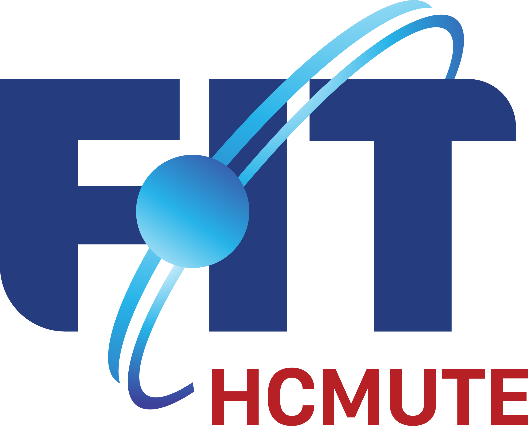
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**🙞 🕮 🙜**

****

**ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL**

**TÊN ĐỀ TÀI:**

**QUẢN LÝ TIỆM TRÀ SỮA**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Lớp HP:** *DBMS330284\_23\_1\_03*

**Nhóm thực hiện:** *Nhóm 10*

**Học kỳ:** *1*

**Năm học:** *2023 - 2024*

*Thành phố Hồ Chí Minh, tháng 11, năm 2023*

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KỲ I NĂM HỌC 2023-2024**

**NHÓM 10**

***Đề tài: Quản lý tiệm trà sữa***

|  |  |
| --- | --- |
| 21110521 | Bùi Thiên Kim |
| 21110548 | Nguyễn Quỳnh My |
| 21110638 | Phạm Khương Tâm |
| 21110643 | Ngô Thanh Thanh |

**NHẬN XÉT CỦA GV**

Tp. Hồ Chí Minh, ngày …. tháng 11 năm 2023

Giảng viên chấm điểm

MỤC LỤC

DỰ ÁN QUẢN LÝ TIỆM TRÀ SỮA

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 1](#_Toc151615577)

[1. Đặc tả đề tài 1](#_Toc151615578)

[1.1. Mô tả bài toán 1](#_Toc151615579)

[1.2. Nghiệp vụ của bài toán 2](#_Toc151615580)

[1.3. Mô tả chức năng của bài toán 4](#_Toc151615581)

[1.3.1. Quản lý 4](#_Toc151615582)

[1.3.2. Nhân viên 5](#_Toc151615583)

[1.4 Quản lý lợi nhuận 5](#_Toc151615584)

[1.4.1 Quản lý doanh thu 5](#_Toc151615585)

[1.4.2 Quản lý chi phí 7](#_Toc151615586)

[1.4.3 Cách tính lợi nhuận 8](#_Toc151615587)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 10](#_Toc151615588)

[2.1 Thiết kế cơ sở dữ liệu mức quan niệm 10](#_Toc151615589)

[2.2 Thiết kế cơ sở dữ liệu mức logic 11](#_Toc151615590)

[2.3 Các ràng buộc cần có 12](#_Toc151615591)

[2.4 Cài đặt CSDL và các ràng buộc 15](#_Toc151615592)

[2.5 Các view 20](#_Toc151615593)

[2.5.1 Xem ca làm việc của nhân viên 20](#_Toc151615594)

[2.5.2 Xem danh sách sản phẩm 20](#_Toc151615595)

[2.5.3 Xem số lượng sản phẩm đã bán trong ngày 20](#_Toc151615596)

[2.5.4 Xem thông tin nhân viên 20](#_Toc151615597)

[2.5.5 Xem doanh thu theo từng ngày 20](#_Toc151615598)

[2.5.6 Xem doanh thu theo từng tháng 20](#_Toc151615599)

[2.5.7 Xem lợi nhuận theo từng ngày 20](#_Toc151615600)

[2.5.8 Xem lợi nhuận theo từng tháng 21](#_Toc151615601)

[2.5.9 Xem số lượng nguyên liệu còn lại 21](#_Toc151615602)

[2.6 Các trigger 21](#_Toc151615603)

[2.6.1 Kiểm tra xem nguyên liệu trong kho còn đủ để đáp ứng số lượng sản phẩm khách hàng yêu cầu hay không 21](#_Toc151615604)

[2.6.2 Kiểm tra tên sản phẩm có bị trùng không 22](#_Toc151615605)

[2.6.3 Trigger bắt lỗi khi thêm hoặc sửa hóa đơn bán 22](#_Toc151615606)

[2.6.4 Trigger bắt lỗi khi thêm hoặc sửa chi tiết hóa đơn bán 23](#_Toc151615607)

[2.6.5 Trigger bắt lỗi khi thêm hoặc sửa nhà cung cấp 24](#_Toc151615608)

[2.6.6 Trigger bắt lỗi khi thêm hoặc sửa hóa đơn nhập 25](#_Toc151615609)

[2.6.7 Trigger bắt lỗi khi thêm hoặc sửa chi tiết hóa đơn nhập 25](#_Toc151615610)

[2.6.8 Trigger bắt lỗi khi thêm hoặc sửa nhân viên 26](#_Toc151615611)

[2.6.9 Trigger bắt lỗi khi thêm khách hàng 28](#_Toc151615612)

[2.6.9 Trigger bắt lỗi khi thêm ứng dụng 28](#_Toc151615613)

[2.6.10 Trigger bắt lỗi khi thêm loại sản phẩm 29](#_Toc151615614)

[2.6.10 Trigger bắt lỗi khi thêm loại sản phẩm 29](#_Toc151615615)

[2.6.11 Trigger tự động tạo tài khoản khi thêm nhân viên 29](#_Toc151615616)

[2.6.11 Trigger tự động phân quyền cho tài khoản của nhân viên 30](#_Toc151615617)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 31](#_Toc151615618)

[3.1 Kết nối cơ sở dữ liệu 31](#_Toc151615619)

[3.2 Xem thông tin danh mục 31](#_Toc151615620)

[3.2.1. Danh mục nhân viên 31](#_Toc151615621)

[3.2.2. Danh mục bảng phân ca 32](#_Toc151615622)

[3.2.3. Danh mục ca làm việc 32](#_Toc151615623)

[3.2.4. Danh mục hóa đơn 32](#_Toc151615624)

[3.2.4.1. Danh mục hóa đơn bán 33](#_Toc151615625)

[3.2.4.2. Danh mục hóa đơn ứng dụng 33](#_Toc151615626)

[3.2.5. Danh mục loại sản phẩm 34](#_Toc151615627)

[3.2.6 Danh mục sản phẩm 34](#_Toc151615628)

[3.2.7. Danh mục phiếu chi 34](#_Toc151615629)

[3.2.8. Danh mục nhà cung cấp 35](#_Toc151615630)

[3.2.9. Danh mục khách hàng 35](#_Toc151615631)

[3.2.10. Danh mục nguyên liệu 36](#_Toc151615632)

[3.2.11. Danh mục cách chế biến 36](#_Toc151615633)

[3.2.12. Danh mục vị trí làm việc 36](#_Toc151615634)

[3.2.13. Danh mục ứng dụng 37](#_Toc151615635)

[3.3 Quản lý khách hàng 37](#_Toc151615636)

[3.3.1. Thêm khách hàng mới 37](#_Toc151615637)

[3.3.2. Sửa thông tin khách hàng 38](#_Toc151615638)

[3.3.3. Xóa khách hàng 38](#_Toc151615639)

[3.3.4. Tìm kiếm khách hàng thông qua số điện thoại 39](#_Toc151615640)

[3.3.5. Trigger bắt lỗi khi thêm khách hàng mới 39](#_Toc151615641)

[3.4 Quản lý hóa đơn 40](#_Toc151615642)

[3.4.1. Hóa đơn nhập 40](#_Toc151615643)

[3.4.1.1. Thêm hóa đơn nhập 40](#_Toc151615644)

[3.4.1.2. Sửa hóa đơn nhập 40](#_Toc151615645)

[3.4.1.3. Xóa hóa đơn nhập 41](#_Toc151615646)

[3.4.2. Chi tiết hóa đơn nhập 42](#_Toc151615647)

[3.4.2.1. Thêm chi tiết hóa đơn nhập 42](#_Toc151615648)

[3.4.2.2. Sửa chi tiết hóa đơn nhập 43](#_Toc151615649)

[3.4.2.3. Xóa chi tiết hóa đơn nhập 43](#_Toc151615650)

[3.4.3. Hóa đơn bán 44](#_Toc151615651)

[3.4.3.1. Thêm hóa đơn bán 44](#_Toc151615652)

[3.4.3.2. Sửa hóa đơn bán 45](#_Toc151615653)

[3.4.3.3. Xóa hóa đơn bán 46](#_Toc151615654)

[3.4.4. Chi tiết hóa đơn bán 46](#_Toc151615655)

[3.4.4.1. Thêm chi tiết hóa đơn bán 46](#_Toc151615656)

[3.4.4.2. Sửa chi tiết hóa đơn bán 47](#_Toc151615657)

[3.4.4.3. Xóa chi tiết hóa đơn bán 48](#_Toc151615658)

[3.4.4.4. Trigger bắt lỗi khi thêm thông tin hoặc sửa thông tin của hóa đơn bán 49](#_Toc151615659)

[3.4.5. Hóa đơn ứng dụng 50](#_Toc151615660)

[3.4.5.1. Thêm hóa đơn ứng dụng 50](#_Toc151615661)

[3.4.5.2. Sửa hóa đơn ứng dụng 50](#_Toc151615662)

[3.4.5.3. Xóa hóa đơn ứng dụng 51](#_Toc151615663)

[3.4.6. Chi tiết hóa đơn ứng dụng 52](#_Toc151615664)

[3.4.6.1. Thêm chi tiết hóa đơn ứng dụng 52](#_Toc151615665)

[3.4.6.2. Sửa chi tiết hóa đơn ứng dụng 53](#_Toc151615666)

[3.4.6.3. Xóa chi tiết hóa đơn ứng dụng 53](#_Toc151615667)

[3.5 Quản lý kho nguyên liệu 54](#_Toc151615668)

[3.5.1. Thêm nguyên liệu vào kho 54](#_Toc151615669)

[3.5.2. Sửa thông tin nguyên liệu 55](#_Toc151615670)

[3.5.3. Xóa nguyên liệu không dùng 56](#_Toc151615671)

[3.5.4. Tìm kiếm nguyên liệu 56](#_Toc151615672)

[3.5.5. Trigger kiểm tra xem nguyên liệu trong kho còn đủ để pha chế không 57](#_Toc151615673)

[3.6 Quản lý sản phẩm 57](#_Toc151615674)

[3.6.1. Thêm sản phẩm 57](#_Toc151615675)

[3.6.2. Sửa sản phẩm 58](#_Toc151615676)

[3.6.3. Xóa sản phẩm 59](#_Toc151615677)

[3.7 Thống kê doanh thu và chi phí 59](#_Toc151615678)

[3.7.1. Tính tổng tiền bên chi tiết hóa đơn sau đó cập nhật qua hóa đơn bán 59](#_Toc151615679)

[3.7.2. Tính tổng tiền bên chi tiết hóa đơn sau đó cập nhật qua hóa đơn ứng dụng 60](#_Toc151615680)

[3.7.2. Tính tổng tiền bên chi tiết hóa đơn sau đó cập nhật qua hóa đơn nhập 61](#_Toc151615681)

[3.8 Quản lý nhân viên 61](#_Toc151615682)

[3.8.1. Thêm nhân viên mới 61](#_Toc151615683)

[3.8.2. Sửa thông tin nhân viên 62](#_Toc151615684)

[3.8.3. Xóa nhân viên đã nghỉ làm việc 63](#_Toc151615685)

[3.8.4. Trigger bắt lỗi khi thêm hoặc sửa thông tin nhân viên 64](#_Toc151615686)

[3.9 Quản lý phân ca 65](#_Toc151615687)

[3.9.1. Thêm bảng phân ca cho nhân viên 65](#_Toc151615688)

[3.9.2. Sửa bảng phân ca của nhân viên 66](#_Toc151615689)

[3.9.3. Xóa ca làm việc của nhân viên 67](#_Toc151615690)

[CHƯƠNG 4: QUẢN LÝ USER VÀ PHÂN QUYỀN 68](#_Toc151615691)

[4.1 Đối với Role Staff (dành cho nhân viên): 68](#_Toc151615692)

[4. 2 Đối với Role Manager ta sử dụng server role sysadmin. 68](#_Toc151615693)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 71](#_Toc151615694)

[1. Form Home 71](#_Toc151615695)

[2. Form đăng nhập 71](#_Toc151615696)

[3. Các Form Danh mục 72](#_Toc151615697)

[3.1. Danh mục sản phẩm 72](#_Toc151615698)

[3.2. Danh mục Loại Nhân Viên 73](#_Toc151615699)

[3.3. Danh mục Nhân Viên 73](#_Toc151615700)

[3.4. Danh mục Vị Trí Công Việc 73](#_Toc151615701)

[3.5. Danh mục Ca Làm Việc 74](#_Toc151615702)

[3.6. Danh mục Bảng Phân Ca 74](#_Toc151615703)

[3.7. Danh mục Khách Hàng 75](#_Toc151615704)

[3.8. Danh mục Nhà cung cấp 76](#_Toc151615705)

[3.9. Danh mục Loại Sản Phẩm 77](#_Toc151615706)

[3.10. Danh mục Sản Phẩm 78](#_Toc151615707)

[3.11. Danh mục Chế Biến 78](#_Toc151615708)

[3.12. Danh mục Nguyên Liệu 79](#_Toc151615709)

[3.13. Danh mục Ứng Dụng 79](#_Toc151615710)

[3.14. Danh mục Hóa Đơn Nhập 80](#_Toc151615711)

[3.15. Danh mục Chi tiết Hóa Đơn Nhập 81](#_Toc151615712)

[3.16. Danh mục Hóa Đơn Bán 81](#_Toc151615713)

[3.18. Danh mục Hóa Đơn Ứng Dụng 82](#_Toc151615714)

[3.19. Danh mục Chi tiết Hóa Đơn Ứng Dụng 83](#_Toc151615715)

[TÀI LIỆU THAM KHẢO 84](#_Toc151615716)

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến TS. Nguyễn Thành Sơn, người đã truyền đạt cho chúng em những kiến thức vô giá về bộ môn Hệ quản trị CSDL. Trong suốt quá trình học tập và nghiên cứu đề tài này, chúng em đã được thầy hướng dẫn, giúp đỡ tận tình và tâm huyết. Thầy đã dành nhiều thời gian và nỗ lực để chúng em có thể hiểu sâu hơn về các khái niệm, lý thuyết và ứng dụng của Hệ quản trị CSDL.

Những kiến thức và kinh nghiệm mà thầy truyền đạt cho chúng em sẽ luôn là những tài sản quý giá trong sự nghiệp học tập và phát triển sau này. Nhờ vào sự giảng dạy của thầy, chúng em đã có cơ hội tiếp cận và tìm hiểu về những xu hướng mới nhất, những công nghệ tiên tiến trong môn học Hệ quản trị Cơ sở dữ liệu.

Tuy nhiên, sự tiếp nhận kiến thức của mỗi người có những giới hạn và hạn chế riêng. Do đó, trong quá trình hoàn thành đề tài này, chắc chắn không tránh khỏi những thiếu sót và những khuyết điểm. Vì vậy, chúng em mong nhận được những lời góp ý và đánh giá từ thầy để bài tiểu luận của tụi em và các bạn cùng lớp được hoàn thiện hơn.

Cuối cùng, chúng em xin chúc thầy sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy. Thầy là một người thầy tuyệt vời và đầy tâm huyết, luôn là người động viên và truyền cảm hứng cho mọi người. Chúng em chân thành cảm ơn thầy vì sự đóng góp to lớn của thầy trong quá trình học tập và nghiên cứu của chúng em**.**

*Trân trọng*

Phạm Khương Tâm,

Bùi Thiên Kim,

Ngô Thanh Thanh,

Nguyễn Quỳnh My

LỜI NÓI ĐẦU

Xin chào thầy và các bạn,

Nhóm em rất vui được chia sẻ với quý thầy và các bạn về thành quả của đồ án quản lý quán trà sữa, một sản phẩm mà chúng tôi đã dành nhiều thời gian nghiên cứu và xây dựng trong khuôn khổ môn học Hệ quản trị cơ sở dữ liệu. Đây không chỉ là một nhiệm vụ đầy thách thức mà còn là cơ hội để chúng tôi tích lũy kiến thức và kinh nghiệm quý báu.

Trong suốt quá trình triển khai dự án, nhóm em đã đối mặt với nhiều khó khăn và thách thức. Nhưng nhờ sự hỗ trợ chân thành từ phía thầy hướng dẫn và bằng sự cố gắng không ngừng của mình, nhóm em đã hoàn thành đồ án.

Đồ án quản lý quán trà sữa không chỉ là một ứng dụng lý thuyết mà còn là một công cụ thực tế giúp chủ quán quản lý hoạt động kinh doanh một cách thuận tiện và hiệu quả. Với sự áp dụng linh hoạt của kiến thức về hệ quản trị cơ sở dữ liệu, nhóm em đã xây dựng một hệ thống quản lý toàn diện, bao gồm quản lý khách hàng, quản lý nhân viên, quản lý thực đơn và nhiều chức năng khác liên quan đến hoạt động kinh doanh.

Nhóm em hi vọng rằng đồ án của chúng tôi sẽ mang lại cho quý thầy cô và các bạn sinh viên những kiến thức thực tế và trải nghiệm hữu ích. Đồng thời, cũng mong rằng đồ án này sẽ làm nảy sinh những ý tưởng sáng tạo và khám phá trong việc ứng dụng hệ quản trị cơ sở dữ liệu vào thực tế.

Chân thành cảm ơn sự quan tâm và đóng góp ý kiến của thầy cô và các bạn sinh viên. Đây là một hành trình quý báu mà nhóm em đã có cơ hội chia sẻ cùng với mọi người.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

# 1. Đặc tả đề tài

## 1.1. Mô tả bài toán

Một cửa tiệm trà sữa cần có một hệ thống để quản lý. Chương trình quản lý được xây dựng cho hai đối tượng người dùng là người quản lý và nhân viên với các quyền khác nhau. Đối với người dùng là nhân viên, họ sẽ có quyền truy cập vào các chức năng như bán hàng, xem doanh thu, xem ca làm việc. Còn với người dùng là người quản lý chi nhánh, họ sẽ có quyền truy cập vào các chức năng như quản lý kho, quản lý doanh thu, quản lý nhân viên, quản lý ca làm việc. Đối với cửa tiệm trà sữa này cần lưu trữ những dữ liệu sau:

Cửa tiệm trà sữa sẽ có nhiều nhân viên và chỉ có một nhân viên làm quản lý cửa tiệm, mỗi nhân viên chỉ làm một công việc, một công việc có thể có nhiều nhân viên cùng làm. Các vị trí công việc phân biệt với nhau bằng mã vị trí, mỗi vị trí công việc gồm các thông tin như mã vị trí, tên vị trí, tiền phụ cấp vị trí. Các nhân viên phân biệt với nhau bằng mã nhân viên, mỗi nhân viên gồm các thông tin như mã nhân viên, họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, mã vị trí, loại nhân viên, ngày tuyển dụng. Nhân viên được chia làm 2 loại là nhân viên làm việc toàn thời gian hoặc nhân viên làm việc bán thời gian, đối với mỗi loại sẽ có quy định cụ thể về mức lương cơ bản, riêng quản lý cửa tiệm phải là nhân viên toàn thời gian. Các nhân viên sẽ được chia ca để làm việc. Một nhân viên phải có ít nhất một ca làm việc hoặc có thể làm nhiều ca, một ca có ít nhất 1 nhân viên làm việc hoặc nhiều nhân viên làm cùng 1 ca. Các ca làm việc được phân biệt với nhau qua mã ca, mỗi ca có các thông tin như mã ca, ngày, giờ bắt đầu, giờ kết thúc.

