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

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

-----🙞🙞🙞🙞🙞-----

A logo with a blue circle

Description automatically generated

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

**TÊN  ĐỀ  TÀI:**

**QUẢN LÝ CỬA HÀNG QUẦN ÁO**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Lớp HP:**DBMS330284\_23\_1\_05

**Nhóm thực hiện: nhóm 12**

**Học kỳ:** *1*

**Năm học:** *2023 – 2024*

*Thành phố Hồ Chí Minh, ngày 21 tháng 9, năm 2023*

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2023-2024**

**Nhóm 12**

***Đề tài: Quản lý cửa hàng quần áo***

|  |  |
| --- | --- |
| 21133027 | Hoàng Mạnh Đức |
| 21133022 | Phạm Hữu Dũng |
| 21133048 | Nguyễn Phương Khoa |
| 21133092 | Trương Quốc Việt |

**NHẬN XÉT CỦA GV**

Tp. Hồ Chí Minh, ngày 21, tháng 11, năm 2023

**Mục lục**

[LỜI CẢM ƠN 1](#_Toc151463047)

[LỜI NÓI ĐẦU 2](#_Toc151463048)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 3](#_Toc151463049)

[1. Đặc tả đề tài 3](#_Toc151463050)

[1.1 . Mô tả bài toán 3](#_Toc151463051)

[1.2 . Nghiệp vụ của bài toán 4](#_Toc151463052)

[1.3 . Mô tả chức năng của bài toán 5](#_Toc151463053)

[1.4. Tính lợi nhuận của một cửa hàng: 5](#_Toc151463054)

[1.5. Tính lợi nhuận của chuỗi cửa hàng: 6](#_Toc151463055)

[1.6. Tính lương nhân viên: 7](#_Toc151463056)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 10](#_Toc151463057)

[1. Thiết kế cơ sở dữ liệu mức quan niệm 10](#_Toc151463058)

[2. Thiết kế cơ sở dữ liệu mức logic Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ: 10](#_Toc151463059)

[3. Các ràng buộc cần có 12](#_Toc151463060)

[4. Cài đặt các CSDL và các ràng buộc: 13](#_Toc151463061)

[5. Các view 16](#_Toc151463062)

[6. Các Trigger 18](#_Toc151463063)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 20](#_Toc151463064)

[1. Kết nối cơ sở dữ liệu 20](#_Toc151463065)

[2. Xem thông tin danh mục 21](#_Toc151463066)

[2.1.Danh mục nhân viên 21](#_Toc151463067)

[2.2.Danh mục bảng phân ca 22](#_Toc151463068)

[2.3.Danh mục hóa đơn bán hàng 22](#_Toc151463069)

[2.4.Hóa đơn ứng dụng: 23](#_Toc151463070)

[2.5.Danh mục sản phẩm: 23](#_Toc151463071)

[2.6.Danh mục phiếu chi: 24](#_Toc151463072)

[3. Quản lý khách hàng 25](#_Toc151463073)

[3.1. Tìm kiếm khách hàng 25](#_Toc151463074)

[3.2. Tạo khách hàng mới 25](#_Toc151463075)

[3.3. Sửa thông tin khách hàng 26](#_Toc151463076)

[3.4. Trigger bắt lỗi trùng số điện thoại khi thêm và chỉnh sửa khách hàng\ 27](#_Toc151463077)

[4. Quản lý hóa đơn bán hàng và hóa đơn ứng dụng 28](#_Toc151463078)

[4.1. Xuất hóa đơn bán hàng 28](#_Toc151463079)

[4.2.Tìm kiếm hóa đơn bán hàng 30](#_Toc151463080)

[4.3. Xóa hóa đơn bán hàng: 30](#_Toc151463081)

[4.4. In hóa đơn bán hàng: 31](#_Toc151463082)

[4.5. Xuất hóa đơn ứng dụng: 34](#_Toc151463083)

[4.6.Tìm kiếm hóa đơn ứng dụng 36](#_Toc151463084)

[4.7.Xóa hóa đơn ứng dụng: 36](#_Toc151463085)

[4.8.In hóa đơn ứng dụng 37](#_Toc151463086)

[5. Danh mục sản phẩm: 40](#_Toc151463087)

[5.1 Thêm sản phẩm: 40](#_Toc151463088)

[5.2. Sửa sản phẩm 42](#_Toc151463089)

[5.3.Xóa sản phẩm 43](#_Toc151463090)

[5.4. Tìm kiếm sản phẩm 45](#_Toc151463091)

[5.5. Trigger bắt lỗi trùng tên khi thêm và chỉnh sửa sản phẩm 47](#_Toc151463092)

[6. Thống kê doanh thu và chi phí 47](#_Toc151463093)

[6.1. Doanh thu theo ngày, tháng, năm 47](#_Toc151463094)

[7. Quản lý nhân viên 48](#_Toc151463095)

[7.1. Thêm nhân viên 48](#_Toc151463096)

[7.2. Xóa nhân viên 50](#_Toc151463097)

[7.3. Tìm kiếm nhân viên 51](#_Toc151463098)

[7.4. Sửa thông tin nhân viên: 53](#_Toc151463099)

[7.5. Tính lương nhân viên 55](#_Toc151463100)

[7.6. Trigger bắt lỗi trùng số điện thoại khi thêm nhân viên: 57](#_Toc151463101)

[8. Quản lý phiếu chi 59](#_Toc151463102)

[8.1 Thêm phiếu chi 59](#_Toc151463103)

[8.2 Xóa phiếu chi 61](#_Toc151463104)

[9. Quản lý phân ca 62](#_Toc151463105)

[9.1. Thêm ca cho nhân viên 62](#_Toc151463106)

[9.2. Xóa ca làm việc 64](#_Toc151463107)

[CHƯƠNG 4 : Tạo User và phân quyền 66](#_Toc151463108)

[+ Đối với Role Staff (dành cho nhân viên): 66](#_Toc151463109)

[+ Đối với Role Manager ta sử dụng server role sysadmin. 66](#_Toc151463110)

[CHƯƠNG 5 : THIẾT KẾ GIAO DIỆN HỆ THỐNG 71](#_Toc151463111)

[1. Form Home 71](#_Toc151463112)

[2. Form đăng nhập 72](#_Toc151463113)

[3. Các Form Danh mục 73](#_Toc151463114)

[3.1. Danh mục sản phẩm 73](#_Toc151463115)

[3.2. Danh mục nhân viên 74](#_Toc151463116)

[3.3. Danh mục phân ca 76](#_Toc151463117)

[3.4. Danh mục khách hàng 77](#_Toc151463118)

[3.5. Danh mục tài khoản 78](#_Toc151463119)

[4. Các Form hóa đơn và phiếu chi 79](#_Toc151463120)

[4.1. Form Hóa đơn bán hàng 79](#_Toc151463121)

[4.2 Form Hóa đơn ứng dụng 80](#_Toc151463122)

[4.3. Form phiếu chi của đơn nhập hàng 81](#_Toc151463123)

[5. Form tạo tài khoản 83](#_Toc151463124)

# LỜI CẢM ƠN

Kính thưa thầy Nguyễn Thành Sơn,

Nhóm em xin gửi lời cảm ơn chân thành đến thầy vì đã hướng dẫn và giúp đỡ chúng em trong quá trình hoàn thiện đồ án môn Hệ quản trị cơ sở dữ liệu. Nhờ vào sự chỉ dẫn và giúp đỡ của thầy, nhóm em đã có thể hoàn thành đồ án một cách hiệu quả và đạt được kết quả như mong đợi. Chúng em rất biết ơn thầy đã dành thời gian và tâm huyết để truyền đạt kiến thức và kinh nghiệm cho nhóm.

Thầy đã giúp chúng em có được những kiến thức và kỹ năng cần thiết để làm việc với cơ sở dữ liệu một cách hiệu quả, và nhóm em sẽ sử dụng những kiến thức này để phát triển bản thân và đóng góp vào công việc sau này.

Một lần nữa, nhóm em xin chân thành cảm ơn thầy Nguyễn Thành Sơn vì sự giúp đỡ và hướng dẫn của thầy trong quá trình học tập của nhóm.

Trân trọng,

Phạm Hữu Dũng

Trương Quốc Việt

Nguyễn Phương Khoa

Hoàng Mạnh Đức

# LỜI NÓI ĐẦU

Xin chào thầy và các bạn,

Đồ án quản lý cửa hàng quần áo là kết quả của nhóm em sau một thời gian học tập và thực hành môn học Hệ quản trị cơ sở dữ liệu. Đây là một đồ án đầy thử thách và hứa hẹn đem lại cho chúng ta những kiến thức và kinh nghiệm quý giá.

Trong quá trình hoàn thành đồ án, nhóm em đã đối mặt với rất nhiều khó khăn và thách thức. Tuy nhiên, với sự hướng dẫn và hỗ trợ tận tình của giáo viên hướng dẫn cùng với sự nỗ lực không ngừng của chúng em, đồ án đã được hoàn thành một cách thật sự tốt đẹp.

Đồ án quản lý cửa hàng quần áo là một ứng dụng thực tiễn, nhằm mục đích giúp chủ cửa hàng quần áo có thể quản lý hoạt động của quán một cách dễ dàng và hiệu quả hơn. Bằng việc áp dụng những kiến thức về hệ quản trị cơ sở dữ liệu, chúng em đã xây dựng được một hệ thống quản lý hoàn chỉnh, bao gồm các chức năng quản lý khách hàng, quản lý nhân viên, quản lý thực đơn và các hoạt động kinh doanh khác.

Nhóm em hy vọng rằng đồ án của chúng em sẽ đem lại cho quý thầy cô và các bạn sinh viên những kiến thức và trải nghiệm thực tế hữu ích. Chúng em cũng mong muốn rằng đồ án này sẽ có thể truyền cảm hứng và khơi gợi những ý tưởng sáng tạo cho các bạn sinh viên khác trong việc ứng dụng hệ quản trị cơ sở dữ liệu vào thực tế. Cảm ơn quý thầy cô và các bạn sinh viên đã theo dõi và đóng góp ý kiến cho đồ án của chúng em.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

## 1. Đặc tả đề tài

### 1.1 . Mô tả bài toán

  Một chuỗi cửa hàng quần áo 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 hệ thống quản lý chuỗi cửa hàng quần áo cần lưu trữ những dữ liệu sau:

  Chuỗi cửa hàng quần áo gồm nhiều chi nhánh, các chi nhánh phân biệt với nhau bằng mã chi nhánh, mỗi chi nhánh gồm các thông tin như mã chi nhánh, tên chi nhánh, địa chỉ chi nhánh, mã người quản lý. Một nhân viên chỉ làm việc tại một chi nhánh. Một chi nhánh có nhiều nhân viên. Một nhân viên có thể quản lý một chi nhánh hoặc không quản lý chi nhánh nào.

  Mỗi chi nhánh có nhiều nhân viên, một nhân viên chỉ làm một công việc, một công việc có thể có nhiều nhân viên cùng làm. Các công việc phân biệt với nhau bằng mã công việc, mỗi công việc gồm các thông tin như mã công việc, tên công việc, lương. 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 công việc, thưởng, ngày tuyển dụng, mã chi nhánh.

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, mã loại sản phẩm.

Để quản lý sản phẩm cần có thông tin của nhà sản xuất sản phẩm, cũng như lên đơn đặt hàng đảm bảo luôn cung cấp đủ sản phẩm để bán. Một nhà sản xuất có thể sản xuất ra nhiều sản phẩm. Các nhà sản xuất phân biệt với nhau bằng mã nhà sản xuất. Mỗi nhà sản xuất có các thông tin như mã nhà sản xuất, tên nhà sản xuất, 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 sản xuất.

  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, mã khách hàng, trạng thái, trị giá đơn hàng, mã sản phẩm.

  Để tăng doanh thu cho cửa hàng thì cần dùng mô hình bán qua các ứng dụng (Shopee, Lazada, …). 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, mỗi đơn hàng qua ứng dụng có các thông tin như mã đơn hàng, ngày đặt hàng, mã ứng dụng, trạng thái, trị giá đơn hàng, mã sản phẩm.

Các nhân viên sẽ được chia ca để làm việc. Một nhân viên có thể làm nhiều ca, một ca có thể có nhiều nhân viên. 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.

  Bảng thống kê tài chính sẽ là điều tất yếu để duy trì các hoạt động khác của quán  bao gồm mã giao dịch, loại giao dịch (doanh thu, chi phí), ngày giao dịch, số tiền, mô tả.

### 1.2 . Nghiệp vụ của bài toán

Bài toán quản lý cửa hàng quần áo là một nghiệp vụ quản lý kinh doanh trong lĩnh vực dịch vụ mua sắm. Nó liên quan tới việc quản lý nhân viên, quản lý hàng hóa, quản lý doanh thu và điều hành cửa hàng quần áo. Nhân viên là bộ mặt của cửa hàng và giao tiếp với khách hàng từ lúc bước vào cho đến khi khách ra về. Vì vậy chúng ta cần xây dựng tài liệu đào tạo nghiệp vụ cho nhân viên về quy trình tác nghiệp, văn hóa phục vụ, cách ứng xử, xử lý tình huống, …

-        Quản lý nhân viên:

Khi tuyển dụng nhân sự, cửa hàng cần đưa ra các yêu cầu cũng như các điều kiện (ví dụ: làm ca đêm, không đi trễ quá số buổi quy định, xin nghỉ trước số ngày quy định, …) mà nhân sự đồng ý trong suốt quá trình làm việc. Cửa hàng cần chi trả cho nhân sự mức lương hợp lý. Nhân viên đã được tuyển dụng sẽ trải qua đợt đào tạo về nghiệp vụ, quy trình ở vị trí của bản thân. Người quản lý cần theo dõi và quan sát các nhân sự của mình, chỉ ra những điểm còn thiếu sót để nhân sự khắc phục và lời khen cho những điểm tốt của nhân sự để nhân sự lấy đó làm động lực.

* Quy trình thực hiện khi có đơn hàng qua ứng dụng (Shopee, Lazada, …).

  Nhân viên cửa hàng sẽ có một chiếc máy tính đăng nhập vào các ứng dụng. Khi có đơn hàng được đặt qua các ứng dụng, máy tính 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 sản phẩm có trong đơn hàng qua ứng dụng. Sau đó, nhân viên kho sẽ gói hàng và giao cho đơn vị giao hàng.

-        Quản lý hàng hóa, quản lý doanh thu và điều hành cửa hàng chia thành

các giai đoạn:

  Ở giai đoạn đầu, cửa hàng phải tìm được đối tác, nhà sản xuất cung cấp hàng hóa uy tín từ đầu vì có chính sách giá cả, giao hàng, chứng từ rõ ràng để tiện cho việc kiểm soát. Cửa hàng cần thống kê hàng hóa nhập vào và có kho hàng để lưu trữ, bảo quản và kiểm soát hàng hóa. Cửa hàng cần kiểm tra hàng hóa theo hóa đơn nhập hàng và nhập vào chương trình quản lý quán cửa hàng quần áo.

  Tiếp theo, ở giai đoạn vận hành, chúng ta cần xây dựng chi phí của từng sản phẩm và lên giá thành cho các sản phẩm đó (ví dụ: 1 bộ sản phẩm sẽ có nhiều chất liệu khác nhau). Ngoài ra, chúng ta cũng cần xây dựng quy trình làm việc, nghiệp vụ cho các nhân viên thu ngân, nhân viên bán hàng, bảo vệ.

  Cuối cùng, ở giai đoạn kiểm soát, chúng ta cần kiểm soát về nghiệp vụ của các nhân viên, dựa trên những đánh giá của khách hàng về chất lượng phục vụ, trích xuất camera các ca làm việc để quan sát và đưa ra đánh giá về từng nhân sự. Ngoài ra, cửa hàng cần phải kiểm soát được thu chi hàng ngày. Nhân viên thu ngân sẽ được phân quyền để xem bảng doanh thu theo ngày để tổng kết ca. Nhân viên sẽ kiểm tra xem số tiền thu được có đúng với doanh thu của ca đó không rồi mới giao ca cho nhân viên khác. Người quản lý sẽ được phân quyền để chi tiền cho số tiền nhập hàng. Kho hàng của quán cần được kiểm soát chặt chẽ. Kho hàng cần được kiểm tra định kỳ hàng tuần để đối chiếu với số liệu thực tế ghi nhận ở chương trình quản lý quán. Người quản lý cần phải nắm các bảng báo cáo phân tích như báo cáo doanh thu (theo ngày, tháng, năm), báo cáo chi phí (chi tiết các khoản chi), báo cáo thất thoát. Chủ của chuỗi cửa hàng quần áo sẽ dựa trên các báo cáo để nắm rõ doanh thu, chi phí, lương nhân sự,

  Qua đó, chủ cửa hàng sẽ đưa ra những điều chỉnh và phương án có lợi cho những quý tiếp theo.

### 1.3 . Mô tả chức năng của bài toán

* **Đối với nhân viên:**

***Chức năng bán hàng:*** Tạo đơn hàng, thêm sản phẩm, xóa sản phẩm, xuất hóa đơn thanh toán.

***Quản lý khách hàng:*** Cho phép nhân viên tạo thẻ thành viên cho khách hàng, nhập số điện thoại và thông tin 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.

***Xem bảng ca làm việc:*** Cho phép nhân viên xem bảng phân ca làm việc trong tuần.

* **Đối với người quản lý:**

***Quản lý kho:*** Cho phép người quản lý biết được số lượng của từng sản phẩm còn trong kho hàng, ngày nhập hàng của sản phẩm.

***Quản lý nhân viên:*** Cho phép người quản lý nhập, thêm, xóa, sửa thông tin của nhân viên. Người quản lý có thể xem các thông tin để liên lạc được với nhân viên như số điện thoại.

***Quản lý ca làm việc của nhân viên:*** Các nhân viên sẽ đăng ký ca với người quản lý và người quản lý sẽ thêm ca làm việc vào bảng phân ca cho các nhân viên. Người quản lý có quyền thêm, xóa, sửa ca làm việc của các nhân viên.

***Quản lý doanh thu:*** Cho phép cửa hàng quần áo tính toán doanh thu theo ngày, tuần, tháng hoặc năm, dựa trên thông tin về đơn hàng và giao dịch tài chính.

***Quản lý tài chính:*** Cho người quản lý tạo ra các giao dịch tài chính, bao gồm mã giao dịch, loại giao dịch (doanh thu hoặc chi phí), ngày giao dịch, số tiền giao dịch và mô tả giao dịch.

### 1.4. Tính lợi nhuận của một cửa hàng:

  Lợi nhuận của một cửa hàng quần áo phụ thuộc vào nhiều yếu tố, bao gồm doanh thu, chi phí và giá vốn. Dưới đây là công thức cơ bản để tính lợi nhuận của cửa hàng quần áo:

**Lợi nhuận = Doanh thu - Giá vốn - Chi phí**

**Doanh thu:** Đây là tổng số tiền mà cửa hàng quần áo kiếm được từ việc bán sản phẩm. Bạn có thể tính doanh thu bằng cách nhân số lượng sản phẩm bán được với giá bán của mỗi sản phẩm.

**Giá vốn:** Đây là tổng số tiền mà bạn phải trả để mua sản phẩm từ nhà cung cấp hoặc sản xuất. Nếu bạn đã tính toán giá vốn trung bình cho mỗi sản phẩm, bạn có thể nhân số lượng sản phẩm bán được để tính tổng giá vốn.

**Chi phí:** Chi phí này bao gồm tất cả các chi phí liên quan đến việc vận hành cửa hàng quần áo, chẳng hạn như thuê mặt bằng, lương nhân viên, tiền điện, nước, quảng cáo, vận chuyển, và các chi phí khác. Bạn cần tính tổng tất cả các khoản chi phí này.

**Giả sử một cửa hàng quần áo và trong một tháng:**

**Doanh thu:** Cửa hàng đã bán được 500 chiếc áo, quần với giá bán là 300,000 VNĐ mỗi chiếc. Tính toán doanh thu như sau:

Doanh thu = Số lượng sản phẩm bán được x Giá bán

Doanh thu = 500 chiếc x 300,000 VNĐ/chiếc = 150,000,000 VNĐ

**Giá vốn**: Bạn đã mua 200 chiếc áo, quần từ nhà cung cấp với giá trung bình là 150,000 VNĐ mỗi chiếc. Tính toán giá vốn như sau:

Giá vốn = Số lượng sản phẩm bán được x Giá vốn trung bình

Giá vốn = 500 chiếc x 150,000 VNĐ/chiếc = 75,000,000 VNĐ

**Chi phí:** Trong tháng, bạn đã có các chi phí sau đây:

* Thuê mặt bằng: 10,000,000 VNĐ
* Lương nhân viên: 20,000,000 VNĐ
* Tiền điện, nước: 1,000,000 VNĐ
* Chi phí quảng cáo: 2,000,000 VNĐ

Tính tổng chi phí: 10,000,000 VNĐ +20,000,000 VNĐ + 1,000,000 VNĐ + 2,000,000 VNĐ = 33,000,000 VNĐ

Bây giờ, chúng ta có thể sử dụng công thức lợi nhuận:

Lợi nhuận = Doanh thu - Giá vốn - Chi phí

Lợi nhuận = 150,000,000 VNĐ -75,000,000 VNĐ - 33,000,000 VNĐ =42,000,000 VNĐ

Lợi nhuận của cửa hàng quần áo trong tháng này là 42,000,000 VNĐ.

