

lsvDanhGia.Items.Clear();

foreach (DataRow row in dt.Rows)

{

ListViewItem item = new ListViewItem(row["MaNhanVien"].ToString());

item.SubItems.Add(row["SoNgayLamViec"].ToString());

item.SubItems.Add(row["Thuong"].ToString());

item.SubItems.Add(row["TongLuong"].ToString());

lsvDanhGia.Items.Add(item);

}

1. **Danh mục kiểm tra**

**Kiểm tra đăng nhập**

-SQL:

create function scalarfDangNhap(@TenDangNhap nvarchar(30), @MatKhau NVARCHAR(30))

returns nvarchar(30)

as

begin

declare @result nvarchar(30)

set @result = ''

if exists (select \* from DangNhap where TenDangNhap = @TenDangNhap and MatKhau = @MatKhau)

begin

select @result = MaNhanVien from DangNhap where TenDangNhap = @TenDangNhap;

end

return @result

end

go

-C#:

SqlCommand command = new SqlCommand("SELECT dbo.scalarfDangNhap(@TenDangNhap, @MatKhau)", sqlCon);

command.Connection = sqlCon;

command.CommandType = CommandType.Text;

command.Parameters.AddWithValue("@tenDangNhap", this.txtUserName.Text);

command.Parameters.AddWithValue("@matKhau", this.txtPassWord.Text);

string result = command.ExecuteScalar().ToString();

if (result != "")

{

fQuanLy f = new fQuanLy(this.txtUserName.Text, result);

this.Hide();

f.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("Sai tên đăng nhập hoặc mật khẩu!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**Kiểm tra Tài Khoản và Số Điện Thoại có tồn tại không**

-SQL:

-- PROCEDURE kiểm tra xem TaiKhoan và SĐT có tồn tại không

CREATE PROCEDURE CheckAccount\_Phone

@SDT NVARCHAR(20),

@TenDangNhap NVARCHAR(30)

AS

BEGIN

SELECT COUNT(\*) FROM NhanVien JOIN DangNhap ON NhanVien.MaNhanVien = DangNhap.MaNhanVien WHERE SoDienThoai = @SDT AND TenDangNhap = @TenDangNhap

END

GO

-C#:

string connectionString = "Data Source=(local);Initial Catalog=QLCHBS;Integrated Security=True";

// Lấy dữ liệu từ các textbox

string tendangnhap = txtForgotUserName.Text.Trim();

string sdt = txtPhoneNumber.Text.Trim();

string matKhauMoi = txtForgorPW.Text.Trim();

string xacNhanMatKhau = txtConfirmPW.Text.Trim();

int count = 0;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

using (SqlCommand command = new SqlCommand("CheckAccount\_Phone", connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@SDT", sdt);

command.Parameters.AddWithValue("@TenDangNhap", tendangnhap);

count = (int)command.ExecuteScalar();

}

}

1. **Danh mục cập nhật**

**Cập nhật Mật Khẩu**

-SQL:

CREATE PROCEDURE UpdateAccount

@MatKhauMoi NVARCHAR(30),

@TenDangNhap NVARCHAR(30)

AS

BEGIN

DECLARE @command NVARCHAR(MAX) = '';

SET @command = '';

SELECT @command = @command + 'ALTER LOGIN [' + name + '] WITH PASSWORD = '''+@MatKhauMoi+''';'

FROM sys.syslogins

WHERE name LIKE @TenDangNhap;

EXEC (@command);

UPDATE DangNhap SET MatKhau = @MatKhauMoi WHERE TenDangNhap = @TenDangNhap

END

GO

-C#:

//query = "UPDATE DangNhap SET MatKhau = @MatKhauMoi WHERE TenDangNhap = @TenDangNhap";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

using (SqlCommand command = new SqlCommand("UpdateAccount", connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MatKhauMoi", matKhauMoi);

command.Parameters.AddWithValue("@TenDangNhap", tendangnhap);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Đổi mật khẩu thành công.");

txtForgotUserName.Clear();

txtPhoneNumber.Clear();

txtForgorPW.Clear();

txtConfirmPW.Clear();

}

else

{

MessageBox.Show("Đổi mật khẩu thất bại.");

}

}

**Cập nhật Sách**

-SQL:

CREATE PROCEDURE UpdateBook

@MaSach nvarchar(10),

@TenSach nvarchar(100),

@TacGia nvarchar(100),

@NXB nvarchar(100),

@MaTheLoai nvarchar(10),

@MaNCC nvarchar(10),

@SoLuongTrenKe int,

@DonGia decimal(18,2)

AS

BEGIN

UPDATE Sach

SET MaSach = @MaSach, TenSach = @TenSach, TacGia = @TacGia, NXB = @NXB,

MaTheLoai = @MaTheLoai, MaNCC = @MaNCC, SoLuongTrenKe = @SoLuongTrenKe, DonGia = @DonGia

WHERE MaSach = @MaSach

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateBook";

command.Connection = conn;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

//txtMaSach.Text = command.Parameters["@MaSach"].Value.ToString();

command.Parameters.Add("@TenSach", SqlDbType.NVarChar).Value = txtTenSach.Text;

command.Parameters.Add("@TacGia", SqlDbType.NVarChar).Value = txtTacGia.Text;

command.Parameters.Add("@NXB", SqlDbType.NVarChar).Value = txtNXB.Text;

command.Parameters.Add("@MaTheLoai", SqlDbType.NVarChar).Value = txtMaTheLoai.Text;

command.Parameters.Add("@MaNCC", SqlDbType.NVarChar).Value = txtMaNCC.Text;

command.Parameters.Add("@SoluongTrenKe", SqlDbType.Int).Value = int.Parse(txtSoLuong.Text);

command.Parameters.Add("@DonGia", SqlDbType.Decimal).Value = decimal.Parse(txtDonGia.Text);

**Cập nhật Sách vào Kho**

-SQL:

CREATE PROCEDURE UpdateInventory

@MaSach nvarchar(10),

@SoLuong int

AS

BEGIN

Update KhoSach SET SoLuong = @SoLuong WHERE MaSach = @MaSach

END

-C#:

SqlCommand command1 = new SqlCommand();

command1.CommandType = CommandType.StoredProcedure;

command1.CommandText = "UpdateInventory";

command1.Connection = conn;

command1.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command1.Parameters.Add("@Soluong", SqlDbType.Int).Value = int.Parse(txtSoLuongKho.Text);

**Cập nhật Thông Tin Khách Hàng**

-SQL:

CREATE PROCEDURE UpdateCustomer

@MaKhachHang NVARCHAR(10),

@HoTen NVARCHAR(100),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20)

AS

BEGIN

UPDATE KhachHang

SET MaKhachHang = @MaKhachHang, HoTen = @HoTen, DiaChi = @DiaChi, SoDienThoai = @SoDienThoai

WHERE MaKhachHang = @MaKhachHang

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateCustomer";

command.Connection = conn;

command.Parameters.Add("@MaKhachHang", SqlDbType.NVarChar).Value = txtMaKhachHang.Text;

//txtMaSach.Text = command.Parameters["@MaSach"].Value.ToString();

command.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txtHoTen.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChi.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoai.Text;

**Cập nhật Thông tin Nhân Viên**

-SQL:

CREATE PROCEDURE UpdateEmployee

@MaNhanVien NVARCHAR(10),

@HoTen NVARCHAR(100),

@NgaySinh DATE,

@GioiTinh NVARCHAR(10),

@MaChucVu NVARCHAR(10),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20),

@HoatDong nvarchar(10)

AS

BEGIN

-- cập nhật nhân viên

UPDATE NhanVien

SET HoTen = @HoTen, NgaySinh = @NgaySinh,

GioiTinh = @GioiTinh, MaChucVu = @MaChucVu, DiaChi = @DiaChi,

SoDienThoai = @SoDienThoai,HoatDong = @HoatDong

WHERE MaNhanVien = @MaNhanVien

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateEmployee";

command.Connection = conn;

command.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txtMaNhanVien.Text;

command.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txtHoTen.Text;

command.Parameters.Add("@NgaySinh", SqlDbType.DateTime).Value = DateTime.ParseExact(txtNgaySinh.Text, "yyyy/MM/dd", CultureInfo.InvariantCulture);

command.Parameters.Add("@GioiTinh", SqlDbType.NVarChar).Value = txtGioiTinh.Text;

command.Parameters.Add("@MaChucVu", SqlDbType.NVarChar).Value = txtMaChucVu.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChi.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoai.Text;

command.Parameters.Add("@HoatDong", SqlDbType.NVarChar).Value = txtHoatDong.Text;

**Cập nhật Thông tin Nhà Cung Cấp**

-SQL:

CREATE PROCEDURE UpdateSupplier

@MaNCC NVARCHAR(10),

@TenNCC NVARCHAR(100),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20)

AS

BEGIN

UPDATE NhaCungCap

SET MaNCC = @MaNCC, TenNCC = @TenNCC, DiaChi = @DiaChi, SoDienThoai = @SoDienThoai

WHERE MaNCC = @MaNCC

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateSupplier";

command.Connection = conn;

command.Parameters.Add("@MaNCC", SqlDbType.NVarChar).Value = txtMaNCC.Text;

//txtMaSach.Text = command.Parameters["@MaSach"].Value.ToString();

command.Parameters.Add("@TenNCC", SqlDbType.NVarChar).Value = txtTenNCC.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChiNCC.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoaiNCC.Text;

**Cập nhật Thông tin Thể Loại**

-SQL:

CREATE PROCEDURE UpdateGenre

@TenTheLoai NVARCHAR(100),

@MaTheLoai NVARCHAR(10) ,

@MoTa NVARCHAR (255)

AS

UPDATE TheLoai SET

TenTheLoai = @TenTheLoai,

MoTa = @MoTa

Where MaTheLoai = @MaTheLoai

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateGenre";

command.Connection = conn;

//txtMaSach.Text = command.Parameters["@MaSach"].Value.ToString();

command.Parameters.Add("@MaTheLoai", SqlDbType.NVarChar).Value = txtMaTheLoai.Text;

command.Parameters.Add("@TenTheLoai", SqlDbType.NVarChar).Value = txtTenTheLoai.Text;

command.Parameters.Add("@MoTa", SqlDbType.NVarChar).Value = txtMoTa.Text;

**Cập nhật Vị Trí Sách**

-SQL:

CREATE PROCEDURE UpdateBookLocation

@MaViTri nvarchar(10),

@MaSach nvarchar(10),

@Ke int,

@Tang int,

@Ngan int

AS

BEGIN

UPDATE ViTriSach

SET MaSach = @MaSach, Ke = @Ke, Tang = @Tang, Ngan = @Ngan

WHERE MaViTri = @MaViTri

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateBookLocation";

command.Connection = conn;

//txtMaSach.Text = command.Parameters["@MaSach"].Value.ToString();

command.Parameters.Add("@MaViTri", SqlDbType.NVarChar).Value = txtMaViTri.Text;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command.Parameters.Add("@Ke", SqlDbType.Int).Value = int.Parse(txtKe.Text);

command.Parameters.Add("@Tang", SqlDbType.Int).Value = int.Parse(txtTang.Text);

command.Parameters.Add("@Ngan", SqlDbType.Int).Value = int.Parse(txtNgan.Text);

**Cập nhật Phiếu Nhập Sách**

-SQL:

CREATE PROCEDURE UpdateReceipt

@MaPhieuNhap nvarchar(10),

@MaNCC nvarchar(10)

AS

BEGIN

Update PhieuNhap

SET MaNCC = @MaNCC

WHERE MaPhieuNhap = @MaPhieuNhap

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateReceipt";

command.Connection = conn;

command.Parameters.Add("@MaPhieuNhap", SqlDbType.NVarChar).Value = txtMaPhieuNhap.Text;

command.Parameters.Add("@MaNCC", SqlDbType.NVarChar).Value = txtMaNCC.Text;

//command.Parameters.Add("@NgayNhap", SqlDbType.DateTime).Value = DateTime.Parse(txtNgayNhap.Text);

**Cập nhật Chi Tiết Phiếu Nhập**

-SQL:

CREATE PROCEDURE UpdateReceiptDetails

@MaPhieuNhap nvarchar(10),

@MaSach nvarchar(10),

@SoLuongNhap int,

@DonGia int

AS

BEGIN

Update ChiTietNhapSach

SET SoLuongNhap = @SoLuongNhap, DonGia = @DonGia

WHERE MaPhieuNhap = @MaPhieuNhap AND MaSach = @MaSach

UPDATE PhieuNhap

SET ThanhTien = (

SELECT SUM(SoLuongNhap \* DonGia)

FROM ChiTietNhapSach

WHERE PhieuNhap.MaPhieuNhap = ChiTietNhapSach.MaPhieuNhap)

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateReceiptDetails";

command.Connection = conn;

command.Parameters.Add("@MaPhieuNhap", SqlDbType.NVarChar).Value = txtMaPNChiTiet.Text;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command.Parameters.Add("@SoLuongNhap", SqlDbType.Int).Value = int.Parse(txtSoLuongNhap.Text);

command.Parameters.Add("@DonGia", SqlDbType.Int).Value = int.Parse(txtDonGia.Text);

**Cập nhật Hóa Đơn**

-SQL:

CREATE PROCEDURE UpdateBill

@MaHoaDon nvarchar(10),

@MaNhanVien nvarchar(10),

@MaKhachHang nvarchar (10)

AS

BEGIN

Update HoaDon

SET MaNhanVien = @MaNhanVien, MaKhachHang = @MaKhachHang

WHERE MaHoaDon = @MaHoaDon

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateBillDetails";

command.Connection = conn;

command.Parameters.Add("@MaHoaDon", SqlDbType.NVarChar).Value = txtMaHoaDonCT.Text;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = int.Parse(txtSoLuong.Text);

command.Parameters.Add("@DonGia", SqlDbType.Int).Value = int.Parse(txtDonGia.Text);

**Cập nhật Chi Tiết Hóa Đơn**

-SQL:

CREATE PROCEDURE UpdateBillDetails

@MaHoaDon nvarchar(10),

@MaSach nvarchar(10),

@SoLuong int,

@DonGia int

AS

BEGIN

Update ChiTietHoaDon

SET SoLuong = @SoLuong, DonGia = @DonGia

WHERE MaHoaDon = @MaHoaDon AND MaSach = @MaSach

UPDATE HoaDon

SET ThanhTien = (

SELECT SUM(SoLuong \* DonGia)

FROM ChiTietHoaDon

WHERE HoaDon.MaHoaDon = ChiTietHoaDon.MaHoaDon)

END

-C#::

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateBillDetails";

command.Connection = conn;

command.Parameters.Add("@MaHoaDon", SqlDbType.NVarChar).Value = txtMaHoaDonCT.Text;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = int.Parse(txtSoLuong.Text);

command.Parameters.Add("@DonGia", SqlDbType.Int).Value = int.Parse(txtDonGia.Text);

**Cập nhật Thông tin Nhân Viên**

-SQL:

CREATE PROCEDURE UpdateEmployee

@MaNhanVien NVARCHAR(10),

@HoTen NVARCHAR(100),

@NgaySinh DATE,

@GioiTinh NVARCHAR(10),

@MaChucVu NVARCHAR(10),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20),

@HoatDong nvarchar(10)

AS

BEGIN

-- cập nhật nhân viên

UPDATE NhanVien

SET HoTen = @HoTen, NgaySinh = @NgaySinh,

GioiTinh = @GioiTinh, MaChucVu = @MaChucVu, DiaChi = @DiaChi, SoDienThoai = @SoDienThoai,HoatDong = @HoatDong

WHERE MaNhanVien = @MaNhanVien

-- kiểm tra trạng thái hoạt động của nhân viên

IF @HoatDong = N'Nghỉ việc'

BEGIN

DECLARE @TenDangNhap varchar(10)

SELECT @TenDangNhap = DangNhap.TenDangNhap FROM DangNhap WHERE MaNhanVien = @MaNhanVien

DECLARE @command nvarchar(max) = ''

SET @command = 'EXEC DeleteAccountEmployee ' + '"' + @TenDangNhap + '"'

EXEC(@command)

END

END

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateEmployee";

command.Connection = conn;

command.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txtMaNhanVien.Text;

command.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txtHoTen.Text;

command.Parameters.Add("@NgaySinh", SqlDbType.DateTime).Value = DateTime.ParseExact(txtNgaySinh.Text, "yyyy/MM/dd", CultureInfo.InvariantCulture);

command.Parameters.Add("@GioiTinh", SqlDbType.NVarChar).Value = txtGioiTinh.Text;

command.Parameters.Add("@MaChucVu", SqlDbType.NVarChar).Value = txtMaChucVu.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChi.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoai.Text;

command.Parameters.Add("@HoatDong", SqlDbType.NVarChar).Value = txtHoatDong.Text;

**Cập nhật Thông tin phân loại Nhân Viên**

-SQL:

CREATE PROCEDURE UpdateClassifiedStaff

@MaChucVu NVARCHAR(10),

@TenChucVu NVARCHAR(50),

@MoTaCongViec NVARCHAR(200),

@Luong DECIMAL(18,2)

AS

BEGIN

-- cập nhật nhân viên

UPDATE PhanLoaiNhanVien

SET TenChucVu = @TenChucVu, MoTaCongViec = @MoTaCongViec, Luong = @Luong

WHERE MaChucVu = @MaChucVu

END

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateClassifiedStaff";

command.Connection = conn;

command.Parameters.Add("@MaChucVu", SqlDbType.NVarChar).Value = txtMaChucVu.Text;

command.Parameters.Add("@TenChucVu", SqlDbType.NVarChar).Value = txtTenChucVu.Text;

command.Parameters.Add("@MoTaCongViec", SqlDbType.NVarChar).Value = txtMoTaCongViec.Text;

command.Parameters.Add("@Luong", SqlDbType.Decimal).Value = decimal.Parse(txtLuong.Text);

int n = command.ExecuteNonQuery();

if (n > 0)

{

HienThiPhanLoaiNhanVien();

txtMaChucVu.ReadOnly = true;

txtTenChucVu.ReadOnly = true;

txtMoTaCongViec.ReadOnly = true;

txtLuong.ReadOnly = true;

MessageBox.Show("Sửa Thành Công");

}

else

{

MessageBox.Show("Sửa Thất Bại");

}

**Cập nhật Thông tin về Lương**

-SQL:

CREATE PROCEDURE UpdateSalary

@MaNhanVien NVARCHAR(10),

@SoNgayLamViec INT ,

@Thuong DECIMAL(18,2)

AS

BEGIN

UPDATE Luong

SET SoNgayLamViec = @SoNgayLamViec, Thuong = @Thuong,

TongLuong = (

SELECT (Luong / 26 \* SoNgayLamViec + Thuong)

FROM PhanLoaiNhanVien join NhanVien on PhanLoaiNhanVien.MaChucVu = NhanVien.MaChucVu

WHERE NhanVien.MaNhanVien = Luong.MaNhanVien)

WHERE MaNhanVien = @MaNhanVien

END

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "UpdateSalary";

command.Connection = conn;

command.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txtMaNhanVien.Text;

command.Parameters.Add("@SoNgayLamViec", SqlDbType.Int).Value = Int32.Parse(txtSNLV.Text);

command.Parameters.Add("@Thuong", SqlDbType.Decimal).Value = decimal.Parse(txtThuong.Text);

1. **Danh mục hiển thị**

**Hiển thị thanh toán của cửa hàng từ việc nhập sách**

-SQL:

CREATE FUNCTION fn\_PaymentStatistics ()

RETURNS TABLE

AS

RETURN

(

SELECT

CONVERT(date, pn.NgayNhap) AS Ngay,

SUM(ct.SoLuongNhap) AS SoLuongNhap,

SUM(ct.SoLuongNhap \* ct.DonGia) AS TongTien

FROM PhieuNhap pn

JOIN ChiTietNhapSach ct ON pn.MaPhieuNhap = ct.MaPhieuNhap

GROUP BY CONVERT(date, pn.NgayNhap)

)

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = " SELECT \* FROM fn\_PaymentStatistics()";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvPhieuNhap.Items.Clear();

while (reader.Read())

{

DateTime Ngay = reader.GetDateTime(0);

string SoLuongNhap = reader.GetInt32(1).ToString();

string TongTien = reader.GetInt32(2).ToString();

ListViewItem lvi = new ListViewItem(Ngay.ToString("yyyy-MM-dd"));

lvi.SubItems.Add(SoLuongNhap);

lvi.SubItems.Add(TongTien);

lsvPhieuNhap.Items.Add(lvi);

}

**Hiển thị doanh thu cửa hàng từ việc bán sách**

-SQL:

CREATE FUNCTION dbo.ThongKeHoaDon()

RETURNS TABLE

AS

RETURN

(

SELECT

CONVERT(DATE, hd.ThoiGian) AS Ngay,

SUM(ct.SoLuong) AS SoLuongSachBan,

SUM(ct.SoLuong \* ct.DonGia) AS TongTien

FROM HoaDon hd

INNER JOIN ChiTietHoaDon ct ON hd.MaHoaDon = ct.MaHoaDon

GROUP BY CONVERT(DATE, hd.ThoiGian)

)

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = " SELECT \* FROM dbo.ThongKeHoaDon() ";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvDoanhThu.Items.Clear();

while (reader.Read())

{

DateTime Ngay = reader.GetDateTime(0);

string SoLuongSachBan = reader.GetInt32(1).ToString() string TongTien = reader.GetInt32(2).ToString();

ListViewItem lvi = new ListViewItem(Ngay.ToString("yyyy-MM-dd"));

lvi.SubItems.Add(SoLuongSachBan);

lvi.SubItems.Add(TongTien);

lsvDoanhThu.Items.Add(lvi);

}

**Hiển thị Thông tin Nhân Viên**

-SQL:

CREATE PROCEDURE ShowStaffInformation

@TenDangNhap NVARCHAR(30)

AS

BEGIN

SELECT \* FROM NhanVien INNER JOIN DangNhap ON NhanVien.MaNhanVien = DangNhap.MaNhanVien

WHERE DangNhap.TenDangNhap = @TenDangNhap

END

-C#:

using (SqlCommand command = new SqlCommand("ShowStaffInformation", connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@TenDangNhap", MaTK);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

}

**Hiển thị Chi Tiết Sách**

-SQL:

CREATE FUNCTION ShowBookDetails (@MaSach NVARCHAR(10))

RETURNS TABLE

AS

RETURN

SELECT Sach.MaSach AS MaSach, Sach.TenSach, Sach.TacGia, Sach.NXB, Sach.MaTheLoai, Sach.MaNCC, Sach.SoLuongTrenKe,

Sach.DonGia, KhoSach.SoLuong

FROM Sach

JOIN KhoSach ON Sach.MaSach = KhoSach.MaSach

WHERE Sach.MaSach = @MaSach

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM dbo.ShowBookDetails(@MaSach)";

command.Connection = conn;

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = MaSach;

command.Parameters.Add(parMaSach);

**Hiển thị Chi Tiết Thể Loại**

-SQL:

CREATE PROCEDURE ShowGenreDetails

@MaTheLoai NVARCHAR(10)

AS

SELECT \* FROM TheLoai WHERE TheLoai.MaTheLoai =@MaTheLoai

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowGenreDetails";

command.Connection = conn;

SqlParameter parMaTheLoai = new SqlParameter("@MaTheLoai", SqlDbType.NVarChar);

parMaTheLoai.Value = MaTheLoai;

command.Parameters.Add(parMaTheLoai);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết Vị Trí Sách**

-SQL:

CREATE PROCEDURE ShowBookLocationDetails

@MaViTri NVARCHAR(10)

AS

SELECT \* FROM ViTriSach WHERE ViTriSach.MaViTri =@MaViTri

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowBookLocationDetails";

command.Connection = conn;

SqlParameter parMaViTri = new SqlParameter("@MaViTri", SqlDbType.NVarChar);

parMaViTri.Value = MaViTri;

command.Parameters.Add(parMaViTri);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết Kho**

-SQL:

CREATE PROCEDURE ShowInventoryDetails

@MaSach NVARCHAR(10)

AS

SELECT KhoSach.MaSach, Sach.TenSach, KhoSach.SoLuong FROM KhoSach JOIN Sach ON KhoSach.MaSach = Sach.MaSach

WHERE KhoSach.MaSach=@MaSach

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowInventoryDetails";

command.Connection = conn;

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = MaSach;

command.Parameters.Add(parMaSach);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Khách Hàng**

-SQL:

CREATE FUNCTION ShowCustomerDetails(@MaKhachHang NVARCHAR(10))

RETURNS TABLE

AS

RETURN

SELECT \* FROM KhachHang WHERE MaKhachHang=@MaKhachHang

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM dbo.ShowCustomerDetails(@MaKhachHang)";

command.Connection = conn;

SqlParameter parMaKhachHang = new SqlParameter("@MaKhachHang", SqlDbType.NVarChar);

parMaKhachHang.Value = MaKhachHang;

command.Parameters.Add(parMaKhachHang);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết Nhà Cung Cấp**

-SQL:

CREATE FUNCTION ShowSupplierDetails(@MaNCC NVARCHAR(10))

RETURNS TABLE

AS RETURN

(

SELECT \* FROM NhaCungCap WHERE MaNCC = @MaNCC

)

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM ShowSupplierDetails(@MaNCC)";

command.Connection = conn;

SqlParameter parMaNCC = new SqlParameter("@MaNCC", SqlDbType.NVarChar);

parMaNCC.Value = MaNCC;

command.Parameters.Add(parMaNCC);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Phiếu Nhập**

-SQL:

CREATE PROCEDURE ShowReceipt

@MaPhieuNhap NVARCHAR(10)

AS

SELECT \* FROM PhieuNhap Join ChiTietNhapSach on PhieuNhap.MaPhieuNhap = ChiTietNhapSach.MaPhieuNhap

WHERE PhieuNhap.MaPhieuNhap=@MaPhieuNhap

order by PhieuNhap.MaPhieuNhap

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowReceipt";

command.Connection = conn;

SqlParameter parMaPhieuNhap = new SqlParameter("@MaPhieuNhap", SqlDbType.NVarChar);

parMaPhieuNhap.Value = MaPhieuNhap;

command.Parameters.Add(parMaPhieuNhap);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết Phiếu Nhập**

-SQL:

CREATE PROCEDURE ShowReceiptDetails

@MaPhieuNhap NVARCHAR(10),

@MaSach NVARCHAR(10)

AS

SELECT \* FROM PhieuNhap Join ChiTietNhapSach on PhieuNhap.MaPhieuNhap = ChiTietNhapSach.MaPhieuNhap

WHERE ChiTietNhapSach.MaPhieuNhap=@MaPhieuNhap AND ChiTietNhapSach.MaSach = @MaSach

order by PhieuNhap.MaPhieuNhap

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowReceiptDetails";

command.Connection = conn;

SqlParameter parMaPhieuNhap = new SqlParameter("@MaPhieuNhap", SqlDbType.NVarChar);

parMaPhieuNhap.Value = MaPhieuNhap;

command.Parameters.Add(parMaPhieuNhap);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết Hóa Đơn**

-SQL:

CREATE PROCEDURE ShowAllBills

@MaHoaDon NVARCHAR(10)

AS

SELECT \* FROM HoaDon WHERE HoaDon.MaHoaDon=@MaHoaDon

GO

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowAllBills";

command.Connection = conn;

SqlParameter parMaHoaDon = new SqlParameter("@MaHoaDon", SqlDbType.NVarChar);

parMaHoaDon.Value = maHoaDon;

command.Parameters.Add(parMaHoaDon);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết của Chi Tiết Hóa Đơn**

-SQL:

CREATE PROCEDURE ShowAllBillDetails

@MaHoaDon NVARCHAR(10),

@MaSach NVARCHAR(10)

AS

SELECT \* FROM ChiTietHoaDon WHERE ChiTietHoaDon.MaHoaDon=@MaHoaDon AND MaSach = @MaSach

-C#::

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowAllBillDetails";

command.Connection = conn;

