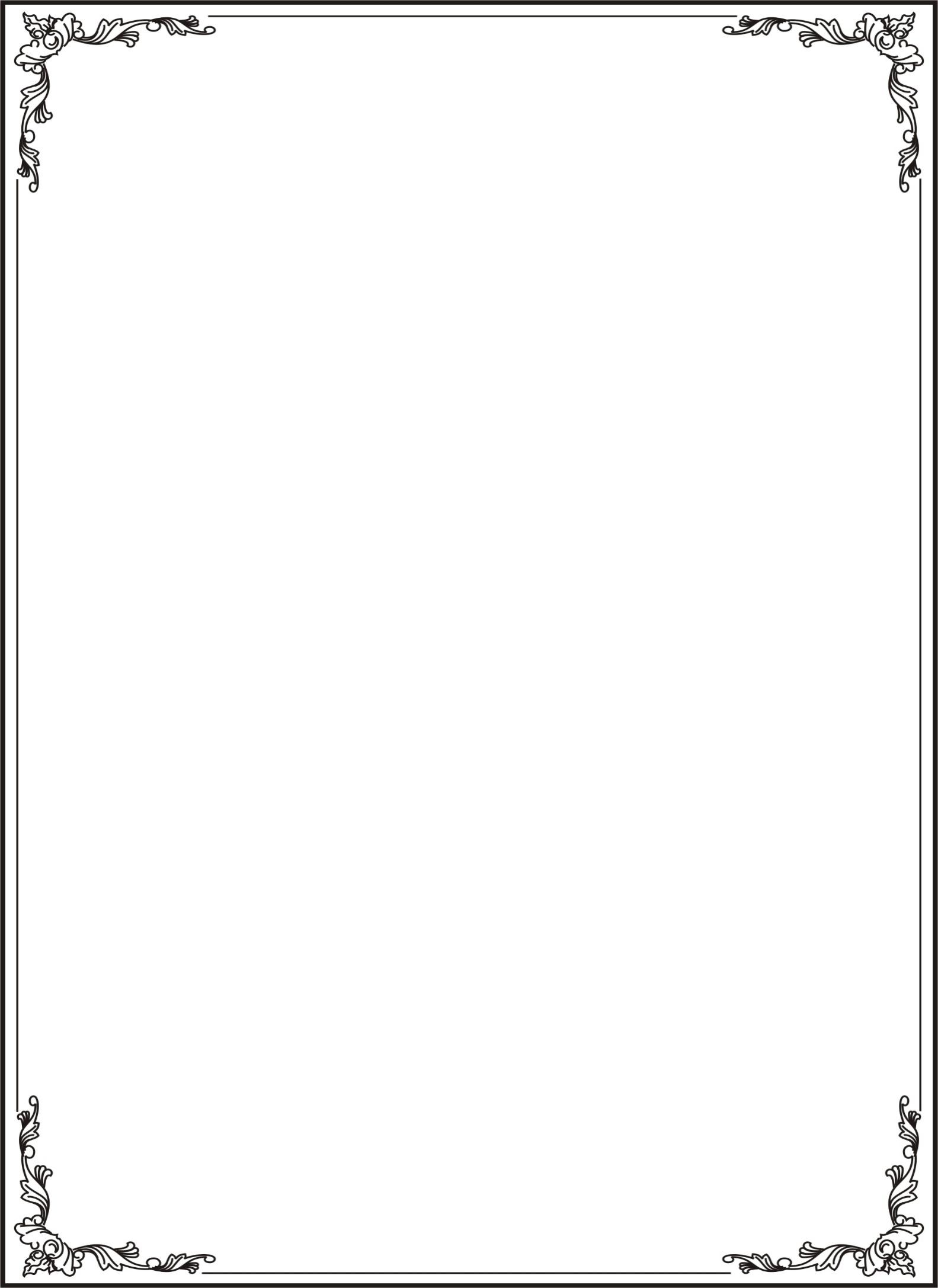
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

🙠🙟🕮🙝🙢



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

**TÊN ĐỀ TÀI:**

**Quản lý hoạt động quán trà sữa - đồ ăn vặt**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Mã học phần:** *DBMS330284\_22\_2\_01CLC*

**Nhóm thực hiện:** 06

**Học kỳ:** 2

**Năm học:** 2022 – 2023

**Thành phố Hồ Chí Minh, Tháng 05 năm 2023**

**DANH SÁCH THÀNH VIÊN THAM GIA VIẾT TIỂU LUẬN**

**HỌC KÌ II, NĂM HỌC 2022–2023**

**Nhóm: 06**

**Tên đề tài:** Quản lý hoạt động quán trà sữa - đồ ăn vặt

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN SINH VIÊN** | **MÃ SỐ SINH VIÊN** | **TỈ LỆ %**  **HOÀN THÀNH** |
| **1** | Trần Nguyễn Trí Đạt | 21110162 | 100% |
| **2** | Nguyễn Việt An | 21110118 | 100% |
| **3** | Trần Viết Trung | 21110859 | 100% |
| **4** | Vương Đình Hiếu | 21110181 | 100% |

***Ghi chú:***

Tỷ lệ %: Mức độ phần trăm hoàn thành của từng sinh viên tham gia.

**Trưởng nhóm:** Vương Đình Hiếu

**Nhận xét của giảng viên:**

*Tp. Hồ Chí Minh – Tháng 05 năm 2023*

Giảng viên chấm điểm

**MỤC LỤC**

[LỜI NÓI ĐẦU 8](#_Toc134652480)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 9](#_Toc134652481)

[**1. Đặc tả đề tài** 9](#_Toc134652482)

[**1.1. Giới thiệu chung về đối tượng khảo sát** 9](#_Toc134652483)

[**1.2. Mô tả bài toán** 9](#_Toc134652484)

[**1.3. Nghiệp vụ bài toán** 10](#_Toc134652485)

[**1.4. Mô tả chức năng của bài toán** 12](#_Toc134652486)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 13](#_Toc134652487)

[**1. Thiết kế CSDL mức quan niệm** 13](#_Toc134652488)

[**2. Thiết kế CSDL mức logic** 14](#_Toc134652489)

[**3. Các ràng buộc cần có** 15](#_Toc134652490)

[**4. Cài đặt CSDL và các ràng buộc** 18](#_Toc134652491)

[**4.1. Cài đặt** 18](#_Toc134652492)

[**4.2. Class diagram** 27](#_Toc134652493)

[**4.3. Trigger kiểm tra các ràng buộc** 28](#_Toc134652494)

[**4.4. Các View** 31](#_Toc134652495)

[CHƯƠNG 3: THIẾT KẾ CÁC THỨC NĂNG 35](#_Toc134652496)

[**1. Kết nối cơ sở dữ liệu** 35](#_Toc134652497)

[**1.1. Chuỗi kết nố**i 35](#_Toc134652498)

[**1.2. Kiểm tra kết nối** 35](#_Toc134652499)

[**2. Code C# phương thức thực hiện các câu lệnh truy vấn và phi truy vấn ở tầng Data Access** 36](#_Toc134652500)

[**2.1. Phương thức thực hiện Hàm SQL trả về một giá trị** 36](#_Toc134652501)

[**2.2. Phương thức thực hiện câu lệnh truy vấn SQL (query) trả về một bảng** 36](#_Toc134652502)

[**2.3. Phương thức thực hiện các câu lệnh phi truy vấn SQL (non-query) trả về đúng sai để kiểm tra hoàn thành** 36](#_Toc134652503)

[**3. Lớp lưu trữ tài khoản và mật khẩu đăng nhập hiện thời** 37](#_Toc134652504)

[**4. Xem thông tin danh mục** 38](#_Toc134652505)

[**5. Quản lý nhân viên** 39](#_Toc134652506)

[**5.1. Lọc nhân viên** 39](#_Toc134652507)

[**5.2. Thêm, sửa, xoá nhân viên** 39](#_Toc134652508)

[**5.3. Quản lý lương** 42](#_Toc134652509)

[5.3.1. *Lọc chi tiết lương* 42](#_Toc134652510)

[5.3.2. *Tìm lương theo mã nhân viên* 42](#_Toc134652511)

[5.3.3. *Thêm, sửa, xoá chi tiết lương* 43](#_Toc134652512)

[**5.4. Quản lý chức vụ** 45](#_Toc134652513)

[5.4.1. *Lọc chức vụ* 45](#_Toc134652514)

[5.4.2. Thêm, sửa chức vụ 45](#_Toc134652515)

[**5.5. Quản lý đăng nhập** 47](#_Toc134652516)

[5.5.1. Tạo tài khoản, thêm tài khoản mới cho nhân viên 47](#_Toc134652517)

[5.5.2. Gán quyền cho tài khoản 48](#_Toc134652518)

[5.5.3. Đổi mật khẩu 48](#_Toc134652519)

[5.5.4. Xóa tài khoản 49](#_Toc134652520)

[**6. Quản lý doanh thu, thao tác bán hàng** 50](#_Toc134652521)

[**6.1. Hóa đơn và chi tiết hóa đơn** 50](#_Toc134652522)

[6.1.1. Lọc hóa đơn 50](#_Toc134652523)

[6.1.2. Tính tổng tiền của nhiều hóa đơn đã được lọc 51](#_Toc134652524)

[6.1.3. Các hàm hỗ trợ chức năng 51](#_Toc134652525)

[6.1.4. Thêm, sửa hóa đơn 53](#_Toc134652526)

[6.1.5. Tìm kiếm chi tiết hóa đơn 55](#_Toc134652527)

[6.1.6. Thêm, sửa chi tiết hóa đơn 55](#_Toc134652528)

[**6.2. Quản lý phiếu mua** 57](#_Toc134652529)

[6.2.1. Lọc chi tiết mua 57](#_Toc134652530)

[6.2.2. Thêm, sửa, xoá chi tiết mua 57](#_Toc134652531)

[6.2.3. Tính tổng số tiền chi tiết phiếu mua của một phiếu mua 59](#_Toc134652532)

[6.2.4. Thêm phiếu mua 60](#_Toc134652533)

[**6.3. Danh mục sản phẩm bán chạy** 60](#_Toc134652534)

[**6.4. Tính toán doanh thu** 61](#_Toc134652535)

[**7. Quản lý khách hàng** 61](#_Toc134652536)

[**7.1. Lọc thông tin khách hàng** 61](#_Toc134652537)

[**7.2. Thêm, sửa khách hàng** 62](#_Toc134652538)

[**8. Quản lý ca làm** 64](#_Toc134652539)

[**8.1. Quản lý ca làm** 64](#_Toc134652540)

[8.1.1. Thêm ca làm 64](#_Toc134652541)

[8.1.2. Lọc ca làm 64](#_Toc134652542)

[**8.2. Phân công** 65](#_Toc134652543)

[8.2.1. Lọc phân công 65](#_Toc134652544)

[8.2.2. Thêm phân công 66](#_Toc134652545)

[8.2.3. Điểm danh phân công 66](#_Toc134652546)

[**9. Quản lý kho** 67](#_Toc134652547)

[**9.1. Lưu trữ** 67](#_Toc134652548)

[**9.2. Nhập kho** 68](#_Toc134652549)

[**9.3. Xuất kho** 69](#_Toc134652550)

[**10. Quản lý thực đơn** 70](#_Toc134652551)

[**10.1. Thêm món** 70](#_Toc134652552)

[**10.2. Chỉnh sửa món** 70](#_Toc134652553)

[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 72](#_Toc134652554)

[**1. Tạo Role và phân quyền cho nhân viên trong SQL** 72](#_Toc134652555)

[**2. Role của người quản lý** 73](#_Toc134652556)

[**3. Các trigger và thủ tục được sử dụng để tạo, xóa User và gán quyền** 74](#_Toc134652557)

[**3.1. Trigger** 74](#_Toc134652558)

[**3.2. Thủ tục** 75](#_Toc134652559)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 77](#_Toc134652560)

[**1. Form Đăng nhập** 77](#_Toc134652561)

[**2. Form Thông tin cá nhân của người đăng nhập** 78](#_Toc134652562)

[**3. Form quản lý** 79](#_Toc134652563)

[**3.1. Form Home** 79](#_Toc134652564)

[**3.2. Form quản lý nhân viên** 80](#_Toc134652565)

[3.2.1. *Form chỉnh sửa thông tin nhân viên, thêm, xóa nhân viên* 80](#_Toc134652566)

[3.2.2. *Form tìm kiếm thông tin nhân viên* 86](#_Toc134652567)

[3.2.3. Form quản lý lương nhân viên 86](#_Toc134652568)

[3.2.4. Form quản lý chức vụ 88](#_Toc134652569)

[3.2.5. Form quản lý đăng nhập 89](#_Toc134652570)

[**3.3. Form Quản lý doanh thu** 90](#_Toc134652571)

[3.3.1. Form Thông tin hóa đơn 90](#_Toc134652572)

[3.3.2. Form Quản lý phiếu mua 91](#_Toc134652573)

[3.3.3. Form Danh mục sản phẩm bán chạy 92](#_Toc134652574)

[3.3.4. Form Tính toán doanh thu 93](#_Toc134652575)

[**3.4. Form Quản lý khách hàng** 93](#_Toc134652576)

[**3.5. Form Quản lý ca làm** 94](#_Toc134652577)

[3.5.1. *Form quản lý ca làm* 94](#_Toc134652578)

[3.5.2. Form Phân công 95](#_Toc134652579)

[**3.6. Form Quản lý kho** 96](#_Toc134652580)

[**3.7. Form Quản lý thực đơn** 98](#_Toc134652581)

[**4. Form bán hàng** 98](#_Toc134652582)

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến thầy *Nguyễn Thành Sơn* vì sự hướng dẫn tận tình của thầy trong suốt học phần, nhờ đó đã giúp chúng em hoàn thành bài báo cáo cuối kì này. Những kiến thức thầy truyền đạt đã giúp chúng em nắm chắc những kiến thức căn bản, nền tảng và tích luỹ thêm nhiều kinh nghiệm để thực hành xây dựng một hệ thống quản trị cơ sở dữ liệu. Thông qua đồ án cuối kì môn học này, nhóm em xin trình bày lại quá trình phát triển một hệ thống quản trị cơ sở dữ liệu, cũng như lập trình một số đồ án liên quan.

Trong quá trình xây dựng đồ án môn học, khó tránh khỏi những sai sót. Vì vậy, nhóm chúng em rất mong sự góp ý đến từ thầy để bài làm của nhóm được hoàn thiện, phát triển đầy đủ hơn. Nhóm chúng em kính chúc thầy sức khoẻ và thành công trên con đường giảng dạy

Cuối cùng nhóm chúng em cũng chân thành cảm ơn các thầy cô và các bạn đã cùng tham gia học phần Hệ quản trị cơ sở dữ liệu đã nhiệt tình giúp đỡ và động viên chúng em trong quá trình học tập.

# **LỜI NÓI ĐẦU**

Cuộc sống xã hội ngày càng tiên tiến và hiện đại hơn nhờ vào ứng dụng rất lớn của công nghệ thông tin vào đời sống xã hội. Với sự ứng dụng rộng rãi và tiện lợi đó công nghệ thông tin đã ngày càng tiến sâu vào đời sống thường nhật của con người. Ngày nay, dễ dàng nhận thấy ứng dụng của công nghệ thông tin trong bất kì lĩnh vực hay ngành nghề nào của xã hội. Chính vì như thế công nghệ thông tin đã và đang giữ vai trò ưu thế trong đời sống kinh tế, xã hội của rất nhiều quốc gia trên thế giới nói chung và Việt Nam nói riêng, bởi vì ứng dụng rộng rãi của ngành này trong đời sống hàng ngày.

Để làm nên sự thành công cho các ứng dụng của công nghệ thông tin phải kể đến các hệ thống quản lý dữ liệu, thông tin một công cụ hỗ trợ đắc lực của nhiều doanh nghiệp. Đặc biệt là lĩnh vực quản lý bán hàng hiện nay, trong công tác quản lý dữ liệu, với sự trợ giúp của công nghệ thông tin quá trình quản lý dữ liệu có thể đạt được hiệu quả cao cũng như giảm thiểu các sai sót về dữ liệu và đồng thời lưu giữ thông tin an toàn hơn so với các cách quản lý thủ công của ngày trước

Chính vì những lý do trên, ứng dụng của Công nghệ thông tin đối với công tác quản lý dữ liệu trong việc quản lý bán hàng đang ngày xu hướng. Cho nên, để làm rõ hơn về các ứng dụng của công nghệ thông tin đối với công tác quản lý trong thực tế ngày nay nhóm chúng em lựa chọn đề tài ***“Quản lý hoạt động quán trà sữa - đồ ăn vặt”*** để nghiên cứu và phát triển các ứng dụng vào việc quản lý hệ thống thông tin trong môn học Hệ quản trị cơ sở dữ liệu.

# **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

## **Đặc tả đề tài**

### **Giới thiệu chung về đối tượng khảo sát**

Chúng em lựa chọn đối tượng khảo sát là một quán trà sữa có tên “Trà Sữa Nhỏ”. Quán có tất cả 10 bàn, lượng khách tối đa cùng một thời điểm là 40 người. Tổng số nhân viên của quán là 12 người cùng với một người quản lý để điều hành. Thời gian hoạt động trong ngày của quán là sáng từ 8h tới 12h, chiều từ 12h-16h và tối từ 16h - 22h. Doanh thu hàng tháng rơi vào khoảng 100 triệu đồng.

### **Mô tả bài toán**

Quán trà sữa “Trà Sữa Nhỏ” cần có một hệ thống để quản lý các hoạt động kinh doanh cũng như tính toán các khoản thu chi. Hệ thống bao gồm các dữ liệu sau:

Các nhân viên được phân biệt với nhau thông qua mã nhân viên và có thông tin cá nhân cơ bản: họ tên nhân viên, ngày sinh, giới tính, ngày bắt đầu làm.

Mỗi nhân viên có một chức vụ. Chức vụ được phân biệt thông qua mã chức vụ, tên chức vụ, lương 1 giờ làm.

Giờ làm việc được phân theo ca, mỗi ca được phân biệt với nhau thông qua mã ca làm việc và có thêm các thông tin: tên ca làm việc, giờ bắt đầu, giờ kết thúc, tổng số giờ.

Chi tiết lương trong ngày của mỗi nhân viên theo chức vụ sẽ gồm có ngày làm, tổng số ca làm trong ngày, thành tiền, kỳ lương.

Mỗi khách hàng tới quán có nhu cầu trở thành hội viên sẽ được nhân viên lưu trữ các thông tin và được phân biệt với nhau thông qua mã khách hàng, các thôg tin cá nhân bao gồm: tên khách hàng, sđt, địa chỉ, điểm tích lũy (10000 vnđ = 1 điểm tính theo hóa đơn). Khách hàng được chia làm hai loại khách hàng VIP và Member được phân biệt với nhau thông qua mã loại khách hàng và các thông tin để mô tả: tên loại khách hàng, ghi chú. Điểm tích lũy đủ 100 sẽ được trở thành hội viên VIP ngược lại sẽ là Member, khách VIP được quyền dùng điểm tích lũy để thanh toán (1điểm = 1000 vnđ).

Các mặt hàng được bán ở quán được phân biệt với nhau thông qua mã mặt hàng và có tên hàng, giá tiền. Mỗi mặt hàng được xếp vào một loại mặt hàng khác nhau. Các loại mặt hàng được phân biệt với nhau thông qua mã loại hàng và có tên loại hàng.

Nhân viên sẽ in hóa đơn để khách hàng thanh toán. Mỗi hóa đơn được phân biệt với nhau thông qua mã hóa đơn và gồm có các thông tin: ngày giờ in hóa đơn, tổng giá tiền, điểm tích lũy được thêm. Một hóa đơn sẽ có một hoặc nhiều mặt hàng và chỉ thuộc về một bàn duy nhất. Mỗi bàn sẽ được phân biệt với nhau thông qua mã bàn và có thông tin số bàn, trạng thái.

Mỗi ngày đều phải sử dụng tiền vốn để chi tiền mua nguyên liệu làm các món trong menu. Những nguyên liệu sẽ được chứa vào trong kho, khi sắp hết sẽ tiếp tục đi mua. Mỗi lần chi tiền mua nguyên vật liệu sẽ xuất một phiếu: mã phiếu, ngày giờ, thành tiền. Các nguyên vật liệu sẽ được phân biệt với nhau thông qua mã nguyên vật liệu và bao gồm tên nguyên vật liệu, đơn vị, giá một đơn vị.

Kho chứa các nguyên vật liệu. Mỗi ngày kho sẽ có số lượng và tình trạng các nguyên vật liệu khác nhau phân biệt với nhau thông qua mã kho và cho biết thông tin về ngày tháng. Nguyên vật liệu khi ở trong kho sẽ có hạn sử dụng, số lượng hiện còn.

Khi nhập một nguyên vật liệu vào kho sẽ có phiếu nhập kho phân biệt qua mã nhập kho, ngày giờ, số lượng. Tương tự với xuất kho ta có mã xuất kho, ngày giờ, số lượng.

### **Nghiệp vụ bài toán**

\* **Nhập, xuất hàng:**

Cuối mỗi ngày thu ngân sẽ kiểm tra hàng tồn trong kho, xem xét các mặt hàng nào sắp hết hoặc quá hạn sử dụng, sau đó báo cáo với quản lý để chi tiền nhập hàng.

*Ví dụ:*

Ống hút, thìa, cốc: dưới 200 cái

Trà đen, lục trà: dưới 5l

Sữa tươi có đường: dưới 8l

Xoài, mãng cầu, bơ, …: dưới 20kg

Khoai lang: dưới 15kg

….…………………

Pha chế sẽ lấy nguyên vật liệu từ kho, mỗi lần lấy sẽ xuất phiếu xuất kho gồm mã phiếu xuất kho, tên các nguyên vật liệu và số lượng xuất.

*Công thức tính lời:*

Tiền lời = Giá bán - giá gốc + giá mặt bằng.

**\* Bán hàng:**

Khách hàng tới quán sẽ được nhân viên sắp xếp vào bàn trống, sau đó tư vấn khách hàng lựa chọn món. Khách hàng sau khi quyết định được món sẽ chốt đơn với nhân viên chạy bàn, sau đó nhân viên sẽ báo món đến cho pha chế và thu ngân. Món sau khi hoàn thành sẽ được nhân viên phục vụ tận bàn, sau đó trạng thái bàn sẽ được chuyển về đang phục vụ. Khách có nhu cầu gọi thêm đồ sẽ order với nhân viên chạy bàn và chi phí sẽ được cộng thêm vào hóa đơn thanh toán.

Khi khách hàng yêu cầu thanh toán thì nhân viên thu ngân sẽ xuất hóa đơn giao cho nhân viên chạy bàn đưa tới cho khách hàng kiểm tra và thanh toán theo 2 hình thức: tiền mặt hoặc chuyển khoản. Khách hàng có thể sử dụng điểm tích lũy để trừ vào tiền hóa đơn với điều kiện là khách hàng vip. Nếu là tiền mặt sẽ giao cho thu ngân cất vào ngăn kéo, tính toán tiền thừa và trả lại cho khách. Sau khi thanh toán bàn sẽ trở lại trạng thái trống.

Trường hợp khách hàng tự ý đổi bàn trong thời gian phục vụ, nhân viên chạy bàn sẽ phải thông báo lại cho thu ngân chuyển trạng thái bàn và đổi bàn cho khách để tính toán hóa đơn.

**\* Báo cáo - thống kê:**

Quản lý cửa hàng sẽ tổng hợp hóa đơn, phiếu nhập hàng, … để lập báo cáo doanh thu định kỳ hằng ngày, tháng, quý, năm. Sau mỗi ca làm việc thu ngân sẽ kết ca, đếm lại tiền mặt trong két cộng với tiền chuyển khoản của khách hàng để so sánh lại với tổng thu trên hóa đơn trong ca đó. Nếu thiếu hụt trên 20.000 VNĐ nhân viên ca đó sẽ phải đền bù tiền tương ứng cho cửa hàng. Ngoài ra, thu ngân phải lấy dữ liệu từ máy chấm công đưa lên hệ thống giờ làm việc của các nhân viên trong ca làm.

Quản lý cửa hàng sẽ thống kê mặt hàng bán chạy, không bán chạy, từ đó lập ra kế hoạch để phát triển menu của cửa hàng.

**\*Quản lý nhân viên:**

Nhân viên quán gồm: 1 quản lý, 2 thu ngân, 4 pha chế, 6 chạy bàn.

Một ca làm có 1 thu ngân, 2 pha chế, 3 chạy bàn. Tất cả nhân viên trong ca làm phải chấm công bằng máy chấm công.

Lương của quản lý: 5 triệu / tháng

Lương của nhân viên thu ngân: 25k/h

### **Mô tả chức năng của bài toán**

***Đối với người quản lý:***

Thêm, xóa, chỉnh sửa thông tin nhân viên, phân quyền cho nhân viên. Phân công lịch làm, điều chỉnh lương, nhập xuất nguyên vật liệu và quản lí kho. Chỉnh sửa menu. Tổng kết doanh thu.