### 1.5. Tính lợi nhuận của chuỗi cửa hàng:

Ta có thể dựa trên công thức tính lợi nhuận của một cửa hàng cụ thể và sau đó tổng hợp lợi nhuận từ tất cả các cửa hàng trong chuỗi.

Công thức:

**Tổng lợi nhuận = Lợi nhuận CH1 + Lợi nhuận CH2 + Lợi nhuận CH3 +...+ Lợi nhuận CHn**

Trong đó:

**Lợi nhuận CH1:** Là lợi nhuận thu được từ quá trình kinh doanh của cửa hàng số 1 (được tính bởi công thức ở mục 1.4).Tương tự ở **Lợi nhuận CH2** cho đến **Lợi nhuận CHn.**

Ví dụ: Chuỗi cửa hàng quần áo có 3 cửa hàng. Trong tháng 9 năm 2023 cửa hàng số 1 có lợi nhuận 42,000,000 VNĐ, cửa hàng số 2 có lợi nhuận 60,000,000 VNĐ, cửa hàng số 3 có lợi nhuận 50,000,000 VNĐ.

Từ công thức trên có thể tính được tổng lợi nhuận của chuỗi:

**Tổng lợi nhuận=** 42,000,000 + 60,000,000 + 50,000,000 = 152,000,000 VNĐ

### 1.6. Tính lương nhân viên:

Có 2 trường hợp cần tính tiền lương: trường hợp 1 tính lương cho nhân viên làm fulltime(làm theo ngày), trường hợp 2 tính lương cho nhân viên làm parttime(làm theo ca 4 tiếng).

Trường hợp 1: bao gồm nhân viên quản lý và nhân viên bảo vệ:

1. **Lương cơ bản:** Là mức lương mà nhân viên quản lý nhận được hàng tháng.
2. **Tiền Thưởng:** đây là khoản tiền thưởng mà nhân viên được trả cho các ngày làm việc vào ngày lễ (mỗi ngày làm việc vào ngày lễ nhân viên này sẽ được nhân đôi lương) . Tuy nhiên đối với nhân viên quản lý khi doanh thu cửa hàng đạt được chỉ tiêu nhất định (ở đây đạt chỉ tiêu của cửa hàng đề ra sẽ được thưởng thêm 3% lợi nhuận). Công thức tính tiền thưởng được tính như sau:

**Tiền Thưởng = Lợi nhuận x 3% + Số ngày thưởng x Lương ngày thưởng**

(trong trường hợp này thưởng doanh thu chỉ áp dụng cho nhân viên quản lý)

1. **Lương tổng:** tổng số lương nhân viên nhận được trong 1 tháng.

**Lương tổng = Lương cơ bản + Tiền thưởng**

Vd: Giả sử cửa hàng có doanh thu đạt 150.000.000đ ,có lợi nhuận là 42.000.000 và trong tháng có 2 ngày lễ. Trong đó lương cơ bản mà nhân viên quản lý nhận được là 10.000.000 VNĐ/tháng(333.333 VNĐ/ngày), lương cơ bản của nhân viên bảo vệ là 7.000.000 VNĐ(233.333 VNĐ/ngày).

+ Lương nhân viên quản lý sẽ được tính như sau:

Tiền thưởng = 42.000.000 x 3% + 2 x 333.333= 1.926.666 VNĐ

Tổng lương = 10.000.000 + 1.926.666= 11.926.666 VNĐ

+ Lương nhân viên bảo vệ sẽ được tính như sau:

Tiền thưởng = 2 x 233.333 = 466.666 VNĐ

Tổng lương = 7.000.000 + 466.666=7.466.666 VNĐ

Trường hợp 2: Bao gồm nhân viên thu ngân và nhân viên bán hàng

1. **Lương theo giờ (lương/giờ):** đây là số tiền nhân viên được trả cho việc làm công việc cố định mà họ được giao. Đây là số tiền cố định mà nhân viên được trả trong 1 giờ. ( ***Quy định mỗi ca sẽ làm là 4 giờ***).

2. **Thời gian làm việc:** đây là số giờ mà nhân viên đã làm việc trong tháng.

**Thời gian làm việc = Số ca x 4 giờ**

3. **Lương cơ bản:** được tính theo công thức:

**Lương cơ bản = Lương theo giờ x thời gian làm việc**

4. **Tiền thưởng:** đây là khoản tiền thưởng mà nhân viên được trả cho các ca làm việc vào ngày lễ(mỗi ca làm việc này nhân viên sẽ được cộng một đơn vị vào ca thưởng) hoặc doanh thu cửa hàng đạt được chỉ tiêu nhất định (ở đây đạt chỉ tiêu của cửa đề ra sẽ được thưởng thêm 1% lợi nhuận) và tiền lương làm trong ca thưởng sẽ được nhân đôi. Công thức tính tiền thưởng được tính như sau:

**Tiền thưởng = lợi nhuận x 1% + số ca thưởng x 4 giờ x Lương theo giờ (quy định mỗi ca làm là 4 giờ).**

5. **Lương tổng:** tổng số lương nhân viên nhận được trong 1 tháng.

**Lương tổng = Lương cơ bản + Tiền thưởng**

VD:

Giả sử cửa hàng có doanh thu đạt 150.000.000đ ,có lợi nhuận là 42.000.000 và trong tháng có 2 ngày lễ. Nhân viên thu ngân, nhân viên bán hàng được tính lương theo giờ làm việc như sau:

Lương theo giờ của nhân viên thu ngân: 30.000 đồng/giờ

Lương theo giờ của nhân viên bán hàng và nhân viên bảo vệ: 25.000 đồng/giờ

Vào tháng này, nhân viên thu ngân A đã làm việc 35 ca và 5 ca thưởng, nhân viên bán hàng B đã làm việc 30 ca và 3 ca thưởng.

**Do đó, lương của nhân viên thu ngân A trong 1 tháng sẽ là:**

Lương cơ bản = Số ca x 4 giờ x Lương theo giờ

= 35 ca x 4h x 30.000 đồng/giờ

= 4.200.000 đồng

Tiền thưởng = lợi nhuận x 1% + số ca thưởng x 4 giờ x Lương/Giờ

= 42.000.000 x 1% +  5 ca thưởng x 4 giờ x 30.000 đồng/giờ

= 1.020.000 đồng

Lương tổng = Lương cơ bản + Tiền thưởng = 4.200.000 + 1.020.000 = 5.220.000đồng

**Tương tự, lương của nhân viên bán hàng B sẽ là:**

Lương cơ bản = Số ca x 4 giờ x Lương/giờ

= 30 ca x 4 giờ x 25.000 đồng/giờ

= 3.000.000 đồng

Tiền thưởng = doanh thu x 1% + số ca thưởng x 4 giờ x Lương/Giờ

=42.000.000 x 1% + 3 ca x 4 giờ x 25.000 đồng/giờ

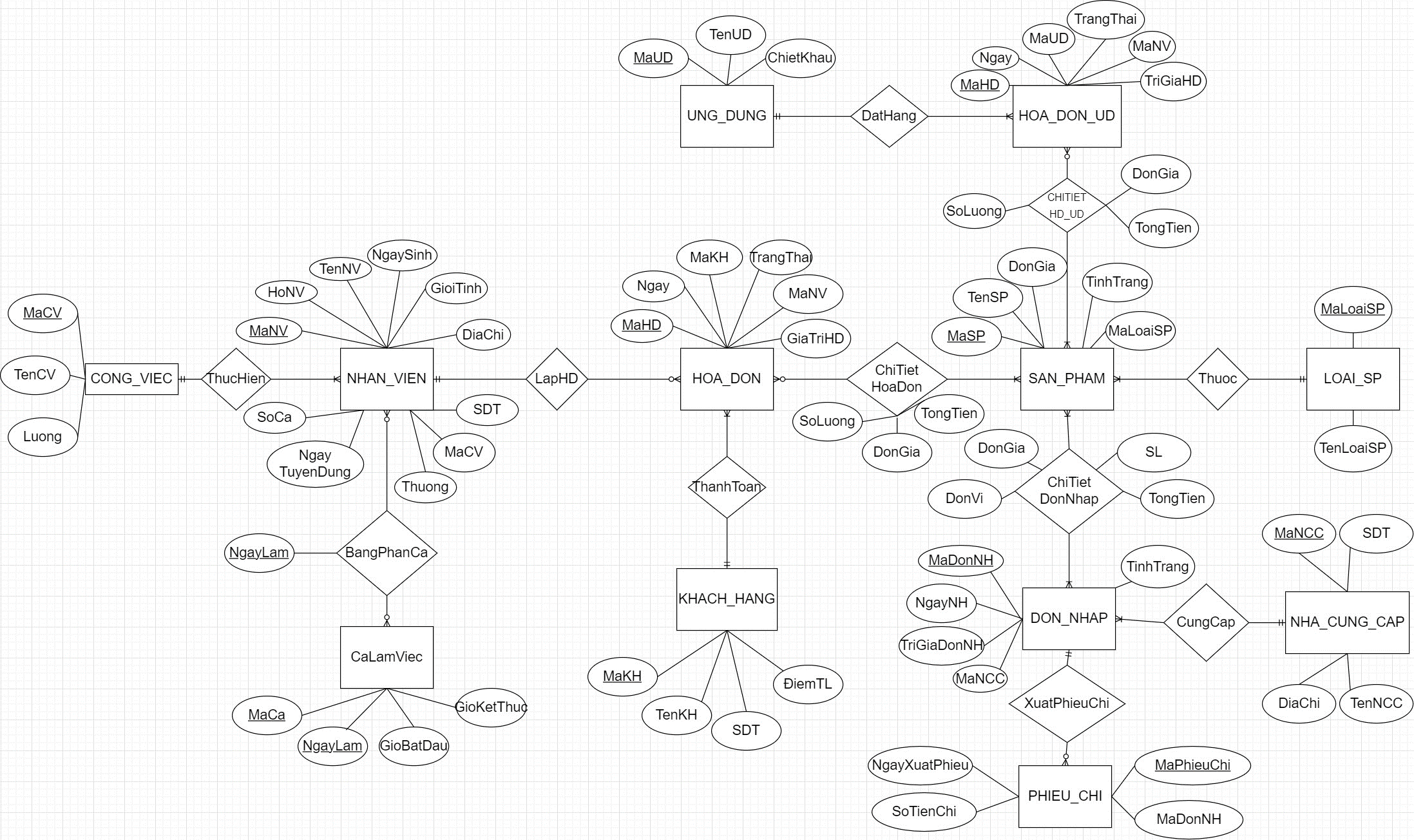
= 720.000 đồng

Lương tổng = Lương cơ bản + Tiền thưởng = 3.000.000 + 720.000 = 3.720.000 đồng

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 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. Thiết kế cơ sở dữ liệu mức logic Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* CONG\_VIEC(MaCV, TenCV, Luong)
* NHAN\_VIEN(MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, NgayTuyenDung, SoCa, Thuong, MaCV, SDT)
* KHACH\_HANG(MaKH, TenKH, SDT, DiemTL)
* NHA\_CUNG\_CAP(MaNCC, SDT, TenNCC, DiaChi)
* DON\_NHAP(MaDonNH, NgayNH, TriGiaDonNH, MaNCC,TinhTrang)
* LOAI\_SP(MaLoaiSP, TenLoaiSP)
* SAN\_PHAM(MaSP, TenSP, DonGia, TinhTrang, MaLoaiSP)
* ChiTietDonNhap(MaSP, MaDonNH, SL, TongTien, DonGia,DonVi)
* UNG\_DUNG(MaUD, TenUD, ChietKhau)
* HOA\_DON\_UD(MaHD, Ngay, MaUD, MaNV, TriGiaHD, TrangThai)
* CHITIETHD\_UD(MaHD\_UD, MaSP, SoLuong, TongTien, DonGia)
* HOA\_DON(MaHD, Ngay, MaKH, MaNV, GiaTriHD, TrangThai)
* ChiTietHoaDon(MaHD, MaSP, SoLuong, TongTien, DonGia)
* PHIEU\_CHI(MaPhieuChi, MaDonNH, SoTienChi, NgayXuatPhieu)
* CaLamViec(MaCa, NgayLam, GioBatDau, GioKetThuc)
* BangPhanCa(MaCa, MaNV, NgayLam)

## 3. Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| STT | Bảng | Ràng buộc |
| 1 | CONG\_VIEC | Khóa chính MaCV,  CHECK Luong > 0 |
| 2 | NHAN\_VIEN | Khóa chính MaNV,  Khóa ngoại MaCV đến bảng CONG\_VIEC,  NOT NULL HoNV,  NOT NULL TenNV,  NOT NULL SDT,  CHECK NgaySinh (>=18 tuổi),  CHECK SDT (=10),  CHECK NgayTuyenDung (>0) |
| 3 | KHACH\_HANG | Khoá chính MaKH,  NOT NULL TenKH,  NOT NULL SDT,  CHECK SDT (=10) |
| 4 | NHA\_CUNG\_CAP | Khoá chính MaNCC,  NOT NULL TenNCC,  NOT NULL SDT,  CHECK SDT(=10) |
| 5 | DON\_NHAP | Khoá chính MaDonNH,  Khoá ngoại MaNCC đến bảng NHA\_CUNG\_CAP.  CHECK NgayNH (>=0),  NOT NULL TriGiaDonNH |
| 6 | LOAI\_SP | Khoá chính MaLoaiSP,  NOT NULL TenLoaiSP |
| 7 | SAN\_PHAM | Khoá chính MaSP,  Khoá ngoại MaLoaiSP đến bảng LOAI\_SP,  NOT NULL TenSP,  NOT NULL DonGia |
| 8 | ChiTietDonNhap | Khóa chính (MaDonNH, MaSP),  Khóa ngoại MaDonNH đến bảng DON\_NHAP,  Khóa ngoại MaSP đến bảng SAN\_PHAM,  CHECK SL(>0), |
| 9 | UNG\_DUNG | Khoá chính MaUngDung,  NOT NULL TenUngDung,  NOT NULL ChietKhauPhanTram |
| 10 | HOA\_DON\_UD | Khoá chính MaHD\_UD,  Khóa ngoại MaUngDung đến bảng UNG\_DUNG,  Khóa ngoại MaNV đến bảng NHAN\_VIEN,  CHECK NgayDatHang (>=0),  NOT NULL TriGiaDH |
| 11 | CHITIETHD\_UD | Khóa chính (MaHD\_UD, MaSP),  Khóa ngoại MaHD\_UD đến bảng HOA\_DON\_UD, Khóa ngoại MaSP đến bảng SAN\_PHAM,  CHECK SL(>0) |
| 12 | HOA\_DON | Khoá chính MaHD,  Khoá ngoại MaKH đến bảng KHACH\_HANG,  Khoá ngoại MaNV đến bảng NHAN\_VIEN,  CHECK NgayDatHang (>=0),  NOT NULL TriGiaHD |
| 13 | ChiTietHoaDon | Khoá chính (MaHD, MaSP),  Khoá ngoại MaHD đến bảng HOA\_DON,  Khoá ngoại MaSP đến bảng SAN\_PHAM,  CHECK SL(>0) |
| 14 | PHIEU\_CHI | Khóa chính MaPhieuChi,  Khóa ngoại MaDonNH đến bảng DON\_NHAP, CHECK NgayXuatPhieu(>=0) |
| 15 | CaLamViec | Khóa chính (MaCa, NgayLam),  CHECK NgayLam(>=0) |
| 16 | BangPhanCa | Khóa chính (MaCa, NgayLam, MaNV),  Khóa ngoại (MaCa, NgayLam) đến bảng CaLamViec, Khóa ngoại MaNV đến bảng NHAN\_VIEN |

## Cài đặt các CSDL và các ràng buộc:

Bảng công việc:

|  |
| --- |
| CREATE TABLE CONG\_VIEC(  MaCV nchar(10) CONSTRAINT PK\_CONG\_VIEC PRIMARY KEY,  TenCV nvarchar(50) NOT NULL,  Luong float check (Luong > 0)  ) |

Bảng nhân viên:

|  |
| --- |
| CREATE TABLE NHAN\_VIEN(  MaNV nchar(10) CONSTRAINT PK\_NHAN\_VIEN PRIMARY KEY,  HoNV nvarchar(10) NOT NULL,  TenNV nvarchar(10) NOT NULL,  NgaySinh date check (DATEDIFF(year, NgaySinh, GETDATE())>=18),  GioiTinh nvarchar(3),  DiaChi nvarchar(100),  SDT nchar(11) check (len(SDT)=10),  MaCV nchar(10) CONSTRAINT FK\_NHAN\_VIEN\_CV FOREIGN KEY REFERENCES  CONG\_VIEC(MaCV),  SoCa int,  Thuong int,  NgayTuyenDung date check (DATEDIFF(day, NgayTuyenDung, GETDATE())>=0)  ) |

Bảng khách hàng:

|  |
| --- |
| CREATE TABLE KHACH\_HANG(  MaKH nchar(10) CONSTRAINT PK\_KHACH\_HANG PRIMARY KEY,  TenKH nvarchar(50) NOT NULL,  SDT nchar(11) NOT NULL check (len(SDT)=10),  DiemTL int  ) |

Bảng nhà cung cấp:

|  |
| --- |
| CREATE TABLE NHA\_CUNG\_CAP(  MaNCC nchar(10) CONSTRAINT PK\_NHA\_CUNG\_CAP PRIMARY KEY,  TenNCC nvarchar(50) NOT NULL,  DiaChi nchar(100),  SDT nchar(10) NOT NULL check (len(SDT)=10)  ) |

Bảng đơn nhập hàng:

|  |
| --- |
| CREATE TABLE DON\_NHAP(  MaDonNH nchar(10) CONSTRAINT PK\_DonNhapHang PRIMARY KEY,  NgayNH date check (DATEDIFF(day, NgayNH, GETDATE())>=0),  TriGiaDonNH float NOT NULL,  MaNCC nchar(10) CONSTRAINT FK\_DonNhapHang\_NCC FOREIGN KEY REFERENCES  NHA\_CUNG\_CAP(MaNCC),  ) |

Bảng loại sản phẩm:

|  |
| --- |
| CREATE TABLE LOAI\_SP(  MaLoaiSP nchar(10) CONSTRAINT PK\_LoaiSanPham PRIMARY KEY,  TenLoaiSP nvarchar(50) NOT NULL  ) |

Bảng sản phẩm:

|  |
| --- |
| CREATE TABLE SAN\_PHAM(  MaSP nchar(10) CONSTRAINT PK\_MaSP PRIMARY KEY,  TenSP nvarchar(50) NOT NULL,  DonGia float NOT NULL,  TinhTrang nchar(10) DEFAULT N'Hết hàng',  MaLoaiSP nchar(10) CONSTRAINT FK\_SanPham\_LOAI\_SP FOREIGN KEY REFERENCES  LOAI\_SP  ) |

Bảng chi tiết đơn nhập hàng:

|  |
| --- |
| CREATE TABLE ChiTietDonNhap(  MaDonNH nchar(10) CONSTRAINT FK\_ChiTietNH\_DonNH FOREIGN KEY REFERENCES  DON\_NHAP(MaDonNH),  MaSP nchar(10) CONSTRAINT FK\_ChiTietNH\_SP FOREIGN KEY REFERENCES  SAN\_PHAM(MaSP),  DonGia float,  SL int check (SL>0),  DonVi nchar(10),  TongTien float  CONSTRAINT PK\_ChiTietDonNhap PRIMARY KEY (MaDonNH, MaSP)  ) |

Bảng ứng dụng:

|  |
| --- |
| CREATE TABLE UNG\_DUNG(  MaUngDung nchar(10) CONSTRAINT PK\_UNG\_DUNG PRIMARY KEY,  TenUngDung nvarchar(50) NOT NULL,  ChietKhauPhanTram int NOT NULL  ) |

Bảng hóa đơn ứng dụng:

|  |
| --- |
| CREATE TABLE HOA\_DON\_UD(  MaHD\_UD nchar(10) CONSTRAINT PK\_HOA\_DON\_UD PRIMARY KEY,  NgayDatHang date check (DATEDIFF(day, NgayDatHang, GETDATE())>=0),  MaUngDung nchar(10) CONSTRAINT FK\_HOA\_DON\_UD FOREIGN KEY REFERENCES  UNG\_DUNG(MaUngDung),  MaNV nchar(10) CONSTRAINT FK\_HoaDonUngDung\_NV FOREIGN KEY REFERENCES  NHAN\_VIEN(MaNV),  TriGiaDH int NOT NULL,  TinhTrang nchar(20)  ) |