Khách hàng đến quán được chia thành hai nhóm khách hàng là nhóm khách hàng vãng lai (không có thông tin) và nhóm khách hàng thành viên (có đăng ký thông tin). Các khách hàng thành viên phân biệt với nhau bằng số điện thoại. Mỗi khách hàng thành viên có các thông tin như họ tên, số điện thoại, địa chỉ.

Sản phẩm được làm ra từ các nguyên liệu. Một sản phẩm được làm ra từ một hoặc nhiều nguyên liệu, một nguyên liệu có thể dùng trong nhiều sản phẩm. Các nguyên liệu phân biệt nhau bằng mã nguyên liệu. Mỗi nguyên liệu có các thông tin như mã nguyên liệu, tên nguyên liệu, đơn vị tính, số lượng. Sản phẩm được chia thành nhiều loại, các loại sản phẩm phân biệt với nhau bằng mã loại sản phẩm. Một loại sản phẩm có thể có nhiều sản phẩm, một sản phẩm chỉ thuộc một loại sản phẩm. Các sản phẩm phân biệt với nhau bằng mã sản phẩm. Mỗi sản phẩm có các thông tin như mã sản phẩm, tên sản phẩm, đơn giá, tình trạng. Để quản lý nguyên liệu cần có thông tin của nhà cung cấp nguyên liệu, cũng như lên đơn đặt hàng nguyên liệu đảm bảo luôn cung cấp đủ nguyên liệu để chế biến sản phẩm. Một nhà cung cấp có thể cung cấp một hoặc nhiều nguyên liệu, mỗi nguyên liệu chỉ được cung cấp từ một nhà cung cấp. Các nhà cung cấp phân biệt với nhau bằng mã nhà cung cấp. Mỗi nhà cung cấp có các thông tin như mã nhà cung cấp, tên nhà cung cấp, địa chỉ, số điện thoại. Đơn nhập hàng được phân biệt bởi mã đơn nhập hàng, mỗi đơn nhập hàng có các thông tin như mã đơn nhập hàng, ngày nhập hàng, trị giá đơn nhập hàng, mã nhà cung cấp.

Các đơn hàng được phân biệt với nhau qua mã đơn hàng, mỗi đơn hàng gồm các thông tin như mã đơn hàng, ngày đặt hàng, số điện thoại người đặt, trị giá đơn hàng.

Để tăng doanh thu cho quán thì còn mô hình bán qua các ứng dụng (Shopee Food, Grab Food, Baemin, …). Các ứng dụng được phân biệt với nhau qua mã ứng dụng. Mỗi ứng dụng có các thông tin như mã ứng dụng, tên ứng dụng, chiết khấu. Và các đơn hàng này được gọi là đơn hàng qua ứng dụng (khác với các đơn hàng nói trên). Các đơn hàng qua ứng dụng này cũng phân biệt theo mã đơn hàng ứng dụng, mỗi đơn hàng qua ứng dụng có các thông tin như mã đơn hàng ứng dụng, ngày đặt hàng, mã nhân viên bán, trị giá đơn hàng.

## 1.2. Nghiệp vụ của bài toán

Bài toán quản lý tiệm trà sữa là một nghiệp vụ quản lý kinh doanh trong lĩnh vực dịch vụ ăn uống. Bài toán này liên quan tới việc quản lý nhân viên, quản lý doanh thu, quản lý hàng hóa, điều hành tiệm trà sữa. Nhân viên đóng vai trò quan trọng trong việc giao tiếp và phục vụ khách hàng từ khi khách hàng bước vào đến khi rời khỏi tiệm. Vì thế, chúng ta cần xây dựng tài liệu đào tạo nghiệp vụ cho nhân viên về quy trình làm việc, văn hóa phục vụ, cách ứng xử, xử lý tình huống, và các yêu cầu khác phù hợp với hoạt động của tiệm trà sữa.

* Quản lý nhân viên: Khi tuyển dụng nhân sự, tiệm trà sữa cần đưa ra yêu cầu và điều kiện phù hợp với công việc, ví dụ như các ca làm việc, đi làm đúng giờ không đi trễ quá số buổi quy định, tuân thủ đồng phục theo quy định, giới hạn thời gian nghỉ phép trong một tháng, ... mà nhân viên phải đồng ý trong quá trình làm việc.  Tiệm cần trả lương hợp lý cho nhân viên. Sau khi được tuyển dụng, nhân viên sẽ được đào tạo về nghiệp vụ và quy trình làm việc tại vị trí của mình. Người quản lý cần theo dõi và đánh giá hiệu suất làm việc của nhân viên, chỉ ra các điểm thiếu sót để nhân viên cải thiện và động viên nhân viên với những thành tích tốt.
* Quy trình phục vụ khách hàng: Khi khách hàng đến tiệm, nhân viên sẽ chào khách và tạo đơn hàng trong hệ thống. Sau đó, nhân viên sẽ hỏi khách hàng muốn gọi sản phẩm nào và có thể tư vấn cho khách hàng nếu cần. Nhân viên sẽ thêm các món vào đơn hàng, kiểm tra thông tin nếu khách hàng là thành viên của tiệm thì mỗi sản phẩm khách hàng mua sẽ được giảm giá. Cuối cùng, nhân viên sẽ xuất hóa đơn cho khách hàng thanh toán và trên mỗi hóa đơn sẽ có mã hóa đơn. Sau đó, nhân viên pha chế sẽ chuẩn bị các món trong đơn hàng và sẽ mang ra cho khách hàng. Trước khi đưa cho khách hàng, nhân viên sẽ thông báo mã hóa đơn để khách hàng dễ dàng nhận ra đơn hàng của mình và nhân viên cần kiểm tra xem đơn hàng đã đủ món hay chưa.
* Quy trình thực hiện khi có đơn hàng qua ứng dụng (Shopee Food, Grab Food, Baemin, …): Nhân viên quán sẽ có một chiếc Ipad đăng nhập vào các ứng dụng. Khi có đơn hàng được đặt qua các ứng dụng, Ipad sẽ nhận được thông báo của các đơn hàng. Nhân viên sẽ bấm tạo đơn ở mục đơn hàng qua ứng dụng trên chương trình, thêm các món có trong đơn hàng qua ứng dụng. Sau đó, nhân viên xuất hóa đơn và đưa cho người giao hàng của ứng dụng đó.
* Quản lý hàng hóa, quản lý doanh thu và vận hành tiệm trà sữa:
* Giai đoạn đầu, tiệm cần tìm đối tác và nhà cung cấp hàng hóa uy tín, có chính sách giá cả và giao hàng đảm bảo và có chứng từ rõ ràng để dễ dàng kiểm soát. Tiệm cần thống kê các hàng hóa nhập vào và có kho để dự trữ, bảo quản và kiểm tra hàng hóa. Kiểm tra hàng hóa dựa trên hóa đơn nhập hàng và nhập thông tin vào hệ thống quản lý tiệm.
* Giai đoạn vận hành, tiệm cần xây dựng chi phí cho từng sản phẩm và tính toán giá thành cho các món. Ngoài ra, cần thiết lập quy trình làm việc và nghiệp vụ cho nhân viên thu ngân, pha chế và phục vụ.
* Cuối cùng, giai đoạn kiểm soát, tiệm trà sữa cần kiểm soát công việc của nhân viên dựa trên đánh giá từ phản hồi của khách hàng về chất lượng dịch vụ. Việc sử dụng hình ảnh từ camera để quan sát và đánh giá hiệu suất làm việc của nhân viên cũng rất quan trọng. Tiệm cần kiểm soát thu chi hàng ngày, nhân viên thu ngân sẽ được phân quyền xem bảng doanh thu theo ngày để tổng kết ca. Nhân viên cần kiểm tra xem số tiền thu được có khớp với doanh thu của ca làm việc hay không trước khi chuyển ca cho nhân viên tiếp theo. Quản lý sẽ được phân quyền chi tiền cho việc nhập hàng hóa. Kho hàng của tiệm cần được kiểm soát chặt chẽ bằng việc kiểm tra định kỳ hàng tuần và so sánh với dữ liệu trong hệ thống quản lý tiệm. Quản lý cần nắm báo cáo doanh thu (theo ngày, tháng, năm), báo cáo chi phí (chi tiết các khoản chi), báo cáo thất thoát (đổ vỡ, hỏng hóc, mất mát). Chủ tiệm sẽ sử dụng các báo cáo này để nắm rõ tình hình doanh thu, chi phí, lương nhân viên và đưa ra các điều chỉnh để tối ưu hóa hoạt động kinh doanh.

## 1.3. Mô tả chức năng của bài toán

### 1.3.1. Quản lý

* Quản lý nhân viên: Cho phép người quản lý xem, nhập, thêm, xóa, sửa thông tin cơ bản của các nhân viên như họ tên, quê quán, địa chỉ, số điện thoại, …
* Quản lý ca làm của nhân viên: Cho phép người quản lý nhập, thêm, xóa, sửa ca làm việc của các nhân viên.
* Đánh giá, xuất báo cáo doanh thu: Người quản lý có thể theo dõi doanh thu của tiệm và xuất báo cáo theo ngày, tháng, quý hoặc năm dựa trên số hóa đơn.
* Quản lý kho hàng: Cho phép người quản lý thông tin về số lượng còn lại của từng nguyên liệu trong kho hàng, ngày nhập hàng của nguyên liệu, hạn sử dụng.

### 1.3.2. Nhân viên

* **Chức năng bán hàng:** cho phép nhân viên tạo hóa đơn bán, thêm món, xóa món, xuất hóa đơn thanh toán.
* **Quản lý khách hàng:** cho phép nhân viên tạo thành viên cho khách hàng, phát phiếu tích điểm và đóng mộc ở bước thanh toán để tích điểm cho khách hàng.
* **Xem doanh thu:** cho phép nhân viên có thể xem được doanh thu bán trong ca làm việc của mình.
* **Đăng ký ca làm việc:** Các nhân viên sẽ đăng ký ca với người quản lý và cho phép nhân viên xem bảng phân ca làm việc trong tuần.

## 1.4 Quản lý lợi nhuận

### 1.4.1 Quản lý doanh thu

- Doanh thu thu được sẽ được tổng hợp theo ngày, giờ chính xác, được ghi lại và quản lý bằng cách sử dụng phần mềm quản lý doanh thu. Theo dõi số tiền thu được từ mỗi giao dịch và tính tổng doanh thu hàng ngày hoặc hàng tháng.

- Ghi lại các giao dịch liên quan đến doanh thu, bao gồm tiền mặt, thẻ và các phương thức thanh toán khác. Ghi chép số lượng và giá bán của từng sản phẩm.

- Xuất báo cáo doanh thu hàng ngày hoặc hàng tháng để theo dõi hiệu suất kinh doanh và phát triển chiến lược tối ưu.

- Dựa trên báo cáo và phân tích, đánh giá hiệu suất kinh doanh và điều chỉnh chiến lược nếu cần thiết.

**Công thức tính tổng doanh thu theo ngày: Tổng doanh thu ngày = Ʃ (đơn giá từng  sản phẩm x số lượng sản phẩm đó bán được trong ngày )**

**Công thức tính tổng doanh thu theo tháng: Tổng doanh thu tháng = Ʃ (Tổng doanh thu các ngày trong tháng  )**

***Ví dụ****:* Một ngày tiệm trà sữa bán ngày thứ nhất bán được 20 ly sữa tươi trân châu đường đen size lớn, 10 ly sữa tươi trân châu đường đen size vừa, 10 ly hồng trà chanh, 30 ly trà sữa khoai môn và 15 ly trà sữa ô long, 3 phần topping thạch đào thêm, 5 phần topping trân châu trắng thêm

Trong đó đơn giá của từng sản phẩm:

* Sữa tươi trân châu đường đen: 25.000/1 ly
* Hồng trà chanh: 20.000/1 ly
* Trà sữa khoai môn: 27.000/1 ly
* Trà sữa ô long: 25.000/1 ly
* Topping thạch đào thêm: 5.000/1 phần
* Topping trân châu trắng thêm: 4.000/1 phần
* Chi phí up size vừa lên lớn: 6.000/1 lần

(**Nếu khách hàng là thành viên vip thì mỗi ly được giảm 3.000 đồng/ly**)

Tổng doanh thu ngày thứ nhất = Ʃ (đơn giá từng sản phẩm x số lượng sản phẩm đó bán được trong ngày )

= 20 ly sữa tươi trân châu đường đen size lớn x (25.000 + 6.000)

+ 10 ly sữa tươi trân châu đường đen size vừa x 25.000

+ 10 ly hồng trà chanh  x 20.000

+ 30 ly trà sữa khoai môn x 27.000

+ 15 ly trà sữa ô long x 25.000

+ 3 phần topping thạch đào thêm x 5.000

+ 5 phần topping trân châu trắng thêm x 4.000

= 2.290.000 đồng

Tương tự, ta tính tổng doanh thu cho ngày thứ hai, thứ ba,... dựa vào số lượng bán ra từng sản phẩm

Tổng doanh thu theo tháng = Ʃ (Tổng doanh thu các ngày trong tháng  )

= Tổng doanh thu ngày thứ nhất + Tổng doanh thu ngày thứ hai + ….

### 1.4.2 Quản lý chi phí

- Tiền nguyên liệu: Là khoản tiền cần chi trả để mua nguyên liệu cần thiết làm trà sữa như bột trà sữa, topping,.... Và tiền nguyên liệu sẽ được tính theo ngày dựa vào số lượng nguyên liệu của từng nguyên liệu đã dùng trong ngày

**Công thức tính tiền nguyên liệu theo tháng: Tiền nguyên liệu mỗi tháng = Ʃ (đơn giá từng nguyên liệu x số lượng nguyên liệu đã mua trong tháng)**

- Tiền lương nhân viên hàng tháng: là tổng số tiền phải trả cho tất cả nhân viên của tiệm trà sữa hàng tháng dựa vào thời gian làm việc của mỗi nhân viên

**Cách tính tiền lương của một nhân viên**

**- Lương cơ bản:** Là số tiền lương cơ bản nhân viên nhận được trên 1 giờ làm phụ thuộc vào dạng làm việc toàn thời gian hay bán thời gian chưa tính phụ cấp. Toàn thời gian là 25.000/1 tiếng và bán thời gian là 20.000/1 tiếng.

**- Thời gian làm việc:** Đây là tổng số giờ làm việc trong tháng của nhân viên

**- Phụ cấp vị trí:** Số tiền phụ cấp sẽ được tính dựa trên số giờ làm việc của nhân viên. Đối với nhân viên phục vụ, pha chế, thu ngân tiền phụ cấp vị trí là 5.000/1 tiếng làm việc còn riêng quản lý cửa hàng là 15.000/1 tiếng làm việc.

**- Lương theo giờ:** Tính theo công thức: (Lương cơ bản + Phụ cấp vị trí) x Thời gian làm việc

**- Tiền thưởng:** Là khoản tiền thưởng mà nhân viên được trả cho các ca làm việc bao gồm các trường hợp sau:

* Các ngày lễ, Tết: (lương cơ bản + phụ cấp vị trí) x 2 x Thời gian làm việc trong ngày lễ
* Doanh thu tiệm đạt chỉ tiêu: Nếu doanh thu của quán đạt được chỉ tiêu nhất định, nhân viên sẽ được thưởng thêm 5% số tiền doanh thu.
* Công thức tính tiền thưởng: Tiền Thưởng = Doanh Thu x 5% + tiền thưởng lễ

**Tổng lương / 1 tháng = Lương theo giờ + Tiền thưởng**

**Ví dụ:** Giả sử tiệm trà sữa có A là nhân viên quản lý. Trong tháng 9 này, nhân viên quản lý A đã làm được 250 tiếng trong đó có 8 tiếng trong ngày lễ 2/9

**Do đó, lương của nhân viên quản lý A sẽ là:**

Lương theo giờ = (Lương cơ bản + Phụ cấp vị trí) x Thời gian làm việc

= (25.000+15.000) đồng/giờ  x 250 giờ

= 10.000.000 đồng

Tổng lương = Lương theo giờ + Tiền thưởng

= 10.000.000 đồng + (25.000+ 15.000) đồng/giờ x 2 x 8 giờ

= 10.000.000 đồng + 640.000 đồng = 10.640.000 đồng

**Công thức tính tổng tiền lương nhân viên hàng tháng: Tổng tiền lương nhân viên hàng tháng = Ʃ (Tổng tiền lương của mỗi nhân viên)**

* Chi phí vận hành: tiền điện, tiền nước
* Chi phí khác

### 1.4.3 Cách tính lợi nhuận

- Lợi nhuận là số tiền dương mà doanh nghiệp kiếm được sau khi trừ đi tất cả các chi phí liên quan đến sản xuất và bán hàng, các khoản phải trả và các chi phí khác.