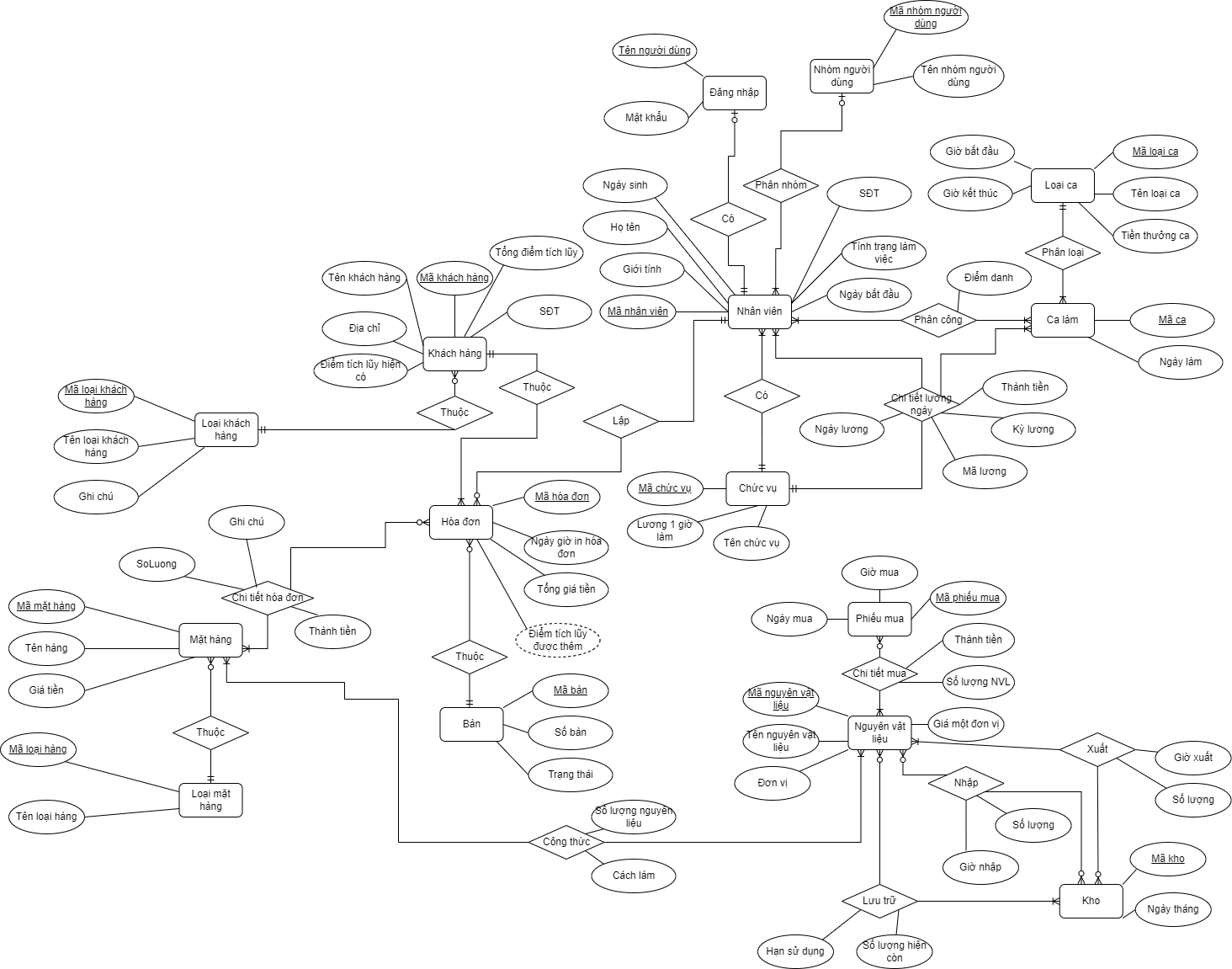
***Đối với nhân viên thu ngân:***

Thêm, xóa, sửa thông tin hội viên (khách hàng), quản lí bàn, in hóa đơn, kiểm tra tồn kho nguyên liệu và báo cáo với quản lý

Phân quyền: quyền admin toàn quyền với hệ thống. Còn nhân viên không thể tham gia vào việc quyền của các tài khoản khác

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

## **Thiết kế CSDL mức quan niệm**



## **Thiết kế CSDL mức logic**

**Các lược đồ quan hệ**

* **NhanVien**(MaNV, MaChucVu, HoTen, NgaySinh, GioiTinh, NgayBatDauLam, SDT, TinhTrangLamViec)
* **ChucVu**(MaChucVu, TenChucVu, Luong1GioLam)
* **CaLam**(MaCa, MaLoaiCa, NgayLam)
* **LoaiCa**(MaLoaiCa, TenLoaiCa, GioBatDau, GioKetThuc, TienThuongCa)
* **ChiTietLuongNgay**(MaLuong, MaNV, MaCa, MaChucVu, NgayLuong, ThanhTien, KyLuong)
* **KhachHang**(MaKH, MaLoaiKH, TenKhachHang, DiaChi, SDT,  
   DiemTichLuyHienTai, TongDiemTichLuy)
* **LoaiKhachHang**(MaLoaiKH, TenLoaiKhachHang, GhiChu)
* **HoaDon**(MaHoaDon, MaNV, MaKH, MaBan, NgayGioInHoaDon, TongGiaTien, DiemTichLuyThem)
* **Ban**(MaBan, SoBan, TrangThai)
* **MatHang**(MaMH, MaLoaiMH, TenHang, GiaTien)
* **LoaiMatHang**(MaLoaiMH, TenLoaiHang)
* **NguyenVatLieu**(MaNVL, TenNVL, DonVi, GiaMotDonVi)
* **PhieuMua**(MaPhieuMua, NgayGioMua)
* **Kho**(MaKho, NgayThang)
* **PhanCong**(MaNV, MaCa, DiemDanh)
* **ChiTietHoaDon**(MaHoaDon, MaMH, SoLuong, ThanhTien)
* **CongThuc**(MaMH, MaNVL, SoLuong)
* **LuuTru**(MaNVL, MaKho,HSD, SoLuongConLai)
* **NhapKho**(MaNVL, MaKho, SoLuong, GioNhap)
* **XuatKho**(MaNVL, MaKho, SoLuong, GioXuat)
* **ChiTietMua**(MaPhieuMua, MaNVL, SoLuongNVL, ThanhTien)
* **DangNhap**(TenNguoiDung, MatKhau, MaNV)
* **NhomNguoiDung**(MaNhomNguoiDung, TenNhomNguoiDung, TenRole)
* **PhanNhom**(MaNV, MaNhomNguoiDung)

## **Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **Table** | **Ràng buộc** |
| 1 | ChucVu | Ràng buộc khoá chính MaChucVu |
| 2 | NhanVien | Ràng buộc khoá chính MaNV  Ràng buộc khoá ngoại MaCV của bảng **NhanVien** tham chiếu tới MaCV của bảng **ChucVu** |
| 3 | LoaiCa | Ràng buộc khoá chính MaLoaiCa |
| 4 | CaLam | Ràng buộc khoá chính MaCa  Ràng buộc khoá ngoại MaLoaiCa của bảng **CaLam** tham chiếu tới MaLoaiCa của bảng **LoaiCa** |
| 5 | LoaiKhachHang | Ràng buộc khoá chính MaLoaiKH |
| 6 | KhachHang | Ràng buộc khoá chính MaKH  Ràng buộc khoá ngoại MaLoaiKH tham chiếu tới MaLoaiKH của bảng **LoaiKH** |
| 7 | Ban | Ràng buộc khoá chính MaBan |
| 8 | HoaDon | Ràng buộc khoá chính MaHD  Ràng buộc khoá ngoại MaNV tham chiếu tới MaNV của bảng **NhanVien**  Ràng buộc khoá ngoại MaKH tham chiếu tới MaKH của bảng **KhachHang**  Ràng buộc khoá ngoại MaBan tham chiếu tới MaBan của bảng **Ban** |
| 9 | LoaiMatHang | Ràng buộc khoá chính MaLoaiMH |
| 10 | MatHang | Ràng buộc khoá chính MaMH  Ràng buộc khoá ngoại MaLoaiMH tham chiếu tới MaLoaiMH của bảng **LoaiMatHang** |
| 11 | NguyenVatLieu | Ràng buộc khoá chính MaNVL |
| 12 | PhieuMua | Ràng buộc khoá chính MaPhieuMua |
| 13 | Kho | Ràng buộc khoá chính MaKho |
| 14 | ChiTietLuongNgay | Ràng buộc khoá chính MaNV, MaCa, MaCV  Ràng buộc khoá ngoại MaNV tham chiếu tới MaNV của bảng **NhanVien**  Ràng buộc khoá ngoại MaCa tham chiếu tới MaCa của bảng **CaLam**  Ràng buộc khoá ngoại MaCV tham chiếu tới MaCV của bảng **ChucVu** |
| 15 | PhanCong | Ràng buộc khoá chính MaNV, MaCa  Ràng buộc khoá ngoại MaNV tham chiếu tới MaNV của bảng **NhanVien**  Ràng buộc khoá ngoại MaCa tham chiếu tới MaCa của bảng **CaLam** |
| 16 | ChiTietHoaDon | Ràng buộc khoá chính MaHoaDon, MaMH  Ràng buộc khoá ngoại MaHoaDon tham chiếu tới MaHoaDon của bảng **HoaDon**  Ràng buộc khoá ngoại MaMH tham chiếu tới MaMH của bảng **MatHang** |
| 17 | CongThuc | Ràng buộc khoá chính MaMH, MaNVL  Ràng buộc khoá ngoại MaMH tham chiếu tới MaMH của bảng **MatHang**  Ràng buộc khoá ngoại MaNVL tham chiếu tới MaNVL của bảng **NguyenVatLieu** |
| 18 | LuuTru | Ràng buộc khoá chính MaNVL, MaKho  Ràng buộc khoá ngoại MaNVL tham chiếu tới MaNVL của bảng **NguyenVatLieu**  Ràng buộc khoá ngoại MaKho tham chiếu tới MaKho của bảng **Kho** |
| 19 | NhapKho | Ràng buộc khoá chính MaNVL, MaKho  Ràng buộc khoá ngoại MaNVL tham chiếu tới MaNVL của bảng **NguyenVatLieu**  Ràng buộc khoá ngoại MaKho tham chiếu tới MaKho của bảng **Kho** |
| 20 | XuatKho | Ràng buộc khoá chính MaNVL, MaKho  Ràng buộc khoá ngoại MaNVL tham chiếu tới MaNVL của bảng **NguyenVatLieu**  Ràng buộc khoá ngoại MaKho tham chiếu tới MaKho của bảng **Kho** |
| 21 | ChiTietMua | Ràng buộc khoá chính MaPhieuMua, MaNVL  Ràng buộc khoá ngoại MaNVL tham chiếu tới MaNVL của bảng **NguyenVatLieu**  Ràng buộc khoá ngoại MaPhieuMua tham chiếu tới MaPhieuMua của bảng **PhieuMua** |
| 22 | DangNhap | Ràng buộc khóa chính TenNguoiDung  Ràng buộc khóa ngoại MaNV tham chiếu tới MaNV của bảng **NhanVien** |
| 23 | NhomNguoiDung | Ràng buộc khóa chính MaNhomNguoiDung |
| 24 | PhanNhom | Ràng buộc khóa chính MaNV  Ràng buộc khóa ngoại MaNhomNguoiDung tham chiếu tới MaNhomNguoiDung của bảng **NhomNguoiDung** |

## **Cài đặt CSDL và các ràng buộc**

### **Cài đặt**

|  |
| --- |
| ***Bảng Chức vụ*** |
| CREATE TABLE ChucVu(  MaChucVu CHAR(10) NOT NULL,  TenChucVu NVARCHAR(20),  LuongMotGioLam FLOAT,  CONSTRAINT PK\_ChucVu PRIMARY KEY(MaChucVu)  ) |

|  |
| --- |
| ***Bảng Nhân viên*** |
| CREATE TABLE NhanVien(  ID INT IDENTITY(1,1) NOT NULL,  MaNV AS ('NV'+ RIGHT('0000000' + CAST(ID AS VARCHAR(8)), 8)) PERSISTED,  -- VARCHAR(10)  MaChucVu CHAR(10),  HoTen NVARCHAR(40),  NgaySinh DATE,  GioiTinh CHAR(10),  NgayBatDauLam DATE,  SDT CHAR(10),  TinhTrang NVARCHAR(10),  CONSTRAINT PK\_NhanVien PRIMARY KEY(MaNV),  CONSTRAINT FK\_NhanVien\_ChucVu FOREIGN KEY(MaChucVu) REFERENCES ChucVu(MaChucVu)  ON DELETE SET NULL  ON UPDATE CASCADE  ) |

|  |
| --- |
| ***Bảng loại ca*** |
| CREATE TABLE LoaiCa(  MaLoaiCa CHAR(10) NOT NULL ,  TenLoaiCa NVARCHAR(30),  GioBatDau TIME,  GioKetThuc TIME,  TienThuongCa FLOAT,  CONSTRAINT PK\_LoaiCa PRIMARY KEY(MaLoaiCa)  ) |

|  |
| --- |
| ***Bảng Ca làm*** |
| CREATE TABLE CaLam(  MaLoaiCa CHAR(10),  NgayLam DATE,  MaCa AS ('CL' + RIGHT(RTRIM(MaLoaiCa), 1) + convert(varchar(8),cast(NgayLam as date),112)) PERSISTED,  -- VD: CLS20230323 (ca sang ngay: 23/03/2023)  -- VARCHAR(11)  --  CONSTRAINT PK\_CaLam PRIMARY KEY(MaCa),  CONSTRAINT FK\_CaLam\_LoaiCa FOREIGN KEY(MaLoaiCa) REFERENCES LoaiCa(MaLoaiCa)  ON DELETE SET NULL  ON UPDATE CASCADE  ) |

|  |
| --- |
| ***Bảng Loại Khách Hàng*** |
| CREATE TABLE LoaiKhachHang(  MaLoaiKH CHAR(10) NOT NULL,  TenLoaiKH NVARCHAR(30),  GhiChu NVARCHAR(Max),  CONSTRAINT PK\_LoaiKhachHang PRIMARY KEY(MaLoaiKH)  ) |

|  |
| --- |
| ***Bảng Khách Hàng*** |
| CREATE TABLE KhachHang(  ID INT IDENTITY(1,1) NOT NULL,  MaKH AS ('KH'+ RIGHT('00000000' + CAST(ID AS VARCHAR(9)), 9)) PERSISTED,  -- CHAR(11)  MaLoaiKH CHAR(10),  TenKH NVARCHAR(40),  DiaChi NVARCHAR(200),  SDT CHAR(10),  DiemTichLuyHienTai FLOAT DEFAULT 0,  TongDiemTichLuy FLOAT DEFAULT 0,  CONSTRAINT PK\_KhachHang PRIMARY KEY(MaKH),  CONSTRAINT FK\_KhachHang\_LoaiKhachHang FOREIGN KEY (MaLoaiKH) REFERENCES LoaiKhachHang(MaLoaiKH)  ON DELETE SET NULL  ON UPDATE CASCADE  ) |

|  |
| --- |
| ***Bảng Bàn*** |
| CREATE TABLE Ban(  MaBan CHAR(10) NOT NULL ,  SoBan INT,  TrangThai NVARCHAR(30) DEFAULT N'Trống',  CONSTRAINT PK\_Ban PRIMARY KEY(MaBan)  ) |

|  |
| --- |
| ***Bảng Hóa đơn*** |
| CREATE TABLE HoaDon(  ID INT IDENTITY(1,1) NOT NULL,  MaHoaDon AS ('BILL'+ RIGHT('00000000' + CAST(ID AS VARCHAR(9)), 9)) PERSISTED,  -- VARCHAR(13)  --  MaNV VARCHAR(10),  MaKH VARCHAR(11),  MaBan CHAR(10),  NgayGioInHoaDon DATETIME,  TongGiaTien INT,  DiemTichLuyThem AS (CAST(TongGiaTien AS FLOAT) / CAST(10000 AS FLOAT)),  DiemSuDung FLOAT,  CONSTRAINT PK\_HoaDon PRIMARY KEY(MaHoaDon),  CONSTRAINT FK\_HoaDon\_NhanVien FOREIGN KEY(MaNV) REFERENCES NhanVien(MaNV)  ON DELETE SET NULL  ON UPDATE CASCADE,  CONSTRAINT FK\_HoaDon\_KhachHang FOREIGN KEY(MaKH) REFERENCES KhachHang(MaKH)  ON DELETE SET NULL  ON UPDATE CASCADE,  CONSTRAINT FK\_HoaDon\_Ban FOREIGN KEY(MaBan) REFERENCES Ban(MaBan)  ON DELETE SET NULL  ON UPDATE CASCADE  ) |

|  |
| --- |
| ***Bảng Loại mặt hàng*** |
| CREATE TABLE LoaiMatHang(  MaLoaiMH CHAR(10) NOT NULL,  TenLoaiHang NVARCHAR(30),  CONSTRAINT PK\_LoaiMatHang PRIMARY KEY(MaLoaiMH)  ) |

|  |
| --- |
| ***Bảng Mặt hàng*** |
| CREATE TABLE MatHang(  MaMH CHAR(10) NOT NULL,  MaLoaiMH CHAR(10),  TenHang NVARCHAR(30),  GiaTien FLOAT,  CONSTRAINT PK\_MatHang PRIMARY KEY(MaMH),  CONSTRAINT FK\_MatHang\_LoaiMatHang FOREIGN KEY(MaLoaiMH) REFERENCES LoaiMatHang(MaLoaiMH)  ON DELETE SET NULL  ON UPDATE CASCADE  ) |

|  |
| --- |
| ***Bảng Nguyên vật liệu*** |
| CREATE TABLE NguyenVatLieu(  MaNVL CHAR(10) NOT NULL,  TenNVL NVARCHAR(30),  DonVi CHAR(10),  GiaMotDonVi FLOAT,  CONSTRAINT PK\_NguyenVatLieu PRIMARY KEY(MaNVL)  ) |

|  |
| --- |
| ***Bảng Phiếu mua*** |
| CREATE TABLE PhieuMua(  NgayGioMua DATETIME,  MaPhieuMua AS CAST('PM' + replace(convert(varchar(8), NgayGioMua, 112)+convert(varchar(8), NgayGioMua, 114), ':','') AS VARCHAR(16)) PERSISTED  -- VARCHAR(16)  CONSTRAINT PK\_PhieuMua PRIMARY KEY(MaPhieuMua)  ) |

|  |
| --- |
| ***Bảng Kho*** |
| CREATE TABLE Kho(  NgayThang DATE,  MaKho AS CAST('KHO' + convert(varchar(8),NgayThang,112) AS VARCHAR(11)) PERSISTED  CONSTRAINT PK\_Kho PRIMARY KEY(MaKho)  ) |

|  |
| --- |
| ***Bảng Chi tiết lương ngày*** |
| CREATE TABLE ChiTietLuongNgay(  MaNV VARCHAR(10), MaCa VARCHAR(11),  MaChucVu CHAR(10),  MaLuong AS ('ML' + RIGHT(RTRIM(MaNV), 8) + RIGHT(RTRIM(MaCa), 9)) PERSISTED,  NgayLuong DATE,  ThanhTien FLOAT,  KyLuong DATE,  CONSTRAINT PK\_ChiTietLuongNgay PRIMARY KEY (MaNV, MaCa, MaChucVu),  CONSTRAINT FK\_ChiTietLuongNgay\_NhanVien FOREIGN KEY(MaNV) REFERENCES NhanVien(MaNV),  CONSTRAINT FK\_ChiTietLuongNgay\_CaLam FOREIGN KEY(MaCa) REFERENCES CaLam(MaCa),  CONSTRAINT FK\_ChiTietLuongNgay\_ChucVu FOREIGN KEY(MaChucVu) REFERENCES ChucVu(MaChucVu)  ) |

|  |
| --- |
| ***Bảng Phân công*** |
| CREATE TABLE PhanCong(  MaNV VARCHAR(10),  MaCa VARCHAR(11),  DiemDanh INT,  CONSTRAINT PK\_PhanCong PRIMARY KEY(MaNV, MaCa),  CONSTRAINT FK\_PhanCong\_NhanVien FOREIGN KEY(MaNV) REFERENCES NhanVien(MaNV),  CONSTRAINT FK\_PhanCong\_CaLam FOREIGN KEY (MaCa) REFERENCES CaLam(MaCa)  ) |

|  |
| --- |
| ***Bảng Chi tiết hóa đơn*** |
| CREATE TABLE ChiTietHoaDon(  MaHoaDon VARCHAR(13),  MaMH CHAR(10),  SoLuongMH INT,  ThanhTien FLOAT,  CONSTRAINT PK\_ChiTietHoaDon PRIMARY KEY(MaHoaDon, MaMH),  CONSTRAINT FK\_ChiTietHoaDon\_HoaDon FOREIGN KEY(MaHoaDon) REFERENCES HoaDon(MaHoaDon),  CONSTRAINT FK\_ChiTietLuongNgay\_MatHang FOREIGN KEY(MaMH) REFERENCES MatHang(MaMH)  ) |

|  |
| --- |
| ***Bảng Công thức*** |
| CREATE TABLE CongThuc(  MaMH CHAR(10),  MaNVL CHAR(10),  SoLuong INT,  CachLam TEXT,  CONSTRAINT PK\_CongThuc PRIMARY KEY(MaMH, MaNVL),  CONSTRAINT FK\_CongThuc\_MatHang FOREIGN KEY(MaMH) REFERENCES MatHang(MaMH),  CONSTRAINT FK\_CongThuc\_NguyenVatLieu FOREIGN KEY(MaNVL) REFERENCES NguyenVatLieu(MaNVL)  ) |

|  |
| --- |
| ***Bảng Lưu trữ*** |
| CREATE TABLE LuuTru(  MaNVL CHAR(10),  MaKho VARCHAR(11),  HSD DATE,  SoLuongConLai INT,  CONSTRAINT PK\_LuuTru PRIMARY KEY(MaNVL, MaKho),  CONSTRAINT FK\_LuuTru\_NguyenVatLieu FOREIGN KEY(MaNVL) REFERENCES NguyenVatLieu(MaNVL),  CONSTRAINT FK\_LuuTru\_Kho FOREIGN KEY(MaKho) REFERENCES Kho(MaKho)  ) |

|  |
| --- |
| ***Bảng Nhập kho*** |
| CREATE TABLE NhapKho(  MaNVL CHAR(10),  MaKho VARCHAR(11),  SoLuong INT,  GioNhap TIME(0) not null,  CONSTRAINT PK\_NhapKho PRIMARY KEY (MaNVL, MaKho, GioNhap),  CONSTRAINT FK\_NhapKho\_NguyenVatLieu FOREIGN KEY(MaNVL) REFERENCES NguyenVatLieu(MaNVL),  CONSTRAINT FK\_NhapKho\_Kho FOREIGN KEY(MaKho) REFERENCES Kho(MaKho)  ) |

|  |
| --- |
| ***Bảng Xuất kho*** |
| CREATE TABLE XuatKho(  MaNVL CHAR(10),  MaKho VARCHAR(11),  SoLuong INT,  NgayGioXuat DATETIME,  CONSTRAINT PK\_XuatKho PRIMARY KEY(MaNVL, MaKho, NgayGioXuat),  CONSTRAINT FK\_XuatKho\_NguyenVatLieu FOREIGN KEY(MaNVL) REFERENCES NguyenVatLieu(MaNVL),  CONSTRAINT FK\_XuatKho\_Kho FOREIGN KEY(MaKho) REFERENCES Kho(MaKho)  ) |

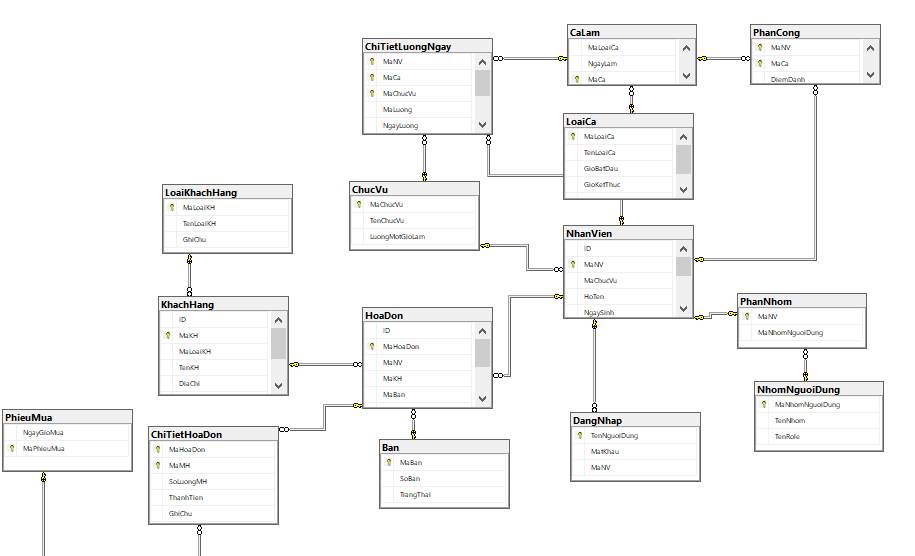
|  |
| --- |
| ***Bảng Chi tiết mua*** |
| CREATE TABLE ChiTietMua(  MaPhieuMua VARCHAR(16),  MaNVL CHAR(10),  SoLuongNVL INT,  ThanhTien FLOAT,  CONSTRAINT PK\_ChiTietMua PRIMARY KEY(MaPhieuMua, MaNVL),  CONSTRAINT FK\_ChiTietMua\_PhieuMua FOREIGN KEY(MaPhieuMua) REFERENCES PhieuMua(MaPhieuMua),  CONSTRAINT FK\_ChiTietMua\_NguyenVatLieu FOREIGN KEY(MaNVL) REFERENCES NguyenVatLieu(MaNVL)  ) |