Bảng chi tiết hoá đơn qua ứng dụng:

|  |
| --- |
| CREATE TABLE CHITIETHD\_UD(  MaHD\_UD nchar(10) CONSTRAINT FK\_ChiTietHD\_UD FOREIGN KEY REFERENCES  HOA\_DON\_UD(MaHD\_UD),  MaSP nchar(10) CONSTRAINT FK\_ChiTietHD\_SP FOREIGN KEY REFERENCES  SAN\_PHAM(MaSP),  SL int check (SL>0),  DonGia float,  TongTien float  CONSTRAINT PK\_CHITIETHD\_UD PRIMARY KEY (MaHD\_UD, MaSP)  ) |

Bảng hoá đơn:

|  |
| --- |
| CREATE TABLE HOA\_DON(  MaHD nchar(10) CONSTRAINT PK\_HOA\_DON PRIMARY KEY,  NgayDatHang date check (DATEDIFF(day, NgayDatHang, GETDATE())>=0),  MaKH nchar(10) CONSTRAINT FK\_HOA\_DON\_KH FOREIGN KEY REFERENCES  KHACH\_HANG(MaKH),  MaNV nchar(10) CONSTRAINT FK\_HOA\_DON\_NV FOREIGN KEY REFERENCES NHAN\_VIEN(MaNV),  TriGiaHD int NOT NULL,  TinhTrang nchar(20)  ) |

Bảng chi tiết hoá đơn:

|  |
| --- |
| CREATE TABLE ChiTietHoaDon(  MaHD nchar(10) CONSTRAINT FK\_ChiTietHoaDon\_HD FOREIGN KEY REFERENCES  HOA\_DON(MaHD),  MaSP nchar(10) CONSTRAINT FK\_ChiTietHoaDon\_SP FOREIGN KEY REFERENCES  SAN\_PHAM(MaSP),  SL int check (SL > 0),  DonGia float,  TongTien float  CONSTRAINT PK\_ChiTietHoaDon PRIMARY KEY (MaHD, MaSP)  ) |

Bảng phiếu chi:

|  |
| --- |
| CREATE TABLE PHIEU\_CHI(  MaPhieuChi nchar(10) CONSTRAINT PK\_PHIEU\_CHI PRIMARY KEY,  MaDonNH nchar(10) CONSTRAINT FK\_PhieuChi\_DonNH FOREIGN KEY REFERENCES  DON\_NHAP(MaDonNH),  NgayXuatPhieu date check (DATEDIFF(day, NgayXuatPhieu, GETDATE())>=0),  SoTienChi float  ) |

Bảng ca làm việc:

|  |
| --- |
| CREATE TABLE CaLamViec(  MaCa nchar(10),  NgayLam date check (DATEDIFF(day, NgayLam, GETDATE())>=0),  GioBatDau nchar(10),  GioKetThuc nchar(10),  CONSTRAINT PK\_CaLamViec PRIMARY KEY (MaCa, NgayLam)  ) |

Bảng phân ca:

|  |
| --- |
| CREATE TABLE BangPhanCa(  MaCa nchar(10),  MaNV nchar(10) CONSTRAINT FK\_PhanCa\_NV FOREIGN KEY REFERENCES NHAN\_VIEN(MaNV),  NgayLam date check (DATEDIFF(day, NgayLam, GETDATE())>=0),  CONSTRAINT PK\_BangPhanCa PRIMARY KEY (MaCa,MaNV, NgayLam),  CONSTRAINT FK\_PhanCa\_Ca FOREIGN KEY (MaCa,NgayLam) REFERENCES CaLamViec(MaCa, NgayLam)  ) |

A computer screen shot of a computer

Description automatically generated

## 5. Các view

a. Xem ca làm việc của nhân viên trong ngày

|  |
| --- |
| CREATE VIEW V\_CaLamTrongNgay AS  SELECT clv.\*, nv.TenNV, nv.MaCV, cv.TenCV  FROM dbo.BangPhanCa pc  INNER JOIN dbo.NHAN\_VIEN nv ON pc.MaNV = nv.MaNV  INNER JOIN dbo.CaLamViec clv ON pc.MaCa = clv.MaCa  INNER JOIN dbo.CONG\_VIEC cv ON nv.MaCV = cv.MaCV  WHERE clv.NgayLam = CONVERT(DATE, GETDATE()) |

b. Xem danh sách sản phẩm

|  |
| --- |
| CREATE VIEW V\_DanhSachSanPham AS  SELECT MaSP, TenSP, DonGia, TinhTrang  FROM dbo.SAN\_PHAM  GO |

c. Xem số lượng sản phẩm đã bán được của mỗi mã sản phẩm:

|  |
| --- |
| CREATE VIEW V\_SoLuongSanPhamDaBan AS  SELECT  sp.MaSP,  ISNULL(SUM(cthd.SL), 0) AS SoLuongDaBanTaiCuaHang,  ISNULL(SUM(cthdUD.SL), 0) AS SoLuongDaBanQuaUD,    ISNULL(SUM(cthd.SL), 0) + ISNULL(SUM(cthdUD.SL), 0) AS TongSoLuongDaBan  FROM  dbo.SAN\_PHAM AS sp  LEFT OUTER JOIN  dbo.ChiTietHoaDon AS cthd ON sp.MaSP = cthd.MaSP  LEFT OUTER JOIN  dbo.HOA\_DON AS hd ON cthd.MaHD = hd.MaHD AND hd.NgayDatHang = CONVERT(DATE, GETDATE())  LEFT OUTER JOIN  dbo.CHITIETHD\_UD AS cthdUD ON sp.MaSP = cthdUD.MaSP  LEFT OUTER JOIN  dbo.HOA\_DON\_UD AS hdUD ON cthdUD.MaHD\_UD = hdUD.MaHD\_UD AND hdUD.NgayDatHang = CONVERT(DATE, GETDATE())  GROUP BY  sp.MaSP  GO |

d. Xem thông tin nhân viên

|  |
| --- |
| CREATE VIEW V\_ThongTinNhanVien AS  SELECT nv.TenNV, nv.NgaySinh, nv.SDT, nv.SoCa, nv.NgayTuyenDung, nv.Thuong,  cv.TenCV, cv.Luong  FROM dbo.NHAN\_VIEN nv JOIN dbo.CONG\_VIEC cv ON nv.MaCV = cv.MaCV  GO |

e. Xem phiếu chi trong ngày

|  |
| --- |
| CREATE VIEW V\_PhieuChiTrongNgay AS  SELECT \*  FROM dbo.PHIEU\_CHI pc  WHERE pc.NgayXuatPhieu = CONVERT(DATE,GETDATE())  GO |

f. Xem đơn nhập hàng chưa giao

|  |
| --- |
| CREATE VIEW vi\_DonNhapHang AS  SELECT \*  FROM dbo.DON\_NHAP dnh  WHERE TinhTrang = N'Chưa giao' |

## 6. Các Trigger

a. Kiểm tra tên sản phẩm có bị trùng không

|  |
| --- |
| CREATE TRIGGER TG\_TrungTenSP  ON dbo.SAN\_PHAM  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.SAN\_PHAM 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 |

b. Kiểm tra số điện thoại khách hàng có bị trùng không

|  |
| --- |
| CREATE TRIGGER TG\_TrungSDT  ON dbo.KHACH\_HANG  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra số điện thoại vừa thêm có bị trùng lặp  IF EXISTS (  SELECT \*  FROM inserted i  WHERE EXISTS (  SELECT \*  FROM dbo.KHACH\_HANG k  WHERE k.SDT = i.SDT AND k.MaKH <> i.MaKH  )  )  BEGIN  -- Nếu trùng thì rollback  RAISERROR( N'Số điện thoại đã tồn tại',16,1)  ROLLBACK;  END  END |

c. Kiểm tra số điện thoại nhân viên có bị trùng không

|  |
| --- |
| CREATE TRIGGER TG\_TrungSDTNhanVien  ON dbo.NHAN\_VIEN  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.NHAN\_VIEN nv  WHERE nv.SDT = i.SDT AND nv.MaNV <> i.MaNV  )  )  BEGIN  -- Nếu trùng thì rollback  RAISERROR ('SDT nhân viên bị trùng', 16, 1)  ROLLBACK;  END  END |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

## Kết nối cơ sở dữ liệu

|  |
| --- |
| public static SqlConnection Con; //Khai báo đối tượng kết nối  public static SqlConnection ConAdmin;  public static void ConnectAdmin()  {  ConAdmin = new SqlConnection(); //Khởi tạo đối tượng  ConAdmin.ConnectionString = @"Data Source=QUOC-VIET\SQLEXPRESS;Initial Catalog=cuaHangQA;Integrated Security=True";  ConAdmin.Open(); //Mở kết nối  //Kiểm tra kết nối  if (ConAdmin.State == ConnectionState.Open)  MessageBox.Show("Kết nối thành công");  else MessageBox.Show("Không thể kết nối với dữ liệu");  }  public static void Connect(string u, string p)  {  Con = new SqlConnection(); //Khởi tạo đối tượng  Con.ConnectionString = @"Data Source=QUOC-VIET;Initial Catalog=cuaHangQA;User ID=" + u + ";Password=" + p;  Con.Open(); //Mở kết nối  //Kiểm tra kết nối  /\*if (Con.State == ConnectionState.Open)  MessageBox.Show("Kết nối thành công");  else MessageBox.Show("Không thể kết nối với dữ liệu");\*/  }  public static void Disconnect()  {  if (Con.State == ConnectionState.Open)  {  Con.Close(); //Đóng kết nối  Con.Dispose(); //Giải phóng tài nguyên  Con = null;  }  }  public static void DisconnectAdmin()  {  if (ConAdmin.State == ConnectionState.Open)  {  ConAdmin.Close(); //Đóng kết nối  ConAdmin.Dispose(); //Giải phóng tài nguyên  ConAdmin = null;  }  }  public static DataTable GetDataToTable(string sql)  {  SqlDataAdapter dap = new SqlDataAdapter(); //Định nghĩa đối tượng thuộc lớp SqlDataAdapter  //Tạo đối tượng thuộc lớp SqlCommand  dap.SelectCommand = new SqlCommand();  dap.SelectCommand.Connection = Functions.Con; //Kết nối cơ sở dữ liệu  dap.SelectCommand.CommandText = sql; //Lệnh SQL  //Khai báo đối tượng table thuộc lớp DataTable  DataTable table = new DataTable();  dap.Fill(table);  return table;  }  public static bool CheckKey(string sql)  {  SqlDataAdapter dap = new SqlDataAdapter(sql, Con);  DataTable table = new DataTable();  dap.Fill(table);  if (table.Rows.Count > 0)  return true;  else return false;  }  public static void RunSQL(string sql)  {  SqlCommand cmd; //Đối tượng thuộc lớp SqlCommand  cmd = new SqlCommand();  cmd.Connection = Con; //Gán kết nối  cmd.CommandText = sql; //Gán lệnh SQL  try  {  cmd.ExecuteNonQuery(); //Thực hiện câu lệnh SQL  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  cmd.Dispose();//Giải phóng bộ nhớ  cmd = null;  } |

## 2. Xem thông tin danh mục

### 2.1.Danh mục nhân viên

|  |
| --- |
| CREATE VIEW [dbo].[v\_LoadEmployee] AS  SELECT MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, CONG\_VIEC.MaCV,TenCV, SoCa,  Thuong, NgayTuyenDung  FROM NHAN\_VIEN, CONG\_VIEC  WHERE NHAN\_VIEN.MaCV= CONG\_VIEC.MaCV |

Thực thi trên C#:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_LoadEmployee";  tblNV = Functions.GetDataToTable(sql); //lấy dữ liệu  dgvNhanVien.DataSource = tblNV;  dgvNhanVien.Columns[0].HeaderText = "Mã nhân viên";  dgvNhanVien.Columns[1].HeaderText = "Họ nhân viên";  dgvNhanVien.Columns[2].HeaderText = "Tên nhân viên";  dgvNhanVien.Columns[3].HeaderText = "Ngày sinh";  dgvNhanVien.Columns[4].HeaderText = "Giới tính";  dgvNhanVien.Columns[5].HeaderText = "Địa chỉ";  dgvNhanVien.Columns[6].HeaderText = "Số điện thoại";  dgvNhanVien.Columns[7].HeaderText = "Tên công việc";  dgvNhanVien.Columns[8].HeaderText = "Tên công việc";  dgvNhanVien.Columns[9].HeaderText = "Số ca";  dgvNhanVien.Columns[10].HeaderText = "Thưởng";  dgvNhanVien.Columns[11].HeaderText = "Ngày tuyển dụng";  dgvNhanVien.Columns[0].Width = 100;  dgvNhanVien.Columns[1].Width = 150;  dgvNhanVien.Columns[2].Width = 100;  dgvNhanVien.Columns[3].Width = 150;  dgvNhanVien.Columns[4].Width = 100;  dgvNhanVien.Columns[5].Width = 100;  dgvNhanVien.Columns[6].Width = 100;  dgvNhanVien.Columns[7].Width = 100;  dgvNhanVien.Columns[8].Width = 100;  dgvNhanVien.Columns[9].Width = 100;  dgvNhanVien.Columns[10].Width = 100;  dgvNhanVien.Columns[11].Width = 150;  dgvNhanVien.AllowUserToAddRows = false;  dgvNhanVien.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

### 2.2.Danh mục bảng phân ca

|  |
| --- |
| CREATE VIEW [dbo].[v\_BangPhanCa]  AS  SELECT BangPhanCa.MaCa, NHAN\_VIEN.MaNV, NHAN\_VIEN.HoNV, NHAN\_VIEN.TenNV, BangPhanCa.NgayLam  FROM BangPhanCa, NHAN\_VIEN  WHERE BangPhanCa.MaNV = NHAN\_VIEN.MaNV |

Thực thi trên C#:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_BangPhanCa";  tblpc = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  dvgPhanCa.DataSource = tblpc; //Nguồn dữ liệu  dvgPhanCa.Columns[0].HeaderText = "Mã ca";  dvgPhanCa.Columns[1].HeaderText = "Mã nhân vien";  dvgPhanCa.Columns[2].HeaderText = "Họ nhân viên";  dvgPhanCa.Columns[3].HeaderText = "Tên nhân viên";  dvgPhanCa.Columns[4].HeaderText = "Ngày làm";  dvgPhanCa.Columns[0].Width = 100;  dvgPhanCa.Columns[1].Width = 300;  dvgPhanCa.Columns[2].Width = 100;  dvgPhanCa.Columns[3].Width = 100;  dvgPhanCa.Columns[4].Width = 100;  dvgPhanCa.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp  dvgPhanCa.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

### 2.3.Danh mục hóa đơn bán hàng

|  |
| --- |
| CREATE VIEW v\_HoaDonView AS  SELECT a.MaHD,a.NgayDatHang,c.TenKH,a.MaKH,a.MaNV,b.MaSP,b.SL, b.DonGia, a.TriGiaHD,c.SDT  FROM HOA\_DON as a  JOIN ChiTietHoaDon as b ON a.MaHD=b.MaHD  JOIN KHACH\_HANG c ON a.MaKH = c.MaKH; |

Thực thi trên C#:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_HoaDonView";  tblCTHDB = Functions.GetDataToTable(sql); //lấy dữ liệu  dgvHDBanHang.DataSource = tblCTHDB;  dgvHDBanHang.Columns[0].HeaderText = "Mã hóa đơn";  dgvHDBanHang.Columns[1].HeaderText = "Ngày mua";  dgvHDBanHang.Columns[2].HeaderText = "Tên khách hàng";  dgvHDBanHang.Columns[3].HeaderText = "Mã khách hàng";  dgvHDBanHang.Columns[4].HeaderText = "Mã nhân viên";  dgvHDBanHang.Columns[5].HeaderText = "Mã sản phẩm";  dgvHDBanHang.Columns[6].HeaderText = "Số lượng";  dgvHDBanHang.Columns[7].HeaderText = "Đơn giá";  dgvHDBanHang.Columns[8].HeaderText = "Giá trị hóa đơn";  dgvHDBanHang.AllowUserToAddRows = false;  dgvHDBanHang.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

### 2.4.Hóa đơn ứng dụng:

|  |
| --- |
| CREATE VIEW v\_HoaDonUngDung AS  SELECT a.MaHD\_UD,a.NgayDatHang,c.TenUngDung,c.MaUngDung,a.MaNV,b.MaSP,b.SL, b.DonGia, a.TriGiaDH  FROM HOA\_DON\_UD as a  JOIN CHITIETHD\_UD as b ON a.MaHD\_UD=b.MaHD\_UD  JOIN UNG\_DUNG c ON a.MaUngDung = c.MaUngDung; |

Thực thi trên C#:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_HoaDonUngDung";  tblHDUng\_Dung = Functions.GetDataToTable(sql); //lấy dữ liệu  dgvHDBanHang.DataSource = tblHDUng\_Dung;  dgvHDBanHang.Columns[0].HeaderText = "Mã hóa đơn ";  dgvHDBanHang.Columns[1].HeaderText = "Ngày mua";  dgvHDBanHang.Columns[2].HeaderText = "Tên ứng dụng";  dgvHDBanHang.Columns[3].HeaderText = "Mã ứng dụng";  dgvHDBanHang.Columns[4].HeaderText = "Mã nhân viên";  dgvHDBanHang.Columns[4].HeaderText = "Mã sản phẩm";  dgvHDBanHang.Columns[5].HeaderText = "Số lượng";  dgvHDBanHang.Columns[6].HeaderText = "Đơn giá";  dgvHDBanHang.Columns[7].HeaderText = "Giá trị hóa đơn";  dgvHDBanHang.Columns[0].Width = 150;  dgvHDBanHang.Columns[1].Width = 130;  dgvHDBanHang.Columns[2].Width = 100;  dgvHDBanHang.Columns[3].Width = 80;  dgvHDBanHang.Columns[4].Width = 100;  dgvHDBanHang.Columns[5].Width = 100;  dgvHDBanHang.Columns[6].Width = 100;  dgvHDBanHang.Columns[7].Width = 100;  dgvHDBanHang.AllowUserToAddRows = false;  dgvHDBanHang.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

### 2.5.Danh mục sản phẩm:

|  |
| --- |
| CREATE VIEW [dbo].[view\_danhSachSanPham]  AS  SELECT MaSP, TenSP, DonGia, TinhTrang, SAN\_PHAM.MaLoaiSP, TenLoaiSP  FROM dbo.SAN\_PHAM INNER JOIN  dbo.LOAI\_SP ON dbo.SAN\_PHAM.MaLoaiSP = dbo.LOAI\_SP.MaLoaiSP |

Thực thi trên C#:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = " SELECT \* FROM view\_danhSachSanPham";  tblsp = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  dgvSanPham.DataSource = tblsp; //Nguồn dữ liệu  dgvSanPham.Columns[0].HeaderText = "Mã sản phẩm";  dgvSanPham.Columns[1].HeaderText = "Tên sản phẩm";  dgvSanPham.Columns[2].HeaderText = "Đơn Giá";  dgvSanPham.Columns[3].HeaderText = "Tình Trạng";  dgvSanPham.Columns[4].HeaderText = "Mã loại sản phẩm";  dgvSanPham.Columns[5].HeaderText = "Tên loại sản phẩm";  dgvSanPham.Columns[0].Width = 100;  dgvSanPham.Columns[1].Width = 300;  dgvSanPham.Columns[2].Width = 100;  dgvSanPham.Columns[3].Width = 100;  dgvSanPham.Columns[4].Width = 100;  dgvSanPham.Columns[5].Width = 100;  dgvSanPham.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp  dgvSanPham.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

### 2.6.Danh mục phiếu chi:

|  |
| --- |
| CREATE VIEW [dbo].[DonNhapHangView]  AS  SELECT DNH.MaDonNH, DNH.NgayNH, DNH.MaNCC, DNH.TinhTrang, PC.MaPhieuChi,  PC.NgayXuatPhieu, CTH.DonGia, CTH.SL, CTH.DonVi, PC.SoTienChi  FROM DON\_NHAP DNH  LEFT JOIN PHIEU\_CHI PC ON DNH.MaDonNH = PC.MaDonNH  LEFT JOIN ChiTietDonNhap CTH ON DNH.MaDonNH = CTH.MaDonNH; |

Thực thi trên C#:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From DonNhapHangView";  tblpc = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  dataGridView1.DataSource = tblpc; //Nguồn dữ liệu  dataGridView1.Columns[0].HeaderText = "Mã đơn nhập hàng";  dataGridView1.Columns[1].HeaderText = "Ngày nhập hàng";  dataGridView1.Columns[2].HeaderText = "Mã ncc";  dataGridView1.Columns[3].HeaderText = "Tình Trạng";  dataGridView1.Columns[4].HeaderText = "Mã phiếu chi";  dataGridView1.Columns[5].HeaderText = "Ngày xuất phiếu";  dataGridView1.Columns[6].HeaderText = "Đơn giá";  dataGridView1.Columns[7].HeaderText = "Số lượng";  dataGridView1.Columns[8].HeaderText = "Đơn vị";  dataGridView1.Columns[9].HeaderText = "Số tiền chi";  dataGridView1.Columns[0].Width = 100;  dataGridView1.Columns[1].Width = 300;  dataGridView1.Columns[2].Width = 100;  dataGridView1.Columns[3].Width = 100;  dataGridView1.Columns[4].Width = 100;  dataGridView1.Columns[5].Width = 100;  dataGridView1.Columns[6].Width = 100;  dataGridView1.Columns[7].Width = 100;  dataGridView1.Columns[8].Width = 100;  dataGridView1.Columns[9].Width = 100;  dataGridView1.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