- Công thức tính lợi nhuận theo tháng:

Lợi nhuận/1 tháng = Tổng doanh thu/1 tháng - Tổng chi phí/1 tháng

- Các khoản chi phí bao gồm:

* Chi phí nguyên liệu
* Chi phí nhân viên
* Chi phí khác

*Ví dụ:*

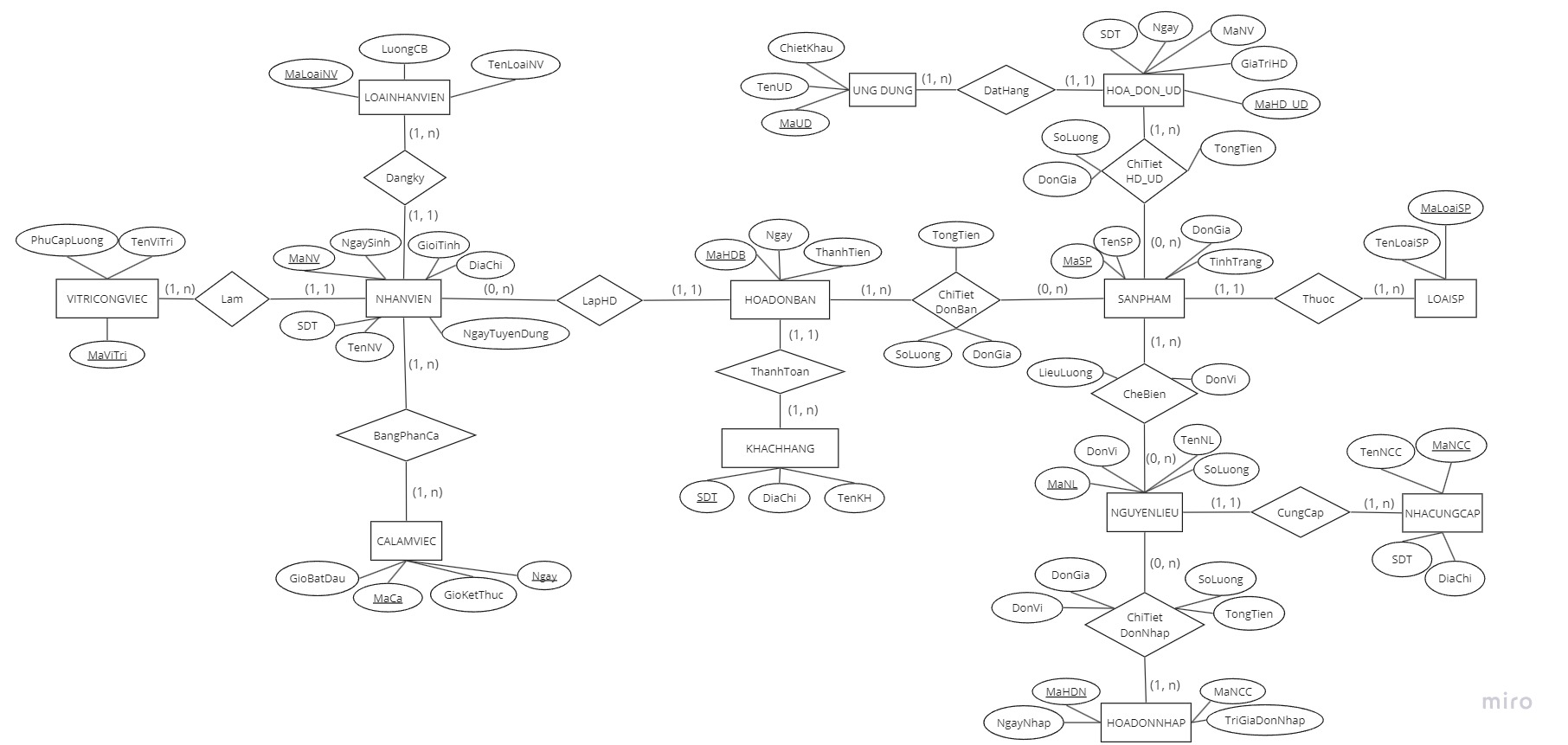
* Nguyên liệu đã mua trong tháng: Bột trà: 7.500.000đ/30kg, trân châu: 1.500.000đ/30kg, dụng cụ kèm theo (ly, ống hút, thìa, túi): 4.500.000đ.
* Chi phí vận hành: Nước: 1.000.000/tháng, Điện: 2.000.000/tháng.
* Lương nhân viên: Khoảng 20.000.000 đồng/tháng cho 3 nhân viên gồm 1 quản lý, 1 pha chế và 1 phục vụ
* Chi phí khác: 2.000.000 đồng/tháng
* Doanh thu của quán tháng đó là khoảng 60.000.000 đồng
* Lợi nhuận = Tổng doanh thu - Tổng chi phí chi ra

= 60.000.000 - (7.500.000 + 1.500.000 + 4.500.000 + 1.000.000 + 2.000.000 + 20.000.000 + 2.000.000) = 21.500.000

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

# 2.1 Thiết kế cơ sở dữ liệu mức quan niệm

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



# 2.2 Thiết kế cơ sở dữ liệu mức logic

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* VITRICONGVIEC(MaViTri, TenViTri, PhuCapLuong)
* NHANVIEN(MaNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, MaLoaiNV, MaViTri, NgayTuyenDung)
* LOAINHANVIEN(MaLoaiNV, TenLoaiNV, LuongCB)
* CALAMVIEC(MaCa, Ngay, GioBatDau, GioKetThuc)
* BangPhanCa(MaCa, MaNV, Ngay)
* HOADONBAN(MaHDB, Ngay, SDT, MaNV, ThanhTien)
* ChiTietDonBan(MaHDB, MaSP, SoLuong, DonGia, TongTien)
* KHACHHANG(SDT, TenKH, DiaChi)
* SANPHAM(MaSP, TenSP, DonGia, TinhTrang, MaLoaiSP)
* LOAISP(MaLoaiSP, TenLoaiSP)
* HOA\_DON\_UD(MaHD\_UD, Ngay, SDT, MaUD, MaNV, GiaTriHD)
* ChiTiet\_HD\_UD(MaHD\_UD, MaSP, SoLuong, DonGia, TongTien)
* UNGDUNG(MaUD, TenUD, ChietKhau)
* NGUYENLIEU(MaNL, TenNL, MaNCC, SoLuong, DonVi)
* CheBien(MaSP, MaNL, LieuLuong, DonVi)
* NHACUNGCAP(MaNCC, TenNCC, DiaChi, SDT)
* ChiTietDonNhap(MaHDN, MaNL, DonGia, SoLuong, DonVi, TongTien)
* HOADONNHAP(MaHDN, NgayNhap, TriGiaDonNhap, MaNCC)

# 2.3 Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | VITRICONGVIEC | Khóa chính MaViTri,  NOT NULL TenViTri,  CHECK PhuCapLuong > 0 |
| 2 | NHANVIEN | Khóa chính MaNV,  NOT NULL TenNV,  NOT NULL SDT,  CHECK NgaySinh (>=18 tuổi),  CHECK SDT (=10),  CHECK NgayTuyenDung (>0)  Khóa ngoại MaViTri đến bảng VITRICONGVIEC  Khóa ngoại MaLoaiNV đến bảng LOAINHANVIEN |
| 3 | LOAINHANVIEN | Khóa chính MaLoaiNV,  NOT NULL TenLoaiNV,  NOT NULL LuongCB |
| 4 | CALAMVIEC | Khóa chính(MaCa, Ngay)  CHECK Ngay (>=0)  NOT NULL GioBatDau,  NOT NULL GioKetThuc |
| 5 | BangPhanCa | Khóa chính(MaCa, Ngay, MaNV),  Khóa ngoại(MaCa, Ngay) đến bảng CALAMVIEC,  Khóa ngoại MaNV đến bảng NHANVIEN |
| 6 | HOADONBAN | Khoá chính (MaHDB)  Khoá ngoại SDT đến bảng KHACHHANG,  Khoá ngoại MaNV đến bảng NHANVIEN,  CHECK Ngay(>=0),  NOT NULL ThanhTien |
| 7 | ChiTietDonBan | Khóa chính (MaHDB, MaSP),  Khoá ngoại MaHDB đến bảng HOADONBAN,  Khoá ngoại MaSP đến bảng SANPHAM,  CHECK SoLuong(>0)  CHECK DonGia(>=0)  CHECK TongTien(>=0) |
| 8 | KHACHHANG | Khóa chính SDT,  NOT NULL TenKH,  CHECK SDT (=10) |
| 9 | SANPHAM | Khoá chính MaSP,  Khoá ngoại MaLoaiSP đến bảng LOAISP,  NOT NULL TenSP,  NOT NULL DonGia  CHECK DonGia(>0) |
| 10 | LOAISP | Khoá chính MaLoaiSP,  NOT NULL TenLoaiSP |
| 11 | HOA\_DON\_UD | Khóa chính MaHD\_UD ,  Khóa ngoại MaUD đến bảng UNGDUNG,  Khóa ngoại MaNV đến bảng NHANVIEN,  Khóa ngoại SDT đến bảng KHACHHANG,  CHECK Ngay (>=0),  NOT NULL GiaTriHD |
| 12 | ChiTiet\_HD\_UD | Khóa chính (MaHD\_UD, MaSP),  Khóa ngoại MaHD\_UD đến bảng HOA\_DON\_UD,  Khóa ngoại MaSP đến bảng SANPHAM,  CHECK SoLuong(>0),  CHECK DonGia(>=0),  CHECK TongTien(>=0) |
| 13 | UNGDUNG | Khoá chính MaUD,  NOT NULL TenUD,  NOT NULL ChietKhau |
| 14 | NGUYENLIEU | Khoá chính MaNL,  Khoá ngoại MaNCC đến bảng NHACUNGCAP,  NOT NULL TenNL,  NOT NULL DonVi,  CHECK SoLuong(>0) |
| 15 | CheBien | Khóa chính(MaSP, MaNL),  Khoá ngoại MaSP đến bảng SANPHAM,  Khoá ngoại MaNL đến bảng NGUYENLIEU,  NOT NULL LieuLuong,  NOT NULL DonVi |
| 16 | NHACUNGCAP | Khóa chính MaNCC,  NOT NULL TenNCC,  NOT NULL SDT,  CHECK SDT (=10) |
| 17 | ChiTietDonNhap | Khóa chính (MaHDN, MaNL),  Khóa ngoại MaHDN đến bảng HOADONNHAP,  Khóa ngoại MaNL đến bảng NGUYENLIEU,  CHECK DonGia(>=0),  CHECK SL(>0),  NOT NULL DonVi  CHECK TongTien(>=0) |
| 18 | HOADONNHAP | Khoá chính MaHDN,  Khoá ngoại MaNCC đến bảng NHACUNGCAP.  CHECK NgayNhap (>=0),  NOT NULL TriGiaDonNhap |

# 2.4 Cài đặt CSDL và các ràng buộc

***Bảng vị trí công việc***

CREATE TABLE ViTriCongViec(

MaViTri nchar(10) CONSTRAINT PK\_ViTri PRIMARY KEY,

TenViTri nvarchar(50) NOT NULL,

PhuCapLuong float check (PhuCapLuong > 0)

)

***Bảng loại nhân viên***

CREATE TABLE LoaiNhanVien(

MaLoaiNV nchar(10) CONSTRAINT PK\_LoaiNV PRIMARY KEY,

TenLoaiNV nvarchar(50) NOT NULL,

LuongCB float check (LuongCB > 0),

)

***Bảng nhân viên***

CREATE TABLE NhanVien(

MaNV nchar(10) CONSTRAINT PK\_NhanVien PRIMARY KEY,

TenNV nvarchar(10) NOT NULL,

NgaySinh date check (DATEDIFF(year, NgaySinh, GETDATE())>=18),

GioiTinh nvarchar(3) NOT NULL,

DiaChi nvarchar(100),

SDT nchar(11) NOT NULL check (len(SDT)=10),

MaLoaiNV nchar(10) CONSTRAINT FK\_NhanVien\_Loai FOREIGN KEY REFERENCES LoaiNhanVien(MaLoaiNV),

MaViTri nchar(10) CONSTRAINT FK\_NhanVien\_ViTri FOREIGN KEY REFERENCES ViTriCongViec(MaViTri),

NgayTuyenDung date check (DATEDIFF(day, NgayTuyenDung, GETDATE())>=0)

)

***Bảng khách hàng***

CREATE TABLE KhachHang(

SDT nchar(11) CONSTRAINT PK\_KhachHang PRIMARY KEY check (len(SDT)=10),

TenKH nvarchar(50) NOT NULL,

DiaChi nchar(100)

)

***Bảng nhà cung cấp***

CREATE TABLE NhaCungCap(

MaNCC nchar(10) CONSTRAINT PK\_NhaCungCap PRIMARY KEY,

TenNCC nvarchar(50) NOT NULL,

DiaChi nchar(100),

SDT nchar(10) NOT NULL check (len(SDT)=10)

)

***Bảng hóa đơn nhập***

CREATE TABLE HoaDonNhap(

MaHDN nchar(10) CONSTRAINT PK\_HoaDonNhap PRIMARY KEY,

NgayNhap date check (DATEDIFF(day, NgayNhap, GETDATE())>=0),

TriGiaDonNhap float NOT NULL,

MaNCC nchar(10) CONSTRAINT FK\_HoaDonNhap\_NCC FOREIGN KEY REFERENCES NhaCungCap(MaNCC),

)

***Bảng nguyên liệu***

CREATE TABLE NguyenLieu(

MaNL nchar(10) CONSTRAINT PK\_NguyenLieu PRIMARY KEY,

TenNL nvarchar(50) NOT NULL,

MaNCC nchar(10) CONSTRAINT FK\_NguyenLieu\_NCC FOREIGN KEY REFERENCES NhaCungCap(MaNCC),

SoLuong int check (SoLuong>0),

DonVi nchar(10) NOT NULL,

)

***Bảng chi tiết đơn nhập***

CREATE TABLE ChiTietDonNhap(

MaHDN nchar(10) CONSTRAINT FK\_ChiTietNH\_DonNH FOREIGN KEY REFERENCES HoaDonNhap(MaHDN),

MaNL nchar(10) CONSTRAINT FK\_ChiTietDN\_NL FOREIGN KEY REFERENCES NguyenLieu(MaNL),

DonGia float check (DonGia>=0),

SoLuong int check (SoLuong>0),

DonVi nchar(10) NOT NULL,

TongTien float check (TongTien>=0),

CONSTRAINT PK\_ChiTietNhapHang PRIMARY KEY (MaHDN, MaNL)

)

***Bảng loại sản phẩm***

CREATE TABLE LoaiSanPham(

MaLoaiSP nchar(10) CONSTRAINT PK\_LoaiSanPham PRIMARY KEY,

TenLoaiSP nvarchar(50) NOT NULL

)

***Bảng sản phẩm***

CREATE TABLE SanPham(

MaSP nchar(10) CONSTRAINT PK\_MaSP PRIMARY KEY,

TenSP nvarchar(50) NOT NULL,

DonGia float check (DonGia>0),

TinhTrang nchar(10) DEFAULT N'Hết hàng',

MaLoaiSP nchar(10) CONSTRAINT FK\_SanPham\_LoaiSP FOREIGN KEY REFERENCES LoaiSanPham(MaLoaiSP)

)

***Bảng ứng dụng***

CREATE TABLE UngDung(

MaUD nchar(10) CONSTRAINT PK\_UngDung PRIMARY KEY,

TenUD nvarchar(50) NOT NULL,

ChietKhau int NOT NULL

)

***Bảng hóa đơn***

CREATE TABLE HoaDonBan(

MaHDB nchar(10) CONSTRAINT PK\_HoaDonBan PRIMARY KEY,

Ngay date check (DATEDIFF(year, Ngay, GETDATE())>=0),

SDT nchar(11) CONSTRAINT FK\_HoaDon\_KH FOREIGN KEY REFERENCES KhachHang(SDT),

MaNV nchar(10) CONSTRAINT FK\_HoaDon\_NV FOREIGN KEY REFERENCES NhanVien(MaNV),

ThanhTien int NOT NULL

)

***Bảng chi tiết đơn bán***

CREATE TABLE ChiTietDonBan(

MaHDB nchar(10) CONSTRAINT FK\_ChiTietDB\_DB FOREIGN KEY REFERENCES HoaDonBan(MaHDB),

MaSP nchar(10) CONSTRAINT FK\_ChiTietDB\_SP FOREIGN KEY REFERENCES SanPham(MaSP),

SoLuong int check (SoLuong>0),

DonGia float check (DonGia>=0),

TongTien float check (TongTien>=0)

CONSTRAINT PK\_ChiTietHD PRIMARY KEY (MaHDB, MaSP)

)

***Bảng hóa đơn ứng dụng***

CREATE TABLE HoaDonUngDung(

MaHD\_UD nchar(10) CONSTRAINT PK\_HoaDonUngDung PRIMARY KEY,

Ngay date check (DATEDIFF(day, Ngay, GETDATE())>=0),

SDT nchar(11) CONSTRAINT FK\_HoaDonUD\_KH FOREIGN KEY REFERENCES KhachHang(SDT),

MaUD nchar(10) CONSTRAINT FK\_HoaDonUngDung\_UD FOREIGN KEY REFERENCES UngDung(MaUD),

MaNV nchar(10) CONSTRAINT FK\_HoaDonUngDung\_NV FOREIGN KEY REFERENCES NhanVien(MaNV),

GiaTriHD int NOT NULL

)

***Bảng chi tiết hóa đơn ứng dụng***

CREATE TABLE ChiTietHoaDonUngDung(

MaHD\_UD nchar(10) CONSTRAINT FK\_ChiTietHDUD\_HDUD FOREIGN KEY REFERENCES HoaDonUngDung(MaHD\_UD),

MaSP nchar(10) CONSTRAINT FK\_ChiTietHDUD\_SP FOREIGN KEY REFERENCES SanPham(MaSP),

SoLuong int check (SoLuong>0),

DonGia float check (DonGia>=0),

TongTien float check (TongTien>=0),

CONSTRAINT PK\_ChiTietHDUD PRIMARY KEY (MaHD\_UD, MaSP)

)

***Bảng ca làm việc***

CREATE TABLE CaLamViec(

MaCa nchar(10),

Ngay date check (DATEDIFF(day, Ngay, GETDATE())>=0),

GioBatDau nchar(10) NOT NULL,

GioKetThuc nchar(10) NOT NULL,

CONSTRAINT PK\_CaLamViec PRIMARY KEY (MaCa, Ngay)

)

***Bảng phân ca làm cho nhân viên***

CREATE TABLE BangPhanCa(

MaCa nchar(10),

MaNV nchar(10) CONSTRAINT FK\_PhanCa\_NV FOREIGN KEY REFERENCES NhanVien(MaNV),

Ngay date check (DATEDIFF(day, Ngay, GETDATE())>=0),

CONSTRAINT PK\_BangPhanCa PRIMARY KEY (MaCa, MaNV, Ngay),

CONSTRAINT FK\_PhanCa\_Ca FOREIGN KEY (MaCa, Ngay) REFERENCES CaLamViec(MaCa, Ngay)

)

***Bảng chế biến***

CREATE TABLE CheBien(

MaSP nchar(10) CONSTRAINT FK\_CheBien\_SP FOREIGN KEY REFERENCES SanPham(MaSP),

MaNL nchar(10) CONSTRAINT FK\_CheBien\_NL FOREIGN KEY REFERENCES NguyenLieu(MaNL),