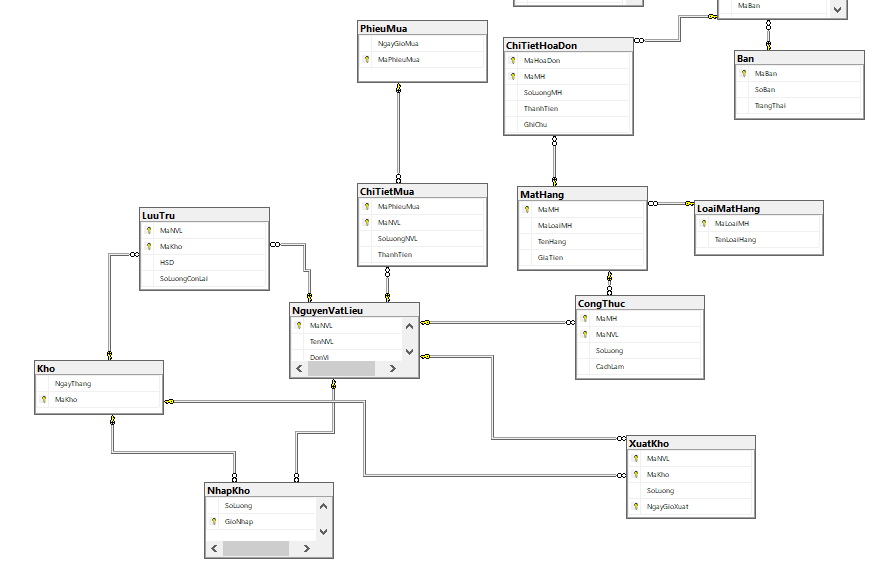
|  |
| --- |
| ***Bảng Đăng nhập*** |
| CREATE TABLE DangNhap(  TenNguoiDung NVARCHAR(30),  MatKhau NVARCHAR(16),  MaNV VARCHAR(10),  CONSTRAINT PK\_DangNhap PRIMARY KEY(TenNguoiDung),  CONSTRAINT FK\_DangNhap\_NhanVien FOREIGN KEY(MaNV) REFERENCES NhanVien(MaNV)  ) |

|  |
| --- |
| ***Bảng Nhóm người dùng*** |
| CREATE TABLE NhomNguoiDung(  MaNhomNguoiDung varchar(10),  TenNhom nvarchar(30),  TenRole varchar(30),  CONSTRAINT PK\_NhomNguoiDung PRIMARY KEY(MaNhomNguoiDung)  ) |

|  |
| --- |
| ***Bảng Phân nhóm*** |
| CREATE TABLE PhanNhom(  MaNV varchar(10),  MaNhomNguoiDung varchar(10),  CONSTRAINT PK\_PhanNhom PRIMARY KEY(MaNV),  CONSTRAINT FK\_PhanNhom\_NhanVien FOREIGN KEY(MaNV) REFERENCES NhanVien(MaNV),  CONSTRAINT FK\_PhanNhom\_NhomNguoiDung FOREIGN KEY(MaNhomNguoiDung) REFERENCES NhomNguoiDung(MaNhomNguoiDung)  ) |

### **Class diagram**





### **Trigger kiểm tra các ràng buộc**

|  |
| --- |
| ***Trigger sau khi thực hiện in hóa đơn (update ngày giờ) thì sửa lại trạng thái bàn là trống và update điểm tích lũy cho khách hàng:*** |
| -- Sau khi thực hiện in hóa đơn (update ngày giờ) thì sửa lại trạng thái bàn là trống và update điểm tích lũy cho khách hàng  CREATE TRIGGER TRG\_HoaDon\_AfterUpdate\_UpdateDTLAndTableStatus on HoaDon  AFTER UPDATE  AS  DECLARE @makh varchar(11), @maban char(10), @dtlAdd float, @dtlUsed float  SELECT @makh = ol.MaKH, @maban = ne.MaBan, @dtlAdd = ne.DiemTichLuyThem, @dtlUsed = ne.DiemSuDung  FROM inserted ne, deleted ol  WHERE ne.MaHoaDon LIKE ol.MaHoaDon  BEGIN  if @makh IS NOT null -- Nếu khách không phải hội viên thì thôi  BEGIN  UPDATE KhachHang  SET DiemTichLuyHienTai += @dtlAdd - @dtlUsed, TongDiemTichLuy += @dtlAdd  WHERE MaKH = @makh  END  if @maban IS NOT null -- Nếu khách mua về thì không sửa trạng thái  BEGIN  UPDATE Ban  SET TrangThai = N'Trống'  WHERE MaBan = @maban  END  END |

|  |
| --- |
| ***Trigger kiểm tra nếu điểm tổng lớn hơn hoặc bằng 100 thì cập nhật thành khách vip:*** |
| -- Nếu điểm tổng lớn hơn hoặc bằng 100 thì set thành khách vip  CREATE TRIGGER TRG\_KhachHang\_AfterUpdate\_UpdateVIP on KhachHang  AFTER UPDATE  AS  DECLARE @maKH char(10), @diemTichLuyHienTai float, @tongDiemTichLuy float, @maLoaiKH char(10)  SELECT @maKH = ne.MaKH, @diemTichLuyHienTai = ne.diemTichLuyHienTai, @tongDiemTichLuy = ne.TongDiemTichLuy, @maLoaiKH = ol.MaLoaiKH  FROM inserted ne, deleted ol  WHERE ne.MaKH = ol.MaKH  IF @maLoaiKH NOT LIKE 'VIP'  BEGIN  IF @tongDiemTichLuy >= 100  BEGIN  UPDATE KhachHang  SET MaLoaiKH = 'VIP'  WHERE MaKH LIKE @maKH  END  END |

|  |
| --- |
| ***Trigger khi điểm danh trong phân công thì tự động insert các thông tin MaNV, MaCV, MaCa, … vào chi tiết lương hoặc xóa chi tiết lương khi hủy điểm danh:*** |
| CREATE TRIGGER [dbo].[TRG\_PhanCong\_AfterUpdate\_GenerateSalaryForPresentEmp] on [dbo].[PhanCong]  AFTER UPDATE  AS  DECLARE @maNV varchar(10), @maChucVu char(10), @maCa varchar(11), @ngayLuong date, @thanhTien float, @diemDanh int,  @luong1GioLam float, @tienThuongCa float, @maLoaiCa char(10), @Flag int, @oldDiemDanh int, @kyLuong date  SELECT @maNV = ne.MaNV, @maCa = ne.MaCa, @diemDanh = ne.DiemDanh, @oldDiemDanh = ol.DiemDanh  FROM inserted ne, deleted ol  WHERE ne.MaNV = ol.MaNV  SET @Flag = 1  IF @diemDanh = @oldDiemDanh  BEGIN  Set @Flag = 0  END  SELECT @maChucVu = nv.MaChucVu  FROM NhanVien nv  WHERE nv.MaNV = @maNV  SELECT @ngayLuong = ca.NgayLam, @maLoaiCa = ca.MaLoaiCa  FROM CaLam ca  WHERE ca.MaCa = @maCa  SET @kyLuong = EOMONTH(@ngayLuong)  SELECT @tienThuongCa = TienThuongCa  FROM LoaiCa  WHERE MaLoaiCa = @maLoaiCa  SELECT @luong1GioLam = LuongMotGioLam  FROM ChucVu  WHERE MaChucVu = @maChucVu  SET @thanhTien = @luong1GioLam \* 4 + @tienThuongCa  IF @diemDanh = 1 AND @Flag = 1 -- flag bằng 0 nghĩa là KHÔNG có sự thay đổi  BEGIN  exec dbo.usp\_ThemChiTietLuong @MaNV = @maNV, @MaCa = @maCa, @NgayLuong = @ngayLuong, @ThanhTien = @thanhTien,  @KyLuong = @kyLuong  END  IF @diemDanh = 0 AND @Flag = 1  BEGIN  exec dbo.usp\_XoaChiTietLuong @MaNV = @maNV, @MaCa = @maCa  END |

|  |
| --- |
| ***Trigger xoá nhân viên thì thay đổi trạng thái làm việc của nhân viên đó và xóa tài khoản của nhân viên đó:*** |
| create trigger TRG\_InsteadDelete\_XoaNhanVien ON NhanVien  Instead of DELETE  AS  DECLARE @MaNV varchar(10)  SELECT @MaNV=ol.MaNV  FROM deleted ol  BEGIN  --Chuyển đổi trạng thái làm việc của nhân viên thành 'Đã nghỉ'  UPDATE NhanVien Set TinhTrang = N'Đã nghỉ' WHERE MaNV = @MaNV  -- Xóa Nhân viên khỏi bảng phân nhóm  DELETE FROM PhanNhom WHERE MaNV = @MaNV  -- Xóa tài khoản SQL  EXEC USP\_XoaTaiKhoanNhanVien @MaNV  -- Xóa tài khoản khỏi bảng đăng nhập  DELETE FROM DangNhap WHERE MaNV = @MaNV  END |

|  |
| --- |
| ***Trigger Tạo tài khoản trong SQL sau khi có một tài khoản được tạo trong bảng đăng nhập:*** |
| CREATE TRIGGER TRG\_AfterInsert\_TaoTaiKhoanSQLSauKhiThemDangNhap ON DangNhap  AFTER INSERT  AS  DECLARE @TenNguoiDung nvarchar(30), @MatKhau nvarchar(16)  SELECT @TenNguoiDung = ne.TenNguoiDung, @MatKhau = ne.MatKhau  FROM inserted ne  DECLARE @sqlString nvarchar(max)  BEGIN  /\* Tạo tài khoản login trên cơ sở dữ liệu cho nhân viên với tên người dùng và tài khoản  được truyền vào\*/  SET @sqlString = 'CREATE LOGIN [' + @TenNguoiDung +'] WITH PASSWORD='''+ @MatKhau  +''', DEFAULT\_DATABASE=[QUANLYQUANTRASUA], CHECK\_EXPIRATION=OFF,  CHECK\_POLICY=OFF'  EXEC (@sqlString)  -- Tạo tài khoản người dùng đối với nhân viên đó trên database (tên ngườidùng trùng với tên login)  SET @sqlString= 'CREATE USER ' + @TenNguoiDung +' FOR LOGIN '+ @TenNguoiDung  EXEC (@sqlString)  END |

### **Các View**

|  |
| --- |
| ***View Lương nhân viên:*** |
| CREATE VIEW VW\_LuongNhanVien AS  SELECT DISTINCT NhanVien.MaNV, HoTen, TongTienLuong, q.KyLuong  FROM (ChiTietLuongNgay JOIN NhanVien  ON ChiTietLuongNgay.MaNV=NhanVien.MaNV)  JOIN (SELECT NhanVien.MaNV, KyLuong, SUM(ThanhTien) AS TongTienLuong  FROM ChiTietLuongNgay JOIN NhanVien ON ChiTietLuongNgay.MaNV=NhanVien.MaNV  GROUP BY NhanVien.MaNV, ChiTietLuongNgay.KyLuong) AS q  ON NhanVien.MaNV = q.MaNV  GO |

|  |
| --- |
| ***View Chi tiết lương:*** |
| CREATE VIEW VW\_ChiTietLuong AS  SELECT MaNV, MaCa, ChucVu.TenChucVu, MaLuong, NgayLuong, ThanhTien, KyLuong  FROM ChiTietLuongNgay join ChucVu ON ChiTietLuongNgay.MaChucVu=ChucVu.MaChucVu  GO |

|  |
| --- |
| ***View Thông tin nhân viên:*** |
| CREATE VIEW VW\_ThongTinNV AS  SELECT MaNV, ChucVu.TenChucVu, HoTen, NgaySinh, GioiTinh, NgayBatDauLam, SDT, TinhTrang  FROM NhanVien, ChucVu  WHERE NhanVien.MaChucVu = ChucVu.MaChucVu  GO |

|  |
| --- |
| ***View Thông tin hoá đơn:*** |
| CREATE VIEW VW\_ThongTinHoaDon AS  SELECT MaHoaDon, MaNV, MaKH, MaBan, NgayGioInHoaDon,  TongGiaTien, DiemTichLuyThem, DiemSuDung  FROM HoaDon  GO |

|  |
| --- |
| ***View Chi tiết hoá đơn:*** |
| CREATE VIEW VW\_ChiTietHoaDon AS  SELECT MaHoaDon, TenHang, SoLuongMH, ThanhTien  FROM MatHang inner join ChiTietHoaDon ON MatHang.MaMH = ChiTietHoaDon.MaMH  GO |

|  |
| --- |
| ***View Chi tiết mua:*** |
| CREATE VIEW VW\_ChiTietMua AS  SELECT MaPhieuMua, TenNVL, SoLuongNVL, ThanhTien, NgayGioMua  FROM NguyenVatLieu, (SELECT ChiTietMua.MaPhieuMua, MaNVL, SoLuongNVL, ThanhTien, NgayGioMua  FROM ChiTietMua INNER JOIN PhieuMua  ON ChiTietMua.MaPhieuMua = PhieuMua.MaPhieuMua) Q  WHERE NguyenVatLieu.MaNVL = Q.MaNVL  GO |

|  |
| --- |
| ***View Danh mục sản phẩm bán chạy:*** |
| CREATE VIEW VW\_DanhMucSanPhamBanChay AS  SELECT TOP(10) P.TenHang, P.TenLoaiHang, sl AS SoLuong, Thang, Nam  FROM ( SELECT distinct MaMH, SUM (SoLuongMH) AS sl, MONTH(NgayGioInHoaDon) as Thang, YEAR(NgayGioInHoaDon) as Nam  FROM HoaDon, ChiTietHoaDon  WHERE HoaDon.MaHoaDon = ChiTietHoaDon.MaHoaDon  GROUP BY MaMH,NgayGioInHoaDon) Q, ( SELECT MatHang.MaMH, TenHang, LoaiMatHang.TenLoaiHang  FROM MatHang, LoaiMatHang  WHERE MatHang.MaLoaiMH = LoaiMatHang.MaLoaiMH ) P  WHERE P.MaMH = Q.MaMH  ORDER BY sl DESC  GO |

|  |
| --- |
| ***View Thông tin khách hàng:*** |
| CREATE VIEW VW\_ThongTinKhachHang AS  SELECT KhachHang.MaKH, LoaiKhachHang.TenLoaiKH, KhachHang.TenKH, DiaChi, SDT, DiemTichLuyHienTai, TongDiemTichLuy  FROM KhachHang, LoaiKhachHang  WHERE KhachHang.MaLoaiKH = LoaiKhachHang.MaLoaiKH |

|  |
| --- |
| ***View Ca làm:*** |
| create view VW\_CaLam AS  select MaCa, MaLoaiCa, NgayLam from CaLam |

|  |
| --- |
| ***View Phân công:*** |
| create view VW\_PhanCong as  select PhanCong.MaNV, HoTen, TenLoaiCa, NgayLam, DiemDanh  from PhanCong, NhanVien, (select MaCa, TenLoaiCa, NgayLam from CaLam, LoaiCa where CaLam.MaLoaiCa = LoaiCa.MaLoaiCa)Q  where PhanCong.MaNV = NhanVien.MaNV and PhanCong.MaCa = Q.MaCa |

|  |
| --- |
| ***View thực đơn:*** |
| CREATE VIEW VW\_Menu  AS  SELECT MaMH, TenLoaiHang, TenHang, GiaTien  FROM MatHang inner join LoaiMatHang on MatHang.MaLoaiMH = LoaiMatHang.MaLoaiMH |

|  |
| --- |
| ***View thông tin đăng nhập:*** |
| create view VW\_ThongTinDangNhap AS  SELECT DangNhap.MaNV, TenNguoiDung, MatKhau, TenRole  FROM DangNhap left join PhanNhom on DangNhap.MaNV = PhanNhom.MaNV left join NhomNguoiDung on  PhanNhom.MaNhomNguoiDung = NhomNguoiDung.MaNhomNguoiDung |

|  |
| --- |
| ***View lưu trữ kho:*** |
| create view VW\_LuuTruKho AS  SELECT LuuTru.MaKho, TenNVL, HSD, SoLuongConLai, NguyenVatLieu.DonVi ,NgayThang  FROM LuuTru inner join Kho on LuuTru.MaKho = Kho.MaKho inner join NguyenVatLieu on LuuTru.MaNVL = NguyenVatLieu.MaNVL |

|  |
| --- |
| ***View lưu trữ kho:*** |
| create view VW\_LuuTruKho AS  SELECT LuuTru.MaKho, TenNVL, HSD, SoLuongConLai, NguyenVatLieu.DonVi ,NgayThang  FROM LuuTru inner join Kho on LuuTru.MaKho = Kho.MaKho inner join NguyenVatLieu on LuuTru.MaNVL = NguyenVatLieu.MaNVL |

|  |
| --- |
| ***View thông tin nhập kho:*** |
| create view VW\_ThongTinNhapKho  AS  SELECT NgayThang, NguyenVatLieu.TenNVL, SoLuong, GioNhap  FROM NhapKho inner join Kho vw on NhapKho.MaKho = vw.MaKho inner join NguyenVatLieu on NhapKho.MaNVL = NguyenVatLieu.MaNVL |

|  |
| --- |
| ***View thông tin xuất kho:*** |
| CREATE VIEW VW\_ThongTinXuatKho  AS  SELECT NgayThang, XuatKho.NgayGioXuat, NguyenVatLieu.TenNVL, SoLuong  FROM XuatKho inner join Kho on XuatKho.MaKho = Kho.MaKho inner join NguyenVatLieu on XuatKho.MaNVL = NguyenVatLieu.MaNVL |

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

## **Kết nối cơ sở dữ liệu**

### **Chuỗi kết nố**i

|  |
| --- |
| string ConnStr = "";  SqlConnection conn = null;  SqlCommand comm = null;  SqlDataAdapter da = null;  public DAO(string userName, string password)  {  ConnStr = @"Data Source=(local)\SQLEXPRESS  ;Initial Catalog=QUANLYQUANTRASUA;  User ID=" + userName + ";Password=" + password;  conn = new SqlConnection(ConnStr);  comm = conn.CreateCommand();  } |

### **Kiểm tra kết nối**

|  |
| --- |
| public bool checkConnect(ref string err)  {  try  {  if (conn.State == ConnectionState.Open)  conn.Close();  conn.Open();  }  catch(SqlException q)  {  if(q.Number == 18456)  {  err = "Tài khoản hoặc mật khẩu không chính xác";  return false;  }  }  conn.Close();  return true;  } |

|  |
| --- |
| ***Kiểm tra tính chính xác của tài khoản, mật khẩu khi đăng nhập*** |
| private void btnDangNhap\_Click(object sender, EventArgs e)  {    DBCurrentLogin\_Singleton currentLogin =  DBCurrentLogin\_Singleton.GetCurrentLoginInfo();  currentLogin.UserName = txtUser.Text;  currentLogin.Password = txtPassword.Text;  DAO db = new DAO(currentLogin.UserName, currentLogin.Password);  string err = "";  if(db.checkConnect(ref err) == false)  {  MessageBox.Show(err, "Sai thông tin đăng nhập" ,  MessageBoxButtons.OK,MessageBoxIcon.Error);  }  else  {  frmThongTinNguoiDangNhap frm = new frmThongTinNguoiDangNhap();  frm.ShowDialog();  }  } |

## **Code C# phương thức thực hiện các câu lệnh truy vấn và phi truy vấn ở tầng Data Access**

### **Phương thức thực hiện Hàm SQL trả về một giá trị**

|  |
| --- |
| public T MyExecuteScalar<T> (string strSQL, CommandType ct)  {  T result = default(T);  if (conn.State == ConnectionState.Open)  conn.Close();  conn.Open();  comm.CommandText= strSQL;  comm.CommandType = ct;  if(comm.ExecuteScalar() != DBNull.Value)  {  result = (T)Convert.ChangeType(comm.ExecuteScalar(), typeof(T));  }  conn.Close() ;  return result;  } |

### **Phương thức thực hiện câu lệnh truy vấn SQL (query) trả về một bảng**

|  |
| --- |
| public DataSet ExecuteQueryDataSet(string strSQL, CommandType ct, params SqlParameter[] p)  {  if (conn.State == ConnectionState.Open)  conn.Close();  conn.Open();  comm.CommandText = strSQL;  comm.CommandType = ct;  da = new SqlDataAdapter(comm);  DataSet ds = new DataSet();  da.Fill(ds);  conn.Close();  return ds;  } |

### **Phương thức thực hiện các câu lệnh phi truy vấn SQL (non-query) trả về đúng sai để kiểm tra hoàn thành**

|  |
| --- |
| public bool MyExecuteNonQuery(string strSQL, CommandType ct, ref string error, params SqlParameter[] param)  {  bool f = false;  if (conn.State == ConnectionState.Open)  conn.Close();  conn.Open();  comm.Parameters.Clear();  comm.CommandText = strSQL;  comm.CommandType = ct;  foreach (SqlParameter p in param)  comm.Parameters.Add(p);  try  {  comm.ExecuteNonQuery();  f = true;  }  catch (SqlException ex)  {  error = ex.Message;  if(ex.Number == 229)  {  error = "Bạn không có quyền thao tác";  }  if(ex.Number == 2627)  {  error = "Trùng khóa chính";  }    }  finally  {  conn.Close();  }  return f;  } |

## **Lớp lưu trữ tài khoản và mật khẩu đăng nhập hiện thời**

Sử dụng Singleton Partern “Lazy Initialization” để đảm bảo rằng trong suốt quá trình sử dụng ứng dụng, chúng ta sẽ chỉ có duy nhất một “Tên đăng nhập” và “Mật khẩu”.

|  |
| --- |
| public sealed class DBCurrentLogin\_Singleton  {  private static DBCurrentLogin\_Singleton LoginInfo = null;  public static DBCurrentLogin\_Singleton GetCurrentLoginInfo()  {  if(LoginInfo == null)  {  LoginInfo = new DBCurrentLogin\_Singleton();  }  return LoginInfo;  }  public string UserName { get; set; }  public string Password { get; set; }  } |

## **Xem thông tin danh mục**

|  |
| --- |
| ***View Thông tin nhân viên:*** |
| CREATE VIEW VW\_ThongTinNV AS  SELECT MaNV, ChucVu.TenChucVu, HoTen, NgaySinh, GioiTinh, NgayBatDauLam, SDT, TinhTrang  FROM NhanVien, ChucVu  WHERE NhanVien.MaChucVu = ChucVu.MaChucVu  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet LayThongTinViewNhanVien()  {  return db.ExecuteQueryDataSet("select \* from VW\_ThongTinNV", CommandType.Text, null);  } |

|  |
| --- |
| ***View Chi tiết hoá đơn trong SQL:*** |
| CREATE VIEW VW\_ChiTietHoaDon AS  SELECT MaHoaDon, TenHang, SoLuongMH, ThanhTien  FROM MatHang inner join ChiTietHoaDon ON MatHang.MaMH = ChiTietHoaDon.MaMH |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet LayViewChiTietHoaDon()  {  return db.ExecuteQueryDataSet("select \* from VW\_ChiTietHoaDon", CommandType.Text, null);  } |

|  |
| --- |
| ***View Ca làm:*** |
| create view VW\_CaLam AS  select MaCa, MaLoaiCa, NgayLam from CaLam |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet LayThongTinViewCaLam()  {  return db.ExecuteQueryDataSet("select \* from VW\_CaLam", CommandType.Text, null);  } |

## **Quản lý nhân viên**

### **Lọc nhân viên**

|  |
| --- |
| ***Function Lọc nhân viên:*** |
| CREATE FUNCTION Table\_FN\_LocNhanVien (@MaNV varchar(10), @HoTen nvarchar(40), @ChucVu nvarchar(20))  RETURNS TABLE  RETURN(  SELECT \* FROM VW\_ThongTinNV  WHERE MaNV= ISNULL(@MaNV, MaNV) AND HoTen LIKE '%' + ISNULL(@HoTen, HoTen) + '%' AND TenChucVu=ISNULL(@ChucVu, TenChucVu)  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinNhanVien(string MaNV, string HoTen, string TenChucVu)  {  string maNV = "NULL", hoTen = "NULL", tenChucVu = "NULL";  if(MaNV != null)  {  maNV = "'" + MaNV +"'";  }  if(HoTen != null)  {  hoTen = "N'" + HoTen +"'";  }  if(TenChucVu != null)  {  tenChucVu = "N'" + TenChucVu +"'";  }  return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_LocNhanVien(" + maNV + ", " +  hoTen + ", " + tenChucVu +")",  CommandType.Text, null);  } |

### **Thêm, sửa, xoá nhân viên**

|  |
| --- |
| ***Procedure Thêm nhân viên:*** |
| CREATE PROCEDURE usp\_ThemNhanVien @MaChucVu Char(10), @HoTen nvarchar(40), @NgaySinh date, @GioiTinh char(10),  @NgayBatDauLam date, @SDT char(10), @TinhTrang nvarchar(10)  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  INSERT INTO NhanVien  VALUES (  @MaChucVu, @HoTen, @NgaySinh, @GioiTinh, @NgayBatDauLam, @SDT, @TinhTrang  )  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemNhanVien(ref string error, string MaChucVu,  string HoTen, DateTime NgaySinh, string GioiTinh, DateTime NgayBatDauLam,  string SDT, string TinhTrang)  {  return db.MyExecuteNonQuery("usp\_ThemNhanVien", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaChucVu", MaChucVu),  new SqlParameter("@HoTen", HoTen),  new SqlParameter("@NgaySinh", NgaySinh.Date),  new SqlParameter("@GioiTinh", GioiTinh),  new SqlParameter("@NgayBatDauLam", NgayBatDauLam.Date),  new SqlParameter("@SDT", SDT),  new SqlParameter("@TinhTrang", TinhTrang));  } |

|  |
| --- |
| ***Sửa nhân viên:*** |
| CREATE PROCEDURE usp\_SuaNhanVien @MaNV varchar(10),@MaChucVu Char(10), @HoTen nvarchar(40), @NgaySinh date, @GioiTinh char(10), @NgayBatDauLam date, @SDT char(10), @TinhTrang nvarchar(10)  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  UPDATE NhanVien  SET MaChucVu=@MaChucVu, HoTen=@HoTen, NgaySinh=@NgaySinh, GioiTinh=@GioiTinh, NgayBatDauLam=@NgayBatDauLam,  SDT=@SDT, TinhTrang=@TinhTrang  WHERE MaNV = @MaNV  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool CapNhatNhanVien(ref string error, string MaNV, string MaChucVu,  string HoTen, DateTime NgaySinh, string GioiTinh, DateTime NgayBatDauLam,  string SDT, string TinhTrang)  {  return db.MyExecuteNonQuery("usp\_SuaNhanVien", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV),  new SqlParameter("@MaChucVu", MaChucVu),  new SqlParameter("@HoTen", HoTen),  new SqlParameter("@NgaySinh", NgaySinh.Date),  new SqlParameter("@GioiTinh", GioiTinh),  new SqlParameter("@NgayBatDauLam", NgayBatDauLam.Date),  new SqlParameter("@SDT", SDT),  new SqlParameter("@TinhTrang", TinhTrang));  } |

|  |
| --- |
| ***Xoá nhân viên:*** |
| CREATE PROCEDURE usp\_XoaNhanVien @MaNV varchar(10)  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  DELETE FROM NhanVien  WHERE MaNV = @MaNV  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool XoaNhanVien(ref string error, string MaNV)  {  return db.MyExecuteNonQuery("usp\_XoaNhanVien", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV));  } |