## 3. Quản lý khách hàng

### 3.1. Tìm kiếm khách hàng

|  |
| --- |
| CREATE FUNCTION [dbo].[SearchTenKHBySDT](@SDT nchar(11))  RETURNS nvarchar(50)  AS  BEGIN  DECLARE @TenKH nvarchar(50);  SELECT @TenKH = TenKH FROM KHACH\_HANG WHERE SDT = @SDT;  RETURN @TenKH;  END |

Thực thi trên C#:

|  |
| --- |
| private void txtTimKiem\_Click(object sender, EventArgs e)  {  string sdt = txtSDT.Text.Trim();  try  {  SqlCommand cmd = new SqlCommand("SELECT dbo.SearchTenKHBySDT(@SDT)", Functions.Con);  cmd.Parameters.AddWithValue("@SDT", sdt);  object result = cmd.ExecuteScalar();  if (result != null && result != DBNull.Value)  {  string tenKH = result.ToString();  // Hiển thị TenKH trong TextBox  txtTenKhach.Text = tenKH;  }  else  {  MessageBox.Show("Không tìm thấy khách hàng với số điện thoại này.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 3.2. Tạo khách hàng mới

|  |
| --- |
| CREATE PROCEDURE [dbo].[InsertNewKhachHang]  @MaKH nchar(10),  @TenKH nvarchar(50),  @SDT nchar(11)  AS  BEGIN  IF EXISTS (SELECT 1 FROM KHACH\_HANG WHERE SDT = @SDT)  BEGIN  RETURN;  END  INSERT INTO KHACH\_HANG (MaKH, TenKH, SDT, DiemTL)  VALUES (@MaKH, @TenKH, @SDT, 0)  END |

Thực thi trên C#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnSua.Enabled = false;  btnXoa.Enabled = false;  btnBoQua.Enabled = true;  btnLuu.Enabled = true;  btnThem.Enabled = false;  ResetValues();  //txtMaKhach.Enabled = true;  txtMaKhach.Focus();  }  private void btnLuu\_Click(object sender, EventArgs e)  {  string MaKH = txtMaKhach.Text;  string TenKH = txtTenKhach.Text;  string SDT = txtSDT.Text;  if (string.IsNullOrEmpty(MaKH))  {  MessageBox.Show("Please enter a valid value for 'MaKH'.");  return;  }  SqlCommand cmd = new SqlCommand("InsertNewKhachHang", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaKH", SqlDbType.NChar).Value = MaKH;  cmd.Parameters.Add("@TenKH", SqlDbType.NVarChar).Value = TenKH;  cmd.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công!", "Add Customer", MessageBoxButtons.OK,  MessageBoxIcon.Information);    }  else  {  MessageBox.Show("Thêm thất bại", "Add Customer", MessageBoxButtons.OK,  MessageBoxIcon.Error);    }  LoadDataGridView();  ResetValues();    } |

### 3.3. Sửa thông tin khách hàng

|  |
| --- |
| CREATE PROCEDURE [dbo].[UpdateKhachHang]  @MaKH nchar(10),  @TenKH nvarchar(50),  @SDT nchar(11)  AS  BEGIN  IF NOT EXISTS (SELECT 1 FROM KHACH\_HANG WHERE MaKH = @MaKH)  BEGIN  -- Nếu không tìm thấy khách hàng với MaKH tương ứng, không thực hiện cập nhật  RETURN;  END  UPDATE KHACH\_HANG  SET TenKH = @TenKH,  SDT = @SDT  WHERE MaKH = @MaKH;  END |

Thực thi trên C#:

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  string MaKH = txtMaKhach.Text;  string TenKH = txtTenKhach.Text;  string SDT = txtSDT.Text;  if (string.IsNullOrEmpty(MaKH))  {  MessageBox.Show("Please enter a valid value for 'MaKH'.");  return;  }  try  {  SqlCommand cmd = new SqlCommand("UpdateKhachHang", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaKH", SqlDbType.NChar).Value = MaKH;  cmd.Parameters.Add("@TenKH", SqlDbType.NVarChar).Value = TenKH;  cmd.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT;  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Cập nhật thành công!", "Update Customer", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Không tìm thấy khách hàng với MaKH này.", "Update Customer", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  btnBoQua.Enabled = false;  } |

### 3.4. Trigger bắt lỗi trùng số điện thoại khi thêm và chỉnh sửa khách hàng\

|  |
| --- |
| CREATE TRIGGER TG\_TrungSDT  ON dbo.KHACH\_HANG  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra số điện thoại vừa thêm có bị trùng lặp  IF EXISTS (  SELECT \*  FROM inserted i  WHERE EXISTS (  SELECT \*  FROM dbo.KHACH\_HANG k  WHERE k.SDT = i.SDT AND k.MaKH <> i.MaKH  )  )  BEGIN  -- Nếu trùng thì rollback  RAISERROR( N'Số điện thoại đã tồn tại',16,1)  ROLLBACK;  END  END |

## 4. Quản lý hóa đơn bán hàng và hóa đơn ứng dụng

### 4.1. Xuất hóa đơn bán hàng

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_AddHoaDon]  @MaHD nchar(10),  @NgayDatHang date,  @MaKH nchar(10),  @MaNV nchar(10),  @TriGiaHD int  AS  BEGIN  INSERT INTO HOA\_DON(MaHD, NgayDatHang, MaKH, MaNV, TriGiaHD)  VALUES (@MaHD, @NgayDatHang, @MaKH, @MaNV, @TriGiaHD);  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnXoa.Enabled = false;  btnLuu.Enabled = true;  btnInHoaDon.Enabled = false;  btnThem.Enabled = false;  btnBoQua.Enabled = true;  ResetValues();  txtMaHDBan.Text = Functions.CreateKey("HD");  LoadDataGridView();  }  private void btnLuu\_Click(object sender, EventArgs e)  {  string sql;  if (txtMaHDBan.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã hóa đơn ứng dụng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtMaHDBan.Focus();  return;  }  if (cboMaSanPham.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã sản phẩm ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  cboMaSanPham.Focus();  return;  }  if (txtSoLuong.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập số lượng sản phẩm ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtSoLuong.Focus();  return;  }  if (cboMaKhach.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã khách hàng ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  cboMaKhach.Focus();  return;  }  if (cboMaNhanVien.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  cboMaNhanVien.Focus();  return;  }  try  {  SqlCommand cmd = new SqlCommand("por\_AddHoaDon", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@MaHD", txtMaHDBan.Text.Trim());  cmd.Parameters.AddWithValue("@MaSP", cboMaSanPham.Text.Trim());  cmd.Parameters.AddWithValue("@SL", txtSoLuong.Text.Trim());  cmd.Parameters.AddWithValue("@NgayDatHang", txtNgayBan.Value);  cmd.Parameters.AddWithValue("@MaKH", cboMaKhach.Text.Trim());  cmd.Parameters.AddWithValue("@MaNV", cboMaNhanVien.Text.Trim());  cmd.Parameters.AddWithValue("@DonGia", txtDonGia.Text.Trim());  cmd.Parameters.AddWithValue("@TriGiaHD", txtThanhTien.Text.Trim());  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Thêm hóa đơn thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  /\* btnSua.Enabled = true;\*/  btnBoQua.Enabled = false;  btnLuu.Enabled = false;  btnInHoaDon.Enabled = true;  } |

### 4.2.Tìm kiếm hóa đơn bán hàng

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_TimKiemHoaDon]  @NgayBatDau DATE,  @NgayKetThuc DATE  AS  BEGIN  SELECT \*  FROM HOA\_DON  WHERE NgayDatHang BETWEEN  @NgayBatDau AND @NgayKetThuc  END |

Thực thi trên C#:

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e)  {  try  {  SqlDataAdapter dataAdapter = new SqlDataAdapter();  SqlCommand cmd = new SqlCommand("proc\_TimKiemHoaDon", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@NgayBatDau", dtpBatDau.Value.Date);  cmd.Parameters.AddWithValue("@NgayKetThuc", dtpKetThuc.Value.Date);  dataAdapter.SelectCommand = cmd;  DataTable dataTable = new DataTable();  dataAdapter.Fill(dataTable);  // Gán DataTable vào DataGridView  dgvHDBanHang.DataSource = dataTable;  MessageBox.Show("Tìm kiếm hóa đơn thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  btnXoa.Enabled = true;  btnLuu.Enabled = true;  btnInHoaDon.Enabled = true;  btnBoQua.Enabled = true;  } |

### Xóa hóa đơn bán hàng:

|  |
| --- |
| CREATE PROCEDURE por\_XoaHoaDon  @MaHD nchar(30)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRANSACTION;  BEGIN TRY  -- Xóa chi tiết hóa đơn từ bảng ChiTietHoaDon  DELETE FROM ChiTietHoaDon  WHERE MaHD = @MaHD;  -- Xóa hóa đơn từ bảng HOA\_DON  DELETE FROM HOA\_DON  WHERE MaHD = @MaHD;  COMMIT TRANSACTION;  PRINT 'Xóa hóa đơn thành công!';  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  PRINT 'Lỗi trong quá trình xóa hóa đơn!';  END CATCH;  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)  {  try  {  SqlCommand command = new SqlCommand("por\_XoaHoaDon", Functions.Con);  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaHD", txtMaHDBan.Text);  command.ExecuteNonQuery();  ResetValues();  LoadDataGridView();  btnXoa.Enabled = false;  btnInHoaDon.Enabled = false;  MessageBox.Show("Đã xóa hóa đơn thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  } |

### In hóa đơn bán hàng:

Thực thi trên C#:

|  |
| --- |
| private void btnInHoaDon\_Click(object sender, EventArgs e)  {  if (string.IsNullOrEmpty(txtMaHDBan.Text))  {  MessageBox.Show("Vui lòng chọn hóa đơn cần in!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  // Khởi động chương trình Excel  COMExcel.Application exApp = new COMExcel.Application();  COMExcel.Workbook exBook;  COMExcel.Worksheet exSheet;  COMExcel.Range exRange;  string sql;  int hang = 0, cot = 0;  DataTable tblThongtinHD, tblThongtinHang;  exBook = exApp.Workbooks.Add(COMExcel.XlWBATemplate.xlWBATWorksheet);  exSheet = exBook.Worksheets[1];  // Code phần định dạng chung, như đã thấy trong mã cũ  exRange = exSheet.Cells[1, 1];  exRange.Range["A1:Z300"].Font.Name = "Times new roman"; //Font chữ  exRange.Range["A1:B3"].Font.Size = 10;  exRange.Range["A1:B3"].Font.Bold = true;  exRange.Range["A1:B3"].Font.ColorIndex = 5; //Màu xanh da trời  exRange.Range["A1:A1"].ColumnWidth = 7;  exRange.Range["B1:B1"].ColumnWidth = 15;  exRange.Range["A1:B1"].MergeCells = true;  exRange.Range["A1:B1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A1:B1"].Value = "Shop DĐKV";  exRange.Range["A2:B2"].MergeCells = true;  exRange.Range["A2:B2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A2:B2"].Value = "Linh Trung - Thủ Đức";  exRange.Range["A3:B3"].MergeCells = true;  exRange.Range["A3:B3"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A3:B3"].Value = "Điện thoại: (04)38526419";  exRange.Range["C2:E2"].Font.Size = 16;  exRange.Range["C2:E2"].Font.Bold = true;  exRange.Range["C2:E2"].Font.ColorIndex = 3; //Màu đỏ  exRange.Range["C2:E2"].MergeCells = true;  exRange.Range["C2:E2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["C2:E2"].Value = "HÓA ĐƠN BÁN";  // Biểu diễn thông tin chung của hóa đơn bán  sql = "SELECT \* From v\_HoaDonView WHERE MaHD = '" + txtMaHDBan.Text + "'";  tblThongtinHD = Functions.GetDataToTable(sql);  // Tiếp tục với phần code, như đã thấy trong mã cũ  exRange.Range["B6:C9"].Font.Size = 12;  exRange.Range["B6:B6"].Value = "Mã hóa đơn:";  exRange.Range["C6:E6"].MergeCells = true;  exRange.Range["C6:E6"].Value = tblThongtinHD.Rows[0][0].ToString();  exRange.Range["B7:B7"].Value = "Khách hàng:";  exRange.Range["C7:E7"].MergeCells = true;  exRange.Range["C7:E7"].Value = tblThongtinHD.Rows[0][2].ToString();  exRange.Range["B9:B9"].Value = "Điện thoại:";  exRange.Range["C9:E9"].MergeCells = true;  exRange.Range["C9:E9"].Value = tblThongtinHD.Rows[0][9].ToString();  //Lấy thông tin các mặt hàng từ hóa đơn đã chọn  sql = "SELECT \* From v\_HoaDonView WHERE MaHD = '" + txtMaHDBan.Text + "'";  tblThongtinHang = Functions.GetDataToTable(sql);  //Tạo dòng tiêu đề bảng  exRange.Range["A11:J11"].Font.Bold = true;  exRange.Range["A11:J11"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["C11:J11"].ColumnWidth = 12;  exRange.Range["C11:C11"].ColumnWidth = 15;  exRange.Range["D11:D11"].ColumnWidth = 15;  exRange.Range["K11:K11"].ColumnWidth = 15;  exRange.Range["A11:A11"].Value = "STT";  exRange.Range["B11:B11"].Value = "Mã hóa đơn";  exRange.Range["C11:C11"].Value = "Ngày mua";  exRange.Range["D11:D11"].Value = "Tên khách hàng";  exRange.Range["E11:E11"].Value = "Mã khách hàng";  exRange.Range["F11:F11"].Value = "Mã nhân viên";  exRange.Range["G11:G11"].Value = "Mã sản phẩm";  exRange.Range["H11:H11"].Value = "Số lượng";  exRange.Range["I11:I11"].Value = "Đơn giá";  exRange.Range["J11:J11"].Value = "Thành tiền";  exRange.Range["K11:K11"].Value = "SĐT";  for (hang = 0; hang < tblThongtinHang.Rows.Count; hang++)  {  //Điền số thứ tự vào cột 1 từ dòng 12  exSheet.Cells[1][hang + 12] = hang + 1;  for (cot = 0; cot < tblThongtinHang.Columns.Count; cot++)  //Điền thông tin hàng từ cột thứ 2, dòng 12  {  exSheet.Cells[cot + 2][hang + 12] = tblThongtinHang.Rows[hang][cot].ToString();  if (cot == 3) exSheet.Cells[cot + 2][hang + 12] = tblThongtinHang.Rows[hang][cot].ToString();  }  }  exRange = exSheet.Cells[cot][hang + 14];  exRange.Font.Bold = true;  exRange.Value2 = "Tổng tiền:";  exRange = exSheet.Cells[cot + 1][hang + 14];  exRange.Font.Bold = true;  exRange.Value2 = tblThongtinHD.Rows[0][7].ToString();  exRange = exSheet.Cells[6][hang + 17]; //Ô A1  exRange.Range["A1:D1"].MergeCells = true;  exRange.Range["A1:D1"].MergeCells = true;  exRange.Range["A1:D1"].Font.Italic = true;  exRange.Range["A1:D1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  DateTime d = Convert.ToDateTime(tblThongtinHD.Rows[0][1]);  exRange.Range["A1:C1"].Value = "HCM, ngày " + DateTime.Now.Day + " tháng " + DateTime.Now.Month + " năm " + DateTime.Now.Year;  exRange.Range["A2:D2"].MergeCells = true;  exRange.Range["A2:D2"].Font.Italic = true;  exRange.Range["A2:D2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A2:C2"].Value = "Nhân viên bán hàng";  exRange.Range["A6:C6"].MergeCells = true;  exRange.Range["A6:C6"].Font.Italic = true;  exRange.Range["A6:C6"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exApp.Visible = true;  btnBoQua.Enabled = true;  } |

### Xuất hóa đơn ứng dụng:

|  |
| --- |
| CREATE PROCEDURE por\_AddHoaDonUngDung  @MaHD\_UD nchar(30),  @NgayDatHang date,  @MaUngDung nchar(10),  @MaNV nchar(10),  @MaSP nchar(10),  @SL int,  @DonGia float,  @TriGiaDH int  AS  BEGIN  SET NOCOUNT ON;  -- Thêm hóa đơn vào bảng HOA\_DON  INSERT INTO HOA\_DON\_UD(MaHD\_UD, NgayDatHang, MaUngDung, MaNV, TriGiaDH)  VALUES (@MaHD\_UD, @NgayDatHang, @MaUngDung, @MaNV, @TriGiaDH);  -- Thêm chi tiết hóa đơn vào bảng ChiTietHoaDon  INSERT INTO CHITIETHD\_UD(MaHD\_UD, MaSP, SL, DonGia, TongTien)  VALUES (@MaHD\_UD, @MaSP, @SL, @DonGia, @SL \* @DonGia);  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnXoa.Enabled = false;  btnLuu.Enabled = true;  btnInHoaDon.Enabled = false;  btnThem.Enabled = false;  btnBoQua.Enabled = true;  ResetValues();  /\* txtMaHDBan.Text = Functions.CreateKey("HD");\*/  LoadDataGridView();  }  private void btnLuu\_Click(object sender, EventArgs e)  {  string sql;  if (txtMaHDBan.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã hóa đơn ứng dụng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtMaHDBan.Focus();  return;  }  if (cboMaSanPham.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã sản phẩm ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  cboMaSanPham.Focus();  return;  }  if (txtSoLuong.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập số lượng sản phẩm ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtSoLuong.Focus();  return;  }  if (cboMaKhach.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã ứng dụng ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  cboMaKhach.Focus();  return;  }  if (cboMaNhanVien.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  cboMaNhanVien.Focus();  return;  }  try  {  SqlCommand cmd = new SqlCommand("por\_AddHoaDonUngDung", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@MaHD\_UD", txtMaHDBan.Text.Trim());  cmd.Parameters.AddWithValue("@MaSP", cboMaSanPham.Text.Trim());  cmd.Parameters.AddWithValue("@SL", txtSoLuong.Text.Trim());  cmd.Parameters.AddWithValue("@NgayDatHang", txtNgayBan.Value);  cmd.Parameters.AddWithValue("@MaUngDung", cboMaKhach.Text.Trim());  cmd.Parameters.AddWithValue("@MaNV", cboMaNhanVien.Text.Trim());  cmd.Parameters.AddWithValue("@DonGia", txtDonGia.Text.Trim());  cmd.Parameters.AddWithValue("@TriGiaDH", txtThanhTien.Text.Trim());  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Thêm hóa đơn thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  /\* btnSua.Enabled = true;\*/  btnBoQua.Enabled = false;  btnLuu.Enabled = false;  btnInHoaDon.Enabled = true;  } |

### 4.6.Tìm kiếm hóa đơn ứng dụng

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_TimKiemHoaDonUngDung]  @NgayBatDau DATE,  @NgayKetThuc DATE  AS  BEGIN  SELECT \*  FROM HOA\_DON\_UD  WHERE NgayDatHang BETWEEN  @NgayBatDau AND @NgayKetThuc  END |

Thực thi trên C#:

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e)  {  try  {  SqlDataAdapter dataAdapter = new SqlDataAdapter();  SqlCommand cmd = new SqlCommand("proc\_TimKiemHoaDonUngDung", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@NgayBatDau", dtpBatDau.Value.Date);  cmd.Parameters.AddWithValue("@NgayKetThuc", dtpKetThuc.Value.Date);  dataAdapter.SelectCommand = cmd;  DataTable dataTable = new DataTable();  dataAdapter.Fill(dataTable);  // Gán DataTable vào DataGridView  dgvHDBanHang.DataSource = dataTable;  MessageBox.Show("Tìm kiếm hóa đơn thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  btnXoa.Enabled = true;  btnLuu.Enabled = true;  btnInHoaDon.Enabled = true;  btnBoQua.Enabled = true;  } |

### 4.7.Xóa hóa đơn ứng dụng:

|  |
| --- |
| CREATE PROCEDURE por\_XoaHoaDonUngDung  @MaHD\_UD nchar(30)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRANSACTION;  BEGIN TRY  -- Xóa chi tiết hóa đơn từ bảng ChiTietHoaDon  DELETE FROM CHITIETHD\_UD  WHERE MaHD\_UD = @MaHD\_UD;  -- Xóa hóa đơn từ bảng HOA\_DON  DELETE FROM HOA\_DON\_UD  WHERE MaHD\_UD = @MaHD\_UD;  COMMIT TRANSACTION;  PRINT 'Xóa hóa đơn thành công!';  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  PRINT 'Lỗi trong quá trình xóa hóa đơn!';  END CATCH;  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)  {  try  {  SqlCommand command = new SqlCommand("por\_XoaHoaDonUngDung", Functions.Con);  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaHD\_UD", txtMaHDBan.Text);  command.ExecuteNonQuery();  ResetValues();  LoadDataGridView();  btnXoa.Enabled = false;  btnInHoaDon.Enabled = false;  MessageBox.Show("Đã xóa hóa đơn thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  } |