SqlParameter parMaHoaDon = new SqlParameter("@MaHoaDon", SqlDbType.NVarChar);

parMaHoaDon.Value = maHoaDon;

command.Parameters.Add(parMaHoaDon);

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = maSach;

command.Parameters.Add(parMaSach);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Chi Tiết Nhân Viên**

-SQL:

CREATE PROCEDURE ShowStaff

@MaNhanVien NVARCHAR(10)

AS

SELECT \* FROM NhanVien

WHERE MaNhanVien=@MaNhanVien

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowStaff";

command.Connection = conn;

SqlParameter parMaNhanVien = new SqlParameter("@MaNhanVien", SqlDbType.NVarChar);

parMaNhanVien.Value = MaNhanVien;

command.Parameters.Add(parMaNhanVien);

SqlDataReader reader = command.ExecuteReader();

**Hiển thị Lương**

-SQL:

--PROCCEDURE Hiển thị Lương

CREATE PROCEDURE ShowSalary

@MaNhanVien NVARCHAR(10)

AS

SELECT \* FROM Luong

WHERE MaNhanVien=@MaNhanVien

-C#:

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowSalary";

command.Connection = conn;

SqlParameter parMaNhanVien = new SqlParameter("@MaNhanVien", SqlDbType.NVarChar);

parMaNhanVien.Value = MaNhanVien;

command.Parameters.Add(parMaNhanVien);

SqlDataReader reader = command.ExecuteReader();

1. **Danh mục tìm kiếm**

**Tìm kiếm Theo Tên sách và Tác Giả**

-SQL:

CREATE FUNCTION SearchBook

(@TenSach nvarchar(100),

@TacGia nvarchar(100))

RETURNS TABLE

AS

RETURN

SELECT \* FROM Sach

WHERE TenSach LIKE @TenSach OR TacGia LIKE @TacGia

-C#:

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchBook(@TenSach, @TacGia)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@TenSach", "%" + tensach + "%");

command.Parameters.AddWithValue("@TacGia", "%" + tacgia + "%");

**Tìm kiếm Sách trong Kho theo Tên Sách và Tác Giả**

-SQL:

-- PROCEDURE tìm kiếm sách theo Tên sách và tác giả

CREATE FUNCTION SearchInventory

(@TenSach nvarchar(100))

RETURNS TABLE

AS

RETURN

SELECT KhoSach.MaSach, Sach.TenSach, KhoSach.SoLuong

FROM KhoSach JOIN Sach ON KhoSach.MaSach = Sach.MaSach

WHERE TenSach LIKE @TenSach

-C#:

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchInventory(@TenSach)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@TenSach", "%" + tensach + "%");

**Tìm kiếm Thông Tin Khách Hàng theo Tên Khách Hàng và Mã Khách Hàng**

-SQL:

-- FUNCTION SearchCustomer: Tìm kiếm thông tin khách hàng trong hệ thống dựa trên tên

CREATE FUNCTION SearchCustomer

(@MaKhachHang nvarchar(10),

@TenKhachHang nvarchar(100))

RETURNS TABLE

AS

RETURN

SELECT \* FROM KhachHang

WHERE HoTen LIKE @TenKhachHang or MaKhachHang LIKE @MaKhachHang

-C#:

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchCustomer(@MaKhachHang, @TenKhachHang)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@MaKhachHang", "%" + MaKH + "%");

command.Parameters.AddWithValue("@TenKhachHang", "%" + HoTen + "%");

**Tìm kiếm Nhà Cung Cấp qua Tên hoặc Mã Nhà Cung Cấp**

-SQL:

-- FUNCTION SearchSupplier: Tìm kiếm thông tin nhà cung cấp trong hệ thống dựa trên tên

CREATE FUNCTION SearchSupplier

(@TenNCC nvarchar(100),

@MaNCC nvarchar(10))

RETURNS TABLE

AS

RETURN

SELECT \* FROM NhaCungCap

WHERE TenNCC LIKE @TenNCC OR MaNCC LIKE @MaNCC

-C#:

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchSupplier(@TenNCC, @MaNCC)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@TenNCC", "%" + tenncc + "%");

command.Parameters.AddWithValue("@MaNCC", "%" + maNCC + "%");

**Tìm kiếm Thể Loại qua Tên Thể Loại**

-SQL:

-- FUNCTION tìm kiếm thông tin dựa trên thể loại

CREATE FUNCTION SearchGenre (@TenTheLoai NVARCHAR(100) ) RETURNS TABLE

AS

RETURN

SELECT \* FROM TheLoai

WHERE TenTheLoai LIKE @TenTheLoai

-C#:

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchGenre(@TenTheLoai)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@TenTheLoai", "%" + tentheloai + "%");

**Tìm kiếm Vị Trí Sách**

-SQL:

-- PROCEDURE tìm kiếm vị trí sách

CREATE FUNCTION SearchBookLocation

(@MaSach nvarchar(10))

RETURNS TABLE

AS

RETURN

SELECT \* FROM ViTriSach

WHERE MaSach LIKE @MaSach

-C#;

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchBookLocation(@MaSach)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@MaSach", "%" + MaSach + "%");

**Tìm kiếm Phiếu Nhập**

-SQL:

-- PROCEDURE Tìm kiếm Phiếu nhập

CREATE FUNCTION SearchReceipt

(@MaPhieuNhap nvarchar(10))

RETURNS TABLE

AS

RETURN

SELECT \* FROM PhieuNhap

WHERE MaPhieuNhap = @MaPhieuNhap

-C#:

SqlCommand commandPN = new SqlCommand();

commandPN.Connection = conn;

commandPN.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

commandPN.CommandText = "SELECT \* FROM SearchReceipt(@MaPhieuNhap)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

commandPN.Parameters.AddWithValue("@MaPhieuNhap", maphieunhap);

**Tìm kiếm Chi Tiết Phiếu Nhập**

-SQL:

-- PROCEDURE Tìm kiếm chi tiết phiếu nhập

CREATE FUNCTION SearchReceiptDetails

(@MaPhieuNhap nvarchar(10))

RETURNS TABLE

AS

RETURN

SELECT \* FROM ChiTietNhapSach

WHERE MaPhieuNhap = @MaPhieuNhap

-C#:

SqlCommand commandCTPN = new SqlCommand();

commandCTPN.Connection = conn;

commandCTPN.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

commandCTPN.CommandText = "SELECT \* FROM SearchReceiptDetails(@MaPhieuNhap)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

commandCTPN.Parameters.AddWithValue("@MaPhieuNhap", maphieunhap);

**Tìm kiếm Hóa Đơn**

-SQL:

--Tìm kiếm Hóa Đơn

CREATE FUNCTION SearchBill

(@MaHoaDon NVARCHAR(10),

@MaKhachHang NVARCHAR(10))

RETURNS TABLE

AS

RETURN

(

SELECT DISTINCT hd.MaHoaDon, hd.MaNhanVien, hd.MaKhachHang, hd.ThoiGian, hd.ThanhTien

FROM HoaDon hd

JOIN ChiTietHoaDon cthd ON hd.MaHoaDon = cthd.MaHoaDon

WHERE hd.MaHoaDon LIKE '%' + @MaHoaDon + '%' OR hd.MaKhachHang LIKE '%' + @MaKhachHang + '%'

)

-C#:

SqlCommand command = new SqlCommand();

command.Connection = conn;

command.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command.CommandText = "SELECT \* FROM SearchBill(@MaHoaDon, @MaKhachHang)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command.Parameters.AddWithValue("@MaHoaDon", "%" + MaHoaDon + "%");

command.Parameters.AddWithValue("@MaKhachHang", "%" + MaKhachHang + "%");

**Tìm kiếm Chi Tiết Hóa Đơn**

-SQL:

--Tìm kiếm chi tiết Hóa Đơn

CREATE FUNCTION SearchBillDetails

(@MaHoaDon NVARCHAR(10),

@MaKhachHang NVARCHAR(10))

RETURNS TABLE

AS

RETURN

(

SELECT cthd.MaHoaDon, cthd.MaSach, cthd.SoLuong, cthd.DonGia

FROM ChiTietHoaDon cthd

JOIN HoaDon hd ON cthd.MaHoaDon = hd.MaHoaDon

WHERE cthd.MaHoaDon LIKE '%' + @MaHoaDon + '%' OR hd.MaKhachHang LIKE '%' + @MaKhachHang + '%'

)

-C#:

SqlCommand command1 = new SqlCommand();

command1.Connection = conn;

command1.CommandType = CommandType.Text; // Thay đổi kiểu CommandType

command1.CommandText = "SELECT \* FROM SearchBillDetails(@MaHoaDon, @MaKhachHang)"; // Sử dụng câu truy vấn SELECT để lấy dữ liệu từ function

command1.Parameters.AddWithValue("@MaHoaDon", "%" + MaHoaDon + "%");

command1.Parameters.AddWithValue("@MaKhachHang", "%" + MaKhachHang + "%");

1. **Danh mục phân loại**

**Phân Loại Sách theo Mã Thể Loại**

CREATE FUNCTION ClassifiedByGenre(@MaTheLoai NVARCHAR(10))

RETURNS TABLE

AS

RETURN

(

SELECT \* FROM Sach WHERE MaTheLoai = @MaTheLoai

)

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM ClassifiedByGenre(@MaTheLoai)";

command.Connection = conn;

command.Parameters.AddWithValue("@MaTheLoai", cbPhanLoai.Text);

**Phân Loại Sách theo Giá Tiền**

-- PROCEDURE Phân loại sách theo giá tiền đã chọn

CREATE PROCEDURE ClassifiedByPrice

@MinPrice int,

@MaxPrice int

AS

SELECT \* FROM Sach WHERE DonGia >= @MinPrice AND DonGia <= @MaxPrice

GO

**Code Gọi thực thi**

// Lọc dữ liệu theo khoảng giá tiền đã chọn

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedByPrice";

command.Connection = conn;

command.Parameters.AddWithValue("@MinPrice", minPrice);

command.Parameters.AddWithValue("@MaxPrice", maxPrice);

**Phân Loại Sách theo Mã Nhà Cung Cấp**

CREATE FUNCTION ClassifiedBySupplier(@MaNCC NVARCHAR(10))

RETURNS TABLE

AS

RETURN

(

SELECT \* FROM Sach WHERE MaNCC = @MaNCC

)

GO

**Code Gọi thực thi**

// Lọc dữ liệu theo Mã NCC đã chọn

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM ClassifiedBySupplier(@MaNCC)";

command.Connection = conn;

command.Parameters.AddWithValue("@MaNCC", cbPhanLoai.Text);

**Phân Loại Sách theo Nhà Xuất Bản**

CREATE FUNCTION ClassifiedByPublishing(@NXB NVARCHAR(100))

RETURNS TABLE

AS

RETURN

(

SELECT \* FROM Sach WHERE NXB = @NXB

)

GO

**Code Gọi thực thi**

// Lọc dữ liệu theo Mã NCC đã chọn

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM ClassifiedByPublishing(@NXB)";

command.Connection = conn;

command.Parameters.AddWithValue("@NXB", cbPhanLoai.Text);

**Phân Loại Vị Trí Sách Theo Kệ**

-- PROCEDURE Phân loại vị trí sách theo kệ đã chọn

CREATE PROCEDURE ClassifiedByShelf

@Ke int

AS

SELECT \* FROM ViTriSach WHERE Ke = @Ke

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedByShelf";

command.Connection = conn;

command.Parameters.AddWithValue("@Ke", cbPhanLoai.Text);

**Phân Loại Hóa Đơn theo Mã Nhân Viên**

-- PROCEDURE Phân loại hóa đơn theo mã nhân viên

CREATE PROCEDURE ClassifiedBillByStaff

@MaNhanVien NVARCHAR(10)

AS

SELECT \* FROM HoaDon WHERE HoaDon.MaNhanVien = @MaNhanVien

GO

**Code Gọi thực thi**

// Lọc dữ liệu theo Mã NCC đã chọn

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedBillByStaff";

command.Connection = conn;

command.Parameters.AddWithValue("@MaNhanVien", cboPhanLoai.Text);

**Phân Loại Nhân Viên theo Chức Vụ**

--PROCEDURE Phân loại Nhân Viên theo chức vụ đã chọn

CREATE PROCEDURE ClassifiedByDuty

@MaChucVu NVARCHAR(10)

AS

SELECT \* FROM NhanVien

WHERE NhanVien.MaChucVu = @MaChucVu

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedByDuty";

command.Connection = conn;

command.Parameters.AddWithValue("@MaChucVu", cbPhanLoai.Text);

**Phân Loại Nhân Viên theo Tình Trạng Hoạt Động**

--PROCEDURE Phân loại Nhân Viên theo Tình Trạng Hoạt Động đã chọn

CREATE PROCEDURE ClassifiedByActivity

@HoatDong NVARCHAR(100)

AS

SELECT \* FROM NhanVien

JOIN PhanLoaiNhanVien ON NhanVien.MaChucVu = PhanLoaiNhanVien.MaChucVu

JOIN Luong ON NhanVien.MaNhanVien = Luong.MaNhanVien

WHERE NhanVien.HoatDong = @HoatDong

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedByActivity";

command.Connection = conn;

command.Parameters.AddWithValue("@HoatDong", cbPhanLoai.Text);

**Phân Loại Nhân Viên theo Giới Tính**

--PROCEDURE Phân loại Nhân Viên theo Giới tính đã chọn

CREATE PROCEDURE ClassifiedBySex

@GioiTinh nvarchar(10)

AS

SELECT \* FROM NhanVien

JOIN PhanLoaiNhanVien ON NhanVien.MaChucVu = PhanLoaiNhanVien.MaChucVu

JOIN Luong ON NhanVien.MaNhanVien = Luong.MaNhanVien

WHERE NhanVien.GioiTinh = @GioiTinh

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedBySex";

command.Connection = conn;

command.Parameters.AddWithValue("@GioiTinh", cbPhanLoai.Text);

**Phân Loại Nhân Viên**

--PROCCEDURE Hiển thị phân loại nhân viên

CREATE PROCEDURE ShowClassifiedStaff

@MaChucVu NVARCHAR(10)