LieuLuong int check (LieuLuong>0),

DonVi nchar(10) NOT NULL

CONSTRAINT PK\_CheBien PRIMARY KEY (MaSP, MaNL))­



# 2.5 Các view

## 2.5.1 Xem ca làm việc của nhân viên

CREATE VIEW [dbo].[v\_LoadBangPhanCa] AS

SELECT CaLamViec.MaCa, NhanVien.MaNV, CaLamViec.Ngay

FROM BangPhanCa, CaLamViec, NhanVien

WHERE BangPhanCa.MaCa = CaLamViec.MaCa AND BangPhanCa.MaNV = NhanVien.MaNV AND BangPhanCa.Ngay = CaLamViec.Ngay

## 2.5.2 Xem danh sách sản phẩm

CREATE VIEW [dbo].[v\_LoadSanPham] AS

SELECT \*

FROM SanPham

## 2.5.3 Xem số lượng sản phẩm đã bán trong ngày

CREATE VIEW V\_SoLuongSanPhamDaBanTrongNgay AS

SELECT sp.MaSP, SUM(cthd.SL) AS SoLuongDaBanTaiQuay, SUM(cthdUD.SoLuong)

AS SoLuongDaBanQuaUD

FROM dbo.SanPham sp

JOIN dbo.ChiTietDonBan cthd ON sp.MaSP = cthd.MaSP

JOIN dbo.HoaDonBan hd ON cthd.mahd = hd.MaHD

JOIN dbo.ChiTietHoaDonUngDung cthdUD ON sp.MaSP = cthdUD.MaSP

JOIN dbo.HoaDonUngDung hdUD ON cthdUD.MaHD\_UD = hdud.MaHD\_UD

WHERE hd.NgayDatHang = CONVERT(DATE, GETDATE()) AND hdud.NgayDatHang = CONVERT(DATE,

GETDATE())

GROUP BY sp.MaSP

## 2.5.4 Xem thông tin nhân viên

CREATE VIEW [dbo].[v\_NhanVien] AS

SELECT \*

FROM dbo.NhanVien

## 2.5.5 Xem doanh thu theo từng ngày

CREATE VIEW V\_DoanhThuNgay AS

SELECT Ngay, SUM(ThanhTien) AS DoanhThuNgay

FROM dbo.HoaDonBan

GROUP BY Ngay

## 2.5.6 Xem doanh thu theo từng tháng

CREATE VIEW V\_DoanhThuThang AS

SELECT MONTH(Ngay) AS Thang, YEAR(Ngay) AS Nam, SUM(ThanhTien) AS DoanhThuThang

FROM dbo.HoaDonBan

GROUP BY MONTH(Ngay), YEAR(Ngay)

## 2.5.7 Xem lợi nhuận theo từng ngày

CREATE VIEW V\_LoiNhuanNgay AS

SELECT Ngay, (DoanhThuNgay - ChiPhiNhap) AS LoiNhuanNgay

FROM V\_DoanhThuNgay vdtn JOIN (SELECT NgayNhap, SUM(TriGiaDonNhap) AS ChiPhiNhap

FROM dbo.HoaDonNhap

GROUP BY NgayNhap)Q ON vdtn.Ngay = Q.NgayNhap

GROUP BY Ngay, DoanhThuNgay, Q.ChiPhiNhap

## 2.5.8 Xem lợi nhuận theo từng tháng

CREATE VIEW V\_LoiNhuanThang AS

SELECT Ngay, SUM (LoiNhuanNgay) AS LoiNhuanThang

FROM V\_LoiNhuanNgay

GROUP BY MONTH(Ngay), YEAR(Ngay)

## 2.5.9 Xem số lượng nguyên liệu còn lại

CREATE VIEW V\_SLNguyenLieuConLai AS

SELECT \*

FROM dbo.NguyenLieu

# 2.6 Các trigger

## 2.6.1 Kiểm tra xem nguyên liệu trong kho còn đủ để đáp ứng số lượng sản phẩm khách hàng yêu cầu hay không

CREATE TRIGGER TG\_SPHetHang

ON ChiTietHoaDon

AFTER INSERT

AS

BEGIN

Declare @newMaSP nchar(10), @newSL int, @MaNLMin nchar(10), @SLMin int, @LieuLuongCan float

Select @newMaSP = ne.MaSP, @newSL = ne.SL

From inserted ne

--

Select @SLMin=min(NguyenLieu.SL)

From NguyenLieu, CheBien

Where NguyenLieu.MaNL = CheBien.MaNL

and CheBien.MaSP = @newMaSP

--

Select TOP 1 @MaNLMin = NguyenLieu.MaNL, @LieuLuongCan = CheBien.LieuLuong

From NguyenLieu, CheBien

Where NguyenLieu.MaNL = CheBien.MaNL

and CheBien.MaSP = @newMaSP

and NguyenLieu.SL = @SLMin

--

IF (@newSL\*@LieuLuongCan - @SLMin\*1000 > 0)

BEGIN

UPDATE SanPham

SET TinhTrang = 'Hết hàng'

WHERE MaSP=@newMaSP

ROLLBACK

END

END

## 2.6.2 Kiểm tra tên sản phẩm có bị trùng không

-- Trigger kiểm tra tên sản phẩm trùng

CREATE TRIGGER TG\_TrungTenSP

ON dbo.SanPham

AFTER INSERT, UPDATE

AS

BEGIN

-- Kiểm tra tên sản phẩm vừa thêm có bị trùng lặp

IF EXISTS (

SELECT \*

FROM inserted i

WHERE EXISTS (

SELECT \*

FROM dbo.SanPham sp

WHERE sp.TenSP = i.TenSP AND sp.MaSP <> i.MaSP

)

)

BEGIN

-- Nếu trùng thì rollback

RAISERROR ('Tên sản phẩm bị trùng', 16, 1)

ROLLBACK;

END

## 2.6.3 Trigger bắt lỗi khi thêm hoặc sửa hóa đơn bán

CREATE TRIGGER trg\_CheckHoaDonBan

ON HoaDonBan

FOR INSERT, UPDATE

AS

BEGIN

-- Check MaHDB

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaHDB) = '')

BEGIN

RAISERROR('Mã hóa đơn không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check Ngay

IF EXISTS (SELECT \* FROM inserted WHERE DATEDIFF(year, Ngay, GETDATE()) < 0)

BEGIN

RAISERROR('Ngày hóa đơn không thể là trong tương lai', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check SDT

IF NOT EXISTS (SELECT \* FROM KhachHang WHERE SDT = (SELECT SDT FROM inserted))

BEGIN

RAISERROR('Số điện thoại không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS (SELECT \* FROM inserted WHERE LEN(SDT) <> 10)

BEGIN

RAISERROR('Số điện thoại phải có đúng 10 ký tự', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check MaNV

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNV) = '')

BEGIN

RAISERROR('Mã nhân viên không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM NhanVien WHERE MaNV = (SELECT MaNV FROM inserted))

BEGIN

RAISERROR('Mã nhân viên không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check ThanhTien

IF EXISTS (SELECT \* FROM inserted WHERE ThanhTien < 0)

BEGIN

RAISERROR('Giá trị Thành Tiền không hợp lệ', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

## 2.6.4 Trigger bắt lỗi khi thêm hoặc sửa chi tiết hóa đơn bán

CREATE TRIGGER trg\_CheckChiTietDonBan

ON ChiTietDonBan

FOR INSERT, UPDATE

AS

BEGIN

-- Check MaHDB

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaHDB) = '')

BEGIN

RAISERROR('Mã hóa đơn không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM HoaDonBan WHERE MaHDB = (SELECT MaHDB FROM inserted))

BEGIN

RAISERROR('Mã hóa đơn không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check MaSP

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaSP) = '')

BEGIN

RAISERROR('Mã sản phẩm không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM SanPham WHERE MaSP = (SELECT MaSP FROM inserted))

BEGIN

RAISERROR('Mã sản phẩm không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check SoLuong

IF EXISTS (SELECT \* FROM inserted WHERE SoLuong <= 0)

BEGIN

RAISERROR('Số lượng phải lớn hơn 0', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check DonGia

IF EXISTS (SELECT \* FROM inserted WHERE DonGia < 0)

BEGIN

RAISERROR('Đơn giá không được âm', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check TongTien

IF EXISTS (SELECT \* FROM inserted WHERE TongTien < 0)

BEGIN

RAISERROR('Tổng tiền không được âm', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

## 2.6.5 Trigger bắt lỗi khi thêm hoặc sửa nhà cung cấp

CREATE TRIGGER trg\_CheckNhaCungCap

ON NhaCungCap

FOR INSERT, UPDATE

AS

BEGIN

-- Check MaNCC

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNCC) = '')

BEGIN

RAISERROR('Mã nhà cung cấp không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check TenNCC

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenNCC) = '')

BEGIN

RAISERROR('Tên nhà cung cấp không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check SDT

IF EXISTS (SELECT \* FROM inserted WHERE LEN(SDT) <> 10)

BEGIN

RAISERROR('Số điện thoại nhà cung cấp phải có đúng 10 ký tự', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

## 2.6.6 Trigger bắt lỗi khi thêm hoặc sửa hóa đơn nhập

CREATE TRIGGER trg\_CheckHoaDonNhap

ON HoaDonNhap

FOR INSERT, UPDATE

AS

BEGIN

-- Check MaHDN

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaHDN) = '')

BEGIN

RAISERROR('Mã hóa đơn nhập không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check NgayNhap

IF EXISTS (SELECT \* FROM inserted WHERE DATEDIFF(DAY, NgayNhap, GETDATE()) < 0)

BEGIN

RAISERROR('Ngày nhập không được lớn hơn ngày hiện tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check TriGiaDonNhap

IF EXISTS (SELECT \* FROM inserted WHERE TriGiaDonNhap < 0)

BEGIN

RAISERROR('Trị giá đơn nhập không được âm', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check MaNCC

IF NOT EXISTS (SELECT \* FROM NhaCungCap WHERE MaNCC = (SELECT MaNCC FROM inserted))

BEGIN

RAISERROR('Mã nhà cung cấp không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

GO

## 2.6.7 Trigger bắt lỗi khi thêm hoặc sửa chi tiết hóa đơn nhập

CREATE TRIGGER trg\_CheckChiTietDonNhap

ON ChiTietDonNhap

FOR INSERT, UPDATE

AS

BEGIN

-- Check MaHDN

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaHDN) = '')

BEGIN

RAISERROR('Mã hóa đơn nhập không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM HoaDonNhap WHERE MaHDN = (SELECT MaHDN FROM inserted))

BEGIN

RAISERROR('Mã hóa đơn nhập không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check MaNL

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNL) = '')

BEGIN

RAISERROR('Mã nguyên liệu không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM NguyenLieu WHERE MaNL = (SELECT MaNL FROM inserted))

BEGIN

RAISERROR('Mã nguyên liệu không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check DonGia

IF EXISTS (SELECT \* FROM inserted WHERE DonGia < 0)

BEGIN

RAISERROR('Đơn giá không được âm', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check SoLuong

IF EXISTS (SELECT \* FROM inserted WHERE SoLuong <= 0)

BEGIN

RAISERROR('Số lượng phải lớn hơn 0', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check DonVi

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(DonVi) = '')

BEGIN

RAISERROR('Đơn vị không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check TongTien

IF EXISTS (SELECT \* FROM inserted WHERE TongTien < 0)

BEGIN

RAISERROR('Tổng tiền không được âm', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

GO

## 2.6.8 Trigger bắt lỗi khi thêm hoặc sửa nhân viên

CREATE TRIGGER trg\_CheckNhanVien

ON NhanVien

FOR INSERT, UPDATE

AS

BEGIN

-- check MaNV

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNV) = ' ')

BEGIN

RAISERROR('Mã nhân viên không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM NhanVien WHERE MaNV IN (SELECT MaNV FROM

inserted))

BEGIN

RAISERROR('Mã nhân viên đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ho ten nhan vien

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenNV) = '' )

BEGIN

RAISERROR('Tên nhân viên không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NhanVien n ON i.MaNV != n.MaNV

AND TRIM(i.TenNV) = TRIM(n.TenNV))

BEGIN

RAISERROR('Tên nhân viên đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ngay sinh

IF EXISTS (SELECT \* FROM inserted WHERE

datediff(year,inserted.NgaySinh,getdate())<(18))

BEGIN

RAISERROR ('Nhân viên phải trên 18 tuổi', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check dia chi

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(DiaChi) = '' )

BEGIN

RAISERROR('Địa chỉ không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check so dien thoai

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(SDT) = '' )

BEGIN

RAISERROR('Số điện thoại không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM inserted WHERE len(TRIM(SDT)) = (10))

BEGIN

RAISERROR('Số điện thoại gồm 10 chữ số', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NhanVien n ON i.MaNV != n.MaNV

AND TRIM(i.SDT) = TRIM(n.SDT))

BEGIN

RAISERROR('Số điện thoại đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ngay tuyen dung

IF NOT EXISTS (SELECT \* FROM inserted WHERE

(datediff(day,[NgayTuyenDung],getdate())>=(0)))

BEGIN

RAISERROR ('Ngày tuyển dụng không thể là trong tương lai', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

## 2.6.9 Trigger bắt lỗi khi thêm khách hàng

CREATE TRIGGER trg\_InsertNewKhachHang

ON KhachHang

FOR INSERT, UPDATE

AS

BEGIN

-- check SDT

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(SDT) = ' ')

BEGIN

RAISERROR('Số điện thoại khách hàng không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM KhachHang WHERE SDT IN (SELECT SDT FROM

inserted))

BEGIN

RAISERROR('SDT dã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END;

## 2.6.9 Trigger bắt lỗi khi thêm ứng dụng

CREATE TRIGGER trg\_ThemUngDungMoi

ON UngDung

FOR INSERT, UPDATE

AS

BEGIN

-- check MaUD

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaUD) = ' ')

BEGIN

RAISERROR('Ma Ung dung khong duoc trung', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM UngDung WHERE MaUD IN (SELECT MaUD FROM

inserted))

BEGIN

RAISERROR('Ma ung dung dã t?n t?i', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ten ung dung

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenUD) = ' ')

BEGIN

RAISERROR('Tên ung dung không du?c d? tr?ng', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END;

## 2.6.10 Trigger bắt lỗi khi thêm loại sản phẩm

CREATE TRIGGER trg\_ThemLoaiSanPham

ON LoaiSanPham

FOR INSERT, UPDATE

AS

BEGIN

-- check MaLoaiSP

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaLoaiSP) = ' ')

BEGIN

RAISERROR('Mã loại sản phẩm không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM LoaiSanPham WHERE MaLoaiSP IN (SELECT MaLoaiSP FROM

inserted))

BEGIN

RAISERROR('Mã loại sản phẩm đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check TenLoaiSP

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenLoaiSP) = ' ')

BEGIN

RAISERROR('Tên loại sản phẩm không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

## 2.6.10 Trigger bắt lỗi khi thêm loại sản phẩm

-- Trigger bắt lỗi trùng tên khi thêm và sửa sản phẩm

CREATE TRIGGER trg\_TrungTenSanPham

ON dbo.SanPham

AFTER INSERT, UPDATE

AS

BEGIN

-- Kiểm tra tên sản phẩm vừa thêm có bị trùng lặp

IF EXISTS (

SELECT \*

FROM inserted i

WHERE EXISTS (

SELECT \*

FROM dbo.SanPham sp

WHERE sp.TenSP = i.TenSP AND sp.MaSP <> i.MaSP

)

)

BEGIN

-- Nếu trùng thì rollback

RAISERROR ('Tên sản phẩm bị trùng', 16, 1)

ROLLBACK;

END

END

## 2.6.11 Trigger tự động tạo tài khoản khi thêm nhân viên

CREATE OR ALTER TRIGGER Trg\_Create\_Account

ON NhanVien

AFTER INSERT

AS

BEGIN

INSERT INTO Account(username, password, MaNV, roles)

SELECT

inserted.MaNV,

'12345678',

inserted.MaNV,

CASE

WHEN NhanVien.MaViTri = 'vt01' THEN 'sysadmin'

ELSE 'Staff'

END AS roles

FROM inserted

INNER JOIN NhanVien ON inserted.MaNV = NhanVien.MaNV;

END;

GO

## 2.6.11 Trigger tự động phân quyền cho tài khoản của nhân viên

CREATE OR ALTER TRIGGER AddQuyen ON Account

AFTER INSERT

AS

DECLARE @tendangnhap nvarchar(10), @matkhau nvarchar(30), @roles nvarchar(20)

SELECT @tendangnhap=nl.username, @matkhau=nl.password, @roles=nl.roles

FROM inserted nl

BEGIN

DECLARE @sqlString nvarchar(2000)

----

SET @sqlString= ' CREATE LOGIN [' + @tendangnhap +'] WITH PASSWORD= '''+ @matkhau

+''', DEFAULT\_DATABASE=[QlQuanTraSua], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

SET @sqlString= ' CREATE USER ' + @tendangnhap +' FOR LOGIN '+ @tendangnhap

EXEC (@sqlString)

if (@roles = 'sysadmin')

SET @sqlString = ' ALTER SERVER ROLE sysadmin ' + ' ADD MEMBER ' + @tendangnhap;

if (@roles = 'Staff')

SET @sqlString = ' ALTER ROLE Staff ADD MEMBER ' + @tendangnhap;

EXEC (@sqlString)

END

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

# 3.1 Kết nối cơ sở dữ liệu

namespace QLQuanTraSua.DB\_Layer

{

class DBMain

{

SqlConnection con = new SqlConnection("Data Source=.;Initial Catalog=QLQuanTraSua;Integrated Security=True");

public SqlConnection getConnection

{

get

{

return con;

}

}

SqlConnection conAdmin = new SqlConnection("Data Source=.;Initial Catalog=QLQuanTraSua;Integrated Security=True");

public SqlConnection getConnectionAdmin

{

get

{

return conAdmin;

}

}

// open the connection

public void openConnection()

{

if (con.State == ConnectionState.Closed)

{

con.Open();

}

}

public void openConnectionAdmin()

{

if (conAdmin.State == ConnectionState.Closed)

{

conAdmin.Open();

}

}

public void closeConnection()

{

if (con.State == ConnectionState.Open)

{

con.Close();

}

}

public void closeConnectionAdmin()

{

if (conAdmin.State == ConnectionState.Open)

{

conAdmin.Close();

}

}

}

}

# 3.2 Xem thông tin danh mục

## 3.2.1. Danh mục nhân viên

CREATE VIEW [dbo].[v\_LoadNhanVien] AS

SELECT MaNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, LoaiNhanVien.MaLoaiNV, ViTriCongViec.MaViTri, NgayTuyenDung

FROM NhanVien, LoaiNhanVien, ViTriCongViec

WHERE NhanVien.MaLoaiNV = LoaiNhanVien.MaLoaiNV AND NhanVien.MaViTri = ViTriCongViec.MaViTri