### 4.8.In hóa đơn ứng dụng

|  |
| --- |
| private void btnInHoaDon\_Click(object sender, EventArgs e)  {  if (string.IsNullOrEmpty(txtMaHDBan.Text))  {  MessageBox.Show("Vui lòng chọn hóa đơn cần in!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  // Khởi động chương trình Excel  COMExcel.Application exApp = new COMExcel.Application();  COMExcel.Workbook exBook;  COMExcel.Worksheet exSheet;  COMExcel.Range exRange;  string sql;  int hang = 0, cot = 0;  DataTable tblThongtinHD, tblThongtinHang;  exBook = exApp.Workbooks.Add(COMExcel.XlWBATemplate.xlWBATWorksheet);  exSheet = exBook.Worksheets[1];  // Code phần định dạng chung, như đã thấy trong mã cũ  exRange = exSheet.Cells[1, 1];  exRange.Range["A1:Z300"].Font.Name = "Times new roman"; //Font chữ  exRange.Range["A1:B3"].Font.Size = 10;  exRange.Range["A1:B3"].Font.Bold = true;  exRange.Range["A1:B3"].Font.ColorIndex = 5; //Màu xanh da trời  exRange.Range["A1:A1"].ColumnWidth = 7;  exRange.Range["B1:B1"].ColumnWidth = 15;  exRange.Range["A1:B1"].MergeCells = true;  exRange.Range["A1:B1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A1:B1"].Value = "Shop DĐKV";  exRange.Range["A2:B2"].MergeCells = true;  exRange.Range["A2:B2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A2:B2"].Value = "Linh Trung - Thủ Đức";  exRange.Range["A3:B3"].MergeCells = true;  exRange.Range["A3:B3"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A3:B3"].Value = "Điện thoại: (04)38526419";  exRange.Range["C2:E2"].Font.Size = 16;  exRange.Range["C2:E2"].Font.Bold = true;  exRange.Range["C2:E2"].Font.ColorIndex = 3; //Màu đỏ  exRange.Range["C2:E2"].MergeCells = true;  exRange.Range["C2:E2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["C2:E2"].Value = "HÓA ĐƠN ỨNG DỤNG";  // Biểu diễn thông tin chung của hóa đơn bán  sql = "SELECT \* From v\_HoaDonUngDung WHERE MaHD\_UD = '" + txtMaHDBan.Text + "'";  tblThongtinHD = Functions.GetDataToTable(sql);  // Tiếp tục với phần code, như đã thấy trong mã cũ  exRange.Range["B6:C9"].Font.Size = 12;  exRange.Range["B6:B6"].Value = "Mã hóa đơn:";  exRange.Range["C6:E6"].MergeCells = true;  exRange.Range["C6:E6"].Value = tblThongtinHD.Rows[0][0].ToString();  exRange.Range["B7:B7"].Value = "Ứng dụng:";  exRange.Range["C7:E7"].MergeCells = true;  exRange.Range["C7:E7"].Value = tblThongtinHD.Rows[0][2].ToString();  //Lấy thông tin các mặt hàng từ hóa đơn đã chọn  sql = "SELECT \* From v\_HoaDonUngDung WHERE MaHD\_UD = '" + txtMaHDBan.Text + "'";  tblThongtinHang = Functions.GetDataToTable(sql);  //Tạo dòng tiêu đề bảng  exRange.Range["A11:J11"].Font.Bold = true;  exRange.Range["A11:J11"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["C11:J11"].ColumnWidth = 12;  exRange.Range["C11:C11"].ColumnWidth = 15;  exRange.Range["D11:D11"].ColumnWidth = 15;  exRange.Range["K11:K11"].ColumnWidth = 15;  exRange.Range["A11:A11"].Value = "STT";  exRange.Range["B11:B11"].Value = "Mã hóa đơn";  exRange.Range["C11:C11"].Value = "Ngày đặt";  exRange.Range["D11:D11"].Value = "Tên ứng dụng";  exRange.Range["E11:E11"].Value = "Mã ứng dụng";  exRange.Range["F11:F11"].Value = "Mã nhân viên";  exRange.Range["G11:G11"].Value = "Mã sản phẩm";  exRange.Range["H11:H11"].Value = "Số lượng";  exRange.Range["I11:I11"].Value = "Đơn giá";  exRange.Range["J11:J11"].Value = "Thành tiền";  for (hang = 0; hang < tblThongtinHang.Rows.Count; hang++)  {  //Điền số thứ tự vào cột 1 từ dòng 12  exSheet.Cells[1][hang + 12] = hang + 1;  for (cot = 0; cot < tblThongtinHang.Columns.Count; cot++)  //Điền thông tin hàng từ cột thứ 2, dòng 12  {  exSheet.Cells[cot + 2][hang + 12] = tblThongtinHang.Rows[hang][cot].ToString();  if (cot == 3) exSheet.Cells[cot + 2][hang + 12] = tblThongtinHang.Rows[hang][cot].ToString();  }  }  exRange = exSheet.Cells[cot][hang + 14];  exRange.Font.Bold = true;  exRange.Value2 = "Tổng tiền:";  exRange = exSheet.Cells[cot + 1][hang + 14];  exRange.Font.Bold = true;  exRange.Value2 = tblThongtinHD.Rows[0][7].ToString();  exRange = exSheet.Cells[6][hang + 17]; //Ô A1  exRange.Range["A1:D1"].MergeCells = true;  exRange.Range["A1:D1"].MergeCells = true;  exRange.Range["A1:D1"].Font.Italic = true;  exRange.Range["A1:D1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  DateTime d = Convert.ToDateTime(tblThongtinHD.Rows[0][1]);  exRange.Range["A1:C1"].Value = "HCM, ngày " + DateTime.Now.Day + " tháng " + DateTime.Now.Month + " năm " + DateTime.Now.Year;  exRange.Range["A2:D2"].MergeCells = true;  exRange.Range["A2:D2"].Font.Italic = true;  exRange.Range["A2:D2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exRange.Range["A2:C2"].Value = "Nhân viên bán hàng";  exRange.Range["A6:C6"].MergeCells = true;  exRange.Range["A6:C6"].Font.Italic = true;  exRange.Range["A6:C6"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;  exApp.Visible = true;  btnBoQua.Enabled = true;  } |

## 5. Danh mục sản phẩm:

### 5.1 Thêm sản phẩm:

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_themSanPhamMoi]  @MaSP nchar(10),  @TenSP nvarchar(50),  @DonGia float,  @TinhTrang nchar(10),  @MaLoaiSP nchar(10),  @TenLoaiSP nvarchar(50)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  -- Kiểm tra xem loại sản phẩm đã tồn tại hay chưa  IF NOT EXISTS (SELECT \* FROM LOAI\_SP WHERE MaLoaiSP =  @MaLoaiSP)  BEGIN  -- Nếu chưa tồn tại, thêm mới loại sản phẩm  INSERT INTO LOAI\_SP (MaLoaiSP, TenLoaiSP)  VALUES (@MaLoaiSP, @TenLoaiSP)  END  -- Thêm mới sản phẩm  INSERT INTO SAN\_PHAM(MaSP, TenSP, DonGia, TinhTrang, MaLoaiSP)  VALUES (@MaSP, @TenSP, @DonGia, @TinhTrang, @MaLoaiSP)  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#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnSua.Enabled = false;  btnXoa.Enabled = false;  btnBoQua.Enabled = true;  btnLuu.Enabled = true;  btnThem.Enabled = false;  ResetValues();  txtMaSP.Enabled = true;  txtMaSP.Focus();  }  private void btnLuu\_Click(object sender, EventArgs e)  {  string sql;  if (txtMaSP.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtMaSP.Focus();  return;  }  if (txtTenSP.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập tên hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtTenSP.Focus();  return;  }  if (txtDonGia.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập giá sản phẩm  {  MessageBox.Show("Bạn phải nhập giá sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtDonGia.Focus();  return;  }  if (txtTinhTrang.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập tình trạng sản phẩm  {  MessageBox.Show("Bạn phải nhập tình trạng sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtTinhTrang.Focus();  return;  }  if (txtMaLoaiSP.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập mã loại sản phẩm  {  MessageBox.Show("Bạn phải nhập mã loại sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtMaLoaiSP.Focus();  return;  }  try  {    SqlCommand cmd = new SqlCommand("proc\_themSanPhamMoi", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text.Trim());  cmd.Parameters.AddWithValue("@TenSP", txtTenSP.Text.Trim());  cmd.Parameters.AddWithValue("@DonGia", txtDonGia.Text.Trim());  cmd.Parameters.AddWithValue("@TinhTrang", txtTinhTrang.Text.Trim());  cmd.Parameters.AddWithValue("@MaLoaiSP", txtMaLoaiSP.Text.Trim());  cmd.Parameters.AddWithValue("@TenLoaiSP", txtTenLoaiSP.Text.Trim());  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Thêm sản phẩm thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  btnSua.Enabled = true;  btnBoQua.Enabled = false;  btnLuu.Enabled = false;  } |

### 5.2. Sửa sản phẩm

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_suaSanPham]  @MaSP nchar(10),  @TenSP nvarchar(50),  @DonGia float,  @TinhTrang nchar(10)  AS  BEGIN  BEGIN TRY  -- Thêm mới sản phẩm  UPDATE dbo.SAN\_PHAM SET MaSP = @MaSP, TenSP = @TenSP, DonGia =  @DonGia,  TinhTrang = @TinhTrang  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#:

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  string sql; //Lưu câu lệnh sql  if (tblsp.Rows.Count == 0)  {  MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (txtMaSP.Text == "") //nếu chưa chọn bản ghi nào  {  MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (txtTenSP.Text.Trim().Length == 0) //nếu chưa nhập tên chất liệu  {  MessageBox.Show("Bạn chưa nhập tên sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (txtDonGia.Text.Trim().Length == 0) //nếu chưa nhập tên chất liệu  {  MessageBox.Show("Bạn chưa nhập tên sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (txtTinhTrang.Text.Trim().Length == 0) //nếu chưa nhập tên chất liệu  {  MessageBox.Show("Bạn chưa nhập tên sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (txtMaLoaiSP.Text.Trim().Length == 0) //nếu chưa nhập tên chất liệu  {  MessageBox.Show("Bạn chưa nhập tên sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  try  {    SqlCommand cmd = new SqlCommand("proc\_suaSanPham", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text.Trim());  cmd.Parameters.AddWithValue("@TenSP", txtTenSP.Text.Trim());  cmd.Parameters.AddWithValue("@DonGia", txtDonGia.Text.Trim());  cmd.Parameters.AddWithValue("@TinhTrang", txtTinhTrang.Text.Trim());  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Sửa sản phẩm thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnBoQua.Enabled = false;  } |

### 5.3.Xóa sản phẩm

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_xoaSAN\_PHAM]  @MaSP nchar(10)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  --Xoá sản phẩm theo @MaSP trong bảng SAN\_PHAM  DELETE FROM dbo.SAN\_PHAM WHERE SAN\_PHAM.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 |

Thực thi trên C#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  string sql;  if (tblsp.Rows.Count == 0)  {  MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (txtMaSP.Text == "") //nếu chưa chọn bản ghi nào  {  MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (MessageBox.Show("Bạn có muốn xoá không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)  {  /\*  \* sql = "DELETE SAN\_PHAM WHERE MaSP=N'" + txtMaSP.Text + "'";  Class.Functions.RunSQL(sql);  LoadDataGridView(); \*/  try  {  Functions.Connect(); // Mở kết nối trước khi thực hiện thủ tục  SqlCommand cmd = new SqlCommand("proc\_xoaSAN\_PHAM", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text.Trim());  cmd.CommandType = CommandType.StoredProcedure;  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("xóa sản phẩm thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  Functions.Disconnect(); // Đảm bảo đóng kết nối sau khi thực hiện thủ tục  }  }  ResetValues();  } |

### 5.4. Tìm kiếm sản phẩm

|  |
| --- |
| --tìm kiếm sản phẩm bằng mã sản phẩm  CREATE PROC [dbo].[pro\_SearchByMaSP]  @MaSP nchar(10)  AS  BEGIN  SELECT \*  FROM view\_danhSachSanPham  WHERE MaSP = @MaSP  END  --tìm kiếm sản phẩm bằng Ten sp  CREATE PROC [dbo].[pro\_SearchByTenSP]  @TenSP nchar(11)  AS  BEGIN  SELECT \*  FROM view\_danhSachSanPham  WHERE TenSP = @TenSP  END  --tìm kiếm sản phẩm bằng mã loại sản phẩm  CREATE PROC [dbo].[pro\_SearchByMaLSP]  @MaLSP nvarchar(50)  AS  BEGIN  SELECT \*  FROM view\_danhSachSanPham  WHERE MaLoaiSP = @MaLSP  END  --tìm kiếm sản phẩm bằng tên loại sản phẩm  CREATE PROC [dbo].[pro\_SearchByTenLSP]  @TenLSP nvarchar(10)  AS  BEGIN  SELECT \*  FROM view\_danhSachSanPham  WHERE TenLoaiSP = @TenLSP  ENDTenLoaiSP = @TenLSP  END |

Thực thi trên C#:

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e)  {  btnBoQua.Enabled = true;  string sql = "";  if ((txtMaSP.Text == "") && (txtTenSP.Text == "") && (txtMaLoaiSP.Text == "") && (txtTenLoaiSP.Text == ""))  {  MessageBox.Show("Bạn hãy nhập điều kiện tìm kiếm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  // Xóa dữ liệu hiển thị trên DataGridView trước khi tải dữ liệu mới  dgvSanPham.DataSource = null;  if (txtMaSP.Text != "")  {  sql = "EXEC pro\_SearchByMaSP @MaSP='" + txtMaSP.Text + "'";  }  else if (txtTenSP.Text != "")  {  sql = "EXEC pro\_SearchByTenSP @TenSP=N'" + txtTenSP.Text + "'";  }  else if (txtMaLoaiSP.Text != "")  {  sql = "EXEC pro\_SearchByMaLSP @MaLSP='" + txtMaLoaiSP.Text + "'";  }  else if (txtTenLoaiSP.Text != "")  {  sql = "EXEC pro\_SearchByTenLSP @TenLSP='" + txtTenLoaiSP.Text + "'";  }  else  {  MessageBox.Show("Không có điều kiện tìm kiếm phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  // Lấy dữ liệu từ CSDL  tblsp = Functions.GetDataToTable(sql);  // Hiển thị dữ liệu lên DataGridView  dgvSanPham.DataSource = tblsp;  // Kiểm tra số bản ghi trả về  if (tblsp.Rows.Count == 0)  {  MessageBox.Show("Không có bản ghi nào phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Có " + tblsp.Rows.Count + " bản ghi phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  // Đặt lại các giá trị và trạng thái  ResetValues();  } |

### 5.5. Trigger bắt lỗi trùng tên khi thêm và chỉnh sửa sản phẩm

|  |
| --- |
| CREATE TRIGGER TG\_TrungTenSP  ON dbo.SAN\_PHAM  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.SAN\_PHAM 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 |

## 6. Thống kê doanh thu và chi phí

### 6.1. Doanh thu theo ngày, tháng, năm

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_tinhDoanhThuNgay](@ngay INT, @thang INT, @nam INT)  RETURNS FLOAT  AS  BEGIN  DECLARE @doanhThu FLOAT = 0;  SELECT @doanhThu = COALESCE(SUM(TriGiaHD), 0)  FROM HOA\_DON  WHERE DAY(NgayDatHang) = @ngay AND MONTH(NgayDatHang) = @thang AND  YEAR(NgayDatHang) = @nam;  RETURN @doanhThu;  END;  CREATE FUNCTION [dbo].[func\_tinhDoanhThuThang](@thang INT, @nam INT) RETURNS  float  BEGIN  DECLARE @doanhThu float = 0;  SELECT @doanhthu = COALESCE(SUM(TriGiaHD), 0)  FROM HOA\_DON  WHERE MONTH(NgayDatHang) = @thang AND YEAR(NgayDatHang) = @nam;  RETURN @doanhThu;  END;  CREATE FUNCTION [dbo].[func\_tinhDoanhThuNam](@nam INT) RETURNS float  BEGIN  DECLARE @doanhThu float = 0;  SELECT @doanhthu = COALESCE(SUM(TriGiaHD), 0)  FROM HOA\_DON  WHERE YEAR(NgayDatHang) = @nam;  RETURN @doanhThu;  END; |

## 7. Quản lý nhân viên

### 7.1. Thêm nhân viên

|  |
| --- |
| CREATE PROC [dbo].[proc\_AddEmployee]  @manv nchar(10), @ho nvarchar(10), @ten nvarchar(10), @ns date, @gt nvarchar(3),  @dc nvarchar(100),  @sdt nchar(11), @macv nchar(10), @soca int, @thuong int, @ntd date  AS  INSERT INTO NHAN\_VIEN(MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT,  MaCV, SoCa, Thuong, NgayTuyenDung)  VALUES(@manv, @ho, @ten, @ns, @gt, @dc, @sdt, @macv, @soca, @thuong, @ntd) |

Thực thi trên C#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnSua.Enabled = false;  btnXoa.Enabled = false;  btnBoQua.Enabled = true;  btnLuu.Enabled = true;  btnThem.Enabled = false;  ResetValues();  txtMaNhanVien.Enabled = true;  txtMaNhanVien.Focus();  }  private void btnLuu\_Click(object sender, EventArgs e)  {  string sql, gt;  if (txtMaNhanVien.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtMaNhanVien.Focus();  return;  }  if (txtHoNhanVien.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập Ho nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtHoNhanVien.Focus();  return;  }  if (txtTenNhanVien.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập tên nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtTenNhanVien.Focus();  return;  }  if (txtDiaChi.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtDiaChi.Focus();  return;  }  if (txtMaCV.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã công việc", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtMaCV.Focus();  return;  }  if (txtSoCa.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập số ca", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtSoCa.Focus();  return;  }  if (txtThuong.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập Thưởng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtThuong.Focus();  return;  }  if (txtSDT.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  txtSDT.Focus();  return;  }  gt = (chkGioiTinh.Checked) ? "Nam" : "Nữ";  try  {  SqlCommand cmd = new SqlCommand("proc\_AddEmployee", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@manv", txtMaNhanVien.Text.Trim());  cmd.Parameters.AddWithValue("@ho", txtHoNhanVien.Text.Trim());  cmd.Parameters.AddWithValue("@ten", txtTenNhanVien.Text.Trim());  cmd.Parameters.AddWithValue("@ns", mskNgaySinh.Value);  cmd.Parameters.AddWithValue("@gt", gt);  cmd.Parameters.AddWithValue("@dc", txtDiaChi.Text.Trim());  cmd.Parameters.AddWithValue("@sdt", txtSDT.Text.Trim());  cmd.Parameters.AddWithValue("@macv", txtMaCV.Text.Trim());  cmd.Parameters.AddWithValue("@soca", txtSoCa.Text.Trim());  cmd.Parameters.AddWithValue("@thuong", txtThuong.Text.Trim());  cmd.Parameters.AddWithValue("@ntd", mskNgaySinhTuyenDung.Value);  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Thêm nhân viên thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  btnSua.Enabled = true;  btnBoQua.Enabled = false;  btnLuu.Enabled = false;  } |

### 7.2. Xóa nhân viên

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_DeleteEmployee]  @MaNV nchar(10)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRANSACTION;  BEGIN TRY  -- Xóa dòng trong bảng OrderDetails có OrderID tương ứng  DELETE FROM BangPhanCa WHERE MaNV = @MaNV;  -- Xóa dòng trong bảng Orders có OrderID tương ứng  CREATE PROCEDURE [dbo].[proc\_DeleteEmployee]  @MaNV nchar(10)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRANSACTION;  BEGIN TRY  -- Xóa dòng trong bảng OrderDetails có OrderID tương ứng  DELETE FROM BangPhanCa WHERE MaNV = @MaNV;  -- Xóa dòng trong bảng Orders có OrderID tương ứng  DELETE FROM NHAN\_VIEN WHERE MaNV = @MaNV;  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#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  string manv = txtMaNhanVien.Text.Trim();  if (tblNV.Rows.Count == 0)  {  MessageBox.Show("Không còn dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (manv == "")  {  MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  try  {  SqlCommand cmd = new SqlCommand("proc\_DeleteEmployee", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số vào SqlCommand  cmd.Parameters.AddWithValue("@MaNV", manv);  // Thực thi câu lệnh SQL  cmd.ExecuteNonQuery();  MessageBox.Show("Xóa nhân viên thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  ResetValues();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 7.3. Tìm kiếm nhân viên

|  |
| --- |
| --tìm kiếm nhân viên bằng mã nhân viên  CREATE PROC [dbo].[pro\_SearchByMaNV]  @MaNV nchar(10)  AS  BEGIN  SELECT \*  FROM v\_LoadEmployee  WHERE MaNV = @MaNV  END  --tìm kiếm nhân viên bằng số điện thoại  CREATE PROC [dbo].[pro\_SearchBySDT]  @SDT nchar(11)  AS  BEGIN  SELECT \*  FROM v\_LoadEmployee  WHERE SDT = @SDT  END  --tìm kiếm nhân viên bằng tên công việc  CREATE PROC [dbo].[pro\_SearchByTenCV]  @TenCV nvarchar(50)  AS  BEGIN  SELECT \*  FROM v\_LoadEmployee  WHERE TenCV = @TenCV  END  --tìm kiếm nhân viên bằng tên nhân viên  CREATE PROC [dbo].[pro\_SearchByTenNV]  @TenNV nvarchar(10)  AS  BEGIN  SELECT \*  FROM v\_LoadEmployee  WHERE TenNV = @TenNV  END |