### **Quản lý lương**

#### *Lọc chi tiết lương*

|  |
| --- |
| ***Function tìm kiếm chi tiết lương theo một hoặc nhiều tham số (lọc):*** |
| CREATE FUNCTION Table\_FN\_LocChiTietLuong (@MaNV varchar(10), @MaCa varchar(11), @TenChucVu nvarchar(20))  RETURNS TABLE  RETURN(  SELECT \* FROM VW\_ChiTietLuong  WHERE MaNV=ISNULL(@MaNV, MaNV) AND MaCa=ISNULL(@MaCa, MaCa) AND TenChucVu=ISNULL(@TenChucVu, TenChucVu)  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemChiTietLuong(string MaNV, string MaCa, string ChucVu)  {  string manv = "NULL", maca = "NULL", chucvu = "NULL";  if (MaNV != null)  {  manv = "'" + MaNV + "'";  }  if (MaCa != null)  {  maca = "'" + MaCa + "'";  }  if (ChucVu != null)  {  chucvu = "N'" + ChucVu + "'";  }  return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_LocChiTietLuong(" + manv + ", " +  maca + ", " + chucvu + ")",  CommandType.Text, null);  } |

#### *Tìm lương theo mã nhân viên*

|  |
| --- |
| ***Tìm bảng lương tổng hợp theo mã nhân viên:*** |
| CREATE FUNCTION Table\_FN\_TimKiemLuongNV (@MaNV varchar(10))  RETURNS TABLE  RETURN(  SELECT \* FROM VW\_LuongNhanVien  WHERE MaNV=@MaNV  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemTongLuong(string MaNV)  {  string manv = "NULL";  if (MaNV != null)  manv = "'" + MaNV + "'";  return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_TimKiemLuongNV(" + manv + ")", CommandType.Text, null);  } |

#### *Thêm, sửa, xoá chi tiết lương*

|  |
| --- |
| ***Thêm chi tiết lương:*** |
| CREATE PROCEDURE usp\_ThemChiTietLuong @MaNV varchar(10), @MaCa varchar(11), @MaChucVu Char(10), @NgayLuong date, @ThanhTien float, @KyLuong date  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  INSERT INTO ChiTietLuongNgay  VALUES (  @MaNV, @MaCa, @MaChucVu, @NgayLuong, @ThanhTien, @KyLuong )  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemChiTietLuong(ref string error, string MaNV, string MaCa,  string MaChucVu, DateTime NgayLuong, float ThanhTien, DateTime KyLuong)  {  return db.MyExecuteNonQuery("usp\_ThemChiTietLuong", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV),  new SqlParameter("@MaCa", MaCa),  new SqlParameter("@MaChucVu", MaChucVu),  new SqlParameter("@NgayLuong", NgayLuong.Date),  new SqlParameter("@ThanhTien", ThanhTien),  new SqlParameter("@KyLuong", KyLuong.Date));  } |

|  |
| --- |
| ***Sửa chi tiết lương:*** |
| CREATE PROCEDURE usp\_SuaChiTietLuong @MaNV varchar(10), @MaCa varchar(11), @MaChucVu Char(10),  @NgayLuong date, @ThanhTien float, @KyLuong date  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  UPDATE ChiTietLuongNgay  SET MaCa=@MaCa, MaChucVu=@MaChucVu, NgayLuong=@NgayLuong, ThanhTien=@ThanhTien, KyLuong=@KyLuong  WHERE MaNV = @MaNV  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool CapNhatChiTietLuong(ref string error, string MaNV, string MaCa,  string MaChucVu, DateTime NgayLuong, float ThanhTien, DateTime KyLuong)  {  return db.MyExecuteNonQuery("usp\_SuaChiTietLuong", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV),  new SqlParameter("@MaCa", MaCa),  new SqlParameter("@MaChucVu", MaChucVu),  new SqlParameter("@NgayLuong", NgayLuong.Date),  new SqlParameter("@ThanhTien", ThanhTien),  new SqlParameter("@KyLuong", KyLuong.Date));  } |

|  |
| --- |
| ***Xoá chi tiết lương:*** |
| CREATE PROCEDURE usp\_XoaChiTietLuong @MaLuong varchar(19)  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  DELETE ChiTietLuongNgay  WHERE MaLuong = @MaLuong  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool XoaChiTietLuong(ref string error, string MaNV, string MaCa)  {  return db.MyExecuteNonQuery("usp\_XoaChiTietLuong", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV),  new SqlParameter("@MaCa", MaCa));  } |

### **Quản lý chức vụ**

#### *Lọc chức vụ*

|  |
| --- |
| ***Function tìm kiếm chức vụ theo một hoặc nhiều tham số:*** |
| CREATE FUNCTION Table\_FN\_LocChucVu (@MaChucVu char(10), @LuongMotGioLam float)  RETURNS TABLE  RETURN(  SELECT \* FROM VW\_ThongTinChucVu  WHERE MaChucVu=ISNULL(@MaChucVu, MaChucVu) AND LuongMotGioLam=ISNULL(@LuongMotGioLam, LuongMotGioLam)  ) |

#### Thêm, sửa chức vụ

|  |
| --- |
| ***Thêm chức vụ:*** |
| CREATE PROCEDURE usp\_ThemChucVu @MaChucVu char(10), @TenChucVu nvarchar(20), @LuongMotGioLam float  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  INSERT INTO ChucVu  VALUES (  @MaChucVu, @TenChucVu, @LuongMotGioLam  )  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemChucVu(ref string error, string MaChucVu, string TenChucVu,  float LuongMotGioLam)  {  return db.MyExecuteNonQuery("usp\_ThemChucVu", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaChucVu", MaChucVu),  new SqlParameter("@TenChucVu", TenChucVu),  new SqlParameter("@LuongMotGioLam", LuongMotGioLam));  } |

|  |
| --- |
| ***Sửa chức vụ:*** |
| CREATE PROCEDURE usp\_SuaChucVu @MaChucVu char(10), @TenChucVu nvarchar(20), @LuongMotGioLam float  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRANSACTION  --  UPDATE ChucVu  SET TenChucVu=@TenChucVu, LuongMotGioLam=@LuongMotGioLam  WHERE MaChucVu=@MaChucVu  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool CapNhatChucVu(ref string error, string MaChucVu, string TenChucVu,  float LuongMotGioLam)  {  return db.MyExecuteNonQuery("usp\_SuaChucVu", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaChucVu", MaChucVu),  new SqlParameter("@TenChucVu",TenChucVu),  new SqlParameter("@LuongMotGioLam", LuongMotGioLam));  } |

### **Quản lý đăng nhập**

#### Tạo tài khoản, thêm tài khoản mới cho nhân viên

Khi sử dụng procedure này, thông tin tài khoản muốn thêm sẽ được thêm vào bảng đăng nhập và bảng phân nhóm. Lúc này “*Trigger Tạo tài khoản trong SQL sau khi có một tài khoản được tạo trong bảng đăng nhập****”*** được kích hoạt

|  |
| --- |
| ***Procedure Tạo tài khoản, thêm tài khoản*** |
| CREATE PROCEDURE USP\_ThemTaiKhoan @MaNV varchar(10), @TenNguoiDung nvarchar(30), @MatKhau nvarchar(16),  @MaNhomNguoiDung varchar(10)  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRAN  INSERT INTO PhanNhom VALUES(@MaNV, @MaNhomNguoiDung)  INSERT INTO DangNhap VALUES(@TenNguoiDung, @MatKhau, @MaNV)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemTaiKhoan(ref string err, string MaNV, string TenNguoiDung,  string MatKhau, string MaNhomNguoiDung)  {  return db.MyExecuteNonQuery("USP\_ThemTaiKhoan",  CommandType.StoredProcedure, ref err,  new SqlParameter("@MaNV", MaNV),  new SqlParameter("@TenNguoiDung", TenNguoiDung),  new SqlParameter("@MatKhau", MatKhau),  new SqlParameter("@MaNhomNguoiDung", MaNhomNguoiDung));  } |

#### Gán quyền cho tài khoản

|  |
| --- |
| ***Procedure Gán quyền cho tài khoản*** |
| CREATE PROCEDURE USP\_GanQuyenTaiKhoan @TenNguoiDung nvarchar(30), @MaNhomNguoiDung varchar(10)  AS  BEGIN  DECLARE @sqlString nvarchar(max), @TenRole varchar(30)  SELECT @TenRole = TenRole FROM NhomNguoiDung WHERE MaNhomNguoiDung = @MaNhomNguoiDung  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRAN  IF @TenRole LIKE 'sysadmin'  BEGIN  SET @sqlString = 'ALTER SERVER ROLE sysadmin' +' ADD MEMBER ' + @TenNguoiDung;  END  ELSE  BEGIN  SET @sqlString = 'ALTER ROLE '+ @TenRole +' ADD MEMBER ' + @TenNguoiDung;  END  EXEC(@sqlString)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW;  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool GanQuyenTaiKhoan(ref string err, string TenNguoiDung,  string MaNhomNguoiDung)  {  return db.MyExecuteNonQuery("USP\_GanQuyenTaiKhoan",  CommandType.StoredProcedure, ref err,  new SqlParameter("@TenNguoiDung", TenNguoiDung),  new SqlParameter("@MaNhomNguoiDung", MaNhomNguoiDung));  } |

#### Đổi mật khẩu

|  |
| --- |
| ***Procedure Đổi mật khẩu*** |
| CREATE PROCEDURE USP\_DoiMatKhau @TenNguoiDung nvarchar(30), @MatKhau nvarchar(16)  AS  BEGIN  SET XACT\_ABORT ON  DECLARE @sqlString nvarchar(max)  SET @sqlString = 'ALTER LOGIN [' + @TenNguoiDung + '] WITH PASSWORD=''' + @MatKhau +''''  BEGIN TRY  BEGIN TRAN  UPDATE DangNhap  SET MatKhau = @MatKhau  WHERE TenNguoiDung = @TenNguoiDung  EXEC(@sqlString)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool DoiMatKhau(ref string err, string TenNguoiDung,  string MatKhau)  {  return db.MyExecuteNonQuery("USP\_DoiMatKhau",  CommandType.StoredProcedure, ref err,  new SqlParameter("@TenNguoiDung", TenNguoiDung),  new SqlParameter("@MatKhau", MatKhau));  } |

#### Xóa tài khoản

Procedure USP\_XoaTaiKhoan được gọi khi “*Trigger xoá nhân viên*” được kích hoạt

|  |
| --- |
| ***Procedure Xóa tài khoản*** |
| create procedure USP\_XoaTaiKhoanNhanVien @MaNV varchar(10)  as  begin  SET XACT\_ABORT ON  declare @TenNguoiDung varchar(20), @stringSQL nvarchar(max), @SID smallint, @sqlKill nvarchar(max);  select @TenNguoiDung = TenNguoiDung from DangNhap where MaNV = @MaNV  begin try  SELECT @SID = session\_id FROM sys.dm\_exec\_sessions WHERE login\_name = @TenNguoiDung  SET @sqlKill = 'KILL ' + CAST(@SID as varchar(4))  EXEC (@sqlKill)  begin tran  set @stringSQL = 'drop user ['+@TenNguoiDung+']';  EXEC(@stringSQL);  set @stringSQL = 'drop login ['+@TenNguoiDung+']';  EXEC(@stringSQL);  commit tran  end try  begin catch  rollback;  throw  end catch  end |

## **Quản lý doanh thu, thao tác bán hàng**

### **Hóa đơn và chi tiết hóa đơn**

#### Lọc hóa đơn

|  |
| --- |
| ***Function Tìm kiếm hóa đơn theo một hoặc nhiều tham số*** |
| create function Table\_FN\_LocHoaDon(@MaNV varchar(10), @MaKH varchar(11), @NgayHoaDon date)  RETURNS TABLE AS  RETURN  (  SELECT \*  FROM VW\_ThongTinHoaDon  WHERE MaNV = ISNULL(@MaNV, MaNV)  and (MaKH = ISNULL(@MaKH, MaKH) OR MaKH IS NULL)  and CAST(NgayGioInHoaDon AS date) = ISNULL(CAST(@NgayHoaDon AS date), NgayGioInHoaDon)  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinHoaDon(string MaNV, string MaKH,  DateTime NgayHoaDon)  {  string maNV = "NULL", maKH = "NULL", ngayHoaDon = "NULL";  if (MaNV != null)  {  maNV = "'" + MaNV + "'";  }  if (MaKH != null)  {  maKH = "'" + MaKH + "'";  }  if (NgayHoaDon != new DateTime(1,1,1))  {  ngayHoaDon = "'" + NgayHoaDon.Date.ToString("yyyy-MM-dd") + "'";  }  return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_LocHoaDon(" + maNV + ", " +  maKH + ", " + ngayHoaDon + ")",  CommandType.Text, null);  } |

#### Tính tổng tiền của nhiều hóa đơn đã được lọc

|  |
| --- |
| ***Function Tính tổng tiền của nhiều hóa đơn được lọc ra*** |
| CREATE FUNCTION Scalar\_FN\_TinhTongTienHoaDon (@MaNV varchar(10), @MaKH varchar(11), @NgayHoaDon date)  RETURNS float  AS  BEGIN  DECLARE @tongTien float  SET @tongTien = 0  SELECT @tongTien = SUM(TongGiaTien) FROM dbo.Table\_FN\_LocHoaDon(@MaNV, @MaKH, @NgayHoaDon)  RETURN @tongTien  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public float TinhTongDanhSachHoaDon(string MaNV, string MaKH,  DateTime NgayHoaDon)  {  string maNV = "NULL", maKH = "NULL", ngayHoaDon = "NULL";  if (MaNV != null)  {  maNV = "'" + MaNV + "'";  }  if (MaKH != null)  {  maKH = "'" + MaKH + "'";  }  if (NgayHoaDon != new DateTime(1, 1, 1))  {  ngayHoaDon = "'" + NgayHoaDon.Date.ToString("yyyy-MM-dd") + "'";  }  return db.MyExecuteScalar<float>(  "SELECT dbo.Scalar\_FN\_TinhTongTienHoaDon(" + maNV + ", " +  maKH + ", " + ngayHoaDon + ")", CommandType.Text);  } |

#### Các hàm hỗ trợ chức năng

|  |
| --- |
| ***Function Lấy mã hóa đơn của bàn đang phục vụ*** |
| create function Scalar\_FN\_LayMaHoaDonCuaBanDangPhucVu(@MaBan char(10))  RETURNS VARCHAR(13)  AS  BEGIN  DECLARE @MaHoaDon varchar(13)  SELECT @MaHoaDon = MaHoaDon FROM HoaDon WHERE MaBan = @MaBan AND NgayGioInHoaDon IS NULL  RETURN @MaHoaDon  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public string LayMaHoaDonCuaBanDangPhucVu(string MaBan)  {  string maBan = "'" + MaBan + "'";  return db.MyExecuteScalar<string>  ("select dbo.Scalar\_FN\_LayMaHoaDonCuaBanDangPhucVu(" +  maBan + ")", CommandType.Text);  } |

|  |
| --- |
| ***Function Lấy chi tiết hóa đơn của bàn đang phục vụ*** |
| CREATE FUNCTION Table\_FN\_LayChiTietHoaDonCuaBanDangPhucVu(@MaBan char(10))  RETURNS TABLE  RETURN(  SELECT VW\_ChiTietHoaDon.MaHoaDon, HoaDon.MaBan, TenHang, SoLuongMH, ThanhTien  FROM VW\_ChiTietHoaDon join HoaDon on VW\_ChiTietHoaDon.MaHoaDon = HoaDon.MaHoaDon  WHERE HoaDon.MaBan = @MaBan and NgayGioInHoaDon IS NULL  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet LayCTHDBanDangPhucVu(string MaBan)  {  string maBan = "'" + MaBan + "'";  return db.ExecuteQueryDataSet("select \* " +  "from Table\_FN\_LayChiTietHoaDonCuaBanDangPhucVu (" + maBan + ")",  CommandType.Text, null);  } |

|  |
| --- |
| ***Function Lấy mã hóa đơn của khách hàng mua mang đi đang được phục vụ*** |
| create function Scalar\_FN\_LayMaHoaDonCuaKhachMangDiDangPhucVu()  RETURNS VARCHAR(13)  AS  BEGIN  DECLARE @MaHoaDon varchar(13)  SELECT @MaHoaDon = MaHoaDon FROM HoaDon WHERE MaBan IS NULL AND NgayGioInHoaDon IS NULL  RETURN @MaHoaDon  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public string LayMaHoaDonCuaKhachMangDiDangPhucVu()  {  return db.MyExecuteScalar<string>  ("select dbo.Scalar\_FN\_LayMaHoaDonCuaKhachMangDiDangPhucVu()",  CommandType.Text);  } |

|  |
| --- |
| ***Function Lấy chi tiết hóa đơn của khách hàng mua mang đi đang được phục vụ*** |
| create function Table\_FN\_LayCTHDCuaKhachMangDiDangPhucVu()  RETURNS TABLE  RETURN(  SELECT VW\_ChiTietHoaDon.MaHoaDon, HoaDon.MaBan, TenHang, SoLuongMH, ThanhTien  FROM VW\_ChiTietHoaDon join HoaDon on VW\_ChiTietHoaDon.MaHoaDon = HoaDon.MaHoaDon  WHERE MaBan IS NULL and NgayGioInHoaDon IS NULL  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet LayCTHDKhachMangDiDangPhucVu()  {  return db.ExecuteQueryDataSet("select \* " +  "from Table\_FN\_LayCTHDCuaKhachMangDiDangPhucVu ()" ,  CommandType.Text, null);  } |

#### Thêm, sửa hóa đơn

|  |
| --- |
| ***Thêm hoá đơn:*** |
| CREATE PROCEDURE [dbo].[USP\_ThemHoaDon] ( @manv varchar(10), @makh varchar(11), @maban char(10))  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  INSERT INTO HoaDon(MaNV, MaKH, MaBan, TongGiaTien) VALUES  (@manv, @makh, @maban, 0);  EXEC USP\_SuaTrangThaiBan @MaBan = @maBan, @TrangThai = N'Có người'  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemHoaDon(ref string error, string MaNV,  string MaKH, string MaBan)  {  object maNV = MaNV, maKH = MaKH, maBan = MaBan;  if(MaNV == null)  {  maNV = DBNull.Value;  }  if(MaKH == null)  {  maKH = DBNull.Value;  }  if (MaBan == null)  {  maBan = DBNull.Value;  }  return db.MyExecuteNonQuery("USP\_ThemHoaDon", CommandType.StoredProcedure,  ref error, new SqlParameter("@manv", maNV),  new SqlParameter("@makh", maKH),  new SqlParameter("@maban", maBan));  } |

|  |
| --- |
| ***Sửa hoá đơn:*** |
| CREATE PROCEDURE [dbo].[USP\_CapNhatHoaDon] ( @mahoadon varchar (13), @maban char(10),  @ngaygioinhoadon DATETIME, @DiemSuDung float)  AS  BEGIN  DECLARE @tongGiaTien float  SELECT @tongGiaTien = dbo.Scalar\_FN\_LayTongSoTienCTHD(@mahoadon)  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  UPDATE dbo.HoaDon  SET MaBan = @maban, NgayGioInHoaDon = @ngaygioinhoadon, DiemSuDung = @DiemSuDung, TongGiaTien = @tongGiaTien  Where MaHoaDon = @mahoadon  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool CapNhatHoaDon(ref string error, string MaHoaDon,  string MaBan, DateTime NgayGioInHoaDon, float DiemSuDung)  {  return db.MyExecuteNonQuery("USP\_CapNhatHoaDon", CommandType.StoredProcedure,  ref error, new SqlParameter("@mahoadon", MaHoaDon),  new SqlParameter("@maban", MaBan),  new SqlParameter("@ngaygioinhoadon", NgayGioInHoaDon),  new SqlParameter("@DiemSuDung", DiemSuDung));  } |

#### Tìm kiếm chi tiết hóa đơn

|  |
| --- |
| ***Tìm kiếm chi tiết hoá đơn dựa vào mã hóa đơn:*** |
| CREATE FUNCTION Table\_FN\_TimKiemChiTietHoaDon(@MaHoaDon varchar(13))  RETURNS TABLE  RETURN(  select \*  from VW\_ChiTietHoaDon  where MaHoaDon = @MaHoaDon  )  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinCTHD(string MaHoaDon)  {  return db.ExecuteQueryDataSet("select \* from " +  "Table\_FN\_TimKiemChiTietHoaDon(" + "'" + MaHoaDon + "'" + ")",  CommandType.Text, null);  } |

#### Thêm, sửa chi tiết hóa đơn

|  |
| --- |
| ***Thêm chi tiết hoá đơn:*** |
| CREATE PROCEDURE [dbo].[USP\_ThemChiTietHoaDon] ( @mahoadon VARCHAR(13), @mamh CHAR(10), @soluongmh INT)  AS  BEGIN  DECLARE @GiaTien FLOAT, @thanhtien FLOAT  SELECT @GiaTien = GiaTien FROM MatHang WHERE MaMH = @mamh  SET @thanhtien = @GiaTien \* @soluongmh  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  INSERT INTO ChiTietHoaDon(MaHoaDon, MaMH, SoLuongMH, ThanhTien) VALUES  (@mahoadon, @mamh, @soluongmh, @thanhtien);  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemChiTietHoaDon(ref string error, string MaHoaDon, string MaMH,  int SoLuongMH)  {  return db.MyExecuteNonQuery("USP\_ThemChiTietHoaDon", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaHoaDon", MaHoaDon),  new SqlParameter("@MaMH", MaMH),  new SqlParameter("@SoLuongMH", SoLuongMH)  );  } |

|  |
| --- |
| ***Sửa chi tiết hoá đơn:*** |
| CREATE PROCEDURE [dbo].[USP\_SuaChiTietHoaDon] ( @mahoadon VARCHAR(13), @mamh CHAR(10), @soluongmh INT)  AS  BEGIN  DECLARE @GiaTien FLOAT, @thanhtien FLOAT  SELECT @GiaTien = GiaTien FROM MatHang WHERE MaMH = @mamh  SET @thanhtien = @GiaTien \* @soluongmh  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  UPDATE dbo.ChiTietHoaDon  SET SoLuongMH = @soluongmh, ThanhTien = @thanhtien  WHERE MaHoaDon = @mahoadon and MaMH = @mamh  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool CapNhatChiTietHoaDon(ref string error, string MaHoaDon, string MaMH,  int SoLuongMH)  {  return db.MyExecuteNonQuery("USP\_SuaChiTietHoaDon", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaHoaDon", MaHoaDon),  new SqlParameter("@MaMH", MaMH),  new SqlParameter("@SoLuongMH", SoLuongMH));  } |

### **Quản lý phiếu mua**

#### Lọc chi tiết mua

|  |
| --- |
| ***Function tìm kiếm chi tiết mua theo một hoặc nhiều tham số:*** |
| CREATE FUNCTION Table\_FN\_LocChiTietMua(@MaPhieuMua varchar(16), @TenNVL nvarchar(30), @NgayGioMua DATETIME)  RETURNS TABLE  RETURN(  select \*  from VW\_ChiTietMua  where MaPhieuMua = ISNULL(@MaPhieuMua, MaPhieuMua)and TenNVL LIKE '%' + ISNULL(@TenNVL, TenNVL) +'%' and  CAST(NgayGioMua as date) = ISNULL(CAST(@NgayGioMua as date), CAST(NgayGioMua as date))  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemChiTietPhieuMua(string MaPhieuMua, string TenNVL, DateTime NgayGioMua)  {  string maphieumua = "NULL", tennvl = "NULL", ngaygiomua = "NULL";  if (MaPhieuMua != null)  {  maphieumua = "'" + MaPhieuMua + "'";  }  if (TenNVL != null)  {  tennvl = "N'" + TenNVL + "'";  }  if (NgayGioMua != new DateTime(1,1,1))  {  ngaygiomua = "'" + NgayGioMua.Date.ToString("yyyy-MM-dd") + "'";  }  return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_LocChiTietMua(" + maphieumua + ", " +  tennvl + ", " + ngaygiomua + ")", CommandType.Text, null);  } |

#### Thêm, sửa, xoá chi tiết mua

|  |
| --- |
| ***Thêm chi tiết mua:*** |
| CREATE PROCEDURE [dbo].[USP\_ThemChitietMua] ( @MaPhieuMua VARCHAR(16), @MaNVL CHAR(10), @SoLuongNVL INT)  AS  BEGIN  DECLARE @ThanhTien FLOAT, @GiaMotDonVi FLOAT  SELECT @GiaMotDonVi = GiaMotDonVi FROM NguyenVatLieu WHERE MaNVL = @MaNVL  SET @ThanhTien = @GiaMotDonVi \* @SoLuongNVL  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  INSERT INTO ChiTietMua(MaPhieuMua, MaNVL, SoLuongNVL, ThanhTien) VALUES  (@MaPhieuMua, @MaNVL, @SoLuongNVL, @ThanhTien);  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemChiTietMua(ref string err, string MaPhieuMua, string MaNVl, int SoLuongNVL)  {  return db.MyExecuteNonQuery("USP\_ThemChitietMua", CommandType.StoredProcedure,  ref err, new SqlParameter("@MaPhieuMua", MaPhieuMua),  new SqlParameter("@MaNVL", MaNVl),  new SqlParameter("@SoLuongNVL", SoLuongNVL));  } |

|  |
| --- |
| ***Sửa chi tiết mua:*** |
| CREATE PROCEDURE [dbo].[USP\_SuaChiTietMua] (@MaPhieuMua VARCHAR(16), @MaNVL CHAR(10), @SoLuongNVL INT)  AS  BEGIN  DECLARE @ThanhTien FLOAT, @GiaMotDonVi FLOAT  SELECT @GiaMotDonVi = GiaMotDonVi FROM NguyenVatLieu WHERE MaNVL = @MaNVL  SET @ThanhTien = @GiaMotDonVi \* @SoLuongNVL  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  UPDATE dbo.ChiTietMua  SET SoLuongNVL = @SoLuongNVL, ThanhTien = @ThanhTien  Where MaPhieuMua = @MaPhieuMua AND MaNVL = @MaNVL  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool SuaChiTietMua(ref string error, string MaPhieuMua, string MaNVL, int SoLuongNVL)  {  return db.MyExecuteNonQuery("USP\_SuaChiTietMua", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaPhieuMua", MaPhieuMua),  new SqlParameter("@MaNVL", MaNVL),  new SqlParameter("@SoLuongNVL", SoLuongNVL));  } |

|  |
| --- |
| ***Xoá chi tiết mua:*** |
| CREATE PROCEDURE [dbo].[USP\_XoaChiTietMua] ( @MaPhieuMua VARCHAR(16), @MaNVL CHAR(10))  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  DELETE dbo.chitietmua  WHERE MaPhieuMua = @MaPhieuMua AND MaNVL = @MaNVL  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool XoaChiTietMua(ref string err, string MaPhieuMua, string MaNVL)  {  return db.MyExecuteNonQuery("USP\_XoaChiTietMua", CommandType.StoredProcedure,  ref err, new SqlParameter("@MaPhieuMua", MaPhieuMua),  new SqlParameter("@MaNVL", MaNVL));  } |