GO

Thực thi trên C#:

public DataTable LayNhanVien()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadNhanVien", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.2. Danh mục bảng phân ca

CREATE VIEW [dbo].[v\_LoadBangPhanCa] AS

SELECT CaLamViec.MaCa, NhanVien.MaNV, CaLamViec.Ngay

FROM BangPhanCa, CaLamViec, NhanVien

WHERE BangPhanCa.MaCa = CaLamViec.MaCa AND BangPhanCa.MaNV = NhanVien.MaNV AND BangPhanCa.Ngay = CaLamViec.Ngay

GO

Thực thi trên C#:

public DataTable LayBangPhanCa()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadBangPhanCa", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.3. Danh mục ca làm việc

CREATE VIEW [dbo].[v\_LoadCaLamViec] AS

SELECT \*

FROM CaLamViec

GO

Thực thi trên C#:

public DataTable LayCaLamViec()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadCaLamViec", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.4. Danh mục hóa đơn

### 3.2.4.1. Danh mục hóa đơn bán

CREATE VIEW [dbo].[v\_HoaDonBan] AS

SELECT \*

FROM dbo.HoaDonBan

GO

Thực thi trên C#:

public DataTable LayHoaDonBan()

{

SqlCommand command = new SqlCommand("select \* from v\_HoaDonBan", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

-- View Chi tiết đơn bán

CREATE VIEW [dbo].[v\_ChiTietDonBan] AS

SELECT \*

FROM dbo.ChiTietDonBan

GO

Thực thi trên C#:

public DataTable LayCTHDB()

{

SqlCommand command = new SqlCommand("select \* from v\_ChiTietDonBan", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

### 3.2.4.2. Danh mục hóa đơn ứng dụng

CREATE VIEW [dbo].[v\_HoaDonUngDung] AS

SELECT \*

FROM HoaDonUngDung

GO

Thực thi trên C#:

public DataTable LayHoaDonUngDung()

{

SqlCommand command = new SqlCommand("select \* from v\_HoaDonUngDung", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

-- BẢNG CHI TIẾT ĐƠN UNG DUNG

CREATE VIEW [dbo].[v\_ChiTietHoadonUngDung] AS

SELECT \*

FROM ChiTietHoaDonUngDung

GO

Thực thi trên C#:

public DataTable LayChiTietHoaDonUngDung()

{

SqlCommand command = new SqlCommand("select \* from v\_ChiTietHoaDonUngDung", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.5. Danh mục loại sản phẩm

CREATE VIEW [dbo].[v\_LoadLoaiSanPham] AS

SELECT \*

FROM LoaiSanPham

GO

Thực thi trên C#:

public DataTable LayLoaiSanPham()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadLoaiSanPham", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.6 Danh mục sản phẩm

CREATE VIEW [dbo].[v\_LoadSanPham] AS

SELECT \*

FROM SanPham

GO

Thực thi trên C#:

public DataTable LaySanPham()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadSanPham", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.7. Danh mục phiếu chi

-- View Hóa đơn nhập

CREATE VIEW [dbo].[v\_HoaDonNhap] AS

SELECT \*

FROM dbo.HoaDonNhap

GO

Thực thi trên C#:

public DataTable LayHoaDonNhap()

{

SqlCommand command = new SqlCommand("select \* from v\_HoaDonNhap", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

-- View chi tiết đơn nhập

CREATE VIEW [dbo].[v\_ChiTietDonNhap] AS

SELECT \*

FROM dbo.ChiTietDonNhap

GO

Thực thi trên C#:

public DataTable LayCTHDN()

{

SqlCommand command = new SqlCommand("select \* from v\_ChiTietDonNhap", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.8. Danh mục nhà cung cấp

CREATE VIEW [dbo].[v\_NhaCungCap] AS

SELECT \*

FROM dbo.NhaCungCap

GO

Thực thi trên C#:

public DataTable LayNhaCungCap()

{

SqlCommand command = new SqlCommand("select \* from v\_NhaCungCap", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.9. Danh mục khách hàng

CREATE VIEW [dbo].[v\_KhachHang] AS

SELECT \*

FROM KhachHang

GO

Thực thi trên C#:

public DataTable LayKhachHang()

{

SqlCommand command = new SqlCommand("select \* from v\_KhachHang", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.10. Danh mục nguyên liệu

CREATE VIEW [dbo].[v\_NguyenLieu] AS

SELECT \*

FROM dbo.NguyenLieu

GO

Thực thi trên C#:

public DataTable LayNguyenLieu()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadNguyenLieu", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.11. Danh mục cách chế biến

CREATE VIEW [dbo].[v\_LoadCachCheBien] AS

SELECT SanPham.TenSP, NguyenLieu.TenNL, CheBien.LieuLuong, CheBien.DonVi

FROM dbo.CheBien

INNER JOIN dbo.SanPham ON SanPham.MaSP = CheBien.MaSP

INNER JOIN dbo.NguyenLieu ON CheBien.MaNL = NGUYENLIEU.MaNL

GO

Thực thi trên C#:

public DataTable LayCheBien()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadCheBien", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.12. Danh mục vị trí làm việc

CREATE VIEW [dbo].[v\_LoadViTriCongViec] AS

SELECT \*

FROM ViTriCongViec

GO

Thực thi trên C#:

public DataTable LayViTriCV()

{

SqlCommand command = new SqlCommand("select \* from v\_LoadViTriCongViec", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

## 3.2.13. Danh mục ứng dụng

CREATE VIEW [dbo].[v\_UngDung] AS

SELECT \*

FROM UngDung

GO

Thực thi trên C#:

public DataTable LayUngDung()

{

SqlCommand command = new SqlCommand("select \* from v\_UngDung", db.getConnection);

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

return dt;

}

# 3.3 Quản lý khách hàng

## 3.3.1. Thêm khách hàng mới

CREATE or alter PROCEDURE [dbo].[proc\_ThemKhachHang]

@SDT nchar(11),

@TenKH nvarchar(50),

@DiaChi nvarchar(100)

AS

BEGIN

IF EXISTS (SELECT 1 FROM KhachHang WHERE SDT = @SDT)

BEGIN

RETURN;

END

INSERT INTO KhachHang (SDT, TenKH, DiaChi)

VALUES (@SDT, @TenKH, @DiaChi)

END

Thực thi trên C#:

public bool ThemKhachHang(string SDT, string TenKH, string DiaChi)

{

SqlCommand command = new SqlCommand("proc\_ThemKhachHang", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@TenKH", SqlDbType.NVarChar).Value = TenKH;

command.Parameters.Add("@DiaChi", SqlDbType.NChar).Value = DiaChi;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.3.2. Sửa thông tin khách hàng

CREATE PROCEDURE [dbo].[proc\_SuaNhaKhachHang]

@SDT nchar(11),

@TenKH nvarchar(50),

@DiaChi nchar(100)

AS

BEGIN

BEGIN TRY

-- Sửa thông tin hóa đơn bán

UPDATE dbo.KhachHang

SET SDT = @SDT, TenKH = @TenKH, DiaChi = @DiaChi

WHERE SDT = @SDT

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatKhachHang(string SDT, string TenKH, string DiaChi)

{

SqlCommand command = new SqlCommand("proc\_SuaNhaKhachHang", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@TenKH", SqlDbType.NVarChar).Value = TenKH;

command.Parameters.Add("@DiaChi", SqlDbType.NChar).Value = DiaChi;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.3.3. Xóa khách hàng

CREATE PROCEDURE [dbo].[proc\_XoaKhachHang]

@SDT nchar(11)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRANSACTION;

BEGIN TRY

DELETE FROM KhachHang WHERE SDT = @SDT;

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

ROLLBACK TRANSACTION;

THROW;

END CATCH

COMMIT TRANSACTION;

END

Thực thi trên C#:

public bool XoaKhachHang(string SDT)

{

SqlCommand command = new SqlCommand("proc\_XoaKhachHang", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.3.4. Tìm kiếm khách hàng thông qua số điện thoại

CREATE FUNCTION [dbo].[TimKiemKHBangSDT](@SDT nchar(11))

RETURNS TABLE

AS

RETURN

(

SELECT \*

FROM KhachHang

WHERE SDT = @SDT

);

## 3.3.5. Trigger bắt lỗi khi thêm khách hàng mới

CREATE TRIGGER trg\_ThemMoiKhachHang

ON KhachHang

FOR INSERT, UPDATE

AS

BEGIN

-- check SDT

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(SDT) = ' ')

BEGIN

RAISERROR('Số điện thoại khách hàng không được trùng', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM KhachHang WHERE SDT IN (SELECT SDT FROM

inserted))

BEGIN

RAISERROR('SDT đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END;

# 3.4 Quản lý hóa đơn

## 3.4.1. Hóa đơn nhập

### 3.4.1.1. Thêm hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_ThemHoaDonNhap]

@MaHDN nchar(10),

@NgayNhap date,

@TriGiaDonNhap float,

@MaNCC nchar(10)

AS

BEGIN

INSERT INTO HoaDonNhap (MaHDN, NgayNhap, TriGiaDonNhap, MaNCC)

VALUES (@MaHDN, @NgayNhap, @TriGiaDonNhap, @MaNCC)

END

GO

Thực thi trên C#:

public bool ThemHDN(string MaHDN, string NgayNhap, string TriGiaDN, string MaNCC)

{

SqlCommand command = new SqlCommand("proc\_ThemHoaDonNhap", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDN", SqlDbType.NChar).Value = MaHDN;

command.Parameters.Add("@NgayNhap", SqlDbType.Date).Value = NgayNhap;

command.Parameters.Add("@TriGiaDonNhap", SqlDbType.Float).Value = TriGiaDN;

command.Parameters.Add("@MaNCC", SqlDbType.NChar).Value = MaNCC;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.1.2. Sửa hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_SuaHoaDonNhap]

@MaHDN nchar(10),

@NgayNhap date,

@TriGiaDonNhap float,

@MaNCC nchar(10)

AS

BEGIN

BEGIN TRY

-- Sửa thông tin hóa đơn nhập

UPDATE dbo.HoaDonNhap

SET NgayNhap = @NgayNhap, TriGiaDonNhap = @TriGiaDonNhap, MaNCC = @MaNCC

WHERE MaHDN = @MaHDN

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatHDN(string MaHDN, string NgayNhap, string MaNCC, string TriGiaDN)

{

SqlCommand command = new SqlCommand("proc\_SuaHoaDonNhap", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDN", SqlDbType.NChar).Value = MaHDN;

command.Parameters.Add("@NgayNhap", SqlDbType.Date).Value = NgayNhap;

command.Parameters.Add("@TriGiaDonNhap", SqlDbType.Float).Value = TriGiaDN;

command.Parameters.Add("@MaNCC", SqlDbType.NChar).Value = MaNCC;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.1.3. Xóa hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_XoaHoaDonNhap]

@MaHDN nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá các chi tiết hóa đơn nhập theo @MaHDN trong bảng ChiTietDonNhap

DELETE FROM dbo.ChiTietDonNhap WHERE ChiTietDonNhap.MaHDN = @MaHDN

-- Xoá hóa đơn nhập theo @MaHDN trong bảng HoaDonNhap

DELETE FROM dbo.HoaDonNhap WHERE HoaDonNhap.MaHDN = @MaHDN

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool XoaHDN(string MaHDN)

{

SqlCommand command = new SqlCommand("proc\_XoaHoaDonNhap", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDN", SqlDbType.NChar).Value = MaHDN;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.4.2. Chi tiết hóa đơn nhập

### 3.4.2.1. Thêm chi tiết hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_ThemChiTietDonNhap]

@MaHDN nchar(10),

@MaNL nchar(10),

@DonGia float,

@SoLuong int,

@DonVi nchar(10),

@TongTien float

AS

BEGIN

INSERT INTO ChiTietDonNhap (MaHDN, MaNL, DonGia, SoLuong, DonVi, TongTien)

VALUES (@MaHDN, @MaNL, @DonGia, @SoLuong, @DonVi, @TongTien)

END

GO

Thực thi trên C#:

public bool ThemCTHDN(string MaHDN, string MaNL, float DonGia, int SoLuong, string DonVi, float TongTien)

{

SqlCommand command = new SqlCommand("proc\_ThemChiTietDonNhap", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDN", SqlDbType.NChar).Value = MaHDN;

command.Parameters.Add("@MaNL", SqlDbType.NChar).Value = MaNL;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonVi", SqlDbType.NChar).Value = DonVi;

command.Parameters.Add("@TongTien", SqlDbType.Float).Value = TongTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.2.2. Sửa chi tiết hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_SuaChiTietDonNhap]

@MaHDN nchar(10),

@MaNL nchar(10),

@DonGia float,

@SoLuong int,

@DonVi nchar(10),

@TongTien float

AS

BEGIN

BEGIN TRY

-- Sửa thông tin chi tiết đơn nhập

UPDATE dbo.ChiTietDonNhap

SET DonGia = @DonGia, SoLuong = @SoLuong, DonVi = @DonVi, TongTien = @TongTien

WHERE MaHDN = @MaHDN AND MaNL = @MaNL

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatCTHDN(string MaHDN, string MaNL, float DonGia, int SoLuong, string DonVi, float TongTien)

{

SqlCommand command = new SqlCommand("proc\_SuaChiTietDonNhap", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDN", SqlDbType.NChar).Value = MaHDN;

command.Parameters.Add("@MaNL", SqlDbType.NChar).Value = MaNL;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonVi", SqlDbType.NChar).Value = DonVi;

command.Parameters.Add("@TongTien", SqlDbType.Float).Value = TongTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.2.3. Xóa chi tiết hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_XoaChiTietDonNhap]

@MaHDN nchar(10),

@MaNL nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá chi tiết đơn nhập theo @MaHDN và @MaNL trong bảng ChiTietDonNhap

DELETE FROM dbo.ChiTietDonNhap WHERE ChiTietDonNhap.MaHDN = @MaHDN AND ChiTietDonNhap.MaNL = @MaNL

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

Thực thi trên C#:

public bool XoaCTHDN(string MaHDN, string MaNL)

{

SqlCommand command = new SqlCommand("proc\_XoaChiTietDonNhap", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDN;

command.Parameters.Add("@MaNL", SqlDbType.NChar).Value = MaNL;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.4.3. Hóa đơn bán

### 3.4.3.1. Thêm hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_ThemHoaDonBan]

@MaHDB nchar(10),

@Ngay date,

@SDT nchar(11),

@MaNV nchar(10),

@ThanhTien int

AS

BEGIN

INSERT INTO HoaDonBan (MaHDB, Ngay, SDT, MaNV, ThanhTien)

VALUES (@MaHDB, @Ngay, @SDT, @MaNV, @ThanhTien)

END

GO

Thực thi trên C#:

public bool ThemHDB(string MaHDB, string Ngay, string SDT, string MaNV, int ThanhTien)

{

SqlCommand command = new SqlCommand("proc\_ThemHoaDonBan", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

command.Parameters.Add("@Ngay", SqlDbType.Date).Value = Ngay;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@ThanhTien", SqlDbType.Int).Value = ThanhTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.3.2. Sửa hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_SuaHoaDonBan]

@MaHDB nchar(10),

@Ngay date,

@SDT nchar(11),

@MaNV nchar(10),

@ThanhTien int

AS

BEGIN

BEGIN TRY

-- Sửa thông tin hóa đơn bán

UPDATE dbo.HoaDonBan

SET Ngay = @Ngay, SDT = @SDT, MaNV = @MaNV, ThanhTien = @ThanhTien

WHERE MaHDB = @MaHDB

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatHDBan(string MaHDB, string Ngay, string SDT, string MaNV, int ThanhTien)

{

SqlCommand command = new SqlCommand("proc\_SuaHoaDonBan", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

command.Parameters.Add("@Ngay", SqlDbType.Date).Value = Ngay;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@ThanhTien", SqlDbType.Int).Value = ThanhTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.3.3. Xóa hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_XoaHoaDonBan]

@MaHDB nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá các chi tiết hóa đơn bán theo @MaHDB trong bảng ChiTietDonBan

DELETE FROM dbo.ChiTietDonBan WHERE ChiTietDonBan.MaHDB = @MaHDB

-- Xoá hóa đơn bán theo @MaHDB trong bảng HoaDonBan

DELETE FROM dbo.HoaDonBan WHERE HoaDonBan.MaHDB = @MaHDB

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực hiện trên C#:

public bool XoaHDB(string MaHDB)

{

SqlCommand command = new SqlCommand("proc\_XoaHoaDonBan", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.4.4. Chi tiết hóa đơn bán

### 3.4.4.1. Thêm chi tiết hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_ThemChiTietDonBan]

@MaHDB nchar(10),

@MaSP nchar(10),

@SoLuong int,

@DonGia float,

@TongTien float

AS

BEGIN

INSERT INTO ChiTietDonBan (MaHDB, MaSP, SoLuong, DonGia, TongTien)

VALUES (@MaHDB, @MaSP, @SoLuong, @DonGia, @TongTien)

END

GO

Thực thi trên C#:

public bool ThemCTHDB(string MaHDB, string MaSP, int SoLuong, float DonGia, float TongTien)

{

SqlCommand command = new SqlCommand("proc\_ThemChiTietDonBan", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@TongTien", SqlDbType.Float).Value = TongTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.4.2. Sửa chi tiết hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_SuaChiTietDonBan]

@MaHDB nchar(10),

@MaSP nchar(10),

@SoLuong int,

@DonGia float,

@TongTien float

AS

BEGIN

BEGIN TRY

-- Sửa thông tin chi tiết đơn bán

UPDATE dbo.ChiTietDonBan

SET SoLuong = @SoLuong, DonGia = @DonGia, TongTien = @TongTien

WHERE MaHDB = @MaHDB AND MaSP = @MaSP

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatCTHDB(string MaHDB, string MaSP, int SoLuong, float DonGia, float TongTien)

{

SqlCommand command = new SqlCommand("proc\_SuaChiTietDonBan", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@TongTien", SqlDbType.Float).Value = TongTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.4.3. Xóa chi tiết hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_XoaChiTietDonBan]

@MaHDB nchar(10),

@MaSP nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá chi tiết đơn bán theo @MaHDB và @MaSP trong bảng ChiTietDonBan

DELETE FROM dbo.ChiTietDonBan WHERE ChiTietDonBan.MaHDB = @MaHDB AND ChiTietDonBan.MaSP = @MaSP

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool XoaCTHDB(string MaHDB, string MaSP)

{

SqlCommand command = new SqlCommand("proc\_XoaChiTietDonBan", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.4.4. Trigger bắt lỗi khi thêm thông tin hoặc sửa thông tin của hóa đơn bán

CREATE TRIGGER trg\_CheckHoaDonBan

ON HoaDonBan

FOR INSERT, UPDATE

AS

BEGIN

-- Check MaHDB

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaHDB) = '')