AS

SELECT \* FROM PhanLoaiNhanVien

WHERE MaChucVu=@MaChucVu

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowClassifiedStaff";

command.Connection = conn;

SqlParameter parMaChucVu = new SqlParameter("@MaChucVu", SqlDbType.NVarChar);

parMaChucVu.Value = MaChucVu;

command.Parameters.Add(parMaChucVu);

SqlDataReader reader = command.ExecuteReader();

**Phân Loại Nhân Viên theo Chức vụ**

----PROCEDURE Phân loại Nhân Viên theo chức vụ đã chọn

CREATE PROCEDURE FindDuty

@TenChucVu NVARCHAR(100)

AS

SELECT \* FROM PhanLoaiNhanVien

WHERE TenChucVu = @TenChucVu

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "FindDuty";

command.Connection = conn;

command.Parameters.AddWithValue("@TenChucVu", cbPhanLoai.Text);

**Phân Loại Nhân Viên theo Lương**

-- PROCEDURE Phân loại theo Lương đã chọn

CREATE PROCEDURE ClassifiedBySalary

@MinPrice int,

@MaxPrice int

AS

SELECT \* FROM PhanLoaiNhanVien WHERE Luong >= @MinPrice AND Luong <= @MaxPrice

GO

**Code Gọi thực thi**

// Lọc dữ liệu theo khoảng giá tiền đã chọn

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedBySalary";

command.Connection = conn;

command.Parameters.AddWithValue("@MinPrice", minPrice);

command.Parameters.AddWithValue("@MaxPrice", maxPrice);

**Phân Loại Nhân Viên theo Số Ngày Làm Việc**

--PROCEDURE Phân loại Nhân Viên theo Số ngày làm việc đã chọn

CREATE PROCEDURE ClassifiedByDayOfWork

@SoNgayLamViec int

AS

SELECT \* FROM Luong

WHERE SoNgayLamViec = @SoNgayLamViec

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedByDayOfWork";

command.Connection = conn;

command.Parameters.AddWithValue("@SoNgayLamViec", cbPhanLoai.Text);

**Phân Loại Nhân Viên theo lương**

-- PROCEDURE Phân loại Nhân Viên theo lương đã chọn

CREATE PROCEDURE ClassifiedBySumSalary

@MinPrice int,

@MaxPrice int

AS

SELECT \* FROM Luong

WHERE TongLuong >= @MinPrice AND TongLuong <= @MaxPrice

GO

**Code Gọi thực thi**

// Lọc dữ liệu theo khoảng giá tiền đã chọn

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ClassifiedBySumSalary";

command.Connection = conn;

command.Parameters.AddWithValue("@MinPrice", minPrice);

command.Parameters.AddWithValue("@MaxPrice", maxPrice);

1. **Danh mục thêm**

**Thêm Sách vào Sách**

-- PROCEDURE thêm sách

CREATE PROCEDURE AddBook

@TenSach nvarchar(100),

@TacGia nvarchar(100),

@NXB nvarchar(100),

@MaTheLoai nvarchar(10),

@MaNCC nvarchar(10),

@SoLuongTrenKe int,

@DonGia decimal(18,2)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaSach) FROM Sach)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'MS' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)), 3)

INSERT INTO Sach (MaSach, TenSach, TacGia, NXB, MaTheLoai, MaNCC, SoLuongTrenKe, DonGia)

VALUES (@NewCode, @TenSach, @TacGia, @NXB, @MaTheLoai, @MaNCC, @SoLuongTrenKe, @DonGia)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddBook";

command.Connection = conn;

command.Parameters.Add("@TenSach", SqlDbType.NVarChar).Value txtTenSach.Text;

command.Parameters.Add("@TacGia", SqlDbType.NVarChar).Value = txtTacGia.Text;

command.Parameters.Add("@NXB", SqlDbType.NVarChar).Value = txtNXB.Text;

command.Parameters.Add("@MaTheLoai", SqlDbType.NVarChar).Value = txtMaTheLoai.Text;

command.Parameters.Add("@MaNCC", SqlDbType.NVarChar).Value = txtMaNCC.Text;

command.Parameters.Add("@SoluongTrenKe", SqlDbType.Int).Value = int.Parse(txtSoLuong.Text);

command.Parameters.Add("@DonGia", SqlDbType.Decimal).Value = decimal.Parse(txtDonGia.Text);

**Thêm Sách vào Kho**

**Thêm Thông Tin Khách Hàng mới**

-- PROCEDURE AddCustomer: Thêm thông tin khách hàng mới vào hệ thống

CREATE PROCEDURE AddCustomer

@MaKhachHang NVARCHAR(10),

@HoTen NVARCHAR(100),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaKhachHang) FROM KhachHang)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'KH' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)), 3)

INSERT INTO KhachHang (MaKhachHang, HoTen, DiaChi, SoDienThoai)

VALUES (@NewCode, @HoTen, @DiaChi, @SoDienThoai)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddCustomer";

command.Connection = conn;

command.Parameters.Add("@MaKhachHang", SqlDbType.NVarChar).Value = txtMaKhachHang.Text;

command.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txtHoTen.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChi.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoai.Text;

**Thêm Thông Tin Nhân Viên Mới**

-- PROCEDURE AddEmployee: Thêm thông tin nhân viên mới vào hệ thống

CREATE PROCEDURE AddEmployee

@HoTen NVARCHAR(100),

@NgaySinh DATE,

@GioiTinh NVARCHAR(10),

@MaChucVu NVARCHAR(10),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20),

@HoatDong NVARCHAR(10)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaNhanVien) FROM NhanVien)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

DECLARE @TongLuong DECIMAL(18,2)

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'NV' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)), 3)

--Thêm vào Nhân Viên

INSERT INTO NhanVien(MaNhanVien, HoTen, NgaySinh, GioiTinh, MaChucVu, DiaChi, SoDienThoai,HoatDong)

VALUES (@NewCode, @HoTen, @NgaySinh, @GioiTinh, @MaChucVu, @DiaChi, @SoDienThoai,@HoatDong)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddEmployee";

command.Connection = conn;

command.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txtHoTen.Text;

command.Parameters.Add("@NgaySinh", SqlDbType.DateTime).Value = DateTime.ParseExact(txtNgaySinh.Text, "yyyy/MM/dd", CultureInfo.InvariantCulture);

command.Parameters.Add("@GioiTinh", SqlDbType.NVarChar).Value = txtGioiTinh.Text;

command.Parameters.Add("@MaChucVu", SqlDbType.NVarChar).Value = txtMaChucVu.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChi.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoai.Text;

command.Parameters.Add("@HoatDong", SqlDbType.NVarChar).Value = txtHoatDong.Text;

**Thêm Thông Tin Nhà Cung Cấp mới**

-- PROCEDURE AddSupplier: Thêm thông tin nhà cung cấp mới vào hệ thống

CREATE PROCEDURE AddSupplier

@MaNCC NVARCHAR(10),

@TenNCC NVARCHAR(100),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaNCC) FROM NhaCungCap)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'NCC' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)), 3)

INSERT INTO NhaCungCap(MaNCC, TenNCC, DiaChi, SoDienThoai)

VALUES (@NewCode, @TenNCC, @DiaChi, @SoDienThoai)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddSupplier";

command.Connection = conn;

command.Parameters.Add("@MaNCC", SqlDbType.NVarChar).Value = txtMaNCC.Text;

command.Parameters.Add("@TenNCC", SqlDbType.NVarChar).Value = txtTenNCC.Text;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txtDiaChiNCC.Text;

command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar).Value = txtSoDienThoaiNCC.Text;

**Thêm Thông Tin Phiếu Nhập**

-- PROCEDURE thêm phiếu nhập

CREATE PROCEDURE AddReceipt

@MaNCC NVARCHAR(10)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaPhieuNhap) FROM PhieuNhap)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'PN' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)), 3)

INSERT INTO PhieuNhap (MaPhieuNhap, MaNCC, NgayNhap, ThanhTien)

VALUES (@NewCode, @MaNCC, GETDATE(), 0)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddReceipt";

command.Connection = conn;

command.Parameters.Add("@MaNCC", SqlDbType.NVarChar).Value = txtMaNCC.Text;

**Thêm Thông Tin Thể Loại Mới**

-- PROCEDURE Thêm thể loại mới vào hệ thống

CREATE PROCEDURE AddGenre

@MaTheLoai NVARCHAR(10),

@TenTheLoai NVARCHAR(100),

@MoTa NVARCHAR(255)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaTheLoai) FROM TheLoai)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'TL' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)), 3)

INSERT INTO TheLoai (MaTheLoai, TenTheLoai, MoTa )

VALUES

(@NewCode, @TenTheLoai, @MoTa)

END

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddGenre";

command.Connection = conn;

command.Parameters.Add("@MaTheLoai", SqlDbType.NVarChar).Value = txtMaTheLoai.Text;

command.Parameters.Add("@TenTheLoai", SqlDbType.NVarChar).Value = txtTenTheLoai.Text;

command.Parameters.Add("@MoTa", SqlDbType.NVarChar).Value = txtMoTa.Text;

**Thêm Thông Tin Chi Tiết Phiếu Nhập**

-- PROCEDURE Thêm chi tiết Phiếu Nhập

CREATE PROCEDURE AddReceiptDetails

@MaPhieuNhap NVARCHAR(10),

@MaSach NVARCHAR(10),

@SoLuongNhap INT,

@DonGia int

AS

BEGIN

INSERT INTO ChiTietNhapSach (MaPhieuNhap, MaSach, SoLuongNhap, DonGia)

VALUES (@MaPhieuNhap, @MaSach, @SoLuongNhap , @DonGia)

UPDATE PhieuNhap

SET ThanhTien = (

SELECT SUM(SoLuongNhap \* DonGia)

FROM ChiTietNhapSach

WHERE PhieuNhap.MaPhieuNhap = ChiTietNhapSach.MaPhieuNhap)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddReceiptDetails";

command.Connection = conn;

command.Parameters.Add("@MaPhieuNhap", SqlDbType.NVarChar).Value = txtMaPNChiTiet.Text;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command.Parameters.Add("@SoLuongNhap", SqlDbType.Int).Value = int.Parse(txtSoLuongNhap.Text);

command.Parameters.Add("@DonGia", SqlDbType.Int).Value = int.Parse(txtDonGia.Text);

**Thêm Thông Tin Hóa Đơn**

--Thêm Hóa Đơn

CREATE PROCEDURE AddBill

@MaNhanVien NVARCHAR(50),

@MaKhachHang NVARCHAR(50)

AS

BEGIN

DECLARE @MaHoaDon NVARCHAR(50)

DECLARE @MaHoaDonMax NVARCHAR(50)

SELECT @MaHoaDonMax = MAX(MaHoaDon) FROM HoaDon

-- Tăng giá trị mã hóa đơn lên 1

SET @MaHoaDon = 'HD' + RIGHT('000' + CAST(CAST(SUBSTRING(@MaHoaDonMax, 3, 3) AS INT)

+ 1 AS NVARCHAR(50)), 3)

-- Lấy thời gian thực

DECLARE @ThoiGian DATETIME = GETDATE()

INSERT INTO HoaDon (MaHoaDon, MaNhanVien, MaKhachHang, ThoiGian, ThanhTien)

VALUES (@MaHoaDon, @MaNhanVien, @MaKhachHang, @ThoiGian, 0)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddBill";

command.Connection = conn;

command.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txtMaNhanVien.Text;

command.Parameters.Add("@MaKhachHang", SqlDbType.NVarChar).Value = txtMaKhachHang.Text;

**Thêm Thông Tin Chi Tiết Hóa Đơn**

-- PROCEDURE Thêm chi tiết Hóa Đơn

CREATE PROCEDURE AddBillDetails

@MaHoaDon NVARCHAR(10),

@MaSach NVARCHAR(10),

@SoLuong INT

AS

BEGIN

-- Lấy đơn giá từ bảng Sách

DECLARE @DonGia DECIMAL(18, 2)

SELECT @DonGia = DonGia FROM Sach WHERE MaSach = @MaSach

INSERT INTO ChiTietHoaDon (MaHoaDon, MaSach, SoLuong, DonGia)

VALUES (@MaHoaDon, @MaSach, @SoLuong , @DonGia)

UPDATE HoaDon

SET ThanhTien = (

SELECT SUM(SoLuong \* DonGia)

FROM ChiTietHoaDon

WHERE HoaDon.MaHoaDon = ChiTietHoaDon.MaHoaDon)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddBillDetails";

command.Connection = conn;

command.Parameters.Add("@MaHoaDon", SqlDbType.NVarChar).Value = txtMaHoaDonCT.Text;

command.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txtMaSach.Text;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = int.Parse(txtSoLuong.Text);

**Thêm Thông Tin Phân Loại Nhân Viên Mới**

-- PROCEDURE AddClassifiedStaff: Thêm thông tin nhân viên mới vào hệ thống

CREATE PROCEDURE AddClassifiedStaff

@MaChucVu NVARCHAR(10),

@TenChucVu NVARCHAR(50),

@MoTaCongViec NVARCHAR(200),

@Luong DECIMAL(18,2)

AS

BEGIN

DECLARE @OldCode VARCHAR(10) = (SELECT MAX(MaChucVu) FROM PhanLoaiNhanVien)

DECLARE @NewCode VARCHAR(10)

DECLARE @Suffix INT

SET @Suffix = CAST(RIGHT(@OldCode, 3) AS INT) + 1

SET @NewCode = 'CV' + RIGHT('00' + CAST(@Suffix AS VARCHAR(3)),3)

--Thêm vào Phân Loại Nhân Viên

INSERT INTO PhanLoaiNhanVien (MaChucVu, TenChucVu, MoTaCongViec, Luong)

VALUES (@NewCode, @TenChucVu, @MoTaCongViec, @Luong)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddClassifiedStaff";

command.Connection = conn;

command.Parameters.Add("@MaChucVu", SqlDbType.NVarChar).Value = txtMaChucVu.Text;

command.Parameters.Add("@TenChucVu", SqlDbType.NVarChar).Value = txtTenChucVu.Text;

command.Parameters.Add("@MoTaCongViec", SqlDbType.NVarChar).Value = txtMoTaCongViec.Text;

command.Parameters.Add("@Luong", SqlDbType.Decimal).Value = decimal.Parse(txtLuong.Text);

**Thêm Thông Tin Lương Nhân Viên Mới**

-- PROCEDURE AddSalary: Thêm thông tin nhân viên mới vào hệ thống

CREATE PROCEDURE AddSalary

@MaNhanVien NVARCHAR(10),

@SoNgayLamViec INT ,

@Thuong DECIMAL(18,2)

AS

BEGIN

--Thêm vào Lương

INSERT INTO Luong(MaNhanVien, SoNgayLamViec, Thuong)

VALUES (@MaNhanVien, @SoNgayLamViec, @Thuong)

UPDATE Luong

SET TongLuong = (

SELECT (Luong / 26 \* SoNgayLamViec + Thuong)

FROM PhanLoaiNhanVien join NhanVien on PhanLoaiNhanVien.MaChucVu = NhanVien.MaChucVu

WHERE NhanVien.MaNhanVien = Luong.MaNhanVien)

END

GO

**Code Gọi thực thi**

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "AddSalary";

command.Connection = conn;

command.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txtMaNhanVien.Text;

command.Parameters.Add("@SoNgayLamViec", SqlDbType.Int).Value = Int32.Parse(txtSNLV.Text);

command.Parameters.Add("@Thuong", SqlDbType.Decimal).Value = decimal.Parse(txtThuong.Text);

# **CHƯƠNG 4: THIẾT KẾ GIAO DIỆN**

## **Môi trường cài đặt và công nghệ sử dụng**

- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 18.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.