#### Tính tổng số tiền chi tiết phiếu mua của một phiếu mua

|  |
| --- |
| ***Function tính tổng số tiền chi tiết phiếu mua của một phiếu mua :*** |
| ALTER function [dbo].[Scalar\_FN\_LayTongSoTienCTPM] (@MaPhieuMua varchar(16))  RETURNS float  BEGIN  DECLARE @tongtien float  SELECT @tongtien = SUM(ThanhTien) FROM ChiTietMua WHERE MaPhieuMua = @MaPhieuMua  RETURN @tongtien  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool XoaChiTietMua(ref string err, string MaPhieuMua, string MaNVL)  {  return db.MyExecuteNonQuery("USP\_XoaChiTietMua", CommandType.StoredProcedure,  ref err, new SqlParameter("@MaPhieuMua", MaPhieuMua),  new SqlParameter("@MaNVL", MaNVL));  } |

#### Thêm phiếu mua

|  |
| --- |
| ***Thêm phiếu mua:*** |
| create procedure USP\_ThemPhieuMua @NgayGioMua datetime  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  INSERT INTO PhieuMua(NgayGioMua) VALUES  (@NgayGioMua);  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  throw  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemPhieuMua(ref string error, DateTime NgayGioMua)  {  return db.MyExecuteNonQuery("USP\_ThemPhieuMua", CommandType.StoredProcedure,  ref error, new SqlParameter("@NgayGioMua", NgayGioMua));  } |

### **Danh mục sản phẩm bán chạy**

|  |
| --- |
| ***Function Đưa ra danh sách sản phẩm bán chạy dựa vào tháng năm*** |
| CREATE FUNCTION [dbo].[Table\_FN\_LocSanPhamBanChay](@thang int, @nam int)  RETURNS TABLE AS  RETURN  (  SELECT TOP(10)  TenHang,  TenLoaiHang,  SoLuong,  Thang,  Nam  FROM VW\_DanhMucSanPhamBanChay  WHERE Thang = @thang and Nam = @nam  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemSanPhamBanChay(string Thang, string Nam)  {  string thang = null, nam = null;  thang = "'" + Thang + "'";  nam = "'" + Nam + "'";  return db.ExecuteQueryDataSet("select \* from " +  "Table\_FN\_LocSanPhamBanChay(" + thang + ", " + nam + ")", CommandType.Text, null);  } |

### **Tính toán doanh thu**

|  |
| --- |
| ***Function Tính toán doanh thu*** |
| CREATE FUNCTION [dbo].[Scalar\_FN\_TinhTongTienDoanhThu] (@year int, @month int)  RETURNS FLOAT  AS  BEGIN  DECLARE @TongTien float, @TongTienHoaDon float, @TongDiemSuDung float, @TongTienPhieuMua float, @TongTienLuong float;    SET @TongTienHoaDon = 0;  SET @TongDiemSuDung = 0;  SET @TongTienPhieuMua = 0;  SET @TongTienLuong = 0;  SET @TongTien = 0;  SELECT @TongTienHoaDon = dbo.Scalar\_FN\_TinhTongTienHoaDonTheoNgayThang(@year, @month)  SELECT @TongDiemSuDung = dbo.Scalar\_FN\_TinhTongDTLSuDung(@year, @month)  SELECT @TongTienPhieuMua = dbo.Scalar\_FN\_TinhTongTienPhieuMuaTheoThangNam(@year, @month)  SELECT @TongTienLuong = dbo.Scalar\_FN\_TinhTongTienLuongTheoThangNam(@year, @month)  SET @TongTien = @TongTienHoaDon - @TongDiemSuDung \* 1000 - @TongTienPhieuMua - @TongTienLuong  RETURN @TongTien  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public float TinhTongDoanhThu(int Year, int Month)  {  return db.MyExecuteScalar<float>  ("SELECT dbo.Scalar\_FN\_TinhTongTienDoanhThu(" + Year + ", " + Month + ")",  CommandType.Text);  } |

## **Quản lý khách hàng**

### **Lọc thông tin khách hàng**

|  |
| --- |
| ***Lọc thông tin khách hàng:*** |
| CREATE FUNCTION [dbo].[Table\_FN\_LocThongTinKhachHang](@makh char(11), @tenkh nvarchar(40), @tenloaikh nvarchar(30) )  RETURNS TABLE AS  RETURN  (  SELECT MaKH, TenLoaiKH, TenKH, DiaChi, SDT, DiemTichLuyHienTai, TongDiemTichLuy  FROM VW\_ThongTinKhachHang  WHERE MaKH = ISNULL(@makh, MaKH) and TenKH = ISNULL(@tenkh, TenKH) and  TenLoaiKH = ISNULL(@tenloaikh, TenLoaiKH)  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinKhachHang(string MaKH, string TenKH, string TenLoaiKH)  {  string maKH = "NULL", tenKH = "NULL", tenLoaiKH = "NULL";  if (MaKH != null)  {  maKH = "'" + MaKH + "'";  }  if (TenKH != null)  {  tenKH = "N'" + TenKH + "'";  }  if (TenLoaiKH != null)  {  tenLoaiKH = "N'" + TenLoaiKH + "'";  }  return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_LocThongTinKhachHang(" + maKH + ", "+  tenKH + ", " + tenLoaiKH + ")",  CommandType.Text, null);  } |

### **Thêm, sửa khách hàng**

|  |
| --- |
| ***Thêm khách hàng:*** |
| CREATE PROCEDURE [dbo].[USP\_ThemKhachHang] ( @maloaikh char(10), @tenkh nvarchar(40), @diachi nvarchar(200), @sdt char(10), @currentpoint float, @totalpoint float )  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  INSERT INTO KhachHang(MaLoaiKH, TenKH, DiaChi, SDT, DiemTichLuyHienTai, TongDiemTichLuy) VALUES (  @maloaikh, @tenkh, @diachi, @sdt, @currentpoint, @totalpoint  );  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemKhachHang(ref string error, string MaLoaiKH,  string TenKH, string DiaChi, string SDT, float DiemTichLuyHienTai,  float TongDiemTichLuy)  {  return db.MyExecuteNonQuery("USP\_ThemKhachHang", CommandType.StoredProcedure,  ref error, new SqlParameter("@maloaikh", MaLoaiKH),  new SqlParameter("@tenkh", TenKH),  new SqlParameter("@diachi", DiaChi),  new SqlParameter("@sdt", SDT),  new SqlParameter("@currentpoint", DiemTichLuyHienTai),  new SqlParameter("@totalpoint", TongDiemTichLuy));  } |

|  |
| --- |
| ***Sửa khách hàng:*** |
| CREATE PROCEDURE [dbo].[USP\_SuaKhachHang] ( @makh char(11), @tenkh nvarchar(40), @maloaikh char(10), @diachi nvarchar(200), @sdt char(10),  @currentpoint float, @totalpoint float )  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRAN  BEGIN TRY  --  UPDATE KhachHang  SET TenKH = @tenkh, MaLoaiKH = @maloaikh, DiaChi = @diachi, SDT = @sdt, DiemTichLuyHienTai = @currentpoint,  TongDiemTichLuy = @totalpoint  WHERE MaKH = @makh;  --  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool CapNhatKhachHang(ref string error, string MaKH, string TenKH,  string MaLoaiKH, string DiaChi, string SDT, float DiemTichLuyHienTai,  float TongDiemTichLuy)  {  return db.MyExecuteNonQuery("USP\_SuaKhachHang", CommandType.StoredProcedure,  ref error, new SqlParameter("@makh", MaKH),  new SqlParameter("@tenkh", TenKH),  new SqlParameter("@maloaikh", MaLoaiKH),  new SqlParameter("@diachi", DiaChi),  new SqlParameter("@sdt", SDT),  new SqlParameter("@currentpoint", DiemTichLuyHienTai),  new SqlParameter("@totalpoint", TongDiemTichLuy));  } |

## **Quản lý ca làm**

### **Quản lý ca làm**

#### Thêm ca làm

|  |
| --- |
| ***Thêm ca làm:*** |
| CREATE PROCEDURE USP\_ThemCaLam @MaLoaiCa char(10), @NgayLam date  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRAN  INSERT INTO CaLam VALUES(@MaLoaiCa, @NgayLam)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemCaLam(ref string error, string MaLoaiCa, DateTime NgayLam)  {  return db.MyExecuteNonQuery("USP\_ThemCaLam", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaLoaiCa", MaLoaiCa),  new SqlParameter("@NgayLam", NgayLam.Date));  } |

#### Lọc ca làm

|  |
| --- |
| ***Lọc ca làm:*** |
| CREATE FUNCTION [dbo].[Table\_FN\_LocCaLam](@MaLoaiCa char(10), @NgayLam date)  RETURNS TABLE  RETURN(  select \*  from VW\_CaLam  where MaLoaiCa = ISNULL(@MaLoaiCa, MaLoaiCa) and NgayLam = ISNULL(@NgayLam, NgayLam)  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinCaLam(string MaLoaiCa, DateTime NgayLam)  {  string maLoaiCa = "NULL", ngayLam = "NULL";  if (MaLoaiCa != null)  {  maLoaiCa = "'" + MaLoaiCa + "'";  }  if (NgayLam != new DateTime(1,1,1))  {  ngayLam = "'" + NgayLam.ToString("yyyy-MM-dd") + "'" ;  }    return db.ExecuteQueryDataSet  ("select \* from dbo.Table\_FN\_LocCaLam(" + maLoaiCa + ", " + ngayLam + ")",  CommandType.Text, null);  } |

### **Phân công**

#### Lọc phân công

|  |
| --- |
| ***Lọc phân công:*** |
| CREATE FUNCTION Table\_FN\_LocPhanCong(@MaNV varchar(10), @TenLoaiCa nvarchar(30), @NgayLam date)  RETURNS TABLE  AS  RETURN  SELECT \*  FROM VW\_PhanCong  WHERE  MaNV = ISNULL(@MaNV, MaNV) and TenLoaiCa = ISNULL(@TenLoaiCa, TenLoaiCa) and  NgayLam = ISNULL(@NgayLam, NgayLam);  GO |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinPhanCong(string MaNV, string TenLoaiCa,  DateTime NgayLam)  {  string maNV = "NULL", tenLoaiCa = "NULL", ngayLam = "NULL";  if (MaNV != null)  {  maNV = "'" + MaNV + "'";  }  if (TenLoaiCa != null)  {  tenLoaiCa = "N'" + TenLoaiCa + "'";  }  if (NgayLam != new DateTime(1, 1, 1))  {  ngayLam = "'" + NgayLam.ToString("yyyy-MM-dd") + "'";  }  return db.ExecuteQueryDataSet("select \* from Table\_FN\_LocPhanCong(" +  maNV + "," + tenLoaiCa + "," + ngayLam + ")", CommandType.Text,  null);  } |

#### Thêm phân công

|  |
| --- |
| ***Thêm phân công:*** |
| CREATE PROCEDURE USP\_ThemPhanCong @MaNV varchar(10), @MaCa varchar(11)  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRAN  INSERT INTO PhanCong VALUES(@MaNV, @MaCa, 0)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemPhanCong(ref string error, string MaNV, string MaCa)  {  return db.MyExecuteNonQuery("USP\_ThemPhanCong", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV),  new SqlParameter("@MaCa", MaCa));  } |

#### Điểm danh phân công

|  |
| --- |
| ***Điểm danh phân công:*** |
| CREATE procedure [dbo].[USP\_DiemDanhPhanCong] @MaNV varchar(10), @MaLoaiCa char(10), @NgayLam date, @DiemDanh int  as  begin  declare @MaCa varchar(11)  set @MaCa = 'CL' + RIGHT(RTRIM(@MaLoaiCa), 1) + convert(varchar(8),cast(@NgayLam as date),112)  set xact\_abort on  begin try  begin tran  update PhanCong  set DiemDanh = @DiemDanh  where MaNV = @MaNV and @MaCa = MaCa  commit tran  end try  begin catch  rollback;  throw  end catch  end |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool DiemDanhPhanCong(ref string error, string MaNV, string MaLoaiCa,  DateTime NgayLam, int DiemDanh)  {  return db.MyExecuteNonQuery("USP\_DiemDanhPhanCong", CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNV", MaNV),  new SqlParameter("@MaLoaiCa", MaLoaiCa),  new SqlParameter("@NgayLam", NgayLam),  new SqlParameter("@DiemDanh", DiemDanh));  } |

## **Quản lý kho**

### **Lưu trữ**

|  |
| --- |
| ***Tìm kiếm thông tin kho lưu trữ:*** |
| CREATE function [dbo].[Table\_FN\_LocKhoLuuTru](@NgayThang date, @TenNVL nvarchar(30))  RETURNS TABLE  RETURN(  SELECT \* FROM VW\_LuuTruKho  WHERE NgayThang = ISNULL(@NgayThang, NgayThang) and  TenNVL LIKE '%' + ISNULL(@TenNVL, TenNVL) + '%'  ) |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public DataSet TimKiemThongTinKhoLuuTru(DateTime NgayThang, string TenNVL)  {  string ngayThang = "NULL", tenNVL = "NULL";  if (NgayThang != new DateTime(1, 1, 1))  {  ngayThang = "'" + NgayThang.Date.ToString("yyyy-MM-dd") + "'";  }  if (TenNVL != null)  {  tenNVL = "N'" + TenNVL + "'";  }  return db.ExecuteQueryDataSet  ("select \* from Table\_FN\_LocKhoLuuTru(" + ngayThang + ", " +  tenNVL + ")",  CommandType.Text, null);  } |

### **Nhập kho**

|  |
| --- |
| ***Thao tác nhập kho*** |
| CREATE procedure [dbo].[USP\_ThaoTacNhapKho] @MaNVL char(10), @SoLuongNhap int, @GioNhap time(0), @NgayThang date, @HSD date  AS  BEGIN  DECLARE @MaKho varchar(11)  SET XACT\_ABORT ON  BEGIN TRY  --Nếu chưa tồn tại => thêm kho mới  IF NOT EXISTS (SELECT 1 FROM Kho WHERE NgayThang = @NgayThang)  BEGIN  BEGIN TRY  BEGIN TRAN  INSERT INTO Kho(NgayThang) VALUES (@NgayThang)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END  --  BEGIN TRAN  SELECT @MaKho = dbo.Scalar\_FN\_LayMaKhoTuNgayThang(@NgayThang)  INSERT INTO NhapKho VALUES (@MaNVL, @MaKho, @SoLuongNhap, @GioNhap)  -- Nếu đã có nguyên vật liệu đó ở trong mã kho ngày hôm nay  -- thì thêm số lượng vào số lượng còn lại trong kho  IF EXISTS (SELECT 1 FROM LuuTru WHERE MaKho = @MaKho and MaNVL = @MaNVL)  BEGIN  UPDATE LuuTru  SET SoLuongConLai += @SoLuongNhap  WHERE MaKho = @MaKho and MaNVL = @MaNVL  END  -- Ngược lại thêm vào lưu trữ  ELSE  BEGIN  INSERT INTO LuuTru VALUES (@MaNVL, @MaKho, @HSD, @SoLuongNhap)  END  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThaoTacNhapKho(ref string error, string MaNVL,  int SoLuong, DateTime GioNhap, DateTime NgayThang, string HSD)  {  string gioNhap = GioNhap.ToString("HH:mm:ss");  string ngayThang = NgayThang.ToString("yyyy-MM-dd");  return db.MyExecuteNonQuery("USP\_ThaoTacNhapKho",  CommandType.StoredProcedure,  ref error, new SqlParameter("@MaNVL", MaNVL),  new SqlParameter("@SoLuongNhap", SoLuong),  new SqlParameter("@GioNhap", gioNhap),  new SqlParameter("@NgayThang", ngayThang),  new SqlParameter("@HSD", HSD));  } |

### **Xuất kho**

|  |
| --- |
| ***Thao tác xuất kho*** |
| ALTER PROCEDURE [dbo].[USP\_ThaoTacXuatKho] @MaNVL char(10), @MaKho varchar(11), @SoLuongXuat int, @NgayGioXuat datetime  AS  BEGIN  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRAN  --  INSERT INTO XuatKho VALUES(@MaNVL, @MaKho, @SoLuongXuat, @NgayGioXuat)  --  UPDATE LuuTru  SET SoLuongConLai -= @SoLuongXuat  WHERE MaKho = @MaKho AND MaNVL = @MaNVL  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThaoTacXuatKho(ref string err, string MaNVL, string MaKho,  int SoLuongXuat, DateTime NgayGioXuat)  {  string ngayGioXuat = NgayGioXuat.ToString("yyyy-MM-dd HH:mm:ss");  return db.MyExecuteNonQuery("USP\_ThaoTacXuatKho",  CommandType.StoredProcedure, ref err,  new SqlParameter("@MaNVL", MaNVL),  new SqlParameter("@MaKho", MaKho),  new SqlParameter("@SoLuongXuat", SoLuongXuat),  new SqlParameter("@NgayGioXuat", ngayGioXuat));  } |

## **Quản lý thực đơn**

### **Thêm món**

|  |
| --- |
| ***Thêm món*** |
| CREATE PROCEDURE [dbo].[USP\_ThemMatHang] @MaMH char(10), @MaLoaiMH char(10), @TenHang nvarchar(30), @GiaTien float  AS  BEGIN  BEGIN TRY  BEGIN TRAN  INSERT INTO MatHang VALUES(@MaMH, @MaLoaiMH, @TenHang, @GiaTien)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool ThemMatHang(ref string err, string MaMH, string MaLoaiMH,  string TenHang, float GiaTien)  {  return db.MyExecuteNonQuery  ("USP\_ThemMatHang", CommandType.StoredProcedure, ref err,  new SqlParameter("@MaMH", MaMH),  new SqlParameter("@MaLoaiMH", MaLoaiMH),  new SqlParameter("@TenHang", TenHang),  new SqlParameter("@GiaTien", GiaTien));  } |

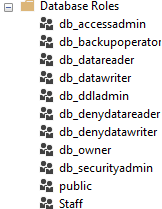
### **Chỉnh sửa món**

|  |
| --- |
| ***Sửa món*** |
| CREATE PROCEDURE [dbo].[USP\_SuaMatHang] @MaMH char(10), @MaLoaiMH char(10), @TenHang nvarchar(30), @GiaTien float  AS  BEGIN  BEGIN TRY  BEGIN TRAN  UPDATE MatHang  SET MaLoaiMH = @MaLoaiMH, TenHang = @TenHang, GiaTien = @GiaTien  WHERE MaMH = @MaMH  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |
| ***Code C#*** |
| ***Tầng Bussiness Access*** |
| public bool SuaMatHang(ref string err, string MaMH, string MaLoaiMH,  string TenHang, float GiaTien)  {  return db.MyExecuteNonQuery  ("USP\_SuaMatHang", CommandType.StoredProcedure, ref err,  new SqlParameter("@MaMH", MaMH),  new SqlParameter("@MaLoaiMH", MaLoaiMH),  new SqlParameter("@TenHang", TenHang),  new SqlParameter("@GiaTien", GiaTien));  } |

# **CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

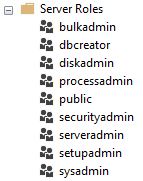
## **Tạo Role và phân quyền cho nhân viên trong SQL**

|  |
| --- |
| CREATE ROLE Staff  --Gán quyền thao tác trên các bảng cho Nhân viên  GRANT SELECT, REFERENCES ON Ban TO Staff  GRANT SELECT, REFERENCES ON CaLam TO Staff  GRANT SELECT, REFERENCES ON ChiTietHoaDon TO Staff  GRANT SELECT, REFERENCES ON ChiTietLuongNgay TO Staff  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON ChiTietMua TO Staff  GRANT SELECT, REFERENCES ON ChucVu TO Staff  GRANT SELECT, REFERENCES ON DangNhap TO Staff  GRANT SELECT, INSERT, UPDATE, REFERENCES ON HoaDon TO Staff  GRANT SELECT, INSERT, UPDATE, REFERENCES ON KhachHang TO Staff  GRANT SELECT, REFERENCES ON DangNhap TO Staff  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON Kho TO Staff  GRANT SELECT, REFERENCES ON LoaiCa TO Staff  GRANT SELECT, REFERENCES ON LoaiKhachHang TO Staff  GRANT SELECT, REFERENCES ON LoaiMatHang TO Staff  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON LuuTru TO Staff  GRANT SELECT, REFERENCES ON NguyenVatLieu TO Staff  GRANT SELECT, REFERENCES ON NhanVien TO Staff  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON NhapKho TO Staff  GRANT SELECT, REFERENCES ON NhomNguoiDung TO Staff  GRANT SELECT, REFERENCES ON PhanCong TO Staff  GRANT SELECT, REFERENCES ON PhanNhom TO Staff  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON PhieuMua TO Staff  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON XuatKho TO Staff  -- Gán quyền thực thi các thủ tục và hàm cho Nhân viên  GRANT EXECUTE TO Staff  GRANT SELECT TO Staff  /\* Không cho thực thi thủ tục trên Ca làm \*/  DENY EXECUTE ON USP\_SuaCaLam to Staff;  DENY EXECUTE ON USP\_ThemCaLam to Staff;  DENY EXECUTE ON USP\_XoaCaLam to Staff;  /\* Không cho thực thi thủ tục trên Chi tiết lương \*/  DENY EXECUTE ON usp\_SuaChiTietLuong to Staff;  DENY EXECUTE ON usp\_ThemChiTietLuong to Staff;  DENY EXECUTE ON usp\_XoaChiTietLuong to Staff;  /\* Không cho thực thi thủ tục trên Chi tiết mua \*/  DENY EXECUTE ON USP\_SuaChiTietMua to Staff;  DENY EXECUTE ON USP\_ThemChiTietMua to Staff;  DENY EXECUTE ON USP\_XoaChiTietMua to Staff;  /\* Không cho thực thi thủ tục trên Chức vụ \*/  DENY EXECUTE ON usp\_SuaChucVu to Staff;  DENY EXECUTE ON usp\_ThemChucVu to Staff;  /\* Không cho thực thi thủ tục trên Đăng nhập \*/  DENY EXECUTE ON USP\_DoiMatKhau to Staff;  DENY EXECUTE ON USP\_GanQuyenTaiKhoan to Staff;  DENY EXECUTE ON USP\_DoiMatKhau to Staff;  DENY EXECUTE ON USP\_ThemTaiKhoan to Staff;  /\* Không cho thực thi thủ tục trên Nhân viên \*/  DENY EXECUTE ON usp\_SuaNhanVien to Staff;  DENY EXECUTE ON usp\_ThemNhanVien to Staff;  DENY EXECUTE ON usp\_XoaNhanVien to Staff;  /\* Không cho thực thi thủ tục trên Phân công \*/  DENY EXECUTE ON USP\_ThemPhanCong to Staff;  DENY EXECUTE ON USP\_XoaPhanCong to Staff;  DENY EXECUTE ON USP\_DiemDanhPhanCong to Staff; |



## **Role của người quản lý**

Người quản lý sẽ có toàn quyền với hệ thống và có Role là “sysadmin”. Sysadmin là một Server Role được định nghĩa sẵn trong SQL Server.