BEGIN

RAISERROR('Mã hóa đơn không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check Ngay

IF EXISTS (SELECT \* FROM inserted WHERE DATEDIFF(year, Ngay, GETDATE()) < 0)

BEGIN

RAISERROR('Ngày hóa đơn không thể là trong tương lai', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check SDT

IF NOT EXISTS (SELECT \* FROM KhachHang WHERE SDT = (SELECT SDT FROM inserted))

BEGIN

RAISERROR('Số điện thoại không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS (SELECT \* FROM inserted WHERE LEN(SDT) <> 10)

BEGIN

RAISERROR('Số điện thoại phải có đúng 10 ký tự', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check MaNV

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNV) = '')

BEGIN

RAISERROR('Mã nhân viên không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM NhanVien WHERE MaNV = (SELECT MaNV FROM inserted))

BEGIN

RAISERROR('Mã nhân viên không tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- Check ThanhTien

IF EXISTS (SELECT \* FROM inserted WHERE ThanhTien < 0)

BEGIN

RAISERROR('Giá trị Thành Tiền không hợp lệ', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

GO

## 3.4.5. Hóa đơn ứng dụng

### 3.4.5.1. Thêm hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_ThemHoaDonUngDung]

@MaHD\_UD nchar(10),

@Ngay date,

@SDT nchar(11),

@MaUD nchar(10),

@MaNV nchar(10),

@GiaTriHD int

AS

BEGIN

INSERT INTO HoaDonUngDung(MaHD\_UD, Ngay, SDT, MaUD, MaNV, GiaTriHD)

VALUES (@MaHD\_UD, @Ngay, @SDT, @MaUD, @MaNV, @GiaTriHD)

END

GO

Thực thi trên C#:

public bool ThemHoaDonUngDung(string MaHDUD, string Ngay, string SDT, string MaUD, string MaNV, int ThanhTien)

{

SqlCommand command = new SqlCommand("proc\_ThemHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDUD", SqlDbType.NChar).Value = MaHDUD;

command.Parameters.Add("@Ngay", SqlDbType.NVarChar).Value = Ngay;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@MaUD", SqlDbType.NChar).Value = MaUD;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@ThanhTien", SqlDbType.Int).Value = ThanhTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.5.2. Sửa hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_SuaHoaDonUngDung]

@MaHD\_UD nchar(10),

@Ngay date,

@SDT nchar(11),

@MaUD nchar(10),

@MaNV nchar(10),

@GiaTriHD int

AS

BEGIN

BEGIN TRY

-- Sửa thông tin hóa đơn Ung Dung

UPDATE dbo.HoaDonUngDung

SET Ngay = @Ngay, SDT=@SDT, MaUD = @MaUD, MaNV = @MaNV, GiaTriHD = @GiaTriHD

WHERE MaHD\_UD = @MaHD\_UD

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatHoaDonUngDung(string MaHDUD, string Ngay, string SDT, string MaUD, string MaNV, int ThanhTien)

{

SqlCommand command = new SqlCommand("proc\_CapNhatHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDUD", SqlDbType.NChar).Value = MaHDUD;

command.Parameters.Add("@Ngay", SqlDbType.NVarChar).Value = Ngay;

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@MaUD", SqlDbType.NChar).Value = MaUD;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@ThanhTien", SqlDbType.Int).Value = ThanhTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.5.3. Xóa hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_XoaHoaDonUngDung]

@MaHD\_UD nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá các chi tiết hóa đơn bán theo @MaHD\_UD trong bảng ChiTietHoaDonUngDung

DELETE FROM dbo.ChiTietHoaDonUngDung WHERE ChiTietHoaDonUngDung.MaHD\_UD = @MaHD\_UD

-- Xoá hóa đơn bán theo @MaHD\_UD trong bảng HoaDonUngDung

DELETE FROM dbo.HoaDonUngDung WHERE HoaDonUngDung.MaHD\_UD = @MaHD\_UD

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool XoaHoaDonUngDung(string MaHDUD)

{

SqlCommand command = new SqlCommand("proc\_XoaHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDUD", SqlDbType.NChar).Value = MaHDUD;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.4.6. Chi tiết hóa đơn ứng dụng

### 3.4.6.1. Thêm chi tiết hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_ThemChiTietHoaDonUngDung]

@MaHD\_UD nchar(10),

@MaSP nchar(10),

@SoLuong int,

@DonGia float,

@TongTien float

AS

BEGIN

INSERT INTO ChiTietHoaDonUngDung(MaHD\_UD, MaSP, SoLuong, DonGia, TongTien)

VALUES (@MaHD\_UD, @MaSP, @SoLuong, @DonGia, @TongTien)

END

GO

Thực thi trên C#:

public bool ThemChiTietHoaDonUngDung(string MaCTHDUD, string MaSP, int SoLuong, float DonGia, float TongTien)

{

SqlCommand command = new SqlCommand("proc\_ThemChiTietHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaCTHDUD", SqlDbType.NChar).Value = MaCTHDUD;

command.Parameters.Add("@MaSP", SqlDbType.NVarChar).Value = MaSP;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@ThanhTien", SqlDbType.Float).Value = TongTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.6.2. Sửa chi tiết hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_SuaChiTietHoaDonUngDung]

@MaHD\_UD nchar(10),

@MaSP nchar(10),

@SoLuong int,

@DonGia float,

@TongTien float

AS

BEGIN

BEGIN TRY

-- Sửa thông tin chi tiết đơn ung dung

UPDATE dbo.ChiTietHoaDonUngDung

SET SoLuong = @SoLuong, DonGia = @DonGia, TongTien = @TongTien

WHERE MaHD\_UD = @MaHD\_UD AND MaSP = @MaSP

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatChiTietHoaDonUngDung(string MaCTHDUD, string MaSP, int SoLuong, float DonGia, float TongTien)

{

SqlCommand command = new SqlCommand("proc\_SuaChiTietHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaCTHDUD", SqlDbType.NChar).Value = MaCTHDUD;

command.Parameters.Add("@MaSP", SqlDbType.NVarChar).Value = MaSP;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@TongTien", SqlDbType.Float).Value = TongTien;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

### 3.4.6.3. Xóa chi tiết hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_XoaChiTietHoaDonUngDung]

@MaHD\_UD nchar(10),

@MaSP nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá chi tiết đơn ung dung theo @MaHD\_UD và @MaSP trong bảng ChiTietHoaDonUngDung

DELETE FROM dbo.ChiTietHoaDonUngDung WHERE ChiTietHoaDonUngDung.MaHD\_UD = @MaHD\_UD AND ChiTietHoaDonUngDung.MaSP = @MaSP

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool XoaChiTietHoaDonUngDung(string MaCTHDUD, string MaSP)

{

SqlCommand command = new SqlCommand("proc\_XoaChiTietHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaCTHDUD", SqlDbType.NChar).Value = MaCTHDUD;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

# 3.5 Quản lý kho nguyên liệu

## 3.5.1. Thêm nguyên liệu vào kho

create PROCEDURE [dbo].[proc\_ThemNguyenLieu]

@MaNL nchar(10),

@TenNL nvarchar(50),

@MaNCC nchar(10),

@SoLuong int,

@DonVi nchar(10)

AS

BEGIN

-- Thêm mới nguyên liệu

INSERT INTO NguyenLieu(MaNL, TenNL, MaNCC, SoLuong, DonVi)

VALUES (@MaNL, @TenNL, @MaNCC, @SoLuong, @DonVi)

END

GO

Thực thi trên C#:

public bool ThemNguyenLieu(string MaNL, string TenNL, string MaNCC, int SoLuong, string DonVi)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_ThemNguyenLieu", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaNL", SqlDbType.NChar).Value = MaNL;

command.Parameters.Add("@TenNl", SqlDbType.NVarChar).Value = TenNL;

command.Parameters.Add("@MaNCC", SqlDbType.NChar).Value = MaNCC;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonVi", SqlDbType.NChar).Value = DonVi;

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.5.2. Sửa thông tin nguyên liệu

create PROCEDURE [dbo].[proc\_SuaNguyenLieu]

@MaNL nchar(10),

@TenNL nvarchar(50),

@MaNCC nchar(10),

@SoLuong int,

@DonVi nchar(10)

AS

BEGIN

BEGIN TRY

-- Cập nhật thông tin nguyen lieu

UPDATE NguyenLieu

SET TenNL = @TenNL, MaNCC = @MaNCC, SoLuong = @SoLuong, DonVi= @DonVi

WHERE MaNL = @MaNL

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatNguyenLieu(string MaNL, string TenNL, string MaNCC, int SoLuong, string DonVi)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_SuaNguyenLieu", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaNL", SqlDbType.NChar).Value = MaNL;

command.Parameters.Add("@TenNl", SqlDbType.NVarChar).Value = TenNL;

command.Parameters.Add("@MaNCC", SqlDbType.NChar).Value = MaNCC;

command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = SoLuong;

command.Parameters.Add("@DonVi", SqlDbType.NChar).Value = DonVi;

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.5.3. Xóa nguyên liệu không dùng

CREATE PROCEDURE [dbo].[proc\_XoaNguyenLieu]

@MaNL nchar(10)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Xoá nguyên liệu theo @MaNL trong bảng NguyenLieu

DELETE FROM dbo.NguyenLieu WHERE NguyenLieu.MaNL = @MaNL

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Xóa nguyên liệu không thành công' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool XoaNguyenLieu(string MaNL)

{

SqlCommand command = new SqlCommand("proc\_XoaNguyenLieu", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaNL", SqlDbType.NChar).Value = MaNL;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.5.4. Tìm kiếm nguyên liệu

CREATE FUNCTION [dbo].[func\_TimNguyenLieu] (@string NVARCHAR(50))

RETURNS @IngreList TABLE (MaNL VARCHAR(10), TenNL NVARCHAR(50), MaNCC

VARCHAR(10), SoLuong INT, DonVi NVARCHAR(10))

AS

BEGIN

INSERT INTO @IngreList

SELECT \*

FROM dbo.NguyenLieu

WHERE CONCAT(MaNL, TenNL, MaNCC, DonVi) LIKE N'%' + @string + '%'

RETURN

END

## 3.5.5. Trigger kiểm tra xem nguyên liệu trong kho còn đủ để pha chế không

CREATE TRIGGER TG\_SPHetHang

ON ChiTietDonBan

AFTER INSERT

AS

BEGIN

Declare @newMaSP nchar(10), @newSL int, @MaNLMin nchar(10), @SLMin int, @LieuLuongCan float

Select @newMaSP = ne.MaSP, @newSL = ne.SoLuong

From inserted ne

Select @SLMin = min(NguyenLieu.SoLuong)

From NguyenLieu, CheBien

Where NguyenLieu.MaNL = CheBien.MaNL

and CheBien.MaSP = @newMaSP

Select TOP 1 @MaNLMin = NguyenLieu.MaNL, @LieuLuongCan = CheBien.LieuLuong

From NguyenLieu, CheBien

Where NguyenLieu.MaNL = CheBien.MaNL

and CheBien.MaSP = @newMaSP

and NguyenLieu.SoLuong = @SLMin

IF (@newSL\*@LieuLuongCan - @SLMin\*1000 > 0)

BEGIN

UPDATE SanPham

SET TinhTrang = 'Hết hàng'

WHERE MaSP = @newMaSP

ROLLBACK

END

END

GO

# 3.6 Quản lý sản phẩm

## 3.6.1. Thêm sản phẩm

CREATE PROCEDURE [dbo].[proc\_ThemSanPham]

@MaSP nchar(10),

@TenSP nvarchar(50),

@DonGia float,

@TinhTrang nchar(10),

@MaLoaiSP nchar(10)

AS

BEGIN

-- Thêm sản phẩm

INSERT INTO SanPham(MaSP, TenSP, DonGia, TinhTrang, MaLoaiSP)

VALUES (@MaSP, @TenSP, @DonGia, @TinhTrang, @MaLoaiSP)

END

GO

Thực thi trên C#:

public bool ThemSanPham(string MaSP, string TenSP, float DonGia, string TinhTrang, string MaLoaiSP)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_ThemSanPham", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

command.Parameters.Add("@TenSP", SqlDbType.NVarChar).Value = TenSP;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@TinhTrang", SqlDbType.NChar).Value = TinhTrang;

command.Parameters.Add("@MaLoaiSP", SqlDbType.NChar).Value = MaLoaiSP;

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.6.2. Sửa sản phẩm

CREATE PROCEDURE [dbo].[proc\_SuaSanPham]

@MaSP nchar(10),

@TenSP nvarchar(50),

@DonGia float,

@TinhTrang nchar(10),

@MaLoaiSP nchar(10)

AS

BEGIN

BEGIN TRY

-- Cập nhật sản phẩm

UPDATE SanPham

SET TenSP = @TenSP, DonGia = @DonGia, TinhTrang = @TinhTrang, MaLoaiSP = @MaLoaiSP

WHERE MaSP = @MaSP

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

Thực thi trên C#

public bool CapNhatSanPham(string MaSP, string TenSP, float DonGia, string TinhTrang, string MaLoaiSP)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_SuaSanPham", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

command.Parameters.Add("@TenSP", SqlDbType.NVarChar).Value = TenSP;

command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia;

command.Parameters.Add("@MaLoaiSP", SqlDbType.NChar).Value = MaLoaiSP;

command.Parameters.Add("@TinhTrang", SqlDbType.NChar).Value = TinhTrang;

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.6.3. Xóa sản phẩm

CREATE PROCEDURE [dbo].[proc\_XoaSanPham]

@MaSP nchar(10)

AS

BEGIN

BEGIN TRY

BEGIN TRANSACTION

-- Xoá vị trí công việc theo @MaSP trong bảng CheBien

DELETE FROM dbo.CheBien WHERE CheBien.MaSP = @MaSP

-- Xóa vị trí công việc theo @MaSP trong bảng SanPham

DELETE FROM dbo.SanPham WHERE SanPham.MaSP = @MaSP

COMMIT TRANSACTION

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

Thực thi trên C#:

public bool XoaSanPham(string MaSP)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_XoaSanPham", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP;

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

# 3.7 Thống kê doanh thu và chi phí

## 3.7.1. Tính tổng tiền bên chi tiết hóa đơn sau đó cập nhật qua hóa đơn bán

CREATE PROCEDURE [dbo].[proc\_TinhTongTienHDB]

@MaHDB nchar(10)

AS

BEGIN

SET NOCOUNT ON;

UPDATE HoaDonBan

SET ThanhTien = (SELECT SUM(TongTien)

FROM ChiTietDonBan

WHERE ChiTietDonBan.MaHDB = HoaDonBan.MaHDB

GROUP BY MaHDB)

WHERE MaHDB = @MaHDB

AND EXISTS (SELECT 1

FROM ChiTietDonBan

WHERE ChiTietDonBan.MaHDB = HoaDonBan.MaHDB

GROUP BY MaHDB);

END;

GO

Thực thi trên C#:

public bool CapNhatThanhTien(string MaHDB)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_TinhTongTienHDB", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDB", SqlDbType.NChar).Value = MaHDB;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.7.2. Tính tổng tiền bên chi tiết hóa đơn sau đó cập nhật qua hóa đơn ứng dụng

CREATE PROCEDURE [dbo].[proc\_TinhTongTienHoaDonUngDung]

@MaHD\_UD nchar(10)

AS

BEGIN

SET NOCOUNT ON;

UPDATE HoaDonUngDung

SET GiaTriHD = (SELECT SUM(TongTien)

FROM ChiTietHoaDonUngDung

WHERE ChiTietHoaDonUngDung.MaHD\_UD = HoaDonUngDung.MaHD\_UD

GROUP BY MaHD\_UD)

WHERE MaHD\_UD = @MaHD\_UD

AND EXISTS (SELECT 1

FROM ChiTietHoaDonUngDung

WHERE ChiTietHoaDonUngDung.MaHD\_UD = HoaDonUngDung.MaHD\_UD

GROUP BY MaHD\_UD);

END;

GO

Thực thi trên C#:

public bool CapNhatThanhTien(string MaHDUD)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_TinhTongTienHoaDonUngDung", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDUD", SqlDbType.NChar).Value = MaHDUD;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.7.2. Tính tổng tiền bên chi tiết hóa đơn sau đó cập nhật qua hóa đơn nhập

CREATE PROCEDURE [dbo].[proc\_TinhTongTienHDN]

@MaHDN nchar(10)

AS

BEGIN

SET NOCOUNT ON;

UPDATE HoaDonNhap

SET TriGiaDonNhap = (SELECT SUM(TongTien)

FROM ChiTietDonNhap

WHERE ChiTietDonNhap.MaHDN = HoaDonNhap.MaHDN

GROUP BY MaHDN)

WHERE MaHDN = @MaHDN

AND EXISTS (SELECT 1

FROM ChiTietDonNhap

WHERE ChiTietDonNhap.MaHDN = HoaDonNhap.MaHDN

GROUP BY MaHDN);

END;

GO

Thực thi trên C#:

public bool CapNhatThanhTien(string MaHDN)

{

db.openConnection();

SqlCommand command = new SqlCommand("proc\_TinhTongTienHDN", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaHDN", SqlDbType.NChar).Value = MaHDN;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

# 3.8 Quản lý nhân viên

## 3.8.1. Thêm nhân viên mới

CREATE PROCEDURE [dbo].[proc\_ThemNhanVien]

@MaNV nchar(10),

@TenNV nvarchar(10),

@NgaySinh date,

@GioiTinh nvarchar(3),

@DiaChi nvarchar(100),

@SDT nchar(11),

@MaLoaiNV nchar(10),

@MaViTri nchar(10),

@NgayTuyenDung date

AS

BEGIN

-- Thêm nhân viên

INSERT INTO NhanVien (MaNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, MaLoaiNV, MaViTri, NgayTuyenDung)

VALUES (@MaNV, @TenNV, @NgaySinh, @GioiTinh, @DiaChi, @SDT, @MaLoaiNV, @MaViTri, @NgayTuyenDung)

END

GO

Thực thi trên C#:

public bool ThemNhanVien(string MaNV, string TenNV, string NgaySinh, string GioiTinh, string DiaChi, string SDT, string MaLoaiNV, string MaViTri, string NgayTuyenDung)

{

SqlCommand command = new SqlCommand("proc\_ThemNhanVien", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@TenNV", SqlDbType.NVarChar).Value = TenNV;

command.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = NgaySinh;

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

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

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@MaLoaiNV", SqlDbType.NChar).Value = MaLoaiNV;

command.Parameters.Add("@MaViTri", SqlDbType.NChar).Value = MaViTri;

command.Parameters.Add("@NgayTuyenDung", SqlDbType.Date).Value = NgayTuyenDung;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.8.2. Sửa thông tin nhân viên

CREATE PROCEDURE [dbo].[proc\_SuaNhanVien]