Thực thi trên C#:

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e)  {  btnBoQua.Enabled = true;  string sql = "";  if ((txtMaNhanVien.Text == "") && (txtTenNhanVien.Text == "") && (txtSDT.Text == "")&&(txtCongViec.Text ==""))  {  MessageBox.Show("Bạn hãy nhập điều kiện tìm kiếm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  // Xóa dữ liệu hiển thị trên DataGridView trước khi tải dữ liệu mới  dgvNhanVien.DataSource = null;  if (txtMaNhanVien.Text != "")  {  sql = "EXEC pro\_SearchByMaNV @MaNV='" + txtMaNhanVien.Text + "'";  }  else if (txtTenNhanVien.Text != "")  {  sql = "EXEC pro\_SearchByTenNV @TenNV=N'" + txtTenNhanVien.Text + "'";  }  else if (txtSDT.Text != "")  {  sql = "EXEC pro\_SearchBySDT @SDT='" + txtSDT.Text + "'";  }  else if (txtCongViec.Text != "")  {  sql = "EXEC pro\_SearchByTenCV @TenCV='" + txtCongViec.Text + "'";  }  else  {  MessageBox.Show("Không có điều kiện tìm kiếm phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  // Lấy dữ liệu từ CSDL  tblNV = Functions.GetDataToTable(sql);  // Hiển thị dữ liệu lên DataGridView  dgvNhanVien.DataSource = tblNV;  // Kiểm tra số bản ghi trả về  if (tblNV.Rows.Count == 0)  {  MessageBox.Show("Không có bản ghi nào phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Có " + tblNV.Rows.Count + " bản ghi phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  // Đặt lại các giá trị và trạng thái  ResetValues();  } |

### 7.4. Sửa thông tin nhân viên:

|  |
| --- |
| CREATE PROC [dbo].[proc\_EditEmployee]  @manv nchar(10), @honv nvarchar(10), @tennv nvarchar(10), @ngaysinh date,  @gioitinh nvarchar(3), @diachi nvarchar(100), @sdt nchar(11), @macv  nchar(10),  @SoCa int, @Thuong int, @NgayTuyenDung date  AS  BEGIN  BEGIN TRY  UPDATE NHAN\_VIEN  SET  HoNV = @honv, TenNV = @tennv, NgaySinh = @ngaysinh,  GioiTinh = @gioitinh, DiaChi = @diachi, SDT = @sdt, MaCV = @macv,  SoCa = @soca, Thuong = @thuong, NgayTuyenDung = @ngaytuyendung  WHERE MaNV = @manv  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#:

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  string manv = txtMaNhanVien.Text.Trim();  string honv = txtHoNhanVien.Text.Trim();  string tennv = txtTenNhanVien.Text.Trim();  DateTime ngaysinh = mskNgaySinh.Value;  string gioitinh = chkGioiTinh.Checked ? "Nam" : "Nữ";  string diachi = txtDiaChi.Text.Trim();  string sdt = txtSDT.Text.Trim();  string macv = txtMaCV.Text.Trim();  int soca, thuong;  DateTime ngaytuyendung;  if (!int.TryParse(txtSoCa.Text.Trim(), out soca))  {  MessageBox.Show("Số ca không hợp lệ!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  if (!int.TryParse(txtThuong.Text.Trim(), out thuong))  {  MessageBox.Show("Số tiền thưởng không hợp lệ!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  ngaytuyendung = mskNgaySinhTuyenDung.Value;  try  {  SqlCommand cmd = new SqlCommand("proc\_EditEmployee", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@manv", manv);  cmd.Parameters.AddWithValue("@honv", honv);  cmd.Parameters.AddWithValue("@tennv", tennv);  cmd.Parameters.AddWithValue("@ngaysinh", ngaysinh);  cmd.Parameters.AddWithValue("@gioitinh", gioitinh);  cmd.Parameters.AddWithValue("@diachi", diachi);  cmd.Parameters.AddWithValue("@sdt", sdt);  cmd.Parameters.AddWithValue("@macv", macv);  cmd.Parameters.AddWithValue("@soca", soca);  cmd.Parameters.AddWithValue("@thuong", thuong);  cmd.Parameters.AddWithValue("@ngaytuyendung", ngaytuyendung);  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Sửa thông tin nhân viên thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnBoQua.Enabled = false;  } |

### 7.5. Tính lương nhân viên

|  |
| --- |
| --tính lương nhân viên cứng  CREATE FUNCTION func\_tinhLuongCung(@thang INT, @nam INT) RETURNS TABLE  AS RETURN (  SELECT  nv.MaNV,  nv.HoNV,  nv.TenNV,  'Lương cơ bản' = CASE  WHEN cv.MaCV = 'CV01' THEN 10000000  WHEN cv.MaCV = 'CV02' THEN 7000000  ELSE 0  END,  'Tiền thưởng' = CASE  WHEN cv.MaCV = 'CV01' THEN (dbo.func\_tinhDoanhThuThang(@thang, @nam) \* 0.03 + 2 \* 333333)  WHEN cv.MaCV = 'CV02' THEN 2 \* 233333  ELSE 0  END,  'Lương tổng' = CASE  WHEN cv.MaCV = 'CV01' THEN 10000000 + (dbo.func\_tinhDoanhThuThang(@thang, @nam) \* 0.03 + 2 \* 333333)  WHEN cv.MaCV = 'CV02' THEN 7000000 + 2 \* 233333  ELSE 0  END  FROM  NHAN\_VIEN nv  INNER JOIN  CONG\_VIEC cv ON nv.MaCV = cv.MaCV  )  -- tính lương nhân viên làm theo ca  CREATE FUNCTION func\_tinhLuongCa(@thang INT, @nam INT) RETURNS TABLE  AS RETURN (  SELECT  nv.MaNV,  nv.HoNV,  nv.TenNV,  'Lương cơ bản' = nv.SoCa \* 4 \* CASE  WHEN cv.MaCV = 'CV03' THEN 35000  WHEN cv.MaCV = 'CV04' THEN 30000  ELSE 0  END,  'Tiền thưởng' = CASE  WHEN cv.MaCV = 'CV03' THEN (dbo.func\_tinhDoanhThuThang(@thang, @nam) \* 0.01 + nv.SoCa\*35000\*4)  WHEN cv.MaCV = 'CV03' THEN (dbo.func\_tinhDoanhThuThang(@thang, @nam) \* 0.01 + nv.SoCa\*30000\*4)  ELSE 0  END,  'Lương tổng' = (nv.SoCa \* 4 \* CASE  WHEN cv.MaCV = 'CV03' THEN 35000  WHEN cv.MaCV = 'CV04' THEN 30000  ELSE 0  END) + (dbo.func\_tinhDoanhThuThang(@thang, @nam) \* 0.01 + ( nv.SoCa \* CASE  WHEN cv.MaCV = 'CV03' THEN 35000  WHEN cv.MaCV = 'CV04' THEN 30000  ELSE 0  END))  FROM  NHAN\_VIEN nv  INNER JOIN  CONG\_VIEC cv ON nv.MaCV = cv.MaCV  ) |

Thực thi trên C#:

|  |
| --- |
| private void btnTinh\_Click(object sender, EventArgs e)  {  // Kiểm tra xem người dùng đã chọn tháng và năm chưa  if (string.IsNullOrEmpty(txtThang.Text) || string.IsNullOrEmpty(txtNam.Text))  {  MessageBox.Show("Vui lòng nhập tháng và năm để tính lương!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  int thang, nam;  // Kiểm tra xem dữ liệu nhập vào có đúng định dạng số hay không  if (!int.TryParse(txtThang.Text, out thang) || !int.TryParse(txtNam.Text, out nam))  {  MessageBox.Show("Tháng và năm phải là số!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  // Lấy dữ liệu lương từ hàm func\_tinhLuongNV trong CSDL  string sql = "SELECT \* FROM func\_tinhLuongCung(" + thang + ", " + nam + ")";  DataTable result = Functions.GetDataToTable(sql);  // Hiển thị dữ liệu lương trên DataGridView hoặc control tương tự  dgvLuongNhanVien.DataSource = result;  // Thông báo khi hoàn tất tính toán  MessageBox.Show("Đã tính toán lương cứng cho tháng " + thang + "/" + nam, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  private void btnTinhLuongCa\_Click(object sender, EventArgs e)  {  // Kiểm tra xem người dùng đã chọn tháng và năm chưa  if (string.IsNullOrEmpty(txtThang.Text) || string.IsNullOrEmpty(txtNam.Text))  {  MessageBox.Show("Vui lòng nhập tháng và năm để tính lương!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  int thang, nam;  // Kiểm tra xem dữ liệu nhập vào có đúng định dạng số hay không  if (!int.TryParse(txtThang.Text, out thang) || !int.TryParse(txtNam.Text, out nam))  {  MessageBox.Show("Tháng và năm phải là số!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  // Lấy dữ liệu lương từ hàm func\_tinhLuongNV trong CSDL  string sql = "SELECT \* FROM func\_tinhLuongCa(" + thang + ", " + nam + ")";  DataTable result = Functions.GetDataToTable(sql);  // Hiển thị dữ liệu lương trên DataGridView hoặc control tương tự  dgvLuongNhanVien.DataSource = result;  // Thông báo khi hoàn tất tính toán  MessageBox.Show("Đã tính toán lương theo ca cho tháng " + thang + "/" + nam, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 7.6. Trigger bắt lỗi trùng số điện thoại khi thêm nhân viên:

|  |
| --- |
| CREATE TRIGGER TG\_TrungSDTNhanVien  ON dbo.NHAN\_VIEN  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.NHAN\_VIEN nv  WHERE nv.SDT = i.SDT AND nv.MaNV <> i.MaNV  )  )  BEGIN  -- Nếu trùng thì rollback  RAISERROR ('SDT nhân viên bị trùng', 16, 1)  ROLLBACK;  END  END |

## 8. Quản lý phiếu chi

### 8.1 Thêm phiếu chi

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_InsertDonNhapHangAndPhieuChiAndChiTietNhapHang]  @MaDonNH nchar(10),  @NgayNH date,  @TriGiaDonNH float,  @MaNCC nchar(10),  @TinhTrang nvarchar(50),  @MaPhieuChi nchar(10),  @NgayXuatPhieu date,  @SoTienChi float,  @DonGia float,  @SL int,  @DonVi nchar(10),  @TongTien float,  @MaSP nchar(10)  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  -- Insert into DonNhapHang table  INSERT INTO DON\_NHAP(MaDonNH, NgayNH, TriGiaDonNH, MaNCC, TinhTrang)  VALUES (@MaDonNH, @NgayNH, @TriGiaDonNH, @MaNCC, @TinhTrang)  -- Insert into PhieuChi table  INSERT INTO PHIEU\_CHI(MaPhieuChi, MaDonNH, NgayXuatPhieu, SoTienChi)  VALUES (@MaPhieuChi, @MaDonNH, @NgayXuatPhieu, @SoTienChi)  -- Insert into ChiTietNhapHang table  INSERT INTO ChiTietDonNhap(MaSP, MaDonNH, DonGia, SL, DonVi, TongTien)  VALUES (@MaSP, @MaDonNH, @DonGia, @SL, @DonVi, @TongTien)  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  THROW;  END CATCH;  COMMIT TRANSACTION;  END |

Thực thi trên C#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnXuat.Enabled = true;  btnThem.Enabled = false;  ResetValues();  txtMadonNH.Enabled = true;  txtMadonNH.Focus();  }  private void btnXuat\_Click(object sender, EventArgs e)  {  string sql;  if (txtMadonNH.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã nhập hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtMadonNH.Focus();  return;  }  if (txtDonGia.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập giá sản phẩm  {  MessageBox.Show("Bạn phải nhập đơn giá ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtDonGia.Focus();  return;  }  if (txtSoLuong.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập tình trạng sản phẩm  {  MessageBox.Show("Bạn phải nhập số lượng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtSoLuong.Focus();  return;  }  if (txtDV.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập đơn vị", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtDV.Focus();  return;  }  if (txtNCC.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập nhà cung cấp", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtNCC.Focus();  return;  }  if (txtPC.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập phiếu chi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtPC.Focus();  return;  }  try  {  SqlCommand cmd = new SqlCommand("proc\_InsertDonNhapHangAndPhieuChiAndChiTietNhapHang", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  float TongTien = float.Parse(txtDonGia.Text) \* float.Parse(txtSoLuong.Text);  // Thêm các tham số vào Sqlcmd  cmd.Parameters.AddWithValue("@MaDonNH", txtMadonNH.Text);  cmd.Parameters.AddWithValue("@NgayNH", DateTime.Today);  cmd.Parameters.AddWithValue("@TriGiaDonNH", TongTien);  cmd.Parameters.AddWithValue("@MaNCC", txtNCC.Text);  cmd.Parameters.AddWithValue("@TinhTrang", "Đang giao");  // Add parameters for the PhieuChi table  cmd.Parameters.AddWithValue("@MaPhieuChi", txtPC.Text);  cmd.Parameters.AddWithValue("@NgayXuatPhieu", DateTime.Today);  cmd.Parameters.AddWithValue("@SoTienChi", TongTien);  // Add parameters for the ChiTietNhapHang table  cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text);  cmd.Parameters.AddWithValue("@DonGia", float.Parse(txtDonGia.Text));  cmd.Parameters.AddWithValue("@SL", int.Parse(txtSoLuong.Text));  cmd.Parameters.AddWithValue("@DonVi", txtDV.Text);  cmd.Parameters.AddWithValue("@TongTien", TongTien);  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Thêm phiếu chi thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnThem.Enabled = true;  btnXuat.Enabled = false;  } |

### 8.2 Xóa phiếu chi

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_XoaDonNhapHang]  @MaDonNH nchar(10)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRANSACTION;  BEGIN TRY  -- Xóa dòng trong bảng OrderDetails có OrderID tương ứng  DELETE FROM PHIEU\_CHI WHERE MaDonNH = @MaDonNH;  DELETE FROM ChiTietDonNhap WHERE MaDonNH = @MaDonNH;  -- Xóa dòng trong bảng Orders có OrderID tương ứng  DELETE FROM DON\_NHAP WHERE MaDonNH = @MaDonNH;  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#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {    SqlCommand cmd = new SqlCommand("proc\_XoaDonNhapHang", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@MaDonNH", txtMadonNH.Text.Trim());  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Xóa đơn nhập hàng thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  btnXuat.Enabled = false;  } |

## 9. Quản lý phân ca

### 9.1. Thêm ca cho nhân viên

|  |
| --- |
| CREATE PROC [dbo].[proc\_AddPhanCa]  @maca nchar(10),  @manv nchar(10),  @ngaylam date  AS  BEGIN  BEGIN TRY  INSERT INTO BangPhanCa  VALUES(@maca, @manv, @ngaylam)  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#:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  btnSua.Enabled = false;  btnXoa.Enabled = false;  btnBoQua.Enabled = true;  btnLuu.Enabled = true;  btnThem.Enabled = false;  ResetValues();  txtMaCa.Enabled = true;  txtMaCa.Focus();  }  private void btnLuu\_Click(object sender, EventArgs e)  {  string sql;  if (txtMaCa.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập mã ca", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtMaCa.Focus();  return;  }  if (txtBatDau.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập giá sản phẩm  {  MessageBox.Show("Bạn phải mã nhân viên ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtBatDau.Focus();  return;  }  if (txtKetThuc.Text.Trim().Length == 0) // Kiểm tra nếu chưa nhập tình trạng sản phẩm  {  MessageBox.Show("Bạn phải nhập họ nhân viên ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtKetThuc.Focus();  return;  }  if (txtMaNV.Text.Trim().Length == 0)  {  MessageBox.Show("Bạn phải nhập tên nhân viên ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  txtMaNV.Focus();  return;  }  try  {    SqlCommand cmd = new SqlCommand("proc\_AddPhanCa", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.Add("@maca", SqlDbType.NChar, 10).Value = txtMaCa.Text;  cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = txtMaNV.Text;  cmd.Parameters.Add("@ngaylam", SqlDbType.NChar, 10).Value = dtpNgayLam.Text;  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Thêm ca thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  btnSua.Enabled = true;  btnBoQua.Enabled = false;  btnLuu.Enabled = false;  } |

### 9.2. Xóa ca làm việc

|  |
| --- |
| CREATE PROC [dbo].[proc\_XoaPhanCa]  @manv nchar(10),  @maca nchar(10),  @ngaylam date  AS  BEGIN  BEGIN TRY  DELETE FROM BangPhanCa  WHERE MaNV = @manv and  MaCa = @maca and  NgayLam = @ngaylam  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#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {    SqlCommand cmd = new SqlCommand("proc\_XoaPhanCa", Functions.Con);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@maca", txtMaCa.Text.Trim());  cmd.Parameters.AddWithValue("@manv", txtMaNV.Text.Trim());  cmd.Parameters.AddWithValue("@ngaylam", dtpNgayLam.Value);  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Xóa ca thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;  btnThem.Enabled = true;  btnSua.Enabled = true;  btnBoQua.Enabled = false;  btnLuu.Enabled = false;  } |

# CHƯƠNG 4 : Tạo User và phân quyền

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

## + 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 ChiTietHoaDon TO Staff  GRANT SELECT, REFERENCES ON CHITIETHD\_UD TO Staff  GRANT SELECT, REFERENCES ON ChiTietDonNhap TO Staff  GRANT SELECT, REFERENCES ON CONG\_VIEC TO Staff  GRANT SELECT, INSERT, REFERENCES ON DON\_NHAP TO Staff  GRANT SELECT, INSERT, REFERENCES ON HOA\_DON TO Staff  GRANT SELECT, INSERT, REFERENCES ON HOA\_DON\_UD TO Staff  GRANT SELECT, INSERT, REFERENCES ON KHACH\_HANG TO Staff  GRANT SELECT, REFERENCES ON Loai\_SP TO Staff  GRANT SELECT, REFERENCES ON NHA\_CUNG\_CAP TO Staff  GRANT SELECT, REFERENCES ON NHAN\_VIEN TO Staff  GRANT SELECT, INSERT, REFERENCES ON PHIEU\_CHI TO Staff  GRANT SELECT, REFERENCES ON SAN\_PHAM TO Staff  GRANT SELECT, REFERENCES ON UNG\_DUNG 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\_AddEmployee to Staff;  DENY EXECUTE ON proc\_DeleteEmployee to Staff;  DENY EXECUTE ON proc\_EditEmployee to Staff;  DENY EXECUTE ON proc\_suaSanPham to Staff;  DENY EXECUTE ON proc\_xoaSAN\_PHAM to Staff;  DENY EXECUTE ON proc\_xoaPhanCa to Staff; |

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

- Bảng DANGNHAP để lưu trữ tài khoản:

|  |
| --- |
| CREATE TABLE DANGNHAP(  MaNV nchar(10) CONSTRAINT FK\_NhanVien\_DangNhap FOREIGN KEY REFERENCES NHAN\_VIEN,  TenDangNhap nvarchar(30) NOT NULL,  MatKhau nvarchar(10) NOT NULL,  ) |

- Trigger tạo tài khoản:

|  |
| --- |
| CREATE TRIGGER [dbo].[Trigger\_CreateSQLAccount] ON [dbo].[DANGNHAP]  AFTER INSERT  AS  DECLARE @userName nvarchar(30), @passWord nvarchar(10), @manv nvarchar(10)  SELECT @userName=nl.TenDangNhap, @passWord=nl.MatKhau, @manv=nl.maNV  FROM inserted nl  BEGIN  DECLARE @sqlString nvarchar(2000), @macv nvarchar(10)  ----  SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+  @passWord  +''', DEFAULT\_DATABASE=[chqa], CHECK\_EXPIRATION=OFF,  CHECK\_POLICY=OFF'  EXEC (@sqlString)  ----  SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName  EXEC (@sqlString)  ----  SELECT @manv = MaNV  FROM NHAN\_VIEN  WHERE MaNV = @manv  if (@manv = 'NV01')  SET @sqlString = 'ALTER SERVER ROLE sysadmin' + ' ADD MEMBER '  + @userName;  else  SET @sqlString = 'ALTER ROLE Staff ADD MEMBER ' + @userName;  EXEC (@sqlString)  END |