## **Các trigger và thủ tục được sử dụng để tạo, xóa User và gán quyền**

### **Trigger**

|  |
| --- |
| ***Trigger Tạo tài khoản trong SQL sau khi có một tài khoản được tạo trong bảng đăng nhập:*** |
| CREATE TRIGGER TRG\_AfterInsert\_TaoTaiKhoanSQLSauKhiThemDangNhap ON DangNhap  AFTER INSERT  AS  DECLARE @TenNguoiDung nvarchar(30), @MatKhau nvarchar(16)  SELECT @TenNguoiDung = ne.TenNguoiDung, @MatKhau = ne.MatKhau  FROM inserted ne  DECLARE @sqlString nvarchar(max)  BEGIN  /\* Tạo tài khoản login trên cơ sở dữ liệu cho nhân viên với tên người dùng và tài khoản  được truyền vào\*/  SET @sqlString = 'CREATE LOGIN [' + @TenNguoiDung +'] WITH PASSWORD='''+ @MatKhau  +''', DEFAULT\_DATABASE=[QUANLYQUANTRASUA], CHECK\_EXPIRATION=OFF,  CHECK\_POLICY=OFF'  EXEC (@sqlString)  -- Tạo tài khoản người dùng đối với nhân viên đó trên database (tên ngườidùng trùng với tên login)  SET @sqlString= 'CREATE USER ' + @TenNguoiDung +' FOR LOGIN '+ @TenNguoiDung  EXEC (@sqlString)  END |

|  |
| --- |
| ***Trigger xoá nhân viên thì thay đổi trạng thái làm việc của nhân viên đó và xóa tài khoản của nhân viên đó:*** |
| create trigger TRG\_InsteadDelete\_XoaNhanVien ON NhanVien  Instead of DELETE  AS  DECLARE @MaNV varchar(10)  SELECT @MaNV=ol.MaNV  FROM deleted ol  BEGIN  --Chuyển đổi trạng thái làm việc của nhân viên thành 'Đã nghỉ'  UPDATE NhanVien Set TinhTrang = N'Đã nghỉ' WHERE MaNV = @MaNV  -- Xóa Nhân viên khỏi bảng phân nhóm  DELETE FROM PhanNhom WHERE MaNV = @MaNV  -- Xóa tài khoản SQL  EXEC USP\_XoaTaiKhoanNhanVien @MaNV  -- Xóa tài khoản khỏi bảng đăng nhập  DELETE FROM DangNhap WHERE MaNV = @MaNV  END |

### **Thủ tục**

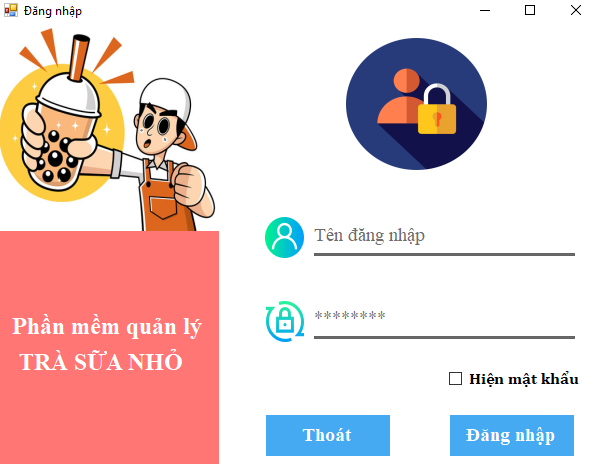
|  |
| --- |
| ***Procedure Đổi mật khẩu*** |
| CREATE PROCEDURE USP\_DoiMatKhau @TenNguoiDung nvarchar(30), @MatKhau nvarchar(16)  AS  BEGIN  SET XACT\_ABORT ON  DECLARE @sqlString nvarchar(max)  SET @sqlString = 'ALTER LOGIN [' + @TenNguoiDung + '] WITH PASSWORD=''' + @MatKhau +''''  BEGIN TRY  BEGIN TRAN  UPDATE DangNhap  SET MatKhau = @MatKhau  WHERE TenNguoiDung = @TenNguoiDung  EXEC(@sqlString)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW  END CATCH  END |

|  |
| --- |
| ***Procedure Xóa tài khoản nhân viên*** |
| create procedure USP\_XoaTaiKhoanNhanVien @MaNV varchar(10)  as  begin  SET XACT\_ABORT ON  declare @TenNguoiDung varchar(20), @stringSQL nvarchar(max), @SID smallint, @sqlKill nvarchar(max);  select @TenNguoiDung = TenNguoiDung from DangNhap where MaNV = @MaNV  begin try  SELECT @SID = session\_id FROM sys.dm\_exec\_sessions WHERE login\_name = @TenNguoiDung  SET @sqlKill = 'KILL ' + CAST(@SID as varchar(4))  EXEC (@sqlKill)  begin tran  set @stringSQL = 'drop user ['+@TenNguoiDung+']';  EXEC(@stringSQL);  set @stringSQL = 'drop login ['+@TenNguoiDung+']';  EXEC(@stringSQL);  commit tran  end try  begin catch  rollback;  throw  end catch  end |

|  |
| --- |
| ***Procedure Gán quyền cho tài khoản*** |
| CREATE PROCEDURE USP\_GanQuyenTaiKhoan @TenNguoiDung nvarchar(30), @MaNhomNguoiDung varchar(10)  AS  BEGIN  DECLARE @sqlString nvarchar(max), @TenRole varchar(30)  SELECT @TenRole = TenRole FROM NhomNguoiDung WHERE MaNhomNguoiDung = @MaNhomNguoiDung  SET XACT\_ABORT ON  BEGIN TRY  BEGIN TRAN  IF @TenRole LIKE 'sysadmin'  BEGIN  SET @sqlString = 'ALTER SERVER ROLE sysadmin' +' ADD MEMBER ' + @TenNguoiDung;  END  ELSE  BEGIN  SET @sqlString = 'ALTER ROLE '+ @TenRole +' ADD MEMBER ' + @TenNguoiDung;  END  EXEC(@sqlString)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK;  THROW;  END CATCH  END |

# **CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG**

## **Form Đăng nhập**



**Khi đăng nhập sai**



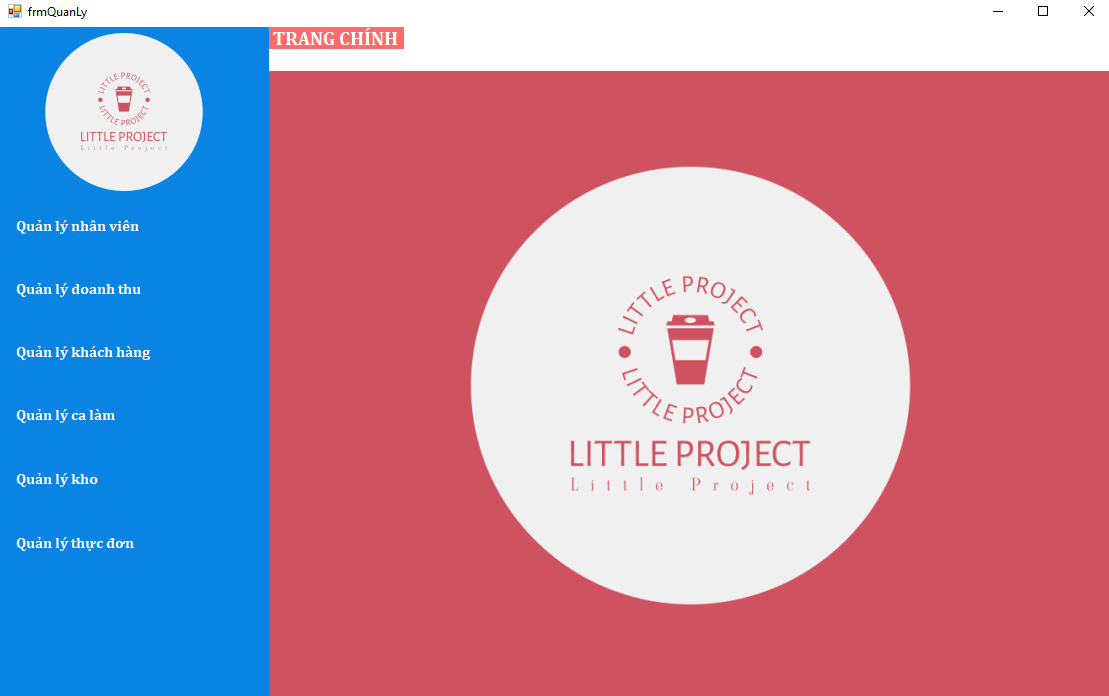
## **Form Thông tin cá nhân của người đăng nhập**

Khi tài khoản mật khẩu đăng nhập là đúng, ứng dụng sẽ xuất hiện Thông tin cá nhân của người đăng nhập



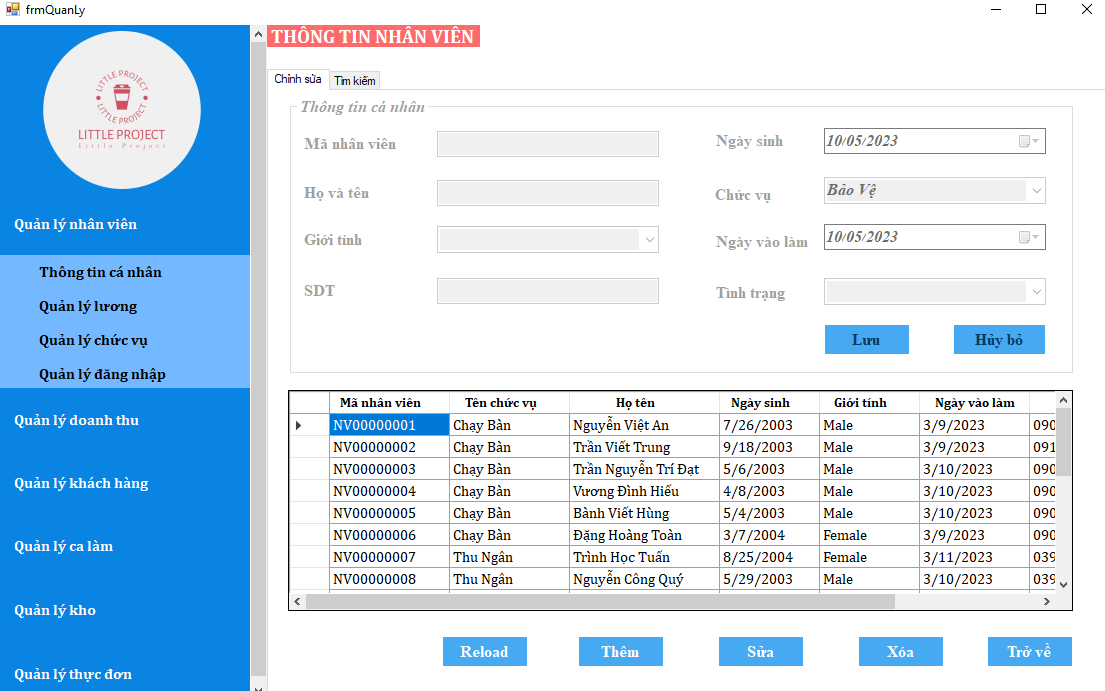
## **Form quản lý**

### **Form Home**



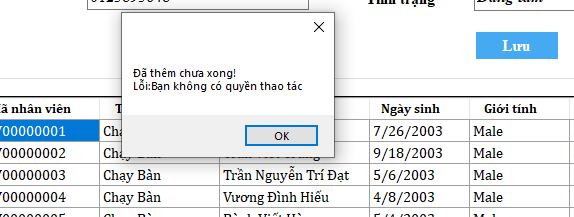
### **Form quản lý nhân viên**

#### *Form chỉnh sửa thông tin nhân viên, thêm, xóa nhân viên*

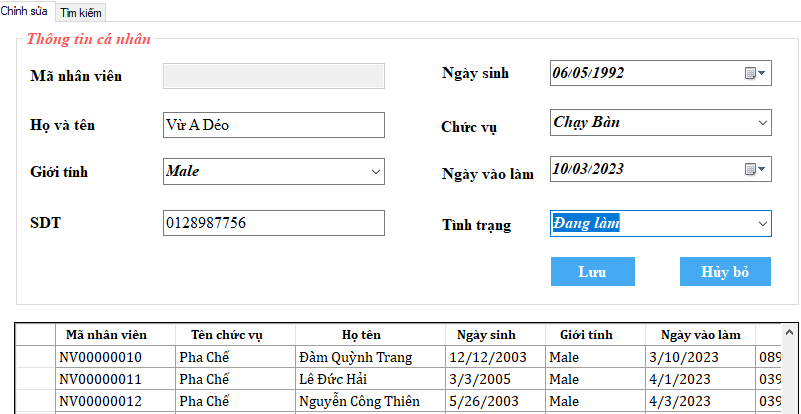


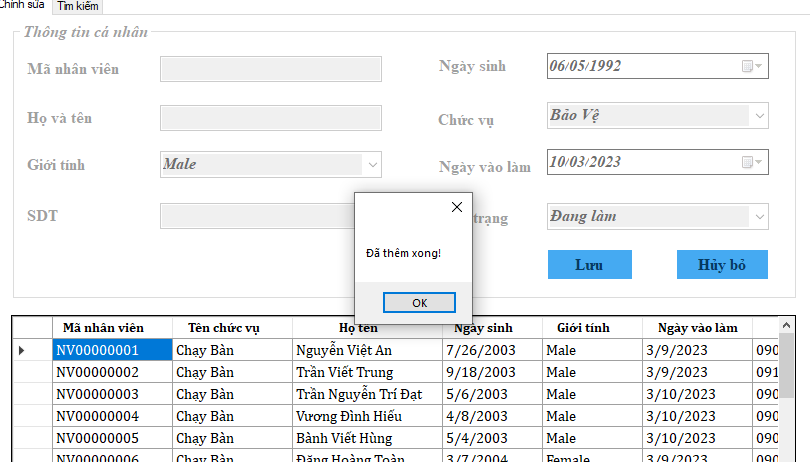
* Khi bấm thêm nhân viên mới

Nếu tài khoản đang đăng nhập là nhân viên thì sẽ không thể thêm và ứng dụng sẽ báo lỗi như sau:



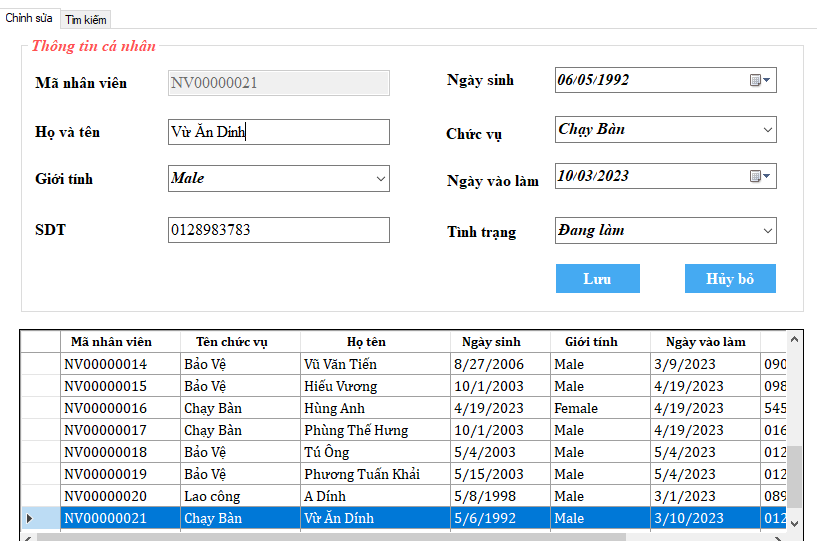
Ngược lại, nếu là quản lý thì sẽ báo thêm thành công nếu không bỏ sót thông tin cần thiết hoặc nhập



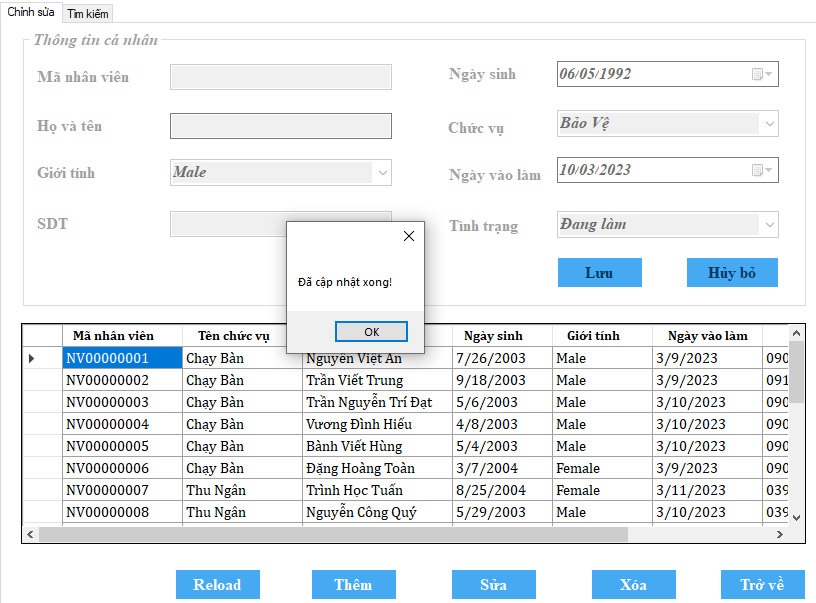


* Khi bấm nút Sửa

Bấm vào hàng có thông tin nhân viên cần chỉnh sửa sau đó bấm nút “Sửa”

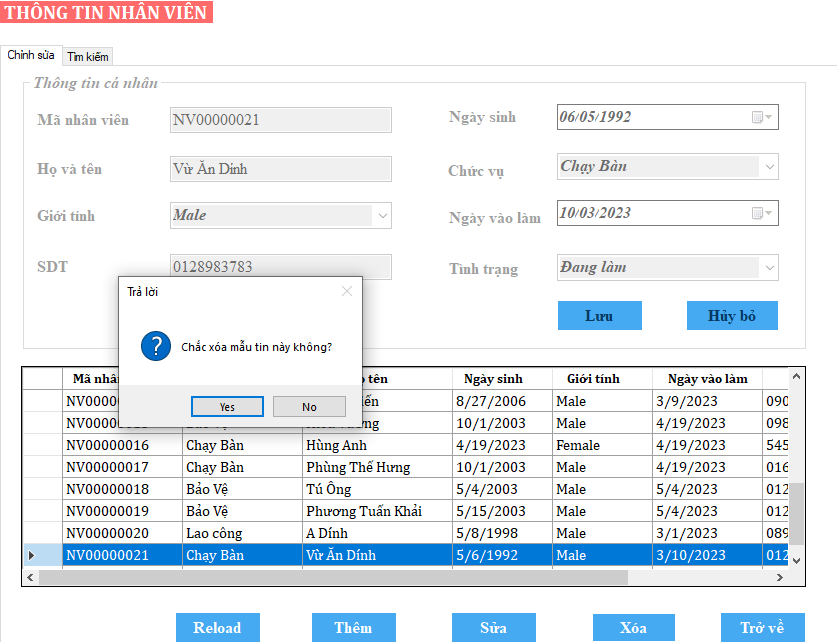


Sau khi chỉnh sửa xong, bấm Lưu sẽ xuất hiện thông báo

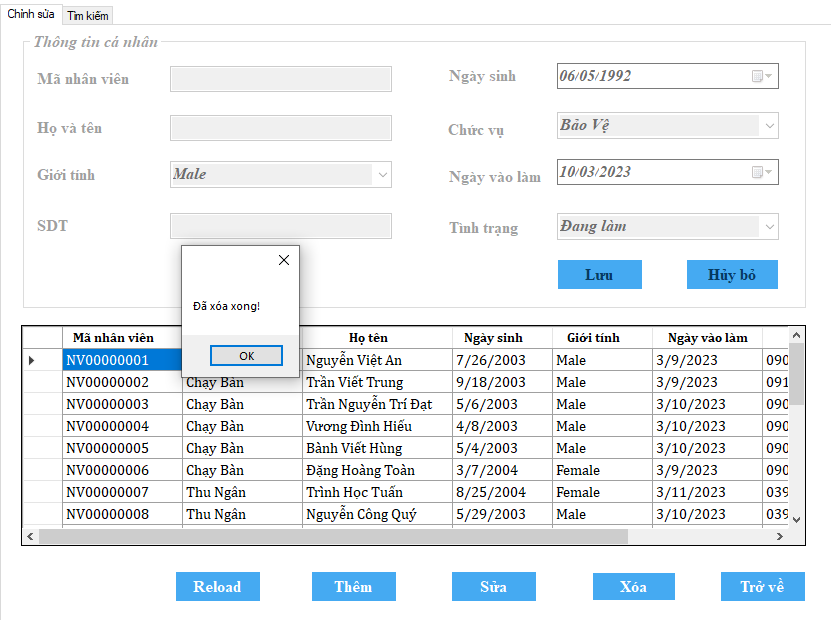


* Khi bấm nút xóa

Chọn vào hàng có thông tin nhân viên cần xóa.

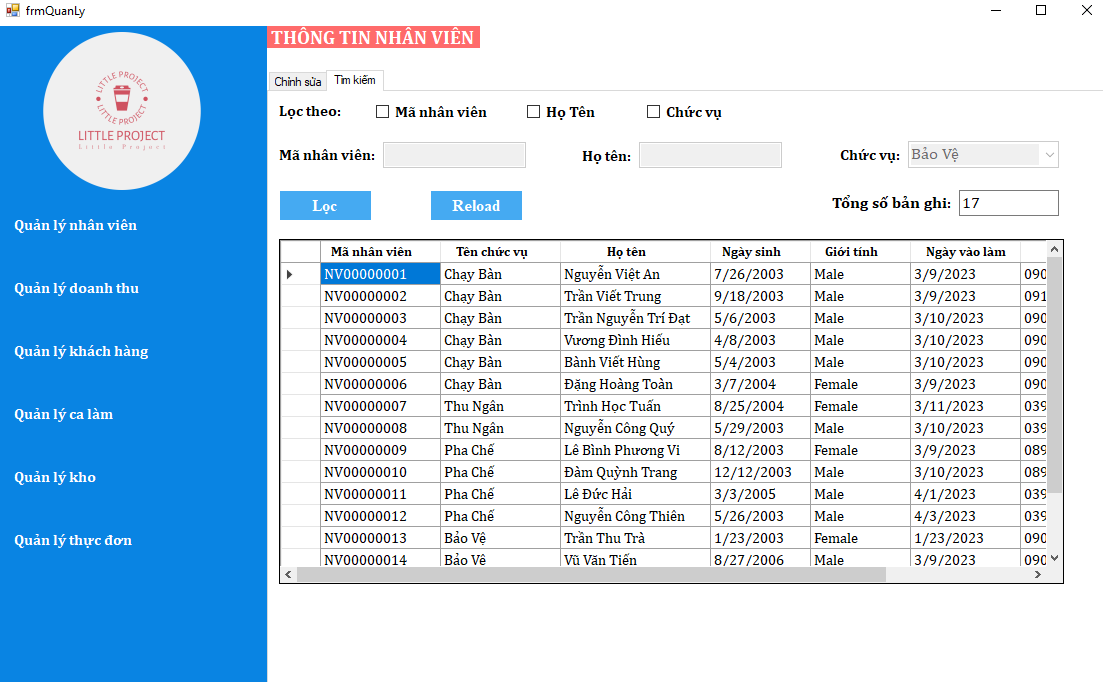


Chọn Yes



Tình trạng làm việc của nhân viên sẽ được đổi thành “Đã nghỉ”. Nếu nhân viên đó có tài khoản thì sẽ tiến hành xóa tài khoản của nhân viên ra khỏi cơ sở dữ liệu.