@MaNV nchar(10),

@TenNV nvarchar(10),

@NgaySinh date,

@GioiTinh nvarchar(3),

@DiaChi nvarchar(100),

@SDT nchar(11),

@MaLoaiNV nchar(10),

@MaViTri nchar(10),

@NgayTuyenDung date

AS

BEGIN

BEGIN TRY

-- Cập nhật thông tin nhân viên

UPDATE NhanVien

SET TenNV = @TenNV, NgaySinh = @NgaySinh, GioiTinh = @GioiTinh, DiaChi = @DiaChi, SDT = @SDT,

MaLoaiNV = @MaLoaiNV, MaViTri = @MaViTri, NgayTuyenDung = @NgayTuyenDung

WHERE MaNV = @MaNV

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool CapNhatNhanVien(string MaNV, string TenNV, string NgaySinh, string GioiTinh, string DiaChi, string SDT, string MaLoaiNV, string MaViTri, string NgayTuyenDung)

{

SqlCommand command = new SqlCommand("proc\_SuaNhanVien", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@TenNV", SqlDbType.NVarChar).Value = TenNV;

command.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = NgaySinh;

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

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

command.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;

command.Parameters.Add("@MaLoaiNV", SqlDbType.NChar).Value = MaLoaiNV;

command.Parameters.Add("@MaViTri", SqlDbType.NChar).Value = MaViTri;

command.Parameters.Add("@NgayTuyenDung", SqlDbType.Date).Value = NgayTuyenDung;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.8.3. Xóa nhân viên đã nghỉ làm việc

CREATE PROCEDURE [dbo].[proc\_XoaNhanVien]

@MaNV nchar(10)

AS

BEGIN

BEGIN TRY

BEGIN TRANSACTION

-- Xoá nhân viên theo @MaNV trong bảng NhanVien

DELETE FROM dbo.NhanVien WHERE NhanVien.MaNV = @MaNV

COMMIT TRANSACTION

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Thực thi trên C#:

public bool XoaNhanVien(string MaNV)

{

SqlCommand command = new SqlCommand("proc\_XoaNhanVien", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.8.4. Trigger bắt lỗi khi thêm hoặc sửa thông tin nhân viên

CREATE TRIGGER trg\_CheckNhanVien

ON NhanVien

FOR INSERT, UPDATE

AS

BEGIN

-- check MaNV

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNV) = ' ')

BEGIN

RAISERROR('Mã nhân viên không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM NhanVien WHERE MaNV IN (SELECT MaNV FROM

inserted))

BEGIN

RAISERROR('Mã nhân viên đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ho ten nhan vien

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenNV) = '' )

BEGIN

RAISERROR('Tên nhân viên không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NhanVien n ON i.MaNV != n.MaNV

AND TRIM(i.TenNV) = TRIM(n.TenNV))

BEGIN

RAISERROR('Tên nhân viên đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ngay sinh

IF EXISTS (SELECT \* FROM inserted WHERE

datediff(year,inserted.NgaySinh,getdate())<(18))

BEGIN

RAISERROR ('Nhân viên phải trên 18 tuổi', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check dia chi

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(DiaChi) = '' )

BEGIN

RAISERROR('Địa chỉ không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check so dien thoai

IF EXISTS (SELECT \* FROM inserted WHERE TRIM(SDT) = '' )

BEGIN

RAISERROR('Số điện thoại không được để trống', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF NOT EXISTS (SELECT \* FROM inserted WHERE len(TRIM(SDT)) = (10))

BEGIN

RAISERROR('Số điện thoại gồm 10 chữ số', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NhanVien n ON i.MaNV != n.MaNV

AND TRIM(i.SDT) = TRIM(n.SDT))

BEGIN

RAISERROR('Số điện thoại đã tồn tại', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

-- check ngay tuyen dung

IF NOT EXISTS (SELECT \* FROM inserted WHERE

(datediff(day,[NgayTuyenDung],getdate())>=(0)))

BEGIN

RAISERROR ('Ngày tuyển dụng không thể là trong tương lai', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END

# 3.9 Quản lý phân ca

## 3.9.1. Thêm bảng phân ca cho nhân viên

CREATE PROCEDURE [dbo].[proc\_ThemBangPhanCa]

@MaCa nchar(10),

@MaNV nchar(10),

@Ngay date

AS

BEGIN

-- Thêm bản ghi vào BangPhanCa

INSERT INTO BangPhanCa (MaCa, MaNV, Ngay)

VALUES (@MaCa, @MaNV, @Ngay)

END

GO

Thực thi trên C#:

public bool ThemBangPhanCa(string MaCa, string MaNV, string Ngay)

{

SqlCommand command = new SqlCommand("proc\_ThemBangPhanCa", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaCa", SqlDbType.NChar).Value = MaCa;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@Ngay", SqlDbType.Date).Value = Ngay;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.9.2. Sửa bảng phân ca của nhân viên

CREATE PROCEDURE [dbo].[proc\_SuaBangPhanCa]

@MaCa nchar(10),

@MaNV nchar(10),

@Ngay date

AS

BEGIN

BEGIN TRY

-- Cập nhật thông bảng phân ca

UPDATE BangPhanCa

SET MaNV = @MaNV

WHERE MaCa = @MaCa AND Ngay = @Ngay

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

END

Thực thi trên C#:

public bool CapNhatBangPhanCa(string MaCa, string MaNV, string Ngay)

{

SqlCommand command = new SqlCommand("proc\_SuaBangPhanCa", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaCa", SqlDbType.NChar).Value = MaCa;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@Ngay", SqlDbType.Date).Value = Ngay;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

## 3.9.3. Xóa ca làm việc của nhân viên

CREATE PROC [dbo].[proc\_XoaBangPhanCa]

@MaNV nchar(10),

@MaCa nchar(10),

@Ngay nchar(10)

AS

BEGIN

BEGIN TRY

DELETE FROM BangPhanCa

WHERE MaNV = @MaNV and

MaCa = @MaCa and

Ngay = @Ngay

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

Thực thi trên C#:

public bool XoaBangPhanCa(string MaCa, string MaNV, string Ngay)

{

SqlCommand command = new SqlCommand("proc\_XoaBangPhanCa", db.getConnection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@MaCa", SqlDbType.NChar).Value = MaCa;

command.Parameters.Add("@MaNV", SqlDbType.NChar).Value = MaNV;

command.Parameters.Add("@Ngay", SqlDbType.Date).Value = Ngay;

db.openConnection();

if (command.ExecuteNonQuery() > 0)

{

db.closeConnection();

return true;

}

else

{

db.closeConnection();

return false;

}

}

# CHƯƠNG 4: QUẢN LÝ USER VÀ PHÂN QUYỀN

Ứng với hai nhóm người dung (Nhân viên và quản lý) là các role riêng

# 4.1 Đối với Role Staff (dành cho nhân viên):

CREATE ROLE Staff

-- Gán các quyền trên table cho role Staff

GRANT SELECT, REFERENCES ON BangPhanCa TO Staff

GRANT SELECT, REFERENCES ON CaLamViec TO Staff

GRANT SELECT, REFERENCES ON CheBien TO Staff

GRANT SELECT, REFERENCES ON ChiTietDonBan TO Staff

GRANT SELECT, REFERENCES ON ChiTietHoaDonUngDung TO Staff

GRANT SELECT, REFERENCES ON ChiTietDonNhap TO Staff

GRANT SELECT, REFERENCES ON ViTriCongViec TO Staff

GRANT SELECT, INSERT, REFERENCES ON HoaDonNhap TO Staff

GRANT SELECT, INSERT, REFERENCES ON HoaDonBan TO Staff

GRANT SELECT, INSERT, REFERENCES ON HoaDonUngDung TO Staff

GRANT SELECT, INSERT, REFERENCES ON KhachHang TO Staff

GRANT SELECT, REFERENCES ON LoaiSanPham TO Staff

GRANT SELECT, REFERENCES ON NguyenLieu TO Staff

GRANT SELECT, REFERENCES ON NhaCungCap TO Staff

GRANT SELECT, REFERENCES ON NhanVien TO Staff

GRANT SELECT, REFERENCES ON SanPham TO Staff

GRANT SELECT, REFERENCES ON UngDung TO Staff

-- Gán quyền thực thi trên các procedure, function cho role Staff

GRANT EXECUTE TO Staff

GRANT SELECT TO Staff

DENY EXECUTE ON proc\_ThemNhanVien to Staff;

DENY EXECUTE ON proc\_XoaNhanVien to Staff;

DENY EXECUTE ON proc\_SuaNhanVien to Staff;

DENY EXECUTE ON proc\_SuaCheBien to Staff;

DENY EXECUTE ON proc\_SuaNguyenLieu to Staff;

DENY EXECUTE ON proc\_SuaSanPham to Staff;

DENY EXECUTE ON proc\_XoaSanPham to Staff;

DENY EXECUTE ON proc\_XoaBangPhanCa to Staff;

# 4. 2 Đối với Role Manager ta sử dụng server role sysadmin.

- Trigger tạo tài khoản

--Them trigger tu dong tao tai khoan khi them nhan vien--

CREATE OR ALTER TRIGGER Trg\_Create\_Account

ON NhanVien

AFTER INSERT

AS

BEGIN

INSERT INTO Account(username, password, MaNV, roles)

SELECT

inserted.MaNV,

'12345678',

inserted.MaNV,

CASE

WHEN NhanVien.MaViTri = 'vt01' THEN 'sysadmin'

ELSE 'Staff'

END AS roles

FROM inserted

INNER JOIN NhanVien ON inserted.MaNV = NhanVien.MaNV;

END;

GO

-- Trigger add quyen cho tai khoan sau khi them nhan vien

CREATE OR ALTER TRIGGER AddQuyen ON Account

AFTER INSERT

AS

DECLARE @tendangnhap nvarchar(10), @matkhau nvarchar(30), @roles nvarchar(20)

SELECT @tendangnhap=nl.username, @matkhau=nl.password, @roles=nl.roles

FROM inserted nl

BEGIN

DECLARE @sqlString nvarchar(2000)

----

SET @sqlString= ' CREATE LOGIN [' + @tendangnhap +'] WITH PASSWORD= '''+ @matkhau

+''', DEFAULT\_DATABASE=[QlQuanTraSua], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

SET @sqlString= ' CREATE USER ' + @tendangnhap +' FOR LOGIN '+ @tendangnhap

EXEC (@sqlString)

if (@roles = 'sysadmin')

SET @sqlString = ' ALTER SERVER ROLE sysadmin ' + ' ADD MEMBER ' + @tendangnhap;

if (@roles = 'Staff')

SET @sqlString = ' ALTER ROLE Staff ADD MEMBER ' + @tendangnhap;

EXEC (@sqlString)

END

GO

- Procedure sửa mật khẩu cho tài khoản

-- Procedure sửa password của tài khoản:

CREATE OR ALTER PROCEDURE proc\_updateAccount

@username NVARCHAR(50),

@password VARCHAR(25),

@roles NVARCHAR(20)

AS

IF EXISTS(SELECT\* FROM Account WHERE username = @username)

BEGIN

BEGIN TRAN

BEGIN TRY

DECLARE @oldUsername nvarchar(50);

SELECT @oldUsername=username FROM ACCOUNT WHERE username=@username;

DECLARE @Str VARCHAR(100);

DECLARE @SessionID INT;

SELECT @SessionID = session\_id

FROM sys.dm\_exec\_sessions

WHERE login\_name = @oldUsername;

IF @SessionID IS NOT NULL

BEGIN

SET @Str = 'kill ' + Convert(NVARCHAR(20), @SessionID)

exec(@Str)

END

UPDATE Account SET username = @username, password = @password, roles = @roles

WHERE username = @username;

DECLARE @sql varchar(100)

SET @sql = 'DROP USER '+ @oldUsername

exec (@sql)

SET @sql = 'DROP LOGIN '+ @oldUsername

exec (@sql)

DECLARE @sqlString NVARCHAR(2000)

-- Tạo tài khoản login cho nhân viên, tên người dùng và mật khẩu là tài khoản được tạo trên bảng Account

SET @sqlString = 'CREATE LOGIN [' + @username + '] WITH PASSWORD=''' + @password + ''', DEFAULT\_DATABASE=[QLQuanTraSua], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

-- Tạo tài khoản người dùng đối với nhân viên đó trên database (tên người dùng trùng với tên login)

SET @sqlString = 'CREATE USER ' + @username + ' FOR LOGIN ' + @username;

EXEC (@sqlString)

IF (@roles = 'sysadmin')

SET @sqlString = 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @username;

ELSE

SET @sqlString = 'ALTER ROLE Staff ADD MEMBER ' + @username;

EXEC (@sqlString)

-- Thêm người dùng vào vai trò quyền tương ứng (Staff hoặc Manager(sysadmin))

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

GO

Procedure xóa nhân viên

-- Procedure xóa nhân viên:

CREATE or ALTER PROCEDURE proc\_deleteEmployee

@MaNV nchar(10)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @username varchar(15);

SELECT @username=username FROM ACCOUNT WHERE MaNV=@MaNV

DECLARE @sql varchar(100)

DECLARE @SessionID INT;

SELECT @SessionID = session\_id

FROM sys.dm\_exec\_sessions

WHERE login\_name = @username;

IF @SessionID IS NOT NULL

BEGIN

SET @sql = 'kill ' + Convert(NVARCHAR(20), @SessionID)

exec(@sql)

END

BEGIN TRANSACTION;

BEGIN TRY

DELETE FROM NhanVien where MaNV = @MaNV

--

SET @sql = 'DROP USER '+ @username

exec (@sql)

--

SET @sql = 'DROP LOGIN '+ @username

exec (@sql)

--

DELETE FROM Account WHERE MaNV=@MaNV;

COMMIT;

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

ROLLBACK TRANSACTION;

THROW;

END CATCH

END;

# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG

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

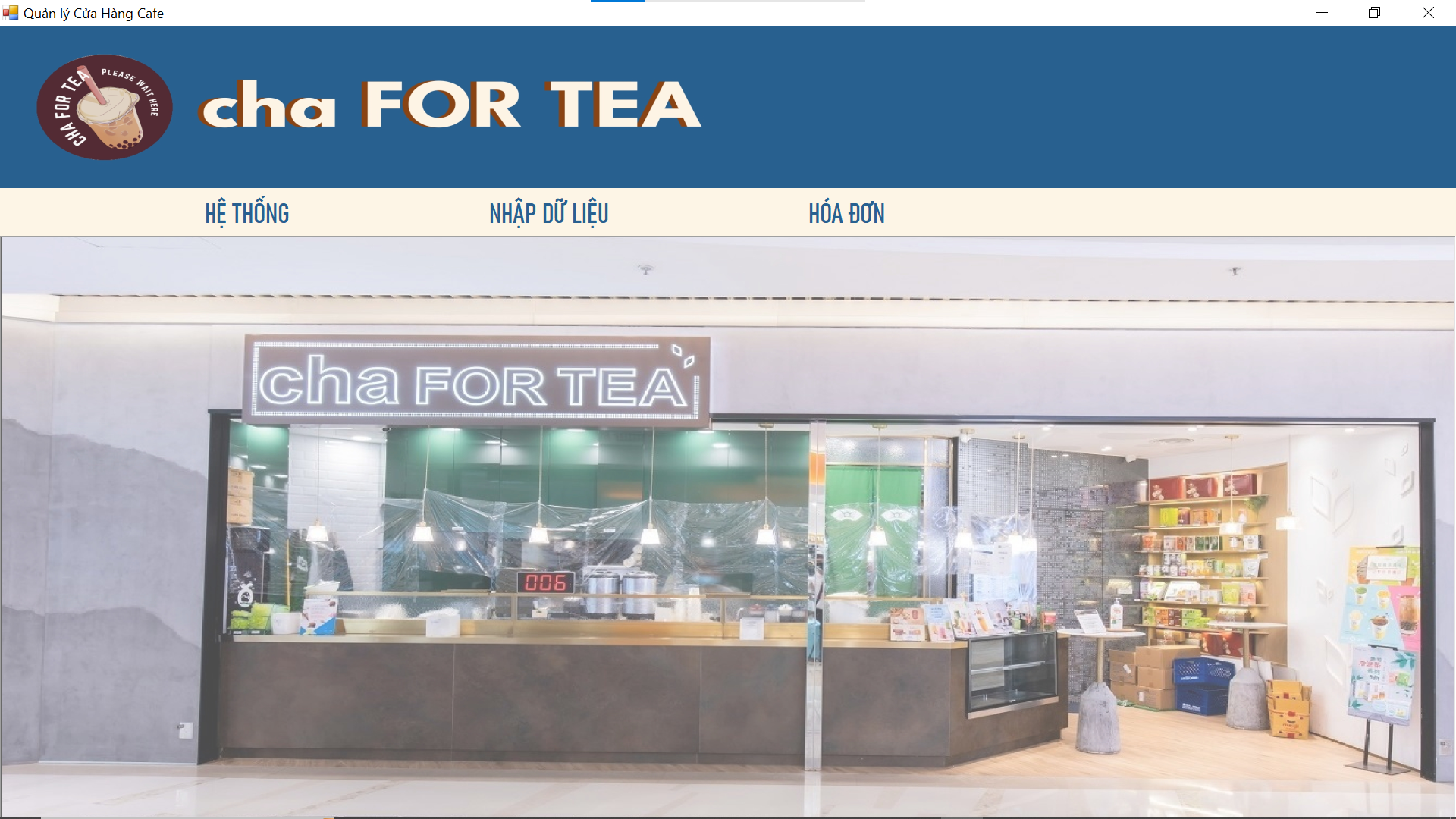
- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên tảng

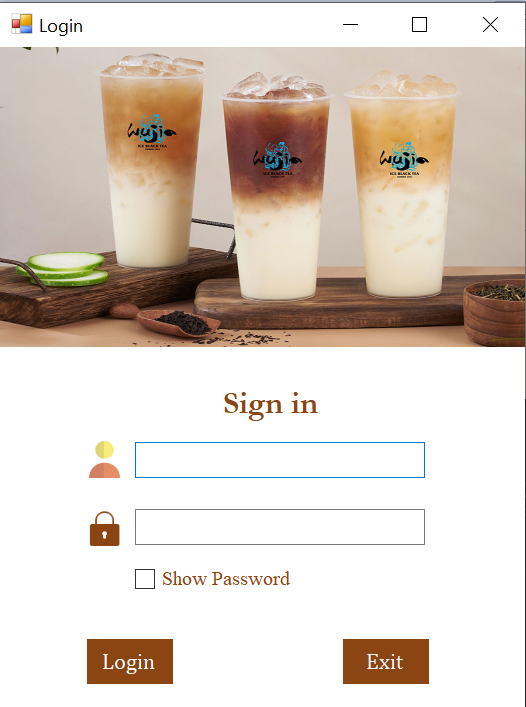
ASP.NET Web Application (.NET Framework) C#.