- Procedure kiểm tra tài khoản đăng nhập:

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_CheckLogin]  @username NVARCHAR(MAX),  @password NVARCHAR(MAX),  @result BIT OUTPUT,  @role NVARCHAR(50) OUTPUT  AS  BEGIN  SET NOCOUNT ON;  SELECT @result = CAST(COUNT(\*) AS BIT)  FROM DANGNHAP  WHERE TenDangNhap = @username AND MatKhau = @password;  -- Kiểm tra và gán giá trị cho biến @role  IF @result = 1  SET @role = CASE  WHEN EXISTS (SELECT TOP 1 1 FROM DANGNHAP WHERE TenDangNhap = @username AND MatKhau = @password AND MaNV = 'NV01') THEN 'Quanly'  ELSE 'Staff'  END;  ELSE  SET @role = NULL; -- Hoặc bất kỳ giá trị mặc định nào bạn muốn đặt  RETURN @result;  END; |

Thực thi trong C#:

|  |
| --- |
| private void btnDN\_Click(object sender, EventArgs e)  {  string username = txtTK.Text.Trim();  string password = txtMK.Text.Trim();  var authResult = AuthenticateUser(username, password);  if (authResult.Item1)  {  MessageBox.Show("Đăng nhập thành công!");  string role = authResult.Item2;  Class.Functions.Connect(username, password);  // Open the main form with appropriate permissions based on the role  fMain mainForm = new fMain(role);  mainForm.Show();  this.Hide();  }  else  {  MessageBox.Show("Đăng nhập không thành công. Vui lòng kiểm tra lại tên đăng nhập và mật khẩu.");  }  }  private Tuple<bool, string> AuthenticateUser(string username, string password)  {  using (SqlCommand cmd = new SqlCommand("dbo.proc\_CheckLogin", Functions.ConAdmin))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@username", username);  cmd.Parameters.AddWithValue("@password", password);  SqlParameter outputParamResult = new SqlParameter("@result", SqlDbType.Bit);  outputParamResult.Direction = ParameterDirection.Output;  cmd.Parameters.Add(outputParamResult);  SqlParameter outputParamRole = new SqlParameter("@role", SqlDbType.NVarChar, 50);  outputParamRole.Direction = ParameterDirection.Output;  cmd.Parameters.Add(outputParamRole);  cmd.ExecuteNonQuery();  bool result = (bool)outputParamResult.Value;  string role = outputParamRole.Value.ToString();  return Tuple.Create(result, role);  }  } |

- Procedure xóa tài khoản:

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_XoaDANGNHAP]  @TenDangNhap nvarchar(30)  AS  BEGIN  SET NOCOUNT ON;  DECLARE @username varchar(15);  SELECT @username=TenDangNhap FROM DANGNHAP WHERE TenDangNhap = @TenDangNhap  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  -- Xóa dòng trong bảng OrderDetails có OrderID tương ứng  --  SET @sql = 'DROP USER '+ @username  exec (@sql)  --  SET @sql = 'DROP LOGIN '+ @username  exec (@sql)  --  DELETE FROM DANGNHAP WHERE TenDangNhap = @TenDangNhap;  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 trong C#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {  SqlCommand cmd = new SqlCommand("proc\_XoaDANGNHAP", Functions.ConAdmin);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào SqlCommand  cmd.Parameters.AddWithValue("@TenDangNhap", txtTenDangNhap.Text.Trim());  cmd.ExecuteNonQuery(); // Thực thi câu lệnh SQL  MessageBox.Show("Xóa tài khoản thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDataGridView();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  ResetValues();  btnXoa.Enabled = true;    } |

- Procedure thêm tài khoản:

|  |
| --- |
| CREATE procedure [dbo].[proc\_ThemTaiKhoan] @tk nvarchar(30), @mk NVARCHAR(10), @manv  NVARCHAR(10)  As  BEGIN  INSERT INTO DANGNHAP(MaNV, TenDangNhap, MatKhau) VALUES (@manv, @tk, @mk)  END |

Thực thi trong C#:

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("proc\_ThemTaiKhoan", Functions.ConAdmin);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@manv", txtMaNV.Text);  cmd.Parameters.AddWithValue("@tk", txtTK.Text);  cmd.Parameters.AddWithValue("@mk", txtMK.Text);  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Tạo tài khoản thành công");    }  else  {  MessageBox.Show("Tạo tài khoản thất bại, vui lòng nhập lại");    }  } |

# CHƯƠNG 5 : THIẾT KẾ GIAO DIỆN HỆ THỐNG

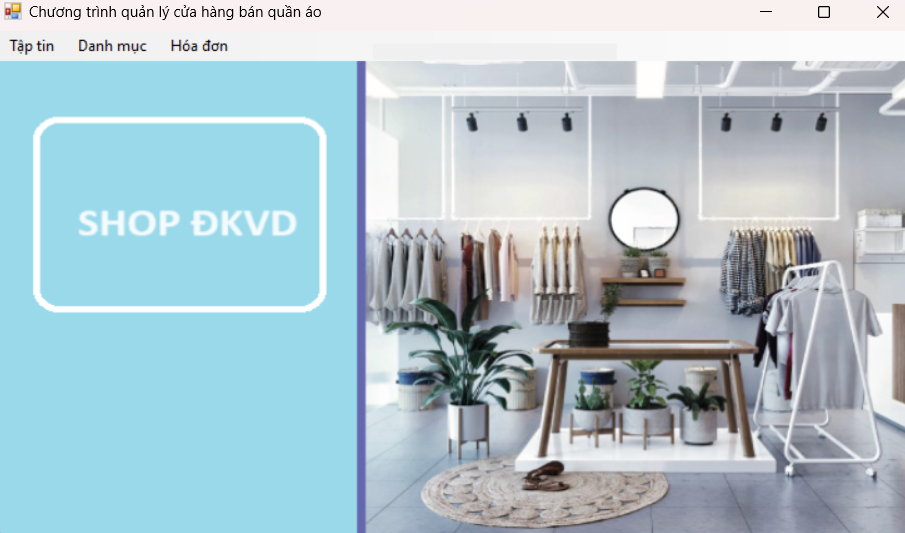
Môi trường cài đặt và công nghệ sử dụng

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

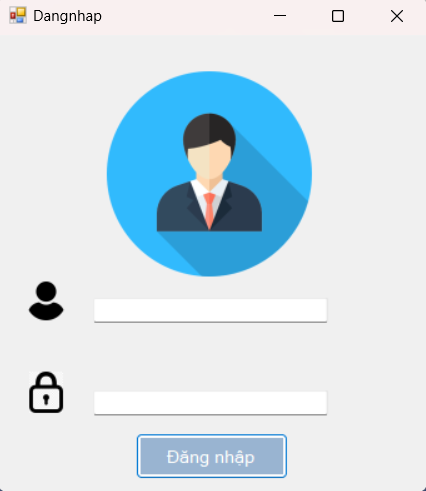
- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 trên nền tảng ASP.NET Web Application (.NET Framework) C#. Giao diện và chức năng của chương trình: 1. Form Home

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

## Form Home

****

## Form đăng nhập



- Kiểm tra đăng nhập:

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_CheckLogin]  @username NVARCHAR(MAX),  @password NVARCHAR(MAX),  @result BIT OUTPUT,  @role NVARCHAR(50) OUTPUT  AS  BEGIN  SET NOCOUNT ON;  SELECT @result = CAST(COUNT(\*) AS BIT)  FROM DANGNHAP  WHERE TenDangNhap = @username AND MatKhau = @password;  -- Kiểm tra và gán giá trị cho biến @role  IF @result = 1  SET @role = CASE  WHEN EXISTS (SELECT TOP 1 1 FROM DANGNHAP WHERE TenDangNhap = @username AND MatKhau = @password AND MaNV = 'NV01') THEN 'Quanly'  ELSE 'Staff'  END;  ELSE  SET @role = NULL; -- Hoặc bất kỳ giá trị mặc định nào bạn muốn đặt  RETURN @result;  END; |

Code C#:

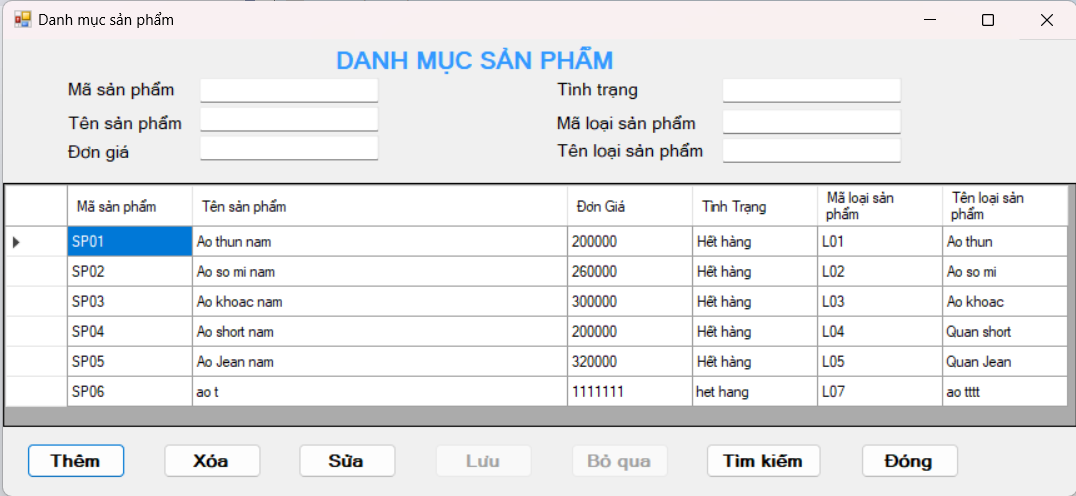
|  |
| --- |
| private void btnDN\_Click(object sender, EventArgs e)  {  string username = txtTK.Text.Trim();  string password = txtMK.Text.Trim();  var authResult = AuthenticateUser(username, password);  if (authResult.Item1)  {  MessageBox.Show("Đăng nhập thành công!");  string role = authResult.Item2;  Class.Functions.Connect(username, password);  // Open the main form with appropriate permissions based on the role  fMain mainForm = new fMain(role);  mainForm.Show();  this.Hide();  }  else  {  MessageBox.Show("Đăng nhập không thành công. Vui lòng kiểm tra lại tên đăng nhập và mật khẩu.");  }  } |

Code tạo bảng DANGNHAP:

|  |
| --- |
| CREATE TABLE DANGNHAP(  MaNV nchar(10) CONSTRAINT FK\_NhanVien\_DangNhap FOREIGN KEY REFERENCES NHAN\_VIEN,  TenDangNhap nvarchar(30) NOT NULL,  MatKhau nvarchar(10) NOT NULL,  ) |

## Các Form Danh mục

### Danh mục sản phẩm



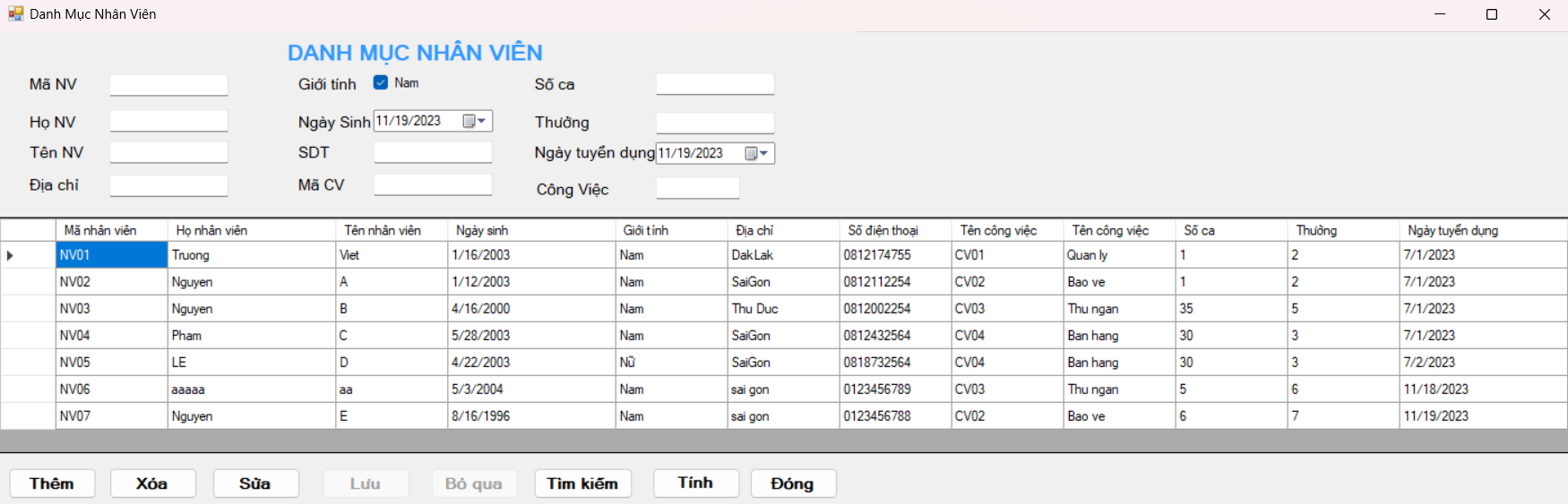
Code C# gọi danh mục:

|  |
| --- |
| private void fSanPham\_Load(object sender, EventArgs e)  {  //txtMaSP.Enabled = false;  btnLuu.Enabled = false;  btnBoQua.Enabled = false;  LoadDataGridView();  }  private void LoadDataGridView()  {  string sql;  sql = " SELECT \* FROM view\_danhSachSanPham";  tblsp = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  dgvSanPham.DataSource = tblsp; //Nguồn dữ liệu  dgvSanPham.Columns[0].HeaderText = "Mã sản phẩm";  dgvSanPham.Columns[1].HeaderText = "Tên sản phẩm";  dgvSanPham.Columns[2].HeaderText = "Đơn Giá";  dgvSanPham.Columns[3].HeaderText = "Tình Trạng";  dgvSanPham.Columns[4].HeaderText = "Mã loại sản phẩm";  dgvSanPham.Columns[5].HeaderText = "Tên loại sản phẩm";  dgvSanPham.Columns[0].Width = 100;  dgvSanPham.Columns[1].Width = 300;  dgvSanPham.Columns[2].Width = 100;  dgvSanPham.Columns[3].Width = 100;  dgvSanPham.Columns[4].Width = 100;  dgvSanPham.Columns[5].Width = 100;  dgvSanPham.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp  dgvSanPham.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW [dbo].[view\_danhSachSanPham]  AS  SELECT MaSP, TenSP, DonGia, TinhTrang, SAN\_PHAM.MaLoaiSP, TenLoaiSP  FROM dbo.SAN\_PHAM INNER JOIN  dbo.LOAI\_SP ON dbo.SAN\_PHAM.MaLoaiSP = dbo.LOAI\_SP.MaLoaiSP |

### Danh mục nhân viên



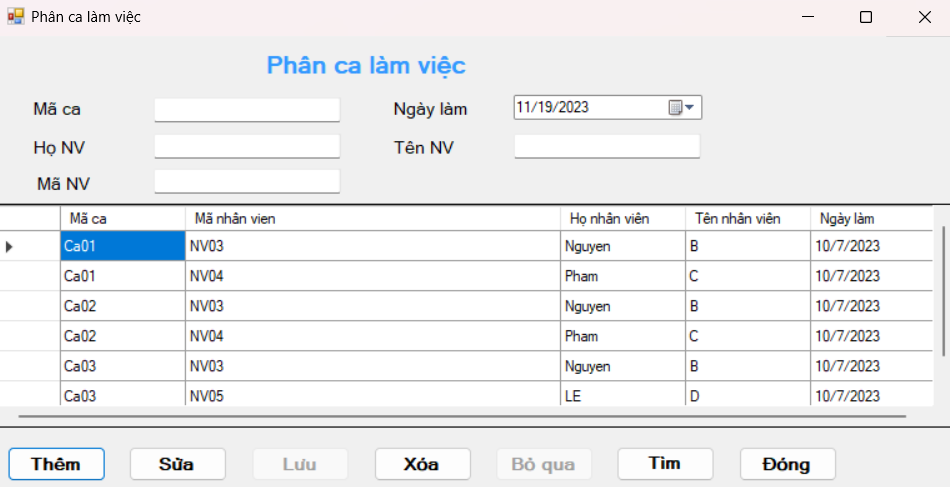
Code C# gọi danh mục:

|  |
| --- |
| private void fDMNhanVien\_Load(object sender, EventArgs e)  {  //txtMaNhanVien.Enabled = false;  btnLuu.Enabled = false;  btnBoQua.Enabled = false;  LoadDataGridView();  }  private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_LoadEmployee";  tblNV = Functions.GetDataToTable(sql); //lấy dữ liệu  dgvNhanVien.DataSource = tblNV;  dgvNhanVien.Columns[0].HeaderText = "Mã nhân viên";  dgvNhanVien.Columns[1].HeaderText = "Họ nhân viên";  dgvNhanVien.Columns[2].HeaderText = "Tên nhân viên";  dgvNhanVien.Columns[3].HeaderText = "Ngày sinh";  dgvNhanVien.Columns[4].HeaderText = "Giới tính";  dgvNhanVien.Columns[5].HeaderText = "Địa chỉ";  dgvNhanVien.Columns[6].HeaderText = "Số điện thoại";  dgvNhanVien.Columns[7].HeaderText = "Tên công việc";  dgvNhanVien.Columns[8].HeaderText = "Tên công việc";  dgvNhanVien.Columns[9].HeaderText = "Số ca";  dgvNhanVien.Columns[10].HeaderText = "Thưởng";  dgvNhanVien.Columns[11].HeaderText = "Ngày tuyển dụng";  dgvNhanVien.Columns[0].Width = 100;  dgvNhanVien.Columns[1].Width = 150;  dgvNhanVien.Columns[2].Width = 100;  dgvNhanVien.Columns[3].Width = 150;  dgvNhanVien.Columns[4].Width = 100;  dgvNhanVien.Columns[5].Width = 100;  dgvNhanVien.Columns[6].Width = 100;  dgvNhanVien.Columns[7].Width = 100;  dgvNhanVien.Columns[8].Width = 100;  dgvNhanVien.Columns[9].Width = 100;  dgvNhanVien.Columns[10].Width = 100;  dgvNhanVien.Columns[11].Width = 150;  dgvNhanVien.AllowUserToAddRows = false;  dgvNhanVien.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW [dbo].[v\_LoadEmployee] AS  SELECT MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, CONG\_VIEC.MaCV,TenCV, SoCa,  Thuong, NgayTuyenDung  FROM NHAN\_VIEN, CONG\_VIEC  WHERE NHAN\_VIEN.MaCV= CONG\_VIEC.MaCV |

### Danh mục phân ca



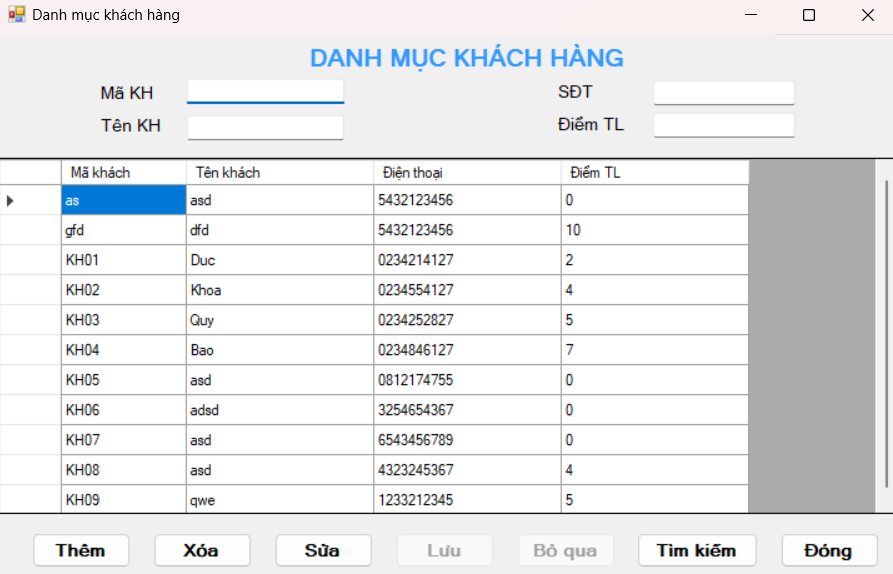
Code C# gọi danh mục:

|  |
| --- |
| private void fPhanCa\_Load(object sender, EventArgs e)  {  btnLuu.Enabled = false;  btnBoQua.Enabled = false;  LoadDataGridView();  }  private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_BangPhanCa";  tblpc = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  dvgPhanCa.DataSource = tblpc; //Nguồn dữ liệu  dvgPhanCa.Columns[0].HeaderText = "Mã ca";  dvgPhanCa.Columns[1].HeaderText = "Mã nhân vien";  dvgPhanCa.Columns[2].HeaderText = "Họ nhân viên";  dvgPhanCa.Columns[3].HeaderText = "Tên nhân viên";  dvgPhanCa.Columns[4].HeaderText = "Ngày làm";  dvgPhanCa.Columns[0].Width = 100;  dvgPhanCa.Columns[1].Width = 300;  dvgPhanCa.Columns[2].Width = 100;  dvgPhanCa.Columns[3].Width = 100;  dvgPhanCa.Columns[4].Width = 100;  dvgPhanCa.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp  dvgPhanCa.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW [dbo].[v\_BangPhanCa]  AS  SELECT BangPhanCa.MaCa, NHAN\_VIEN.MaNV, NHAN\_VIEN.HoNV, NHAN\_VIEN.TenNV, BangPhanCa.NgayLam  FROM BangPhanCa, NHAN\_VIEN  WHERE BangPhanCa.MaNV = NHAN\_VIEN.MaNV |