## **Giao diện và chức năng của chương trình:**

### **Form đăng nhập**

**A screenshot of a login screen

Description automatically generated with medium confidence**

private void KetNoiVoiCSDL()

{

username = this.txtUserName.Text;

password = this.txtPassWord.Text;

try

{

string strCon = $"Data Source=(local);Initial Catalog=QLCHBS;" +

$"User ID={username};" +

$"Password={password};";

sqlCon = new SqlConnection(strCon);

sqlCon.Open();

fQuanLy f = new fQuanLy(username, password);

this.Hide();

f.ShowDialog();

this.Show();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi rồi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnThoat\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void btnDangNhap\_Click(object sender, EventArgs e)

{

try

{

KetNoiVoiCSDL();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi kết nối đến cơ sở dữ liệu: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

sqlCon.Close();

}

}

Hệ thống sẽ sử dụng username và password nhân viên nhập vào để login vào

Database

### **Form quên mật khẩu**

**A screenshot of a login box

Description automatically generated with medium confidence**

private void btnDoiMatKhau\_Click(object sender, EventArgs e)

{

string connectionString = "Data Source=(local);Initial Catalog=QLCHBS;Integrated Security=True";

// Lấy dữ liệu từ các textbox

string tendangnhap = txtForgotUserName.Text.Trim();

string sdt = txtPhoneNumber.Text.Trim();

string matKhauMoi = txtForgorPW.Text.Trim();

string xacNhanMatKhau = txtConfirmPW.Text.Trim();

int count = 0;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

using (SqlCommand command = new SqlCommand("CheckAccount\_Phone", connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@SDT", sdt);

command.Parameters.AddWithValue("@TenDangNhap", tendangnhap);

count = (int)command.ExecuteScalar();

}

}

// Nếu tồn tại, kiểm tra xem MatKhauMoi và XacNhanMatKhau có trùng nhau không

if (count > 0)

{

if (matKhauMoi.Equals(xacNhanMatKhau))

{

//query = "UPDATE DangNhap SET MatKhau = @MatKhauMoi WHERE TenDangNhap = @TenDangNhap";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

using (SqlCommand command = new SqlCommand("UpdateAccount", connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MatKhauMoi", matKhauMoi);

command.Parameters.AddWithValue("@TenDangNhap", tendangnhap);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Đổi mật khẩu thành công.");

txtForgotUserName.Clear();

txtPhoneNumber.Clear();

txtForgorPW.Clear();

txtConfirmPW.Clear();

}

else

{

MessageBox.Show("Đổi mật khẩu thất bại.");

}

}

}

}

else

{

MessageBox.Show("Mật khẩu mới và xác nhận mật khẩu không trùng nhau.");

}

}

else

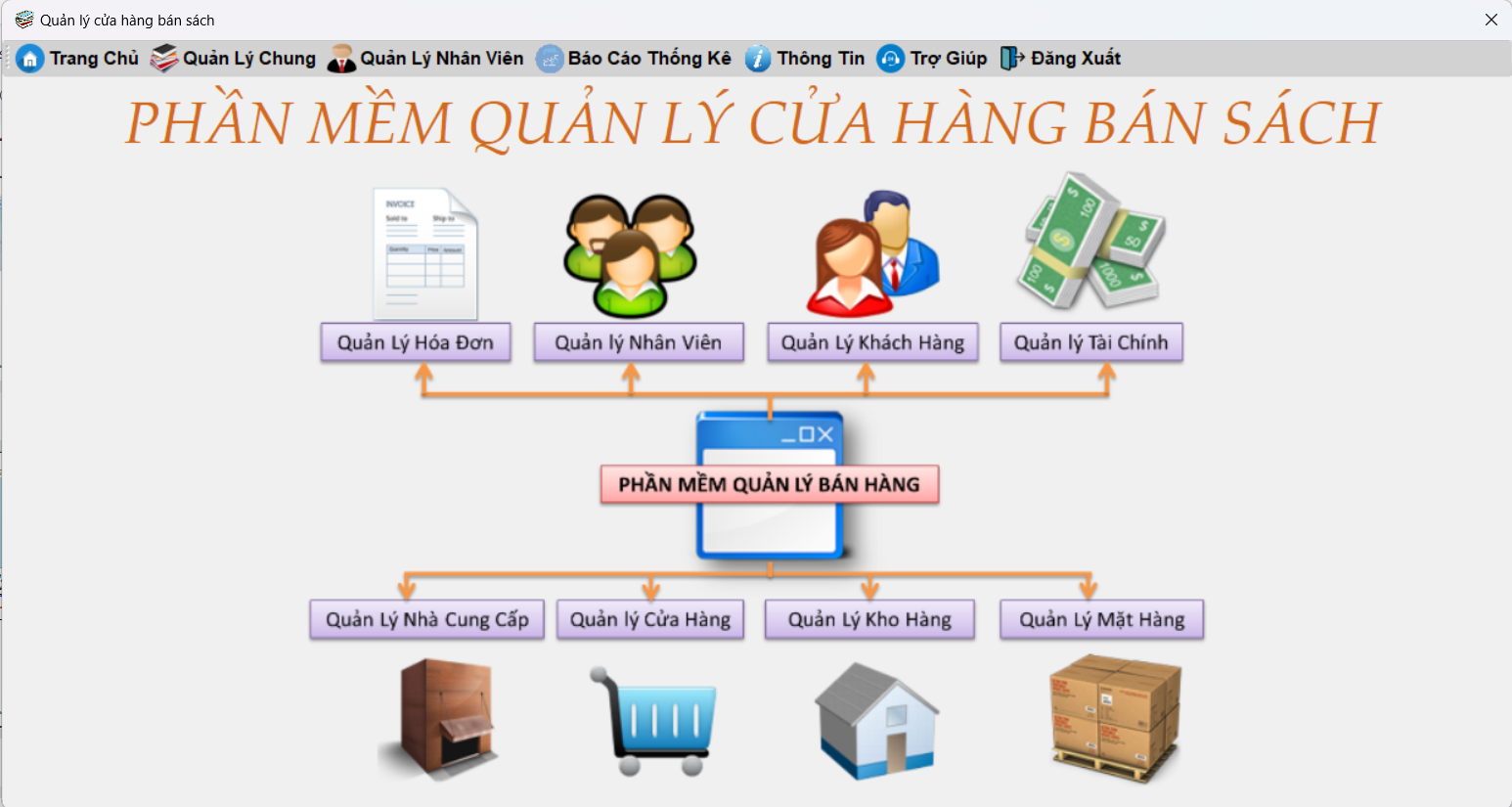
{

MessageBox.Show("Tài khoản hoặc số điện thoại không đúng.");

}

}

**Form home**



### **Form quản lí chung**

#### **3.1 Form sách**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoSach()

{

try

{

if (conn == null)

{

conn = new SqlConnection(connectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Book";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvSach.Items.Clear();

while (reader.Read())

{

string MaSach = reader.GetString(0);

string TenSach = reader.GetString(1);

string TacGia = reader.GetString(2);

string NXB = reader.GetString(3);

string MaTheLoai = reader.GetString(4);

string MaNCC = reader.GetString(5);

int SoLuongTrenKe = reader.GetInt32(6);

decimal DonGia = reader.GetDecimal(7);

ListViewItem lvi = new ListViewItem(MaSach);

lvi.SubItems.Add(TenSach);

lvi.SubItems.Add(TacGia);

lvi.SubItems.Add(NXB);

lvi.SubItems.Add(MaTheLoai);

lvi.SubItems.Add(MaNCC);

lvi.SubItems.Add(SoLuongTrenKe.ToString());

lvi.SubItems.Add(DonGia.ToString());

lsvSach.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvSach\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvSach.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvSach.SelectedItems[0];

string MaSach = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoSach(MaSach);

}

private void HienThiChiTietToanBoSach(string MaSach)

{

try

{

if (conn == null)

{

conn = new SqlConnection(connectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM dbo.ShowBookDetails(@MaSach)";

command.Connection = conn;

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = MaSach;

command.Parameters.Add(parMaSach);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaSach.Text = reader.GetString(0) + "";

txtTenSach.Text = reader.GetString(1) + "";

txtTacGia.Text = reader.GetString(2) + "";

txtNXB.Text = reader.GetString(3) + "";

txtMaTheLoai.Text = reader.GetString(4) + "";

txtMaNCC.Text = reader.GetString(5) + "";

txtSoLuong.Text = reader.GetInt32(6).ToString() + "";

txtDonGia.Text = reader.GetDecimal(7).ToString() + "";

txtSoLuongKho.Text = reader.GetInt32(8).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.2 Form vị trí sách**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoVTS()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_BookLocation";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvVTS.Items.Clear();

while (reader.Read())

{

string MaViTri = reader.GetString(0);

string MaSach = reader.GetString(1);

string Ke = reader.GetInt32(2).ToString();

string Tang = reader.GetInt32(3).ToString();

string Ngan = reader.GetInt32(4).ToString();

ListViewItem lvi = new ListViewItem(MaViTri);

lvi.SubItems.Add(MaSach);

lvi.SubItems.Add(Ke);

lvi.SubItems.Add(Tang);

lvi.SubItems.Add(Ngan);

lsvVTS.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvVTS\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvVTS.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvVTS.SelectedItems[0];

string MaViTri = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoVTS(MaViTri);

}

private void HienThiChiTietToanBoVTS(string MaViTri)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowBookLocationDetails";

command.Connection = conn;

SqlParameter parMaViTri = new SqlParameter("@MaViTri", SqlDbType.NVarChar);

parMaViTri.Value = MaViTri;

command.Parameters.Add(parMaViTri);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaViTri.Text = reader.GetString(0) + "";

txtMaSach.Text = reader.GetString(1) + "";

txtKe.Text = reader.GetInt32(2).ToString() + "";

txtTang.Text = reader.GetInt32(3).ToString() + "";

txtNgan.Text = reader.GetInt32(4).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.3 Form thể loại**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoTheLoai()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Genre";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvTheLoai.Items.Clear();

while (reader.Read())

{

string MaTheLoai = reader.GetString(0);

string TenTheLoai = reader.GetString(1);

string MoTa = reader.GetString(2);

ListViewItem lvi = new ListViewItem(MaTheLoai);

lvi.SubItems.Add(TenTheLoai);

lvi.SubItems.Add(MoTa);

lsvTheLoai.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvTheLoai\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvTheLoai.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvTheLoai.SelectedItems[0];

string MaTheLoai = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoTheLoai(MaTheLoai);

}

private void HienThiChiTietToanBoTheLoai(string MaTheLoai)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowGenreDetails";

command.Connection = conn;

SqlParameter parMaTheLoai = new SqlParameter("@MaTheLoai", SqlDbType.NVarChar);

parMaTheLoai.Value = MaTheLoai;

command.Parameters.Add(parMaTheLoai);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaTheLoai.Text = reader.GetString(0) + "";

txtTenTheLoai.Text = reader.GetString(1) + "";

txtMoTa.Text = reader.GetString(2) + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.4 Form kho sách**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoKhoSach()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Inventory";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvKhoSach.Items.Clear();

while (reader.Read())

{

string MaSach = reader.GetString(0);

string TenSach = reader.GetString(1);

int SoLuongKho = reader.GetInt32(2);

ListViewItem lvi = new ListViewItem(MaSach);

lvi.SubItems.Add(TenSach);

lvi.SubItems.Add(SoLuongKho.ToString());

lsvKhoSach.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvKhoSach\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvKhoSach.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvKhoSach.SelectedItems[0];

string MaSach = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoKhoSach(MaSach);

}

private void HienThiChiTietToanBoKhoSach(string MaSach)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowInventoryDetails";

command.Connection = conn;

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = MaSach;

command.Parameters.Add(parMaSach);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaSach.Text = reader.GetString(0) + "";

txtTenSach.Text = reader.GetString(1) + "";

txtSoLuong.Text = reader.GetInt32(2).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.5 Form khách hàng**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoKH()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Customer";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvKhachHang.Items.Clear();

while (reader.Read())

{

string MaKhachHang = reader.GetString(0);

string HoTen = reader.GetString(1);

string DiaChi = reader.GetString(2);

string SoDienThoai = reader.GetString(3);

ListViewItem lvi = new ListViewItem(MaKhachHang);

lvi.SubItems.Add(HoTen);

lvi.SubItems.Add(DiaChi);

lvi.SubItems.Add(SoDienThoai);

lsvKhachHang.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvKhachHang\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvKhachHang.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvKhachHang.SelectedItems[0];

string MaKhachHang = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoKH(MaKhachHang);