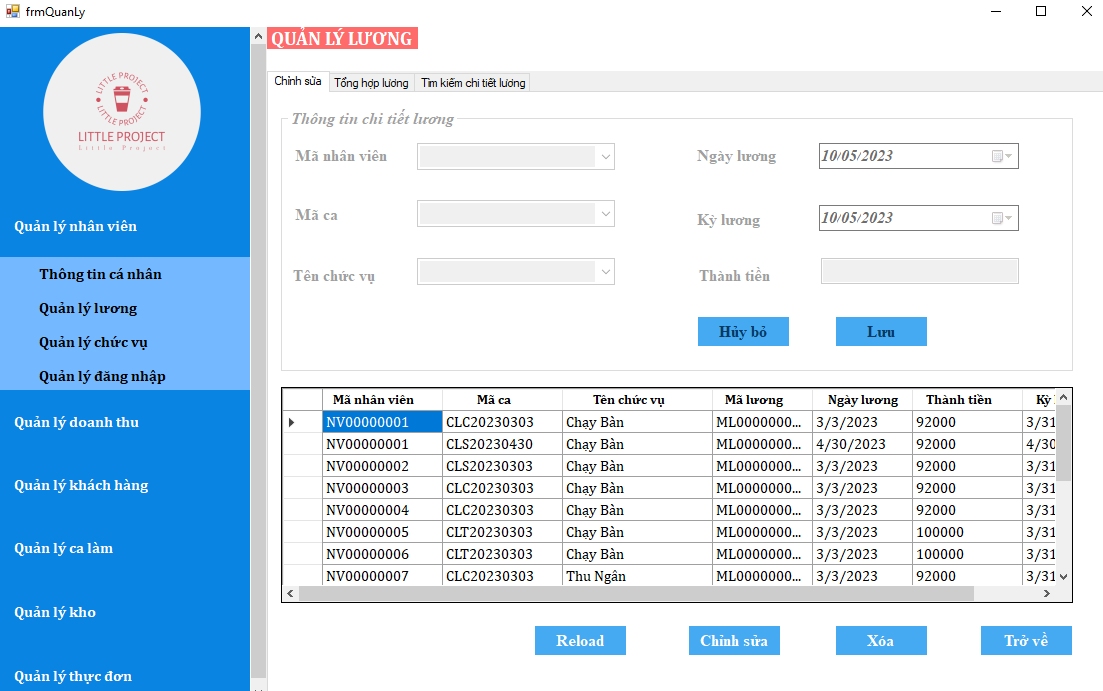
#### *Form tìm kiếm thông tin nhân viên*



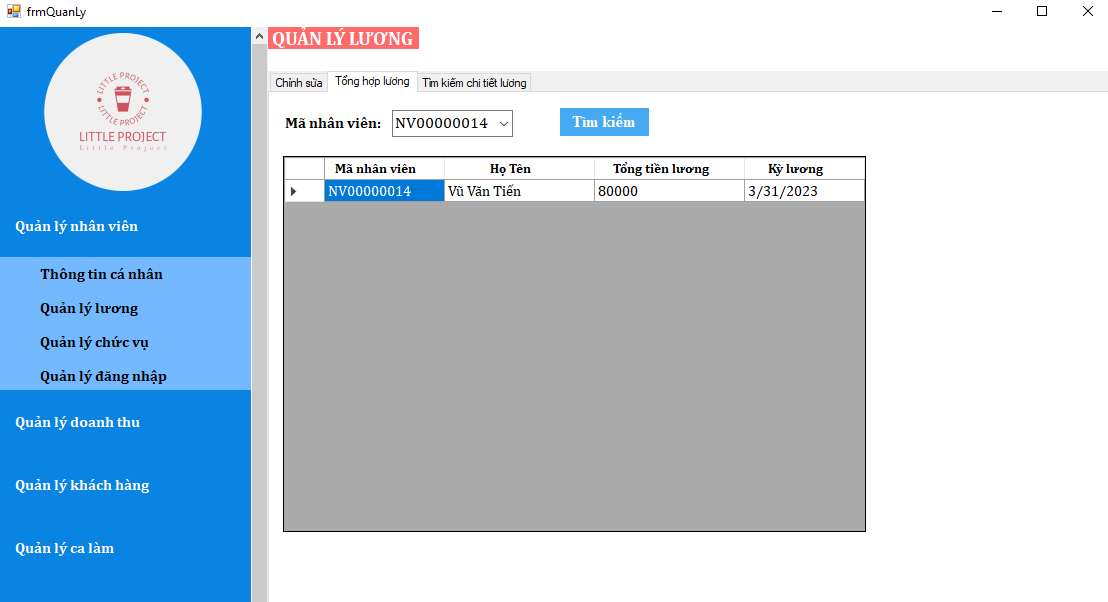
Có thể chọn tìm kiếm theo một hoặc nhiều thuộc tính mà các ô checkbox biểu diễn