### Danh mục khách hàng



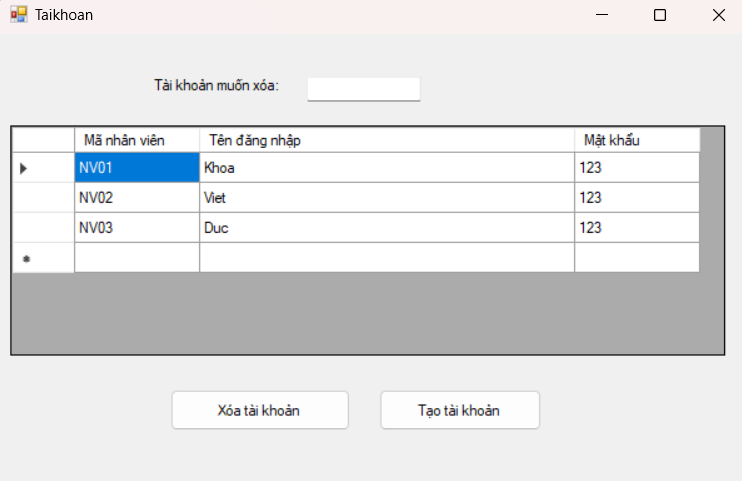
Code C# gọi danh mục:

|  |
| --- |
| private void fDMKhachHang\_Load(object sender, EventArgs e)  {  //txtMaKhach.Enabled = false;  btnLuu.Enabled = false;  btnBoQua.Enabled = false;  LoadDataGridView();  }  private void LoadDataGridView()  {  string sql;  sql = "SELECT \* from KHACH\_HANG";  tblKH = Functions.GetDataToTable(sql); //Lấy dữ liệu từ bảng  dgvKhachHang.DataSource = tblKH; //Hiển thị vào dataGridView  dgvKhachHang.Columns[0].HeaderText = "Mã khách";  dgvKhachHang.Columns[1].HeaderText = "Tên khách";  dgvKhachHang.Columns[2].HeaderText = "Điện thoại";  dgvKhachHang.Columns[3].HeaderText = "Điểm TL";  dgvKhachHang.Columns[0].Width = 100;  dgvKhachHang.Columns[1].Width = 150;  dgvKhachHang.Columns[2].Width = 150;  dgvKhachHang.Columns[3].Width = 150;  dgvKhachHang.AllowUserToAddRows = false;  dgvKhachHang.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

Code tạo bảng trên SQL:

|  |
| --- |
| CREATE TABLE KHACH\_HANG(  MaKH nchar(10) CONSTRAINT PK\_KHACH\_HANG PRIMARY KEY,  TenKH nvarchar(50) NOT NULL,  SDT nchar(11) NOT NULL check (len(SDT)=10),  DiemTL int  ) |

### Danh mục tài khoản



Code C# gọi danh mục:

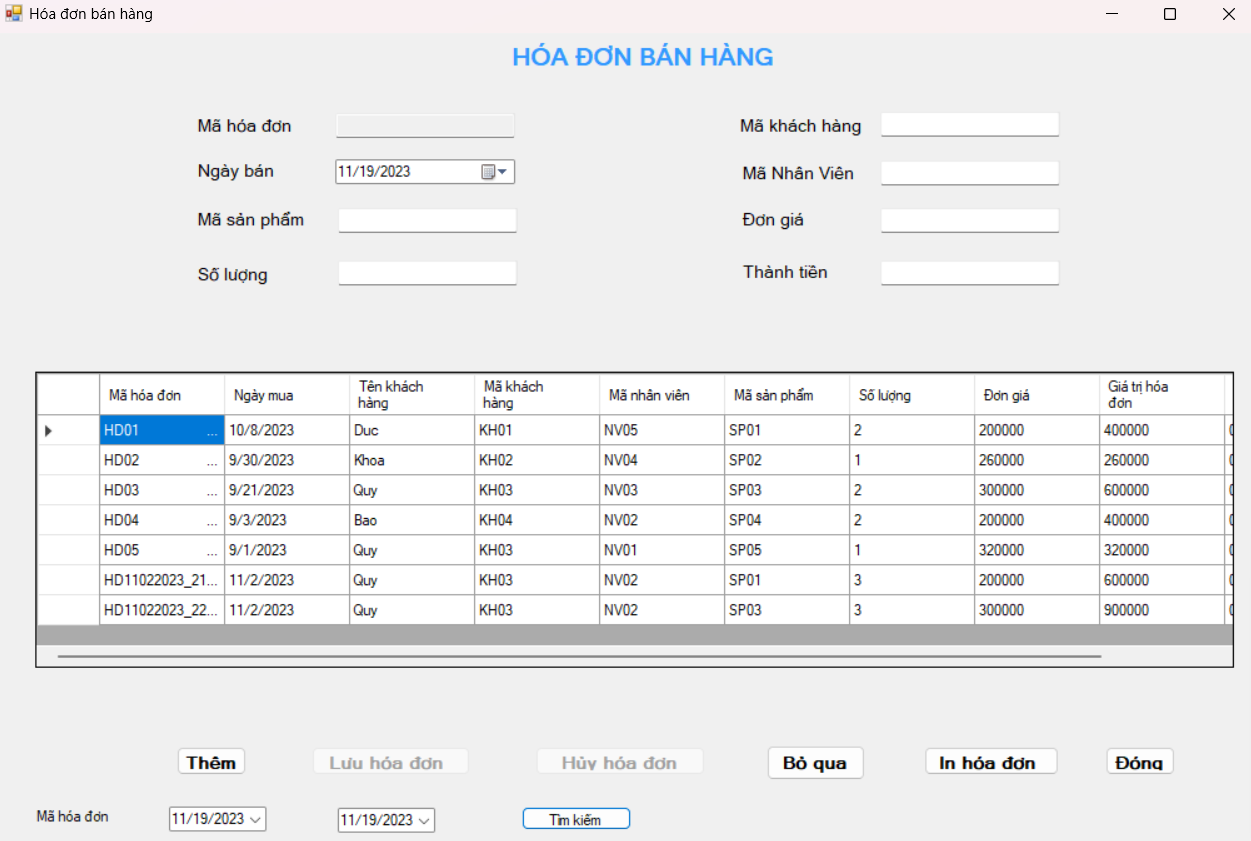
|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From DANGNHAP";  tblDn = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  grid\_TK.DataSource = tblDn; //Nguồn dữ liệu  grid\_TK.Columns[0].HeaderText = "Mã nhân viên";  grid\_TK.Columns[1].HeaderText = "Tên đăng nhập";  grid\_TK.Columns[2].HeaderText = "Mật khẩu";    grid\_TK.Columns[0].Width = 100;  grid\_TK.Columns[1].Width = 300;  grid\_TK.Columns[2].Width = 100;    grid\_TK.EditMode = DataGridViewEditMode.EditProgrammatically;  }  private void Taikhoan\_Load(object sender, EventArgs e)  {  btnXoa.Enabled = true;  LoadDataGridView();  } |

Code tạo bảng trên SQL:

|  |
| --- |
| CREATE TABLE DANGNHAP(  MaNV nchar(10) CONSTRAINT FK\_NhanVien\_DangNhap FOREIGN KEY REFERENCES NHAN\_VIEN,  TenDangNhap nvarchar(30) NOT NULL,  MatKhau nvarchar(10) NOT NULL,  ) |

## Các Form hóa đơn và phiếu chi

### Form Hóa đơn bán hàng



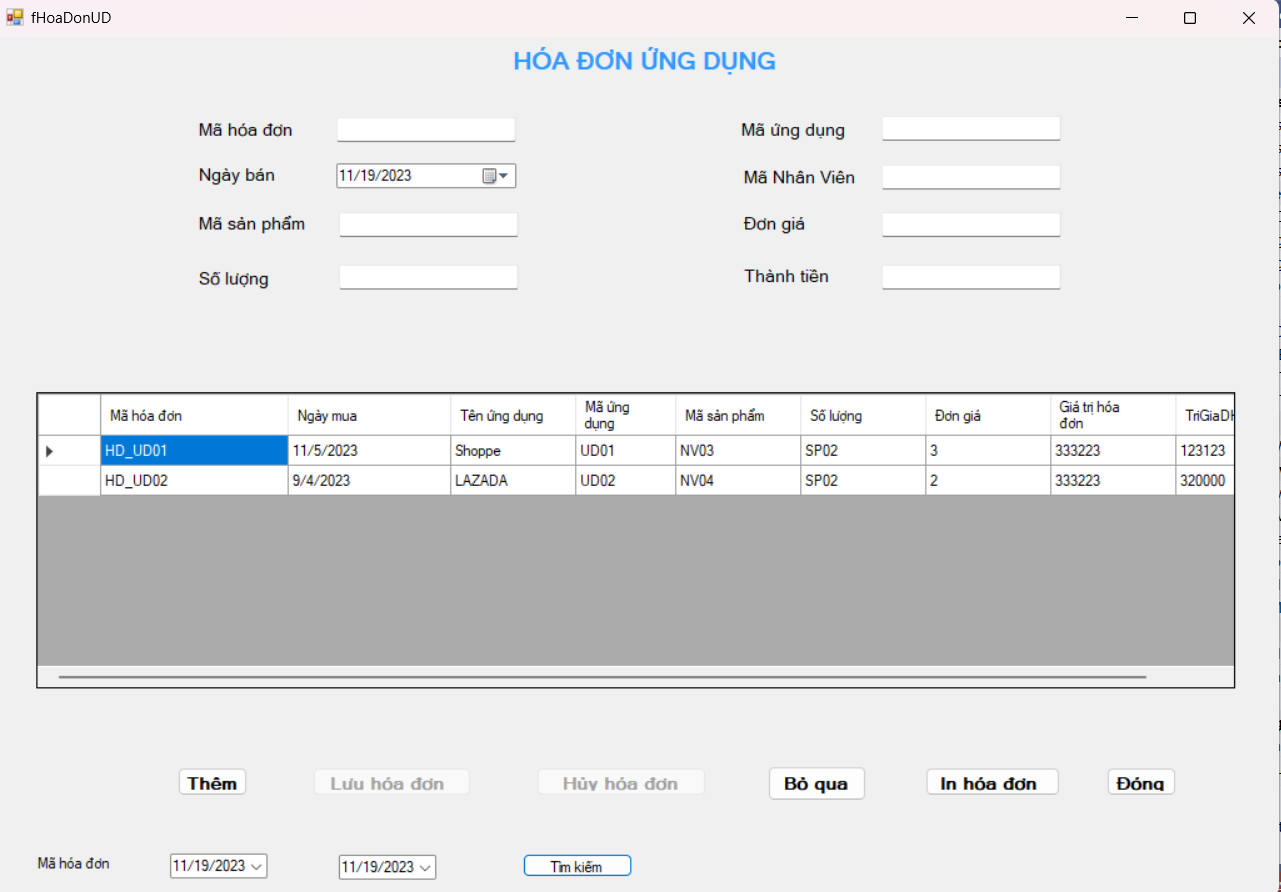
Code C# gọi form:

|  |
| --- |
| private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_HoaDonView";  tblCTHDB = Functions.GetDataToTable(sql); //lấy dữ liệu  dgvHDBanHang.DataSource = tblCTHDB;  dgvHDBanHang.Columns[0].HeaderText = "Mã hóa đơn";  dgvHDBanHang.Columns[1].HeaderText = "Ngày mua";  dgvHDBanHang.Columns[2].HeaderText = "Tên khách hàng";  dgvHDBanHang.Columns[3].HeaderText = "Mã khách hàng";  dgvHDBanHang.Columns[4].HeaderText = "Mã nhân viên";  dgvHDBanHang.Columns[5].HeaderText = "Mã sản phẩm";  dgvHDBanHang.Columns[6].HeaderText = "Số lượng";  dgvHDBanHang.Columns[7].HeaderText = "Đơn giá";  dgvHDBanHang.Columns[8].HeaderText = "Giá trị hóa đơn";  dgvHDBanHang.AllowUserToAddRows = false;  dgvHDBanHang.EditMode = DataGridViewEditMode.EditProgrammatically;  }  private void fHoaDonBanHang\_Load(object sender, EventArgs e)  {  btnThem.Enabled = true;  btnLuu.Enabled = false;  btnXoa.Enabled = false;  txtMaHDBan.ReadOnly = true;  LoadDataGridView();  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW v\_HoaDonView AS  SELECT a.MaHD,a.NgayDatHang,c.TenKH,a.MaKH,a.MaNV,b.MaSP,b.SL, b.DonGia, a.TriGiaHD,c.SDT  FROM HOA\_DON as a  JOIN ChiTietHoaDon as b ON a.MaHD=b.MaHD  JOIN KHACH\_HANG c ON a.MaKH = c.MaKH; |

### Form Hóa đơn ứng dụng



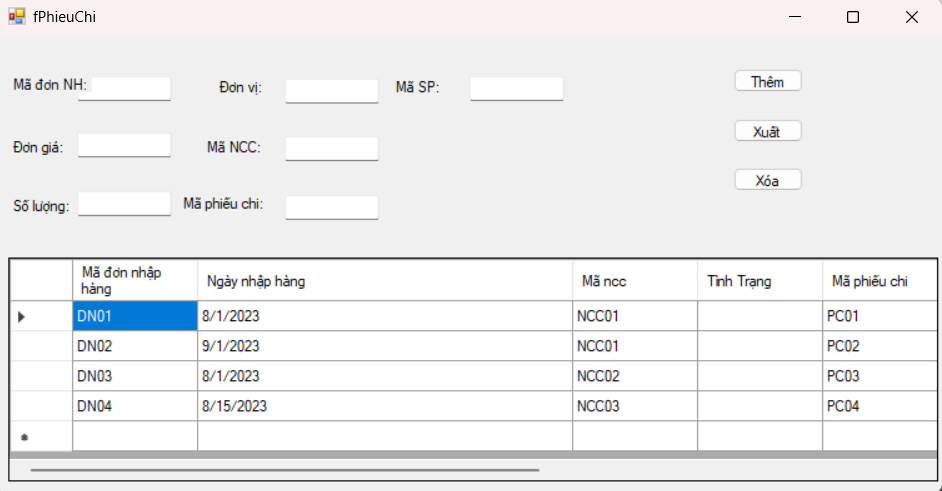
Code C# gọi form:

|  |
| --- |
| private void fHoaDonUD\_Load(object sender, EventArgs e)  {  btnThem.Enabled = true;  btnLuu.Enabled = false;  btnXoa.Enabled = false;  LoadDataGridView();  }  private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From v\_HoaDonUngDung";  tblHDUng\_Dung = Functions.GetDataToTable(sql); //lấy dữ liệu  dgvHDBanHang.DataSource = tblHDUng\_Dung;  dgvHDBanHang.Columns[0].HeaderText = "Mã hóa đơn ";  dgvHDBanHang.Columns[1].HeaderText = "Ngày mua";  dgvHDBanHang.Columns[2].HeaderText = "Tên ứng dụng";  dgvHDBanHang.Columns[3].HeaderText = "Mã ứng dụng";  dgvHDBanHang.Columns[4].HeaderText = "Mã nhân viên";  dgvHDBanHang.Columns[4].HeaderText = "Mã sản phẩm";  dgvHDBanHang.Columns[5].HeaderText = "Số lượng";  dgvHDBanHang.Columns[6].HeaderText = "Đơn giá";  dgvHDBanHang.Columns[7].HeaderText = "Giá trị hóa đơn";  dgvHDBanHang.Columns[0].Width = 150;  dgvHDBanHang.Columns[1].Width = 130;  dgvHDBanHang.Columns[2].Width = 100;  dgvHDBanHang.Columns[3].Width = 80;  dgvHDBanHang.Columns[4].Width = 100;  dgvHDBanHang.Columns[5].Width = 100;  dgvHDBanHang.Columns[6].Width = 100;  dgvHDBanHang.Columns[7].Width = 100;  dgvHDBanHang.AllowUserToAddRows = false;  dgvHDBanHang.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW v\_HoaDonUngDung AS  SELECT a.MaHD\_UD,a.NgayDatHang,c.TenUngDung,c.MaUngDung,a.MaNV,b.MaSP,b.SL, b.DonGia, a.TriGiaDH  FROM HOA\_DON\_UD as a  JOIN CHITIETHD\_UD as b ON a.MaHD\_UD=b.MaHD\_UD  JOIN UNG\_DUNG c ON a.MaUngDung = c.MaUngDung; |

### Form phiếu chi của đơn nhập hàng



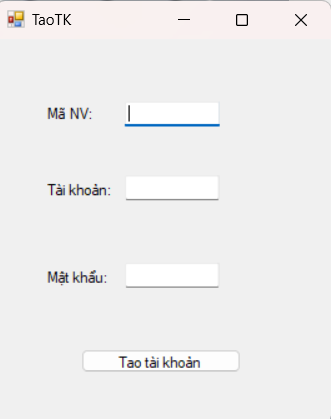
Code C# gọi form:

|  |
| --- |
| private void fPhieuChi\_Load(object sender, EventArgs e)  {  btnThem.Enabled = true;  btnXuat.Enabled = true;  LoadDataGridView();  }  private void LoadDataGridView()  {  string sql;  sql = "SELECT \* From DonNhapHangView";  tblpc = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng  dataGridView1.DataSource = tblpc; //Nguồn dữ liệu  dataGridView1.Columns[0].HeaderText = "Mã đơn nhập hàng";  dataGridView1.Columns[1].HeaderText = "Ngày nhập hàng";  dataGridView1.Columns[2].HeaderText = "Mã ncc";  dataGridView1.Columns[3].HeaderText = "Tình Trạng";  dataGridView1.Columns[4].HeaderText = "Mã phiếu chi";  dataGridView1.Columns[5].HeaderText = "Ngày xuất phiếu";  dataGridView1.Columns[6].HeaderText = "Đơn giá";  dataGridView1.Columns[7].HeaderText = "Số lượng";  dataGridView1.Columns[8].HeaderText = "Đơn vị";  dataGridView1.Columns[9].HeaderText = "Số tiền chi";  dataGridView1.Columns[0].Width = 100;  dataGridView1.Columns[1].Width = 300;  dataGridView1.Columns[2].Width = 100;  dataGridView1.Columns[3].Width = 100;  dataGridView1.Columns[4].Width = 100;  dataGridView1.Columns[5].Width = 100;  dataGridView1.Columns[6].Width = 100;  dataGridView1.Columns[7].Width = 100;  dataGridView1.Columns[8].Width = 100;  dataGridView1.Columns[9].Width = 100;  dataGridView1.EditMode = DataGridViewEditMode.EditProgrammatically;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW [dbo].[DonNhapHangView]  AS  SELECT DNH.MaDonNH, DNH.NgayNH, DNH.MaNCC, DNH.TinhTrang, PC.MaPhieuChi,  PC.NgayXuatPhieu, CTH.DonGia, CTH.SL, CTH.DonVi, PC.SoTienChi  FROM DON\_NHAP DNH  LEFT JOIN PHIEU\_CHI PC ON DNH.MaDonNH = PC.MaDonNH  LEFT JOIN ChiTietDonNhap CTH ON DNH.MaDonNH = CTH.MaDonNH; |

## Form tạo tài khoản



* Tạo tài khoản:

|  |
| --- |
| CREATE procedure [dbo].[proc\_ThemTaiKhoan] @tk nvarchar(30), @mk NVARCHAR(10), @manv  NVARCHAR(10)  As  BEGIN  INSERT INTO DANGNHAP(MaNV, TenDangNhap, MatKhau) VALUES (@manv, @tk, @mk)  END |

Code C#:

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("proc\_ThemTaiKhoan", Functions.ConAdmin);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@manv", txtMaNV.Text);  cmd.Parameters.AddWithValue("@tk", txtTK.Text);  cmd.Parameters.AddWithValue("@mk", txtMK.Text);  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Tạo tài khoản thành công");    }  else  {  MessageBox.Show("Tạo tài khoản thất bại, vui lòng nhập lại");    }  } |