}

private void HienThiChiTietToanBoKH(string MaKhachHang)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM dbo.ShowCustomerDetails(@MaKhachHang)";

command.Connection = conn;

SqlParameter parMaKhachHang = new SqlParameter("@MaKhachHang", SqlDbType.NVarChar);

parMaKhachHang.Value = MaKhachHang;

command.Parameters.Add(parMaKhachHang);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaKhachHang.Text = reader.GetString(0) + "";

txtHoTen.Text = reader.GetString(1) + "";

txtDiaChi.Text = reader.GetString(2) + "";

txtSoDienThoai.Text = reader.GetString(3) + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.6 Form nhà cung cấp**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoNCC()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Supplier";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvNCC.Items.Clear();

while (reader.Read())

{

string MaNCC = reader.GetString(0);

string TenNCC = reader.GetString(1);

string DiaChi = reader.GetString(2);

string SoDienThoai = reader.GetString(3);

ListViewItem lvi = new ListViewItem(MaNCC);

lvi.SubItems.Add(TenNCC);

lvi.SubItems.Add(DiaChi);

lvi.SubItems.Add(SoDienThoai);

lsvNCC.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvNCC\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvNCC.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvNCC.SelectedItems[0];

string MaNCC = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoNCC(MaNCC);

}

private void HienThiChiTietToanBoNCC(string MaNCC)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM ShowSupplierDetails(@MaNCC)";

command.Connection = conn;

SqlParameter parMaNCC = new SqlParameter("@MaNCC", SqlDbType.NVarChar);

parMaNCC.Value = MaNCC;

command.Parameters.Add(parMaNCC);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaNCC.Text = reader.GetString(0) + "";

txtTenNCC.Text = reader.GetString(1) + "";

txtDiaChiNCC.Text = reader.GetString(2) + "";

txtSoDienThoaiNCC.Text = reader.GetString(3) + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.7 Form hóa đơn**

A screenshot of a computer

Description automatically generated

private void HienThiToanBoHoaDon()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Bill";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvHoaDon.Items.Clear();

while (reader.Read())

{

string MaHoaDon = reader.GetString(0);

string MaNhanVien = reader.GetString(1);

string MaKhachHang = reader.GetString(2);

DateTime ThoiGian = reader.GetDateTime(3);

Decimal ThanhTien = reader.GetDecimal(4);

ListViewItem lvi = new ListViewItem(MaHoaDon);

lvi.SubItems.Add(MaNhanVien);

lvi.SubItems.Add(MaKhachHang);

lvi.SubItems.Add(ThoiGian.ToString());

lvi.SubItems.Add(ThanhTien.ToString());

lsvHoaDon.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

};

}

private void HienThiToanBoChiTietHoaDon()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_BillDetails";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvChiTietHoaDon.Items.Clear();

while (reader.Read())

{

string MaHoaDon = reader.GetString(0);

string MaSach = reader.GetString(1);

int SoLuong = reader.GetInt32(2);

int DonGia = reader.GetInt32(3);

ListViewItem lvi = new ListViewItem(MaHoaDon);

lvi.SubItems.Add(MaSach);

lvi.SubItems.Add(SoLuong.ToString());

lvi.SubItems.Add(DonGia.ToString());

lsvChiTietHoaDon.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvHoaDon\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvHoaDon.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvHoaDon.SelectedItems[0];

string MaHoaDon = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoHoaDon(MaHoaDon);

}

private void HienThiChiTietToanBoHoaDon(string maHoaDon)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowAllBills";

command.Connection = conn;

SqlParameter parMaHoaDon = new SqlParameter("@MaHoaDon", SqlDbType.NVarChar);

parMaHoaDon.Value = maHoaDon;

command.Parameters.Add(parMaHoaDon);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaHoaDon.Text = reader.GetString(0) + "";

txtMaNhanVien.Text = reader.GetString(1) + "";

txtMaKhachHang.Text = reader.GetString(2) + "";

txtThoiGian.Text = reader.GetDateTime(3).ToString() + "";

txtThanhTien.Text = reader.GetDecimal(4).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvChiTietHoaDon\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvChiTietHoaDon.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvChiTietHoaDon.SelectedItems[0];

string MaHoaDon = lvi.SubItems[0].Text.Trim();

string MaSach = lvi.SubItems[1].Text.Trim();

HienThiChiTietToanBoChiTietHoaDon(MaHoaDon, MaSach);

}

private void HienThiChiTietToanBoChiTietHoaDon(string maHoaDon, string maSach) //trung

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowAllBillDetails";

command.Connection = conn;

SqlParameter parMaHoaDon = new SqlParameter("@MaHoaDon", SqlDbType.NVarChar);

parMaHoaDon.Value = maHoaDon;

command.Parameters.Add(parMaHoaDon);

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = maSach;

command.Parameters.Add(parMaSach);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaHoaDonCT.Text = reader.GetString(0) + "";

txtMaSach.Text = reader.GetString(1);

txtSoLuong.Text = reader.GetInt32(2).ToString();

txtDonGia.Text = reader.GetInt32(3).ToString();

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **3.8 Form phiếu nhập**

A screenshot of a computer

Description automatically generated

void HienThiAllPhieuNhap()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Receipt";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvPhieuNhap.Items.Clear();

while (reader.Read())

{

string MaPhieuNhap = reader.GetString(0);

string MaNCC = reader.GetString(1);

DateTime NgayNhap = reader.GetDateTime(2);

string ThanhTien = reader.GetDecimal(3).ToString();

ListViewItem lvi = new ListViewItem(MaPhieuNhap);

lvi.SubItems.Add(MaNCC);

lvi.SubItems.Add(NgayNhap.ToString("yyyy-MM-dd HH:mm:ss"));

lvi.SubItems.Add(ThanhTien);

lsvPhieuNhap.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

void HienThiAllChiTietPhieuNhap()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_ReceiptDetails";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvChiTietPhieuNhap.Items.Clear();

while (reader.Read())

{

string MaPhieuNhap = reader.GetString(0);

string MaSach = reader.GetString(1);

string SoLuongNhap = reader.GetInt32(2).ToString();

string DonGia = reader.GetInt32(3).ToString();

ListViewItem lvi = new ListViewItem(MaPhieuNhap);

lvi.SubItems.Add(MaSach);

lvi.SubItems.Add(SoLuongNhap);

lvi.SubItems.Add(DonGia);

lsvChiTietPhieuNhap.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvPhieuNhap\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvPhieuNhap.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvPhieuNhap.SelectedItems[0];

string MaPhieuNhap = lvi.SubItems[0].Text.Trim();

HienThiChiTietCuaPhieuNhap(MaPhieuNhap);

}

private void lsvChiTietPhieuNhap\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvChiTietPhieuNhap.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvChiTietPhieuNhap.SelectedItems[0];

string MaPhieuNhap = lvi.SubItems[0].Text.Trim();

string MaSach = lvi.SubItems[1].Text.Trim();

HienThiChiTietCuaChiTietPhieuNhap(MaPhieuNhap, MaSach);

}

void HienThiChiTietCuaPhieuNhap(string MaPhieuNhap)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowReceipt";

command.Connection = conn;

SqlParameter parMaPhieuNhap = new SqlParameter("@MaPhieuNhap", SqlDbType.NVarChar);

parMaPhieuNhap.Value = MaPhieuNhap;

command.Parameters.Add(parMaPhieuNhap);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaPhieuNhap.Text = reader.GetString(0) + "";

txtMaNCC.Text = reader.GetString(1) + "";

txtNgayNhap.Text = reader.GetDateTime(2).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

void HienThiChiTietCuaChiTietPhieuNhap(string MaPhieuNhap, string maSach)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowReceiptDetails";

command.Connection = conn;

SqlParameter parMaPhieuNhap = new SqlParameter("@MaPhieuNhap", SqlDbType.NVarChar);

parMaPhieuNhap.Value = MaPhieuNhap;

command.Parameters.Add(parMaPhieuNhap);

SqlParameter parMaSach = new SqlParameter("@MaSach", SqlDbType.NVarChar);

parMaSach.Value = maSach;

command.Parameters.Add(parMaSach);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaPNChiTiet.Text = reader.GetString(4) + "";

txtMaSach.Text = reader.GetString(5) + "";

txtSoLuongNhap.Text = reader.GetInt32(6).ToString() + "";

txtDonGia.Text = reader.GetInt32(7).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

### **Form quản lí nhân viên**

#### **4.1 Form thông tin nhân viên**

A screenshot of a computer

Description automatically generated

void HienThiToanBoNhanVien()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Staff";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvNhanVien.Items.Clear();

while (reader.Read())

{

string MaNhanVien = reader.GetString(0);

string HoTen = reader.GetString(1);

string NgaySinh = reader.GetDateTime(2).ToString("yyyy-MM-dd");

string GioiTinh = reader.GetString(3);

string MaChucVu = reader.GetString(4);

string DiaChi = reader.GetString(5);

string SoDienThoai = reader.GetString(6);

string HoatDong = reader.GetString(7);

ListViewItem lvi = new ListViewItem(MaNhanVien);

lvi.SubItems.Add(HoTen);

lvi.SubItems.Add(NgaySinh);

lvi.SubItems.Add(GioiTinh);

lvi.SubItems.Add(MaChucVu);

lvi.SubItems.Add(DiaChi);

lvi.SubItems.Add(SoDienThoai);

lvi.SubItems.Add(HoatDong);

lsvNhanVien.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvNhanVien\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvNhanVien.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvNhanVien.SelectedItems[0];

string MaNhanVien = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoNV(MaNhanVien);

}

void HienThiChiTietToanBoNV(string MaNhanVien)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowStaff";

command.Connection = conn;

SqlParameter parMaNhanVien = new SqlParameter("@MaNhanVien", SqlDbType.NVarChar);

parMaNhanVien.Value = MaNhanVien;

command.Parameters.Add(parMaNhanVien);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaNhanVien.Text = reader.GetString(0) + "";

txtHoTen.Text = reader.GetString(1) + "";

txtNgaySinh.Text = reader.GetDateTime(2).ToString() + "";

txtGioiTinh.Text = reader.GetString(3) + "";

txtMaChucVu.Text = reader.GetString(4) + "";

txtDiaChi.Text = reader.GetString(5) + "";

txtSoDienThoai.Text = reader.GetString(6) + "";

txtHoatDong.Text = reader.GetString(7) + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

#### **4.2 Form phân loại nhân viên**

A screenshot of a computer

Description automatically generated

void HienThiPhanLoaiNhanVien()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_ClassifiedStaff";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvPhanLoaiNhanVien.Items.Clear();

while (reader.Read())

{

string MaChucVu = reader.GetString(0);

string TenChucVu = reader.GetString(1);

string MoTaCongViec = reader.GetString(2);

string Luong = reader.GetDecimal(3).ToString();

ListViewItem lvi = new ListViewItem(MaChucVu);

lvi.SubItems.Add(TenChucVu);

lvi.SubItems.Add(MoTaCongViec);

lvi.SubItems.Add(Luong);

lsvPhanLoaiNhanVien.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

void HienThiToanBoDangNhap()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Account";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvDangNhap.Items.Clear();

while (reader.Read())

{

string MaNhanVien = reader.GetString(0);

string TaiKhoan = reader.GetString(1);

string MatKhau = reader.GetString(2);

ListViewItem lvi = new ListViewItem(MaNhanVien);

lvi.SubItems.Add(TaiKhoan);

lvi.SubItems.Add(MatKhau);

lsvDangNhap.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvPhanLoaiNhanVien\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvPhanLoaiNhanVien.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvPhanLoaiNhanVien.SelectedItems[0];

string MaChucVu = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoPLNV(MaChucVu);

}

void HienThiChiTietToanBoPLNV(string MaChucVu)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowClassifiedStaff";

command.Connection = conn;

SqlParameter parMaChucVu = new SqlParameter("@MaChucVu", SqlDbType.NVarChar);

parMaChucVu.Value = MaChucVu;

command.Parameters.Add(parMaChucVu);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaChucVu.Text = reader.GetString(0) + "";

txtTenChucVu.Text = reader.GetString(1) + "";

txtMoTaCongViec.Text = reader.GetString(2) + "";

txtLuong.Text = reader.GetDecimal(3).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

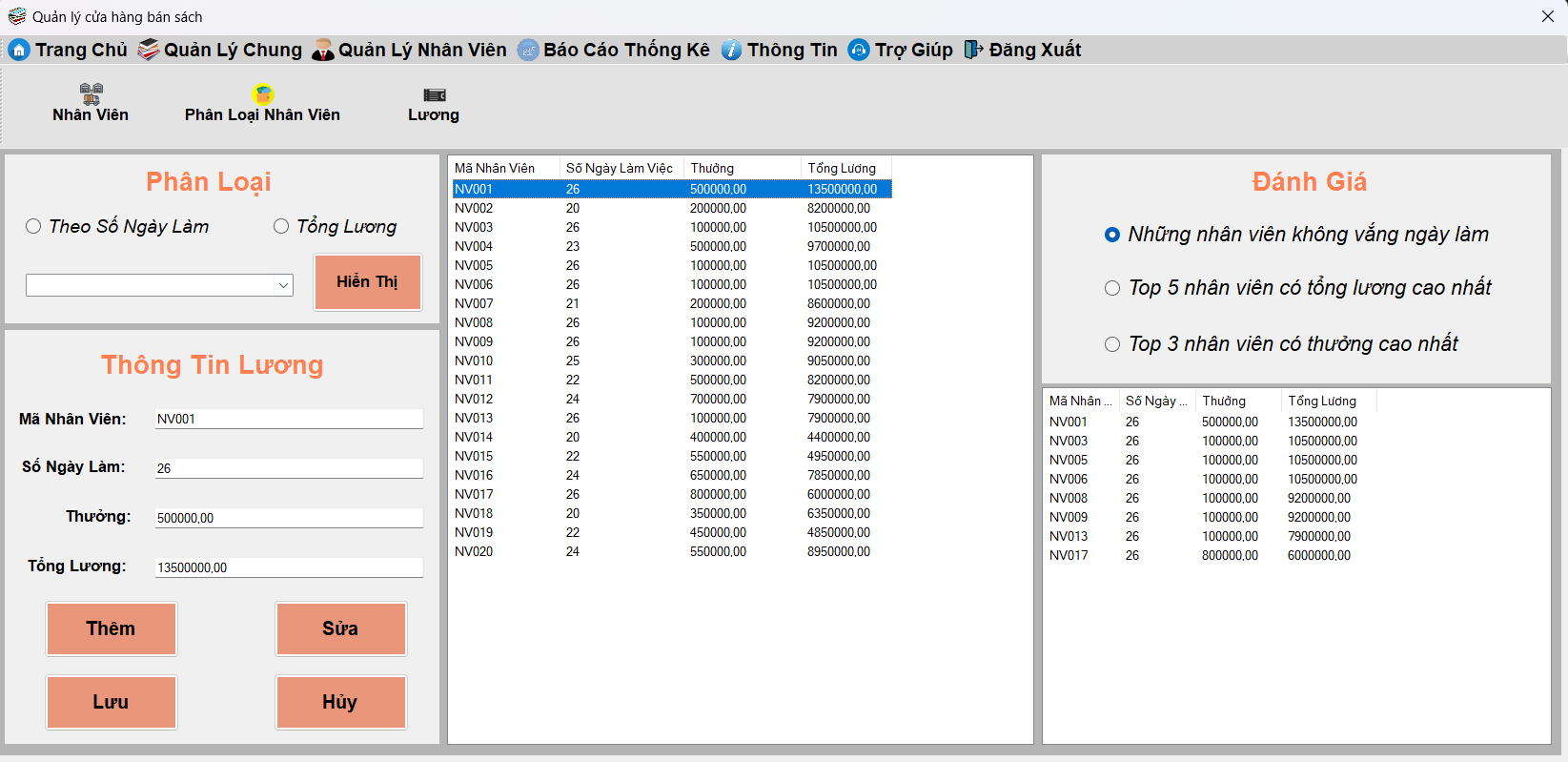
{

conn.Close();