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

# 1. Form Home

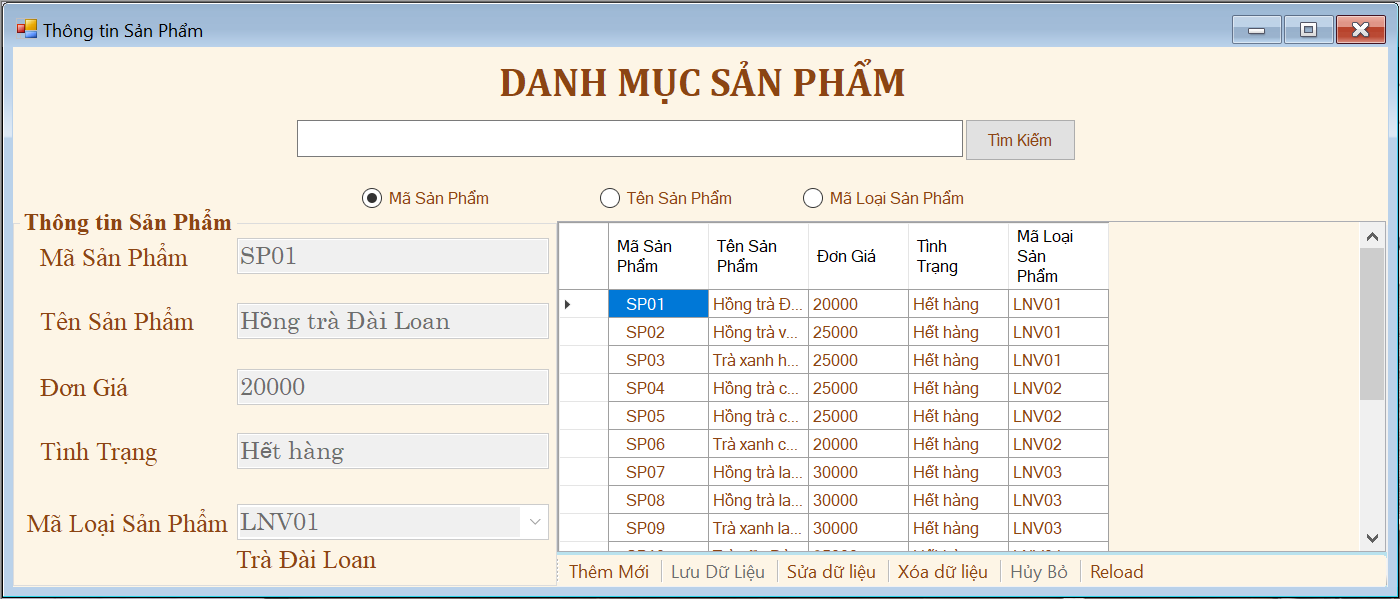


# 2. Form đăng nhập

****

# 3. Các Form Danh mục

## 3.1. Danh mục sản phẩm



## 3.2. Danh mục Loại Nhân Viên



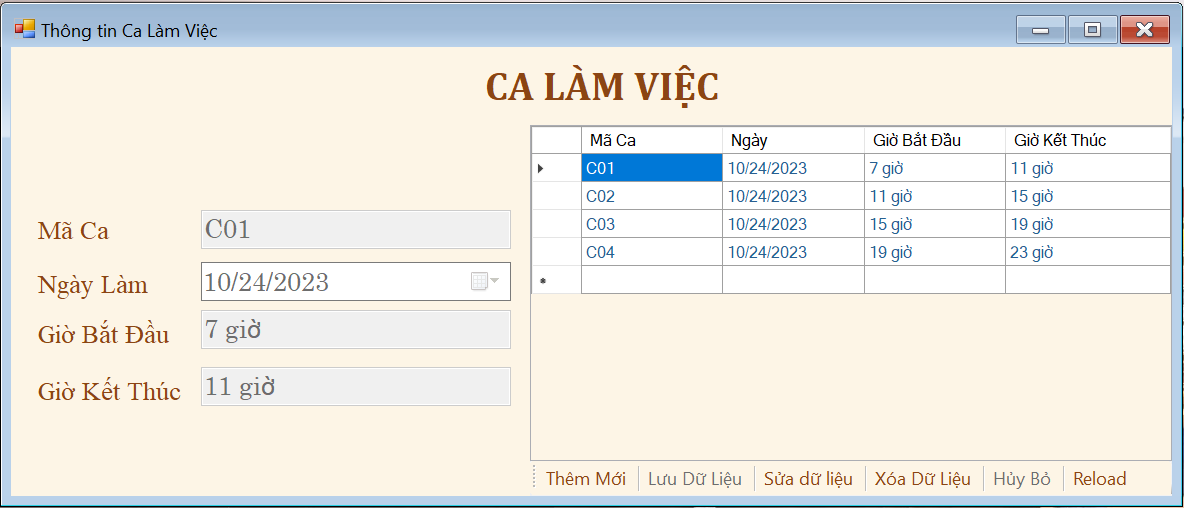
## 3.3. Danh mục Nhân Viên



## 3.4. Danh mục Vị Trí Công Việc



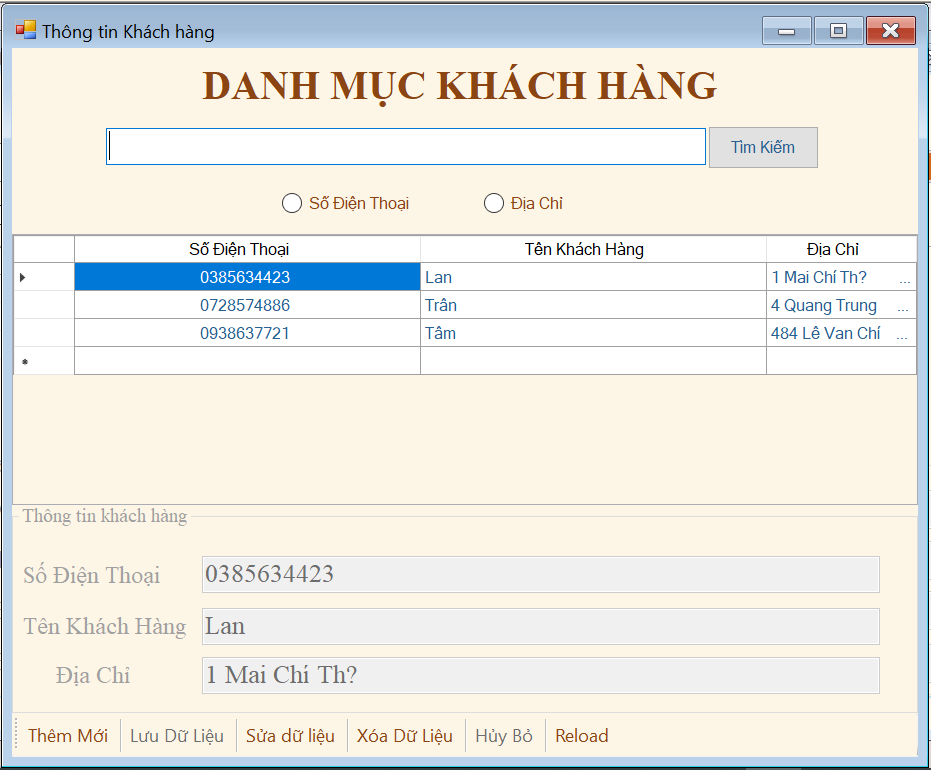
## 3.5. Danh mục Ca Làm Việc



## 3.6. Danh mục Bảng Phân Ca



## 3.7. Danh mục Khách Hàng



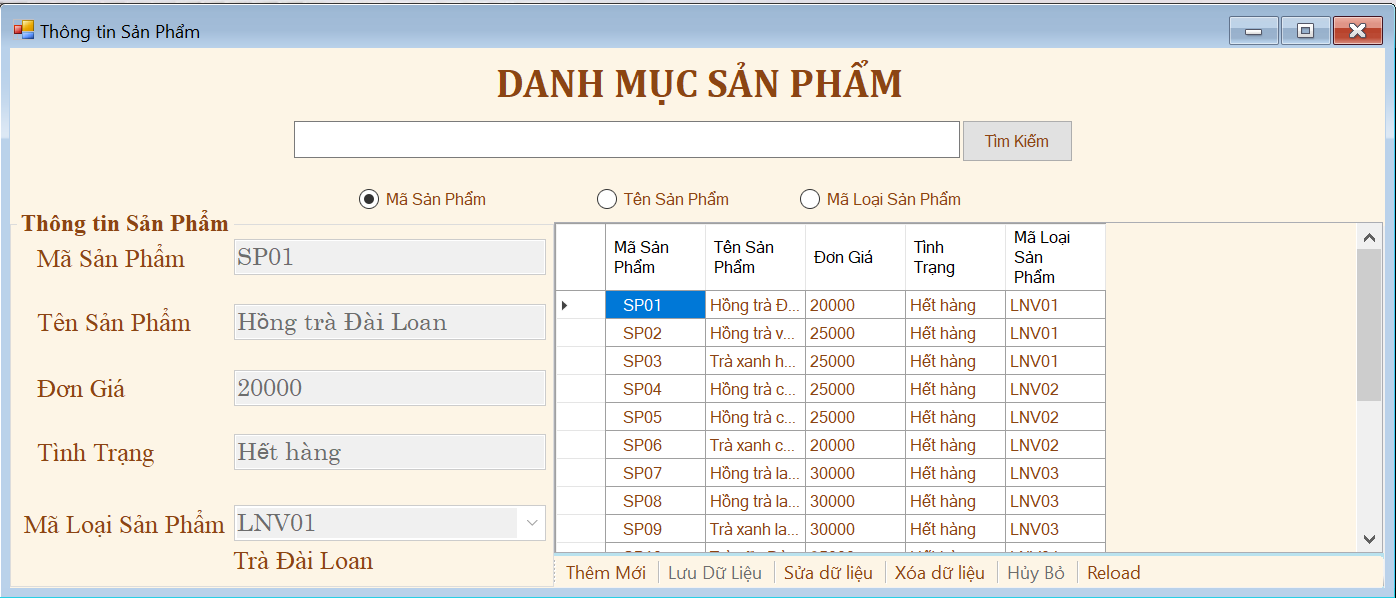
## 3.8. Danh mục Nhà cung cấp



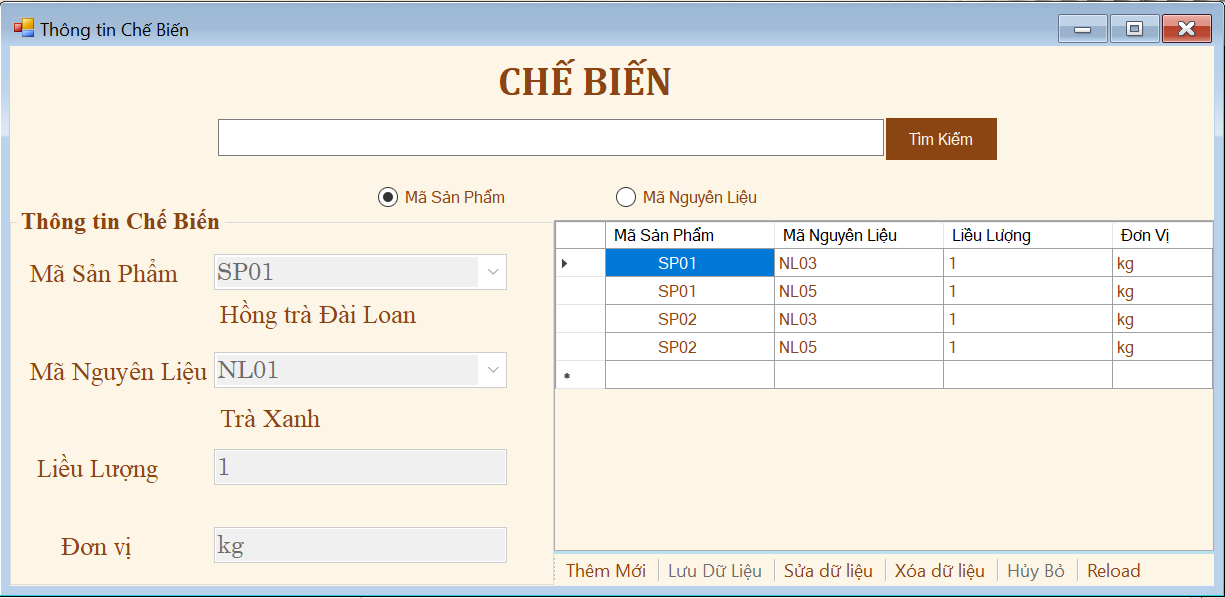
## 3.9. Danh mục Loại Sản Phẩm



## 3.10. Danh mục Sản Phẩm



## 3.11. Danh mục Chế Biến



## 3.12. Danh mục Nguyên Liệu



## 3.13. Danh mục Ứng Dụng



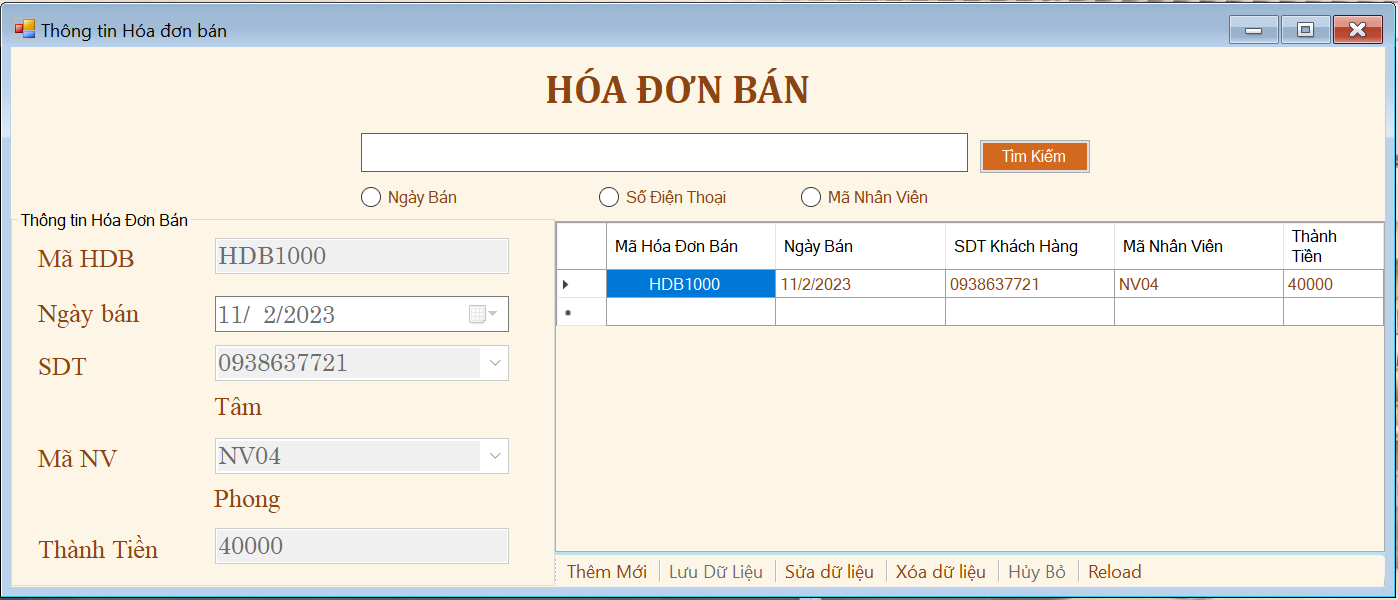
## 3.14. Danh mục Hóa Đơn Nhập



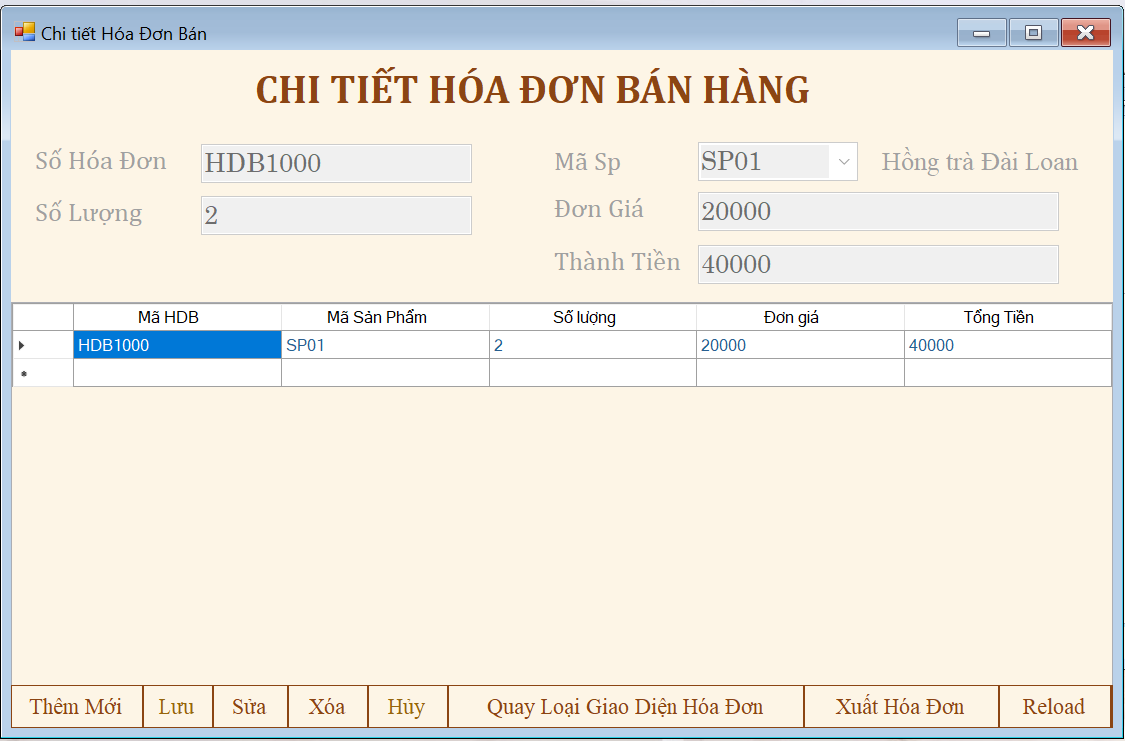
## 3.15. Danh mục Chi tiết Hóa Đơn Nhập

****

## 3.16. Danh mục Hóa Đơn Bán



**3.17. Danh mục Chi tiết Hóa Đơn Bán**



## 3.18. Danh mục Hóa Đơn Ứng Dụng



## 3.19. Danh mục Chi tiết Hóa Đơn Ứng Dụng



# TÀI LIỆU THAM KHẢO