}

}

#### **4.3 Form lương nhân viên**



void HienThiLuong()

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM View\_Salary";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvLuong.Items.Clear();

while (reader.Read())

{

string MaNhanVien = reader.GetString(0);

string SoNgayLamViec = reader.GetInt32(1).ToString();

string Thuong = reader.GetDecimal(2).ToString();

string TongLuong = reader.GetDecimal(3).ToString();

ListViewItem lvi = new ListViewItem(MaNhanVien);

lvi.SubItems.Add(SoNgayLamViec);

lvi.SubItems.Add(Thuong);

lvi.SubItems.Add(TongLuong);

lsvLuong.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

private void lsvLuong\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvLuong.SelectedItems.Count == 0) return;

ListViewItem lvi = lsvLuong.SelectedItems[0];

string MaNhanVien = lvi.SubItems[0].Text.Trim();

HienThiChiTietToanBoLuong(MaNhanVien);

}

void HienThiChiTietToanBoLuong(string MaNhanVien)

{

try

{

if (conn == null)

{

conn = new SqlConnection(strConnectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "ShowSalary";

command.Connection = conn;

SqlParameter parMaNhanVien = new SqlParameter("@MaNhanVien", SqlDbType.NVarChar);

parMaNhanVien.Value = MaNhanVien;

command.Parameters.Add(parMaNhanVien);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtMaNhanVien.Text = reader.GetString(0) + "";

txtSNLV.Text = reader.GetInt32(1).ToString() + "";

txtThuong.Text = reader.GetDecimal(2).ToString() + "";

txtTongLuong.Text = reader.GetDecimal(3).ToString() + "";

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

finally

{

conn.Close();

}

}

### **Form báo cáo thống kê**

A screenshot of a computer

Description automatically generated with medium confidence

private void HienThiTKDoanhThu()

{

try

{

if (conn == null)

{

conn = new SqlConnection(connectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM dbo.ThongKeHoaDon()";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvDoanhThu.Items.Clear();

while (reader.Read())

{

DateTime Ngay = reader.GetDateTime(0);

string SoLuongSachBan = reader.GetInt32(1).ToString();

string TongTien = reader.GetInt32(2).ToString();

ListViewItem lvi = new ListViewItem(Ngay.ToString("yyyy-MM-dd"));

lvi.SubItems.Add(SoLuongSachBan);

lvi.SubItems.Add(TongTien);

lsvDoanhThu.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi rồi");

}

finally

{

conn.Close();

}

}

private void HienThiTKPhieuNhap()

{

try

{

if (conn == null)

{

conn = new SqlConnection(connectionString);

}

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

SqlCommand command = new SqlCommand();

command.CommandType = CommandType.Text;

command.CommandText = "SELECT \* FROM fn\_PaymentStatistics()";

command.Connection = conn;

SqlDataReader reader = command.ExecuteReader();

lsvPhieuNhap.Items.Clear();

while (reader.Read())

{

DateTime Ngay = reader.GetDateTime(0);

string SoLuongNhap = reader.GetInt32(1).ToString();

string TongTien = reader.GetInt32(2).ToString();

ListViewItem lvi = new ListViewItem(Ngay.ToString("yyyy-MM-dd"));

lvi.SubItems.Add(SoLuongNhap);

lvi.SubItems.Add(TongTien);

lsvPhieuNhap.Items.Add(lvi);

}

reader.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi rồi");

}

finally

{

conn.Close();

}

}

private void CreateColumnChartDoanhThu()

{

try

{

// Thiết lập kích thước và vị trí của biểu đồ cột

chartDoanhThu.Series["Series1"].ChartType = SeriesChartType.Column;

chartDoanhThu.Series["Series1"].IsVisibleInLegend = false;

chartDoanhThu.Series["Series1"].XValueMember = "Tuan";

chartDoanhThu.Series["Series1"].YValueMembers = "TongDoanhThu";

chartDoanhThu.ChartAreas[0].AxisX.Interval = 1;

chartDoanhThu.Titles.Add("Title1").Text = "Biểu Đồ Doanh Thu";

string query = "SELECT \* FROM View\_RevenueStatistics\_Week";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

// Thiết lập dữ liệu cho biểu đồ tròn và cột

chartDoanhThu.DataSource = dataTable;

chartDoanhThu.DataBind();

}

// Thêm biểu đồ cột vào form

pDoanhThu.Controls.Add(chartDoanhThu);

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi rồi");

}

}

private void CreateColumnChartPhieuNhap()

{

try

{

// Thiết lập kích thước và vị trí của biểu đồ cột

chartPhieuNhap.Series["Series1"].ChartType = SeriesChartType.Column;

chartPhieuNhap.Series["Series1"].IsVisibleInLegend = false;

chartPhieuNhap.Series["Series1"].Color = System.Drawing.ColorTranslator.FromHtml("#FFA500");

chartPhieuNhap.Series["Series1"].XValueMember = "Thang";

chartPhieuNhap.Series["Series1"].YValueMembers = "TongTien";

chartPhieuNhap.ChartAreas[0].AxisX.Interval = 1;

chartPhieuNhap.Titles.Add("Title1").Text = "Biểu Đồ Phiếu Nhập";

string query = "SELECT \* FROM View\_PaymentStatistics\_Week";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

// Thiết lập dữ liệu cho biểu đồ tròn và cột

chartPhieuNhap.DataSource = dataTable;

chartPhieuNhap.DataBind();

}

// Thêm biểu đồ cột vào form

pPhieuNhap.Controls.Add(chartPhieuNhap);

}

catch (SqlException ex)

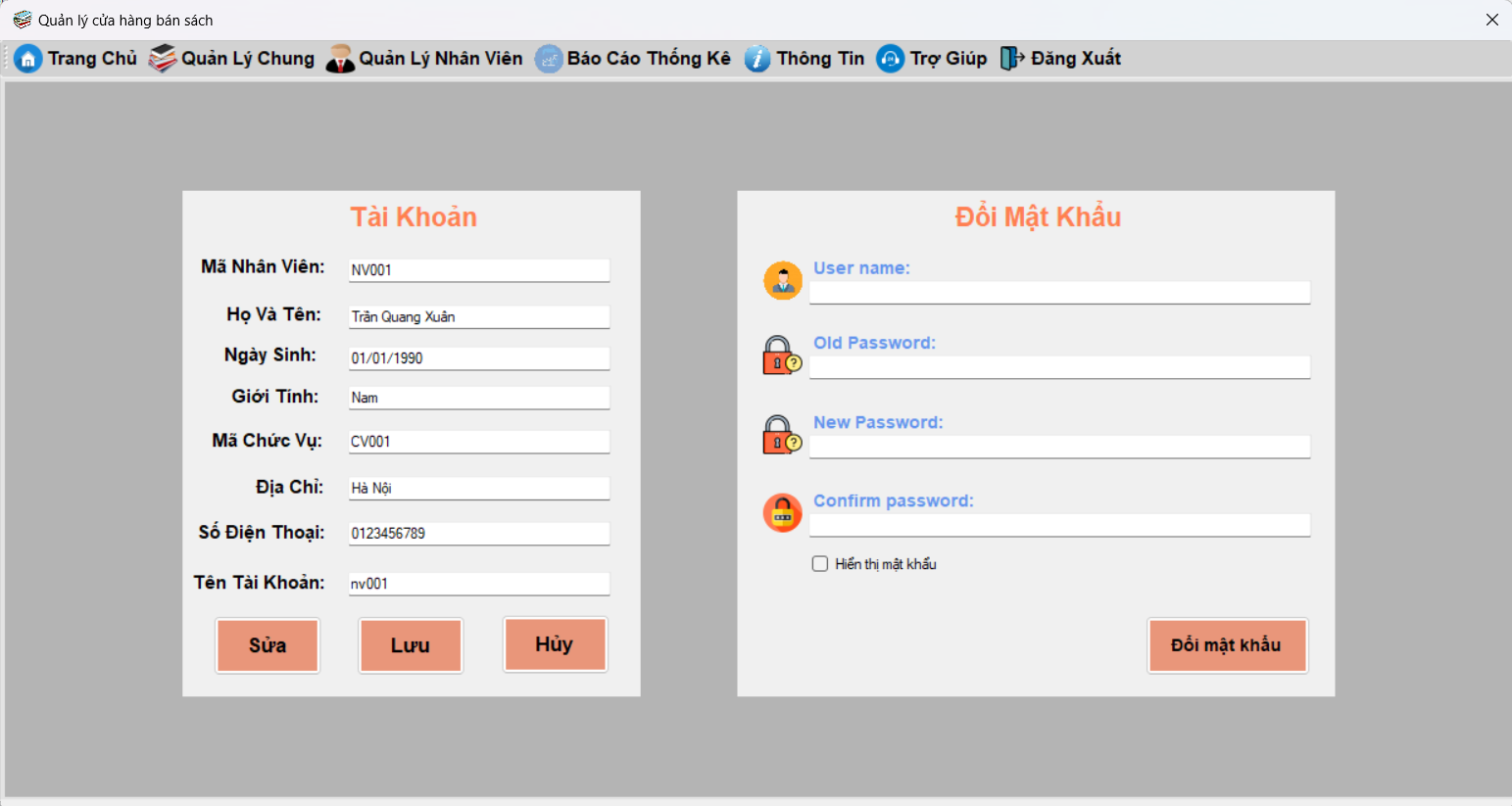
{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi rồi");

}

}

### **Form thông tin**



private void fThongTin\_Load(object sender, EventArgs e)

{

hienThiThongTin();

}

private void hienThiThongTin()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand("ShowStaffInformation", connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@TenDangNhap", taikhoan);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

// Lấy dữ liệu từ SqlDataReader

string maNhanVien = reader.GetString(0);

string hoTen = reader.GetString(1);

DateTime ngaySinh = reader.GetDateTime(2);

string gioiTinh = reader.GetString(3);

string maChucVu = reader.GetString(4);

string diaChi = reader.GetString(5);

string soDienThoai = reader.GetString(6);

string hoatdong = reader.GetString(7);

string taikhoan = reader.GetString(9);

string matkhau = reader.GetString(10);

// Hiển thị thông tin nhân viên trong textbox

txtMaNhanVien.Text = maNhanVien;

txtHoTen.Text = hoTen;

txtNgaySinh.Text = ngaySinh.ToString("dd/MM/yyyy");

txtGioiTinh.Text = gioiTinh;

txtMaChucVu.Text = maChucVu;

txtDiaChi.Text = diaChi;

txtSoDienThoai.Text = soDienThoai;

txtHoatDong.Text = hoatdong;

txtUserName.Text = taikhoan;

txtPassword.Text = matkhau;

}

reader.Close();

}

}

}

### **Form trợ giúp**



# **CHƯƠNG 5: PHÂN QUYỀN**

## **Nhân viên tư vấn**

-- Tư vấn

--CREATE ROLE TuVan

GRANT SELECT, REFERENCES ON Sach TO TuVan

GRANT SELECT, REFERENCES ON TheLoai TO TuVan

GRANT SELECT, REFERENCES ON KhoSach TO TuVan

GRANT SELECT, REFERENCES ON ViTriSach TO TuVan

GRANT SELECT, REFERENCES ON KhachHang TO TuVan

GRANT SELECT, REFERENCES ON NhaCungCap TO TuVan

REVOKE EXECUTE TO TuVan

REVOKE SELECT TO TuVan

--SÁCH

GRANT SELECT ON View\_Book TO TuVan

GRANT SELECT ON ShowBookDetails TO TuVan

GRANT SELECT ON SearchBook TO TuVan

GRANT SELECT ON ClassifiedByGenre TO TuVan

GRANT EXECUTE ON ClassifiedByPrice TO TuVan

GRANT SELECT ON ClassifiedBySupplier TO TuVan

GRANT SELECT ON ClassifiedByPublishing TO TuVan

--THỂ LOẠI

GRANT SELECT ON View\_Genre TO TuVan

GRANT EXECUTE ON ShowGenreDetails TO TuVan

GRANT SELECT ON SearchGenre TO TuVan

GRANT SELECT ON View\_GenreHighestNumber TO TuVan

GRANT SELECT ON View\_GenreLeastNumber TO TuVan

GRANT SELECT ON View\_GenreBestSell TO TuVan

GRANT SELECT ON View\_GenreWorstSell TO TuVan

--KHO SÁCH

GRANT SELECT ON View\_Inventory TO TuVan

GRANT EXECUTE ON ShowInventoryDetails TO TuVan

GRANT SELECT ON SearchInventory TO TuVan

GRANT SELECT ON View\_MostInStock TO TuVan

GRANT SELECT ON View\_LeastInStock TO TuVan

--VỊ TRÍ

GRANT SELECT ON View\_BookLocation TO TuVan

GRANT EXECUTE ON ShowBookLocationDetails TO TuVan

GRANT SELECT ON SearchBookLocation TO TuVan

GRANT EXECUTE ON ClassifiedByShelf TO TuVan

GRANT SELECT ON View\_BookBestSell TO TuVan

GRANT SELECT ON View\_BookWorstSell TO TuVan

--KHÁCH HÀNG

GRANT SELECT ON View\_Customer TO TuVan

GRANT SELECT ON ShowCustomerDetails TO TuVan

GRANT SELECT ON SearchCustomer TO TuVan

GRANT SELECT ON View\_CustomerMostBuy TO TuVan

GRANT SELECT ON View\_CustomerLeastBuy TO TuVan

--NHÀ CUNG CẤP

GRANT SELECT ON View\_Supplier TO TuVan

GRANT SELECT ON ShowSupplierDetails TO TuVan

GRANT SELECT ON SearchSupplier TO TuVan

GRANT EXECUTE ON CheckAccount\_Phone TO TuVan

GRANT EXECUTE ON CheckAccount\_Password TO TuVan

GRANT EXECUTE ON UpdateAccount TO TuVan

GRANT EXECUTE ON updateEmployee TO TuVan

GRANT EXECUTE ON ShowStaffInformation TO TuVan

GO

## **Nhân viên thu ngân**

-- Thu Ngân

--CREATE ROLE ThuNgan

GRANT SELECT, REFERENCES ON Sach TO ThuNgan

GRANT SELECT, REFERENCES ON TheLoai TO ThuNgan

GRANT SELECT, REFERENCES ON KhoSach TO ThuNgan

GRANT SELECT, REFERENCES ON ViTriSach TO ThuNgan

GRANT SELECT, REFERENCES ON KhachHang TO ThuNgan

GRANT SELECT, REFERENCES ON NhaCungCap TO ThuNgan

GRANT SELECT, INSERT, UPDATE, REFERENCES ON HoaDon TO ThuNgan

GRANT SELECT, INSERT, UPDATE, REFERENCES ON ChiTietHoaDon TO ThuNgan

REVOKE EXECUTE TO ThuNgan

REVOKE SELECT TO ThuNgan

--SÁCH

GRANT SELECT ON View\_Book TO ThuNgan

GRANT SELECT ON ShowBookDetails TO ThuNgan

GRANT SELECT ON SearchBook TO ThuNgan

GRANT SELECT ON ClassifiedByGenre TO ThuNgan

GRANT EXECUTE ON ClassifiedByPrice TO ThuNgan

GRANT SELECT ON ClassifiedBySupplier TO ThuNgan

GRANT SELECT ON ClassifiedByPublishing TO ThuNgan

--THỂ LOẠI

GRANT SELECT ON View\_Genre TO ThuNgan

GRANT EXECUTE ON ShowGenreDetails TO ThuNgan

GRANT SELECT ON SearchGenre TO ThuNgan

GRANT SELECT ON View\_GenreHighestNumber TO ThuNgan

GRANT SELECT ON View\_GenreLeastNumber TO ThuNgan

GRANT SELECT ON View\_GenreBestSell TO ThuNgan

GRANT SELECT ON View\_GenreWorstSell TO ThuNgan

--KHO SÁCH

GRANT SELECT ON View\_Inventory TO ThuNgan

GRANT EXECUTE ON ShowInventoryDetails TO ThuNgan

GRANT SELECT ON SearchInventory TO ThuNgan

GRANT SELECT ON View\_MostInStock TO ThuNgan

GRANT SELECT ON View\_LeastInStock TO ThuNgan

--VỊ TRÍ

GRANT SELECT ON View\_BookLocation TO ThuNgan

GRANT EXECUTE ON ShowBookLocationDetails TO ThuNgan

GRANT SELECT ON SearchBookLocation TO ThuNgan

GRANT EXECUTE ON ClassifiedByShelf TO ThuNgan

GRANT SELECT ON View\_BookBestSell TO ThuNgan

GRANT SELECT ON View\_BookWorstSell TO ThuNgan

--KHÁCH HÀNG

GRANT SELECT ON View\_Customer TO ThuNgan

GRANT SELECT ON ShowCustomerDetails TO ThuNgan

GRANT SELECT ON SearchCustomer TO ThuNgan

GRANT SELECT ON View\_CustomerMostBuy TO ThuNgan

GRANT SELECT ON View\_CustomerLeastBuy TO ThuNgan

--NHÀ CUNG CẤP

GRANT SELECT ON View\_Supplier TO ThuNgan

GRANT SELECT ON ShowSupplierDetails TO ThuNgan

GRANT SELECT ON SearchSupplier TO ThuNgan

--HÓA ĐƠN VÀ CHI TIẾT HÓA ĐƠN

GRANT SELECT ON View\_Bill TO ThuNgan

GRANT SELECT ON View\_BillDetails TO ThuNgan

GRANT EXECUTE ON ShowAllBills TO ThuNgan

GRANT EXECUTE ON ShowAllBillDetails TO ThuNgan

GRANT SELECT ON SearchBill TO ThuNgan

GRANT SELECT ON SearchBillDetails TO ThuNgan

GRANT EXECUTE ON ClassifiedBillByStaff TO ThuNgan

GRANT EXECUTE ON AddBill TO ThuNgan

GRANT EXECUTE ON UpdateBill TO ThuNgan

GRANT EXECUTE ON AddBillDetails TO ThuNgan

GRANT EXECUTE ON UpdateBillDetails TO ThuNgan

GRANT EXECUTE ON CheckAccount\_Phone TO ThuNgan

GRANT EXECUTE ON CheckAccount\_Password TO ThuNgan

GRANT EXECUTE ON UpdateAccount TO ThuNgan

GRANT EXECUTE ON updateEmployee TO ThuNgan

GRANT EXECUTE ON ShowStaffInformation TO ThuNgan

GO

**Nhân viên thủ kho**

-- Thủ kho

--CREATE ROLE ThuKho

GRANT SELECT, INSERT, UPDATE, REFERENCES ON Sach TO ThuKho

GRANT SELECT, INSERT, UPDATE, REFERENCES ON KhoSach TO ThuKho

GRANT SELECT, INSERT, UPDATE, REFERENCES ON TheLoai TO ThuKho

GRANT SELECT, UPDATE, REFERENCES ON ViTriSach TO ThuKho

GRANT SELECT, REFERENCES ON NhaCungCap TO ThuKho

GRANT SELECT, REFERENCES ON PhieuNhap TO ThuKho

GRANT SELECT, REFERENCES ON ChiTietNhapSach TO ThuKho

REVOKE EXECUTE TO ThuKho

REVOKE SELECT TO ThuKho

--SÁCH

GRANT SELECT ON View\_Book TO ThuKho

GRANT SELECT ON ShowBookDetails TO ThuKho

GRANT SELECT ON SearchBook TO ThuKho

GRANT SELECT ON ClassifiedByGenre TO ThuKho

GRANT EXECUTE ON ClassifiedByPrice TO ThuKho

GRANT SELECT ON ClassifiedBySupplier TO ThuKho

GRANT SELECT ON ClassifiedByPublishing TO ThuKho

GRANT EXECUTE ON AddBook TO ThuKho

GRANT EXECUTE ON AddInventory TO ThuKho

GRANT EXECUTE ON UpdateBook TO ThuKho

GRANT EXECUTE ON UpdateInventory TO ThuKho

--THỂ LOẠI

GRANT SELECT ON View\_Genre TO ThuKho

GRANT EXECUTE ON ShowGenreDetails TO ThuKho

GRANT SELECT ON SearchGenre TO ThuKho

GRANT SELECT ON View\_GenreHighestNumber TO ThuKho

GRANT SELECT ON View\_GenreLeastNumber TO ThuKho

GRANT SELECT ON View\_GenreBestSell TO ThuKho

GRANT SELECT ON View\_GenreWorstSell TO ThuKho

GRANT EXECUTE ON AddGenre TO ThuKho

GRANT EXECUTE ON UpdateGenre TO ThuKho

--KHO SÁCH

GRANT SELECT ON View\_Inventory TO ThuKho

GRANT EXECUTE ON ShowInventoryDetails TO ThuKho

GRANT SELECT ON SearchInventory TO ThuKho

GRANT SELECT ON View\_MostInStock TO ThuKho

GRANT SELECT ON View\_LeastInStock TO ThuKho

GRANT EXECUTE ON UpdateInventory TO ThuKho

GRANT EXECUTE ON AddInventory TO ThuKho

--VỊ TRÍ

GRANT SELECT ON View\_BookLocation TO ThuKho

GRANT EXECUTE ON ShowBookLocationDetails TO ThuKho

GRANT SELECT ON SearchBookLocation TO ThuKho

GRANT EXECUTE ON ClassifiedByShelf TO ThuKho

GRANT SELECT ON View\_BookBestSell TO ThuKho

GRANT SELECT ON View\_BookWorstSell TO ThuKho

GRANT EXECUTE ON UpdateBookLocation TO ThuKho

--NHÀ CUNG CẤP

GRANT SELECT ON View\_Supplier TO ThuKho

GRANT SELECT ON ShowSupplierDetails TO ThuKho

GRANT SELECT ON SearchSupplier TO ThuKho

--PHIẾU NHẬP VÀ CHI TIẾT PHIẾU NHẬP

GRANT SELECT ON View\_Receipt TO ThuKho

GRANT SELECT ON View\_ReceiptDetails TO ThuKho

GRANT EXECUTE ON ShowReceipt TO ThuKho

GRANT EXECUTE ON ShowReceiptDetails TO ThuKho

GRANT SELECT ON SearchReceipt TO ThuKho

GRANT SELECT ON SearchReceiptDetails TO ThuKho

GRANT EXECUTE ON CheckAccount\_Phone TO ThuKho

GRANT EXECUTE ON CheckAccount\_Password TO ThuKho

GRANT EXECUTE ON UpdateAccount TO ThuKho

GRANT EXECUTE ON updateEmployee TO ThuKho

GRANT EXECUTE ON ShowStaffInformation TO ThuKho

GO

+ Đối với Role Manager, yêu cần cần có quyền gán quyền cho nhân viên, sử dụng server role sysadmin cho Manager.

**Tạo login**

create TRIGGER CreateSQLAccount ON DangNhap

AFTER INSERT

AS

BEGIN

DECLARE @sqlString nvarchar(2000)

DECLARE @userName nvarchar(10)

DECLARE @passWord nvarchar(30)

DECLARE @cv varchar(20)

SELECT @userName = DangNhap.TenDangNhap, @passWord = DangNhap.MatKhau, @cv = NhanVien.MaChucVu

FROM DangNhap

INNER JOIN NhanVien ON DangNhap.MaNhanVien = NhanVien.MaNhanVien

INNER JOIN INSERTED ON DangNhap.MaNhanVien = INSERTED.MaNhanVien

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

+''', DEFAULT\_DATABASE=[QLCHBS], CHECK\_EXPIRATION=OFF,

CHECK\_POLICY=OFF'

EXEC (@sqlString)

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

EXEC (@sqlString)

--role quản lý

if (@cv='CV001')

begin

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

exec (@sqlString)

end

--role thu ngân

else if(@cv='CV002')

begin

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

exec (@sqlString)

end

--role Quản lý kho

else if(@cv='CV003')

begin

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

exec (@sqlString)

end

--role tư vấn

else

begin

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

exec (@sqlString)

end

END

Go

+ Procedure khi cập nhật hoạt động của nhân viên hoặc “Nghỉ việc”, thì sẽ thực hiện procedure xóa tài khoản, user và login nhân viên đó

CREATE PROCEDURE UpdateEmployee

@MaNhanVien NVARCHAR(10),

@HoTen NVARCHAR(100),

@NgaySinh DATE,

@GioiTinh NVARCHAR(10),

@MaChucVu NVARCHAR(10),

@DiaChi NVARCHAR(200) ,

@SoDienThoai NVARCHAR(20),

@HoatDong nvarchar(10)

AS

BEGIN

-- cập nhật nhân viên

UPDATE NhanVien

SET HoTen = @HoTen, NgaySinh = @NgaySinh,

GioiTinh = @GioiTinh, MaChucVu = @MaChucVu, DiaChi = @DiaChi, SoDienThoai = @SoDienThoai,HoatDong = @HoatDong

WHERE MaNhanVien = @MaNhanVien

-- kiểm tra trạng thái hoạt động của nhân viên

IF @HoatDong = N'Nghỉ việc'

BEGIN

DECLARE @TenDangNhap varchar(10)

SELECT @TenDangNhap = DangNhap.TenDangNhap FROM DangNhap WHERE MaNhanVien = @MaNhanVien

DECLARE @command nvarchar(max) = ''

SET @command = 'EXEC DeleteAccountEmployee ' + '"' + @TenDangNhap + '"'

EXEC(@command)

END

END

GO

-- Xóa tài khoản nhân viên

create procedure DeleteAccountEmployee

@TenDangNhap varchar(10)

as

begin

declare @command nvarchar(max) = ''

set @command = '';

SELECT @command = @command + 'KILL ' + CONVERT(varchar(5), session\_id) + ';'

FROM sys.dm\_exec\_sessions

WHERE login\_name = @TenDangNhap

EXEC (@command);

--

set @command = ''

select @command = @command + 'DROP login ['+name+'];'

from sys.syslogins where name like @TenDangNhap

exec(@command)

--

set @command = ''

select @command = @command + 'DROP user ['+name+'];'

from sys.sysusers where name like @TenDangNhap

exec(@command)

-- Xóa nhân viên khỏi phân nhóm

DELETE FROM DangNhap WHERE TenDangNhap = @TenDangNhap;

end

go