#### Form quản lý lương nhân viên

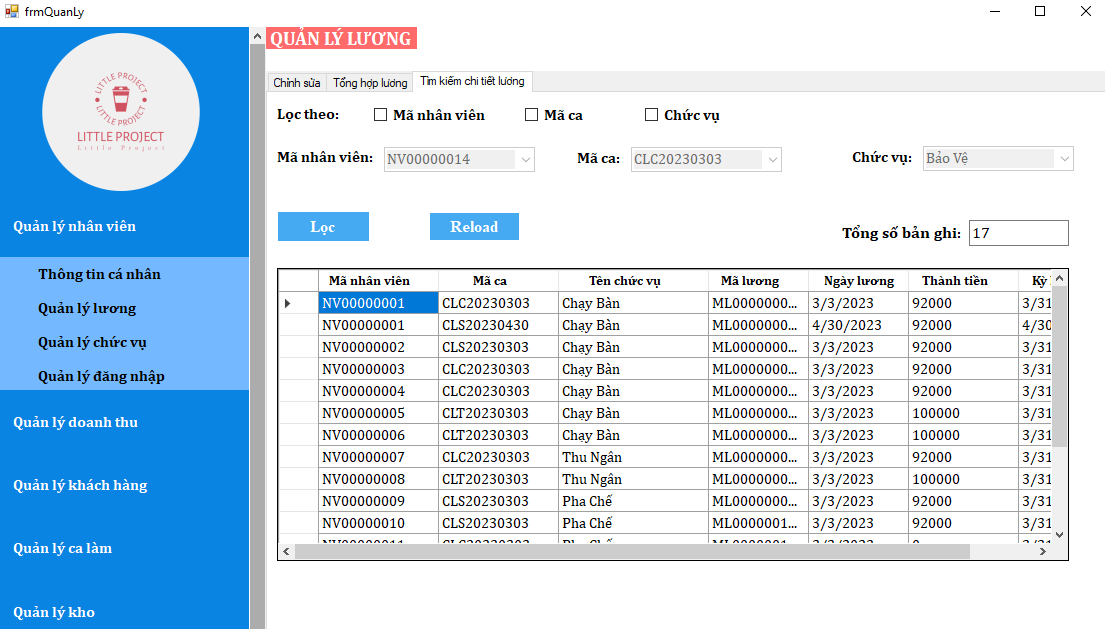
* Tab Chỉnh sửa



* Tab Tổng hợp lương

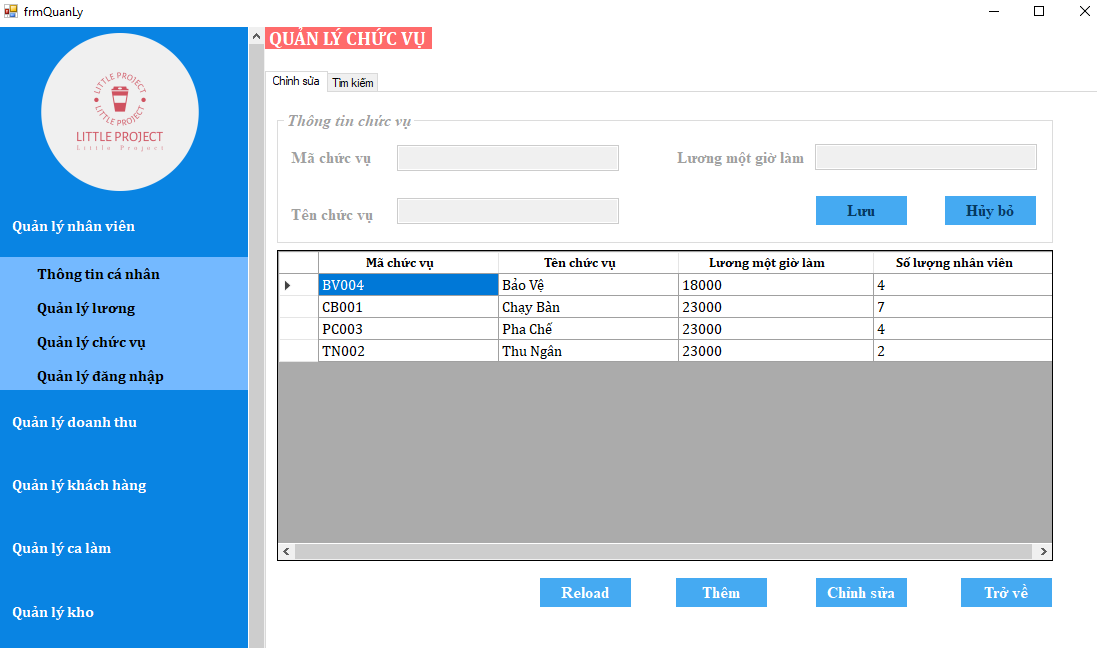


* Tab Tìm kiếm chi tiết lương

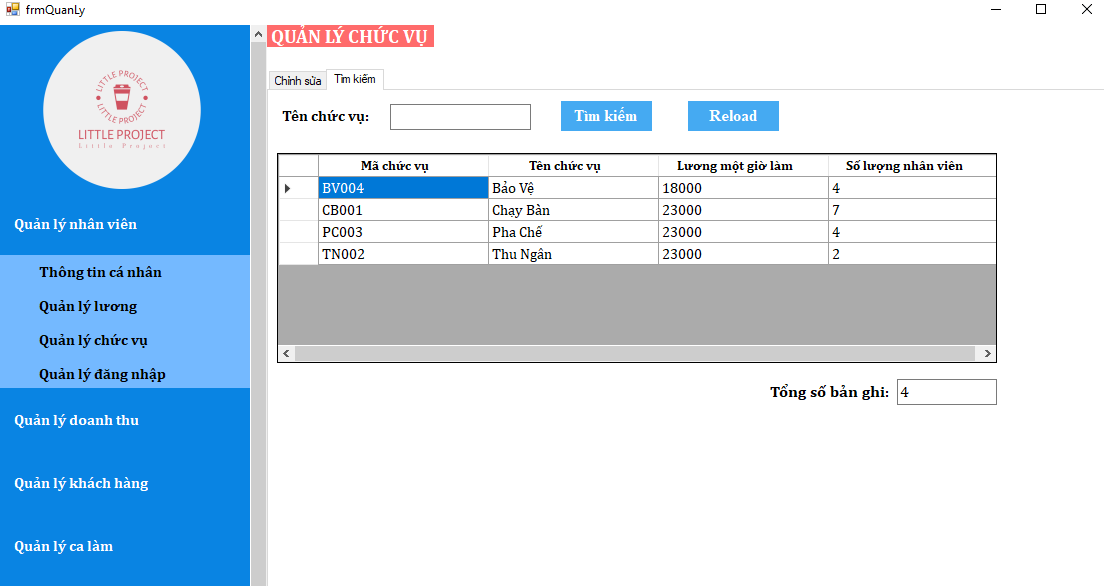


#### Form quản lý chức vụ

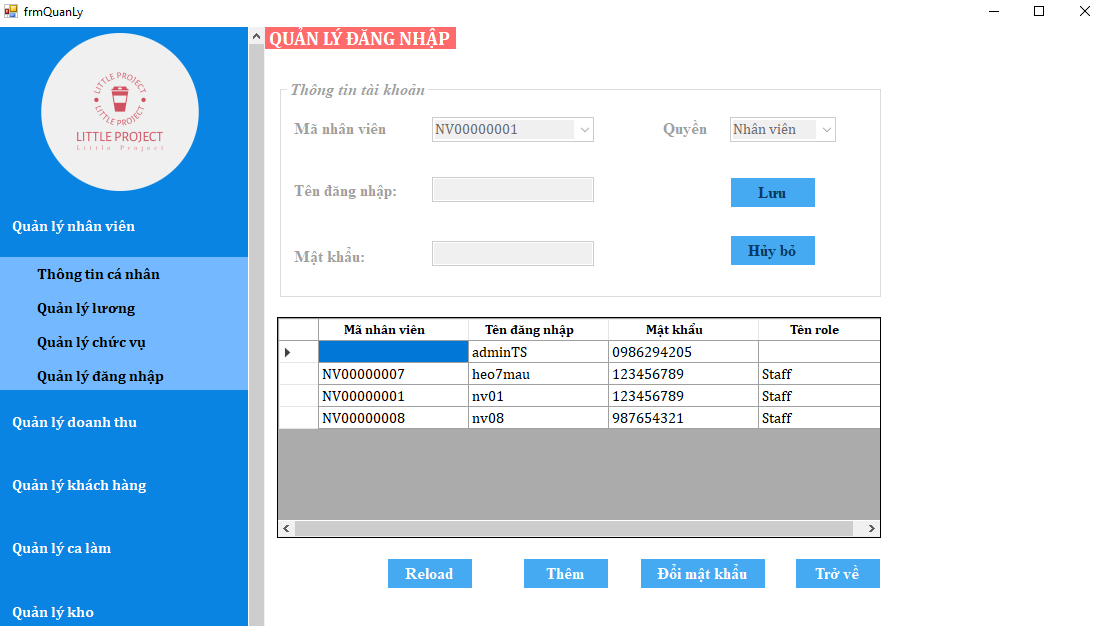
* Tab chỉnh sửa



* Tab Tìm kiếm



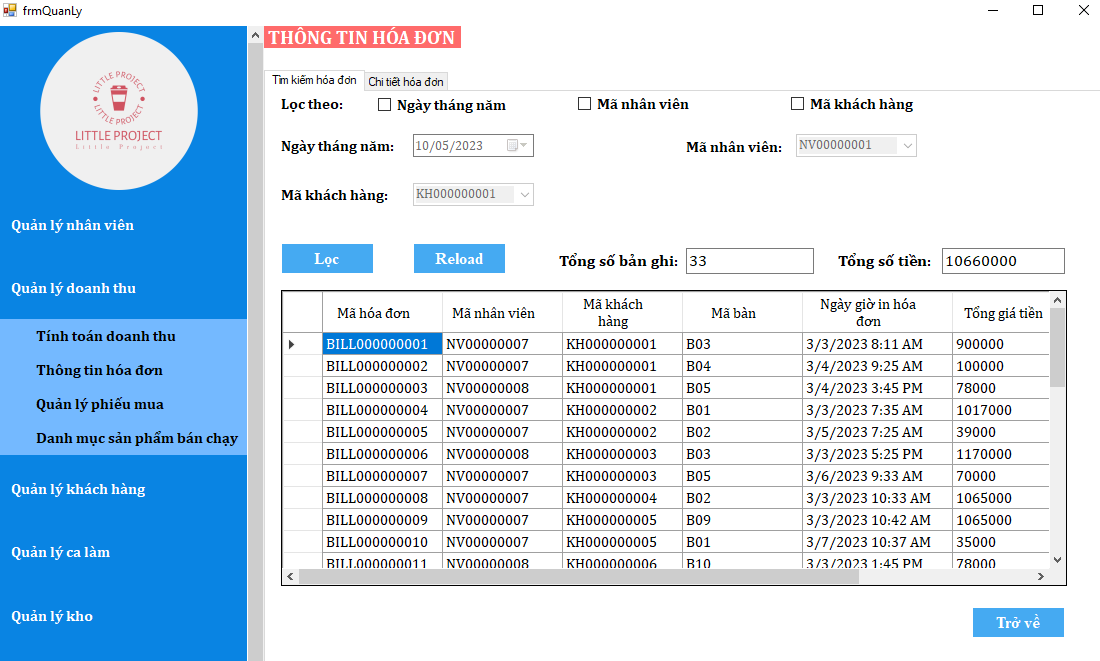
#### Form quản lý đăng nhập



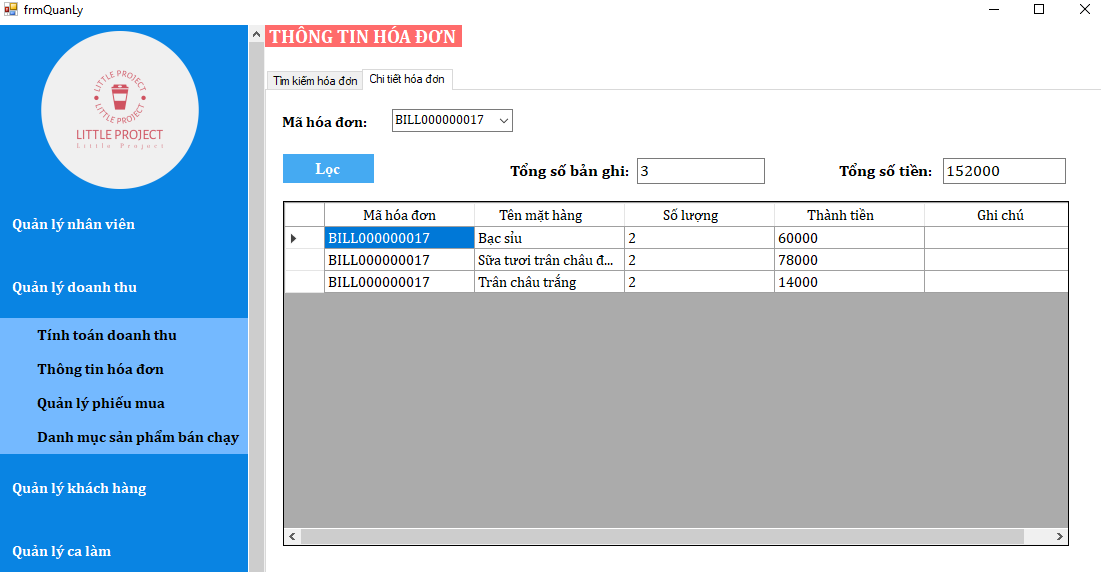
### **Form Quản lý doanh thu**

#### Form Thông tin hóa đơn

* Tab Tìm kiếm hóa đơn

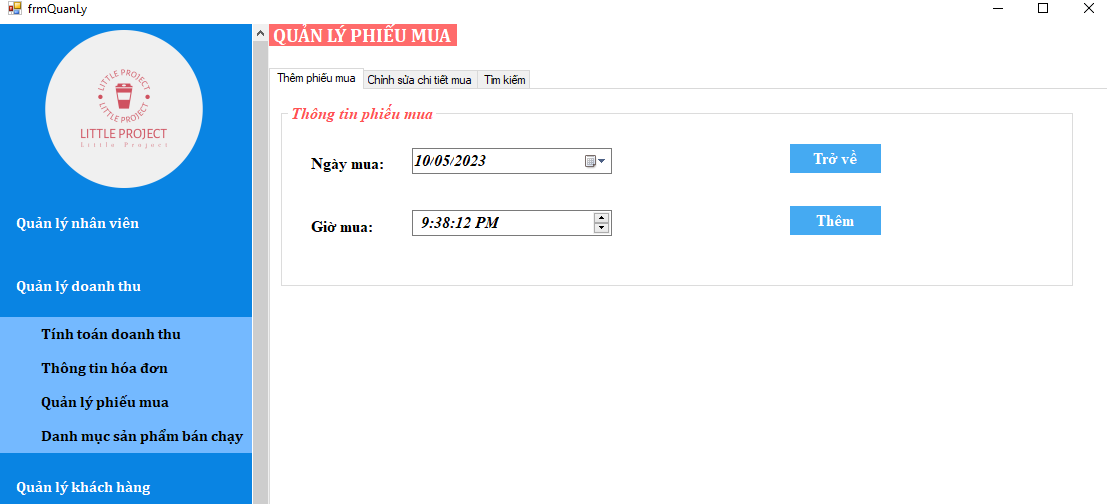


* Tab Chi tiết hóa đơn

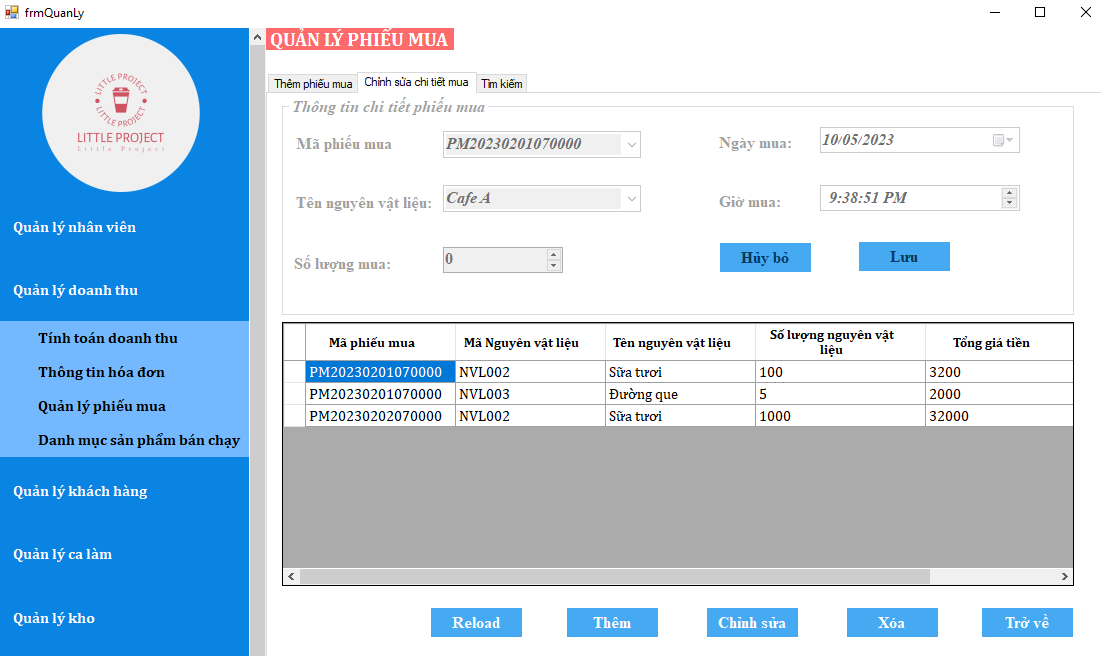


#### Form Quản lý phiếu mua

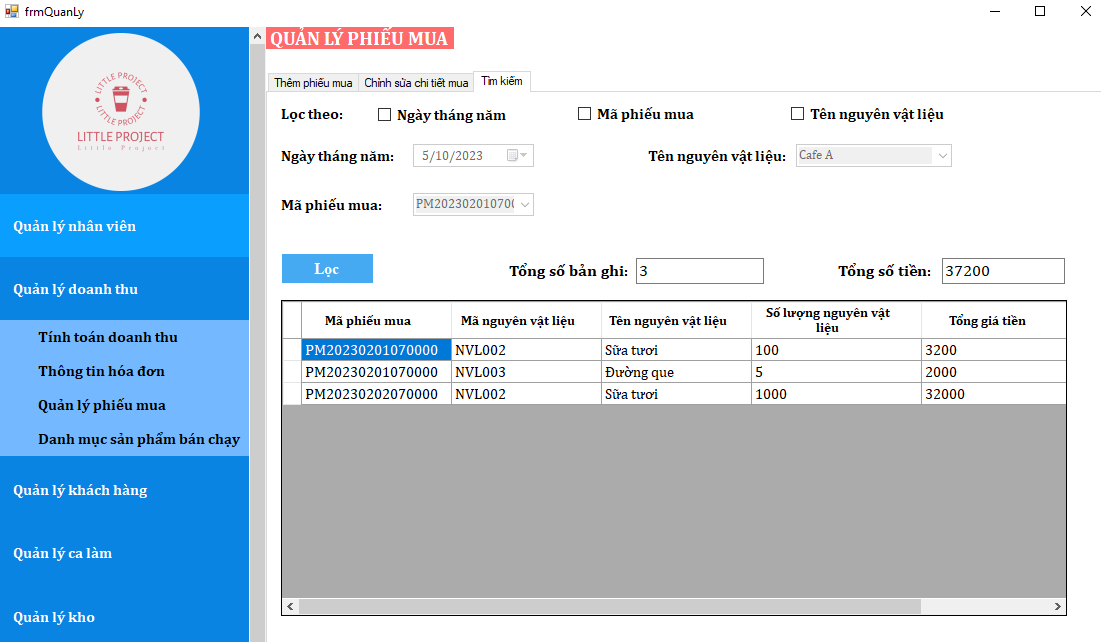
* Tab Thêm phiếu mua



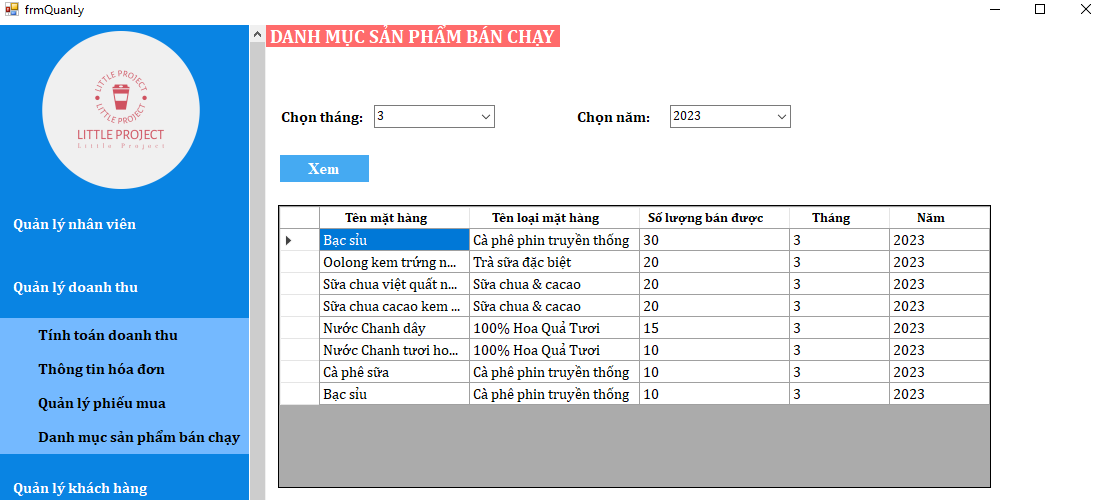
* Tab Chỉnh sửa chi tiết mua



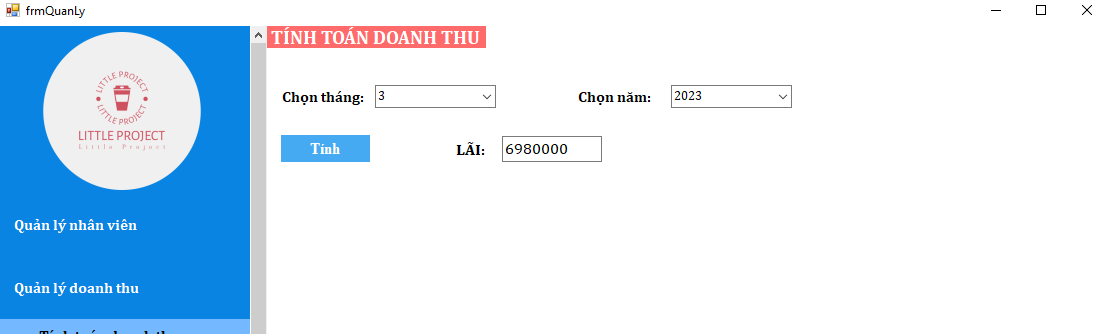
* Tab Tìm kiếm



#### Form Danh mục sản phẩm bán chạy

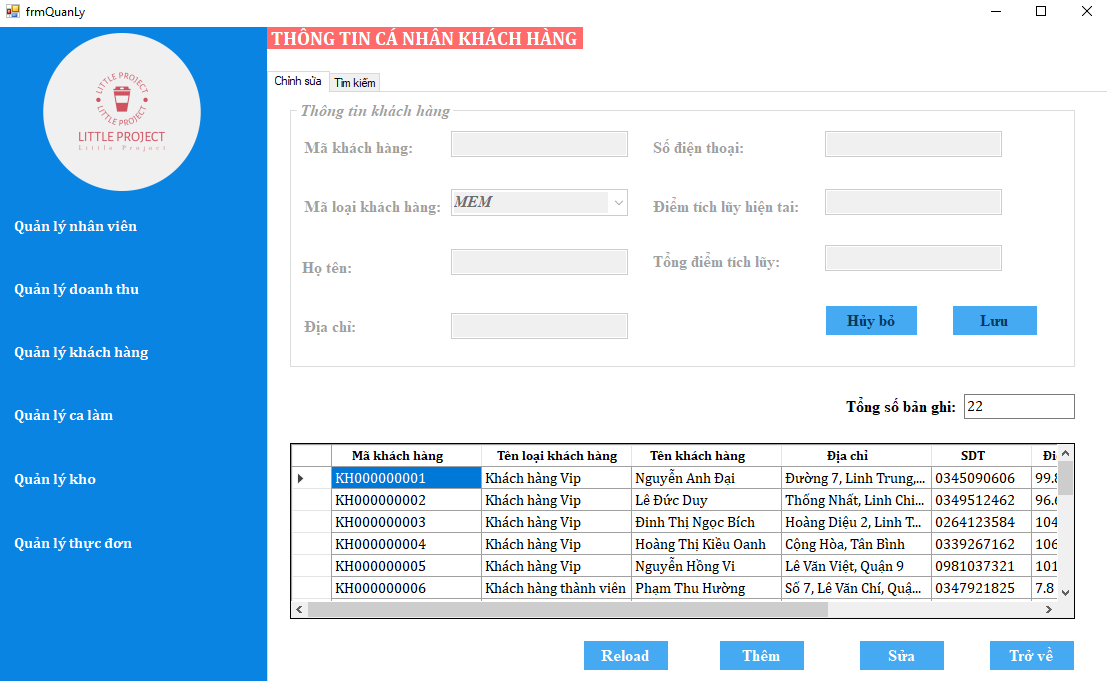


#### Form Tính toán doanh thu

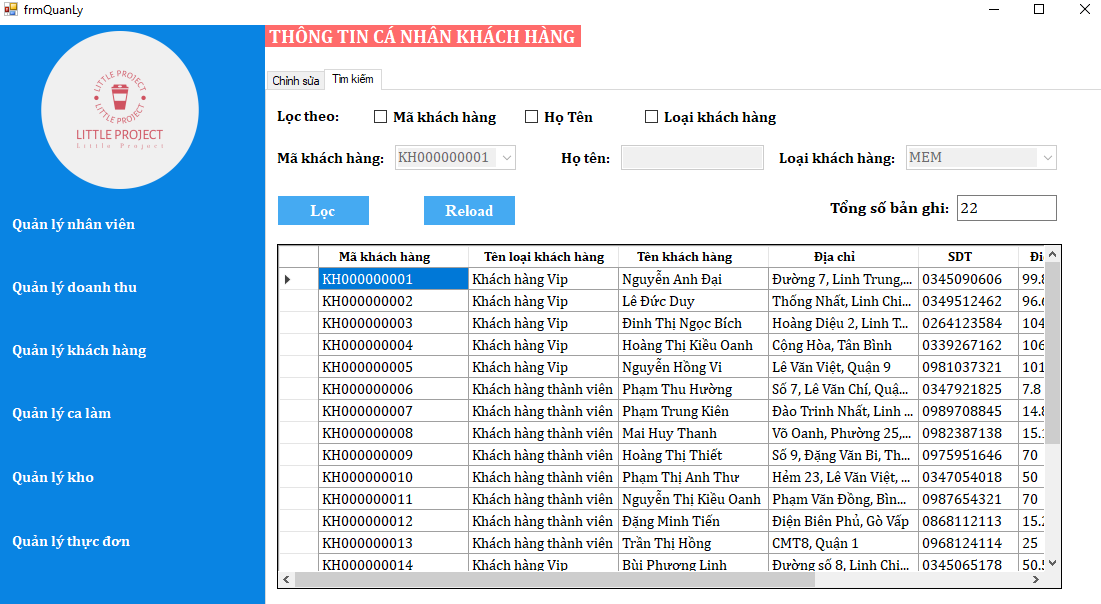


### **Form Quản lý khách hàng**

* Tab Chỉnh sửa



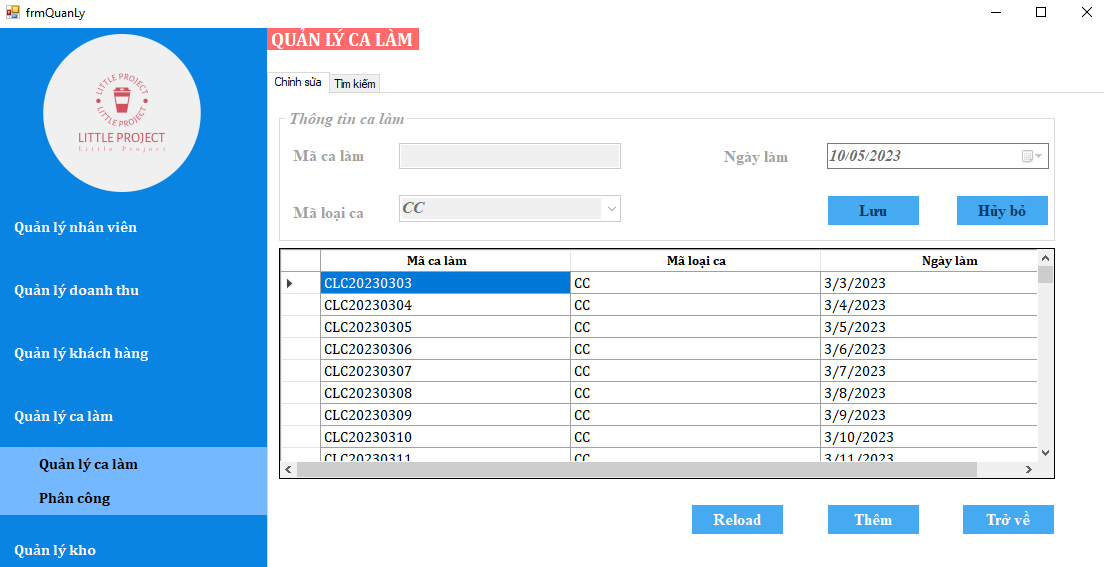
* Tab Tìm kiếm



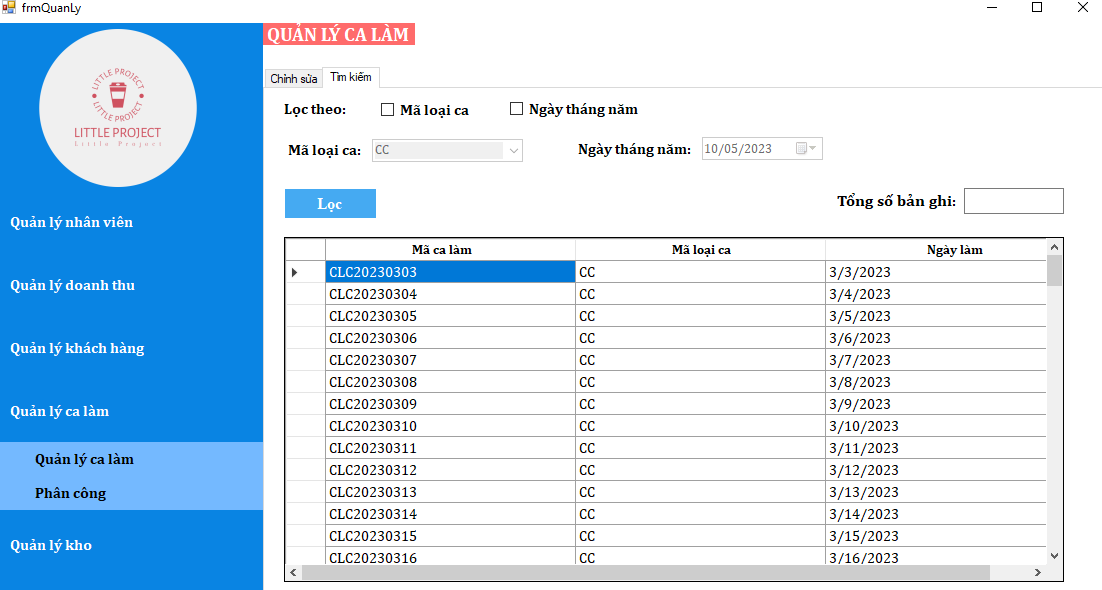
### **Form Quản lý ca làm**

#### *Form quản lý ca làm*

* Tab Chỉnh sửa

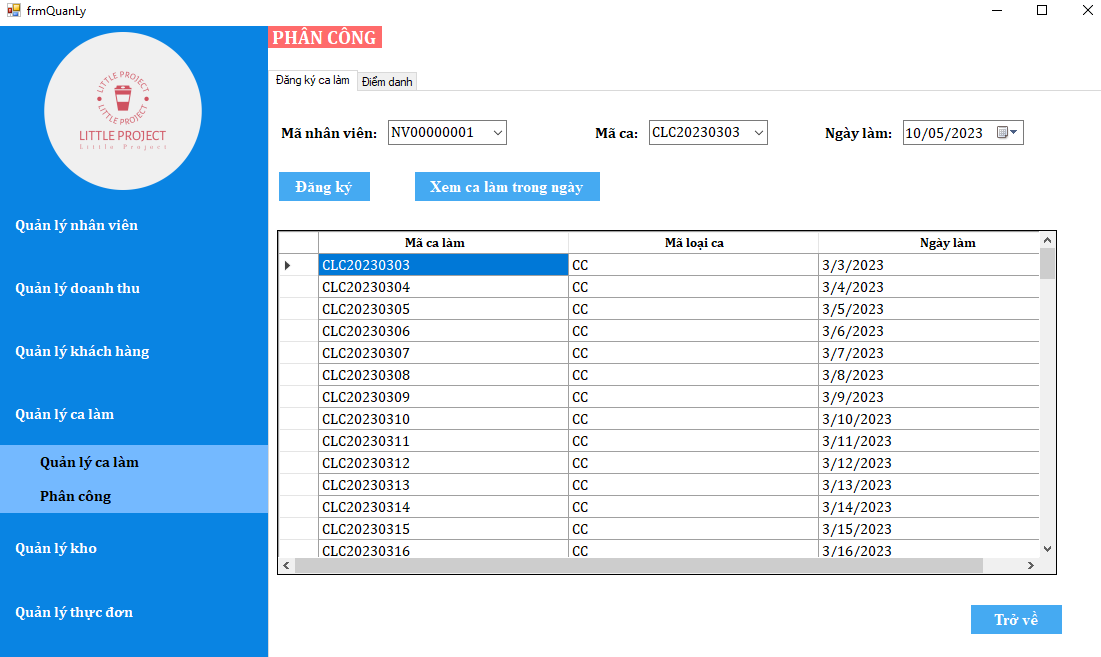


* Tab Tìm kiếm

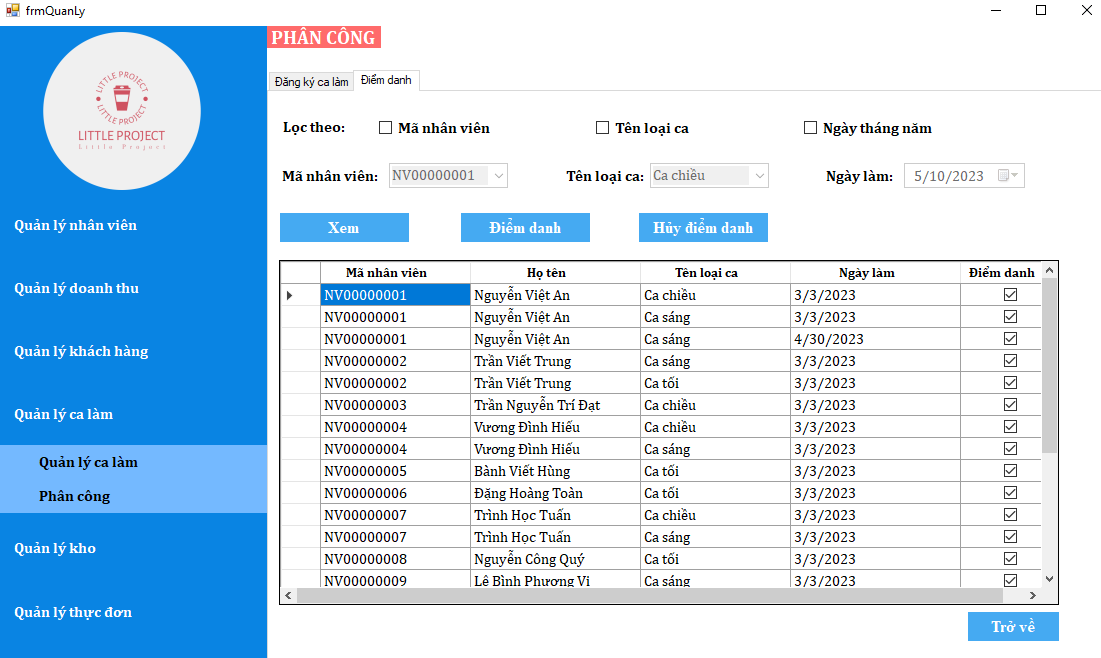


#### Form Phân công

* Tab Đăng ký ca làm

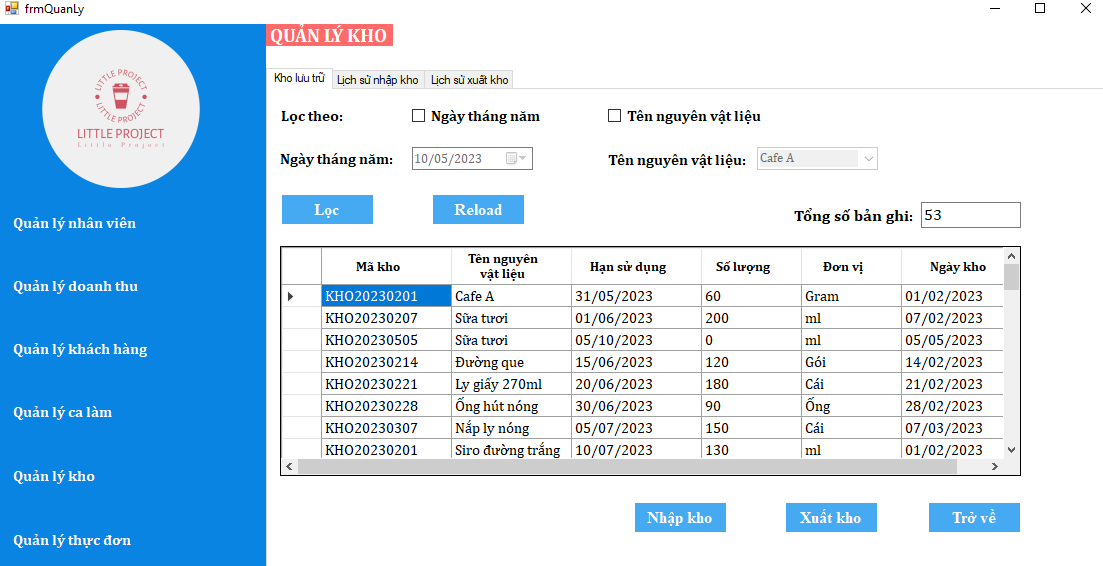


* Tab Điểm danh

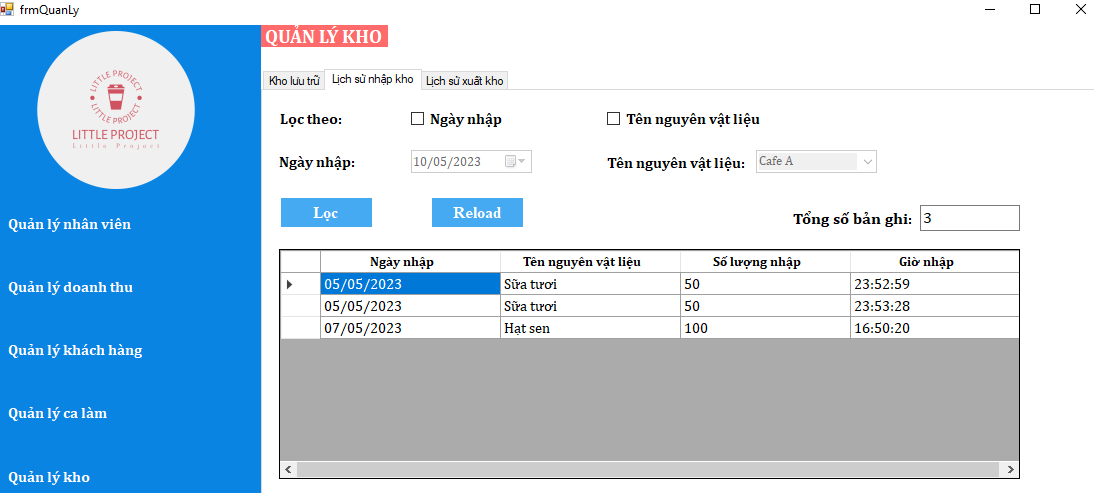


### **Form Quản lý kho**

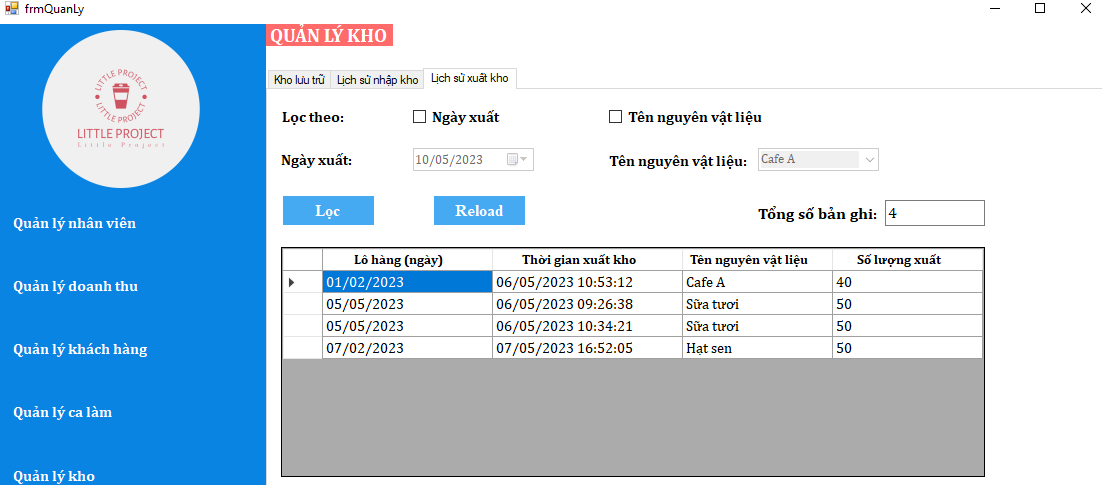
* Tab Kho lưu trữ



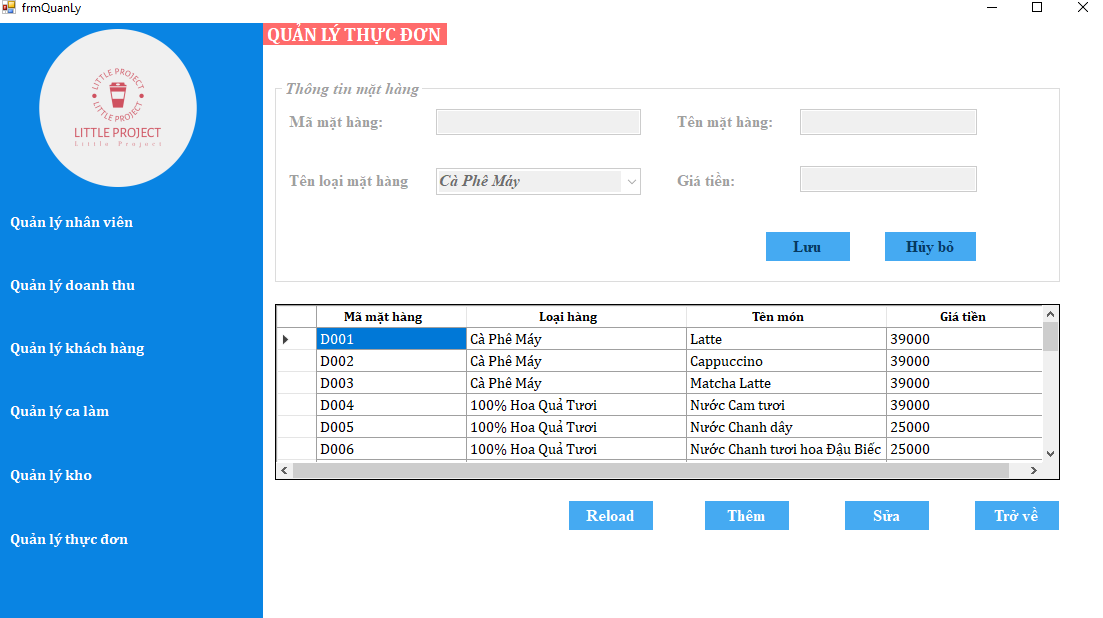
* Tab lịch sử nhập kho



* Tab lịch sử xuất kho

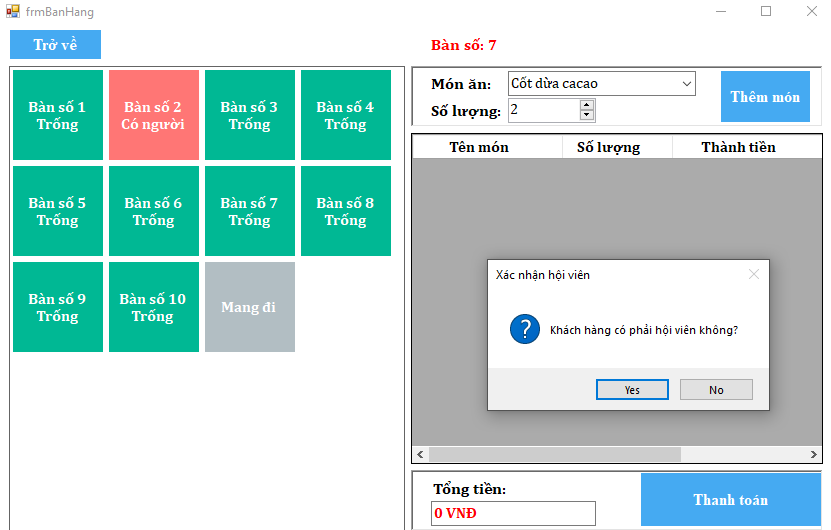


### **Form Quản lý thực đơn**

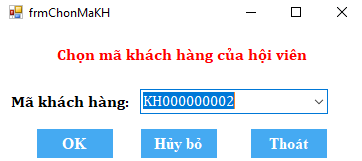


## **Form bán hàng**

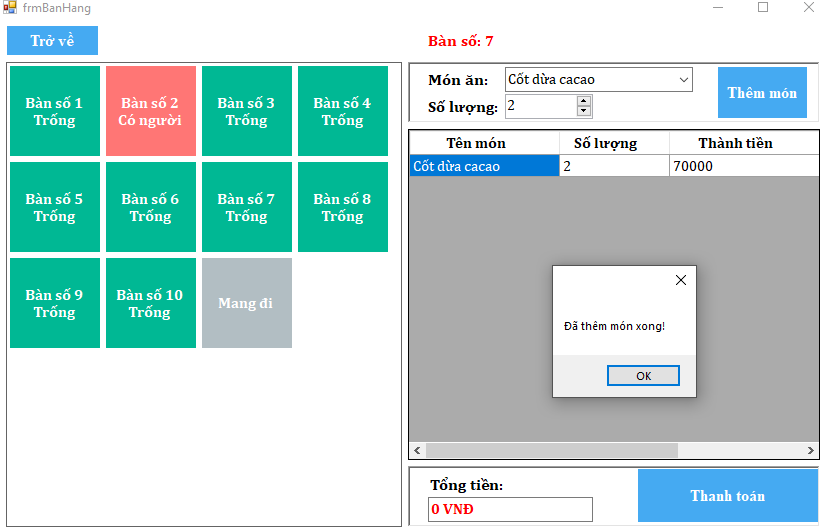
Lần đầu thêm món sẽ hỏi khách hàng có phải hội viên không

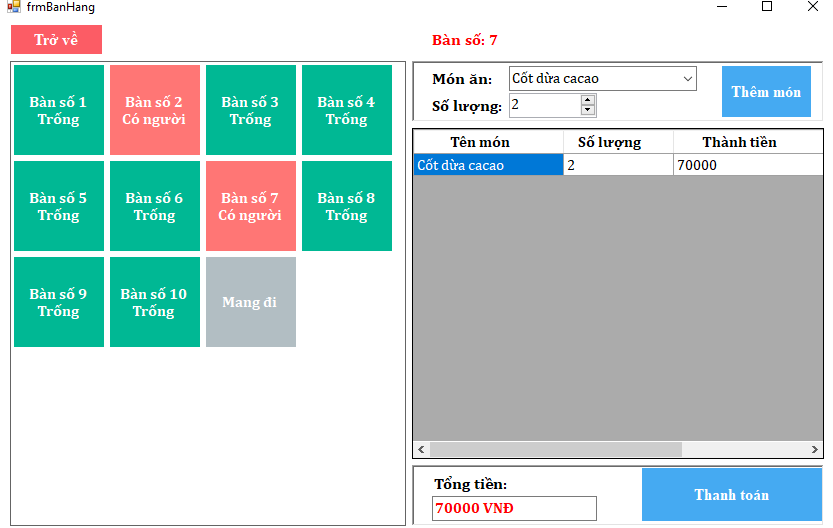


Nếu khách hàng là hội viên, Form chọn mã khách hàng sẽ xuất hiện

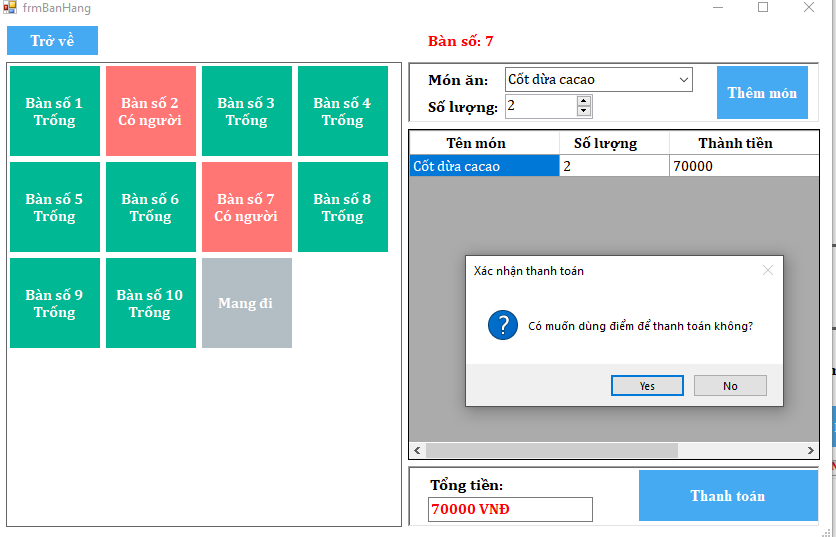


Chọn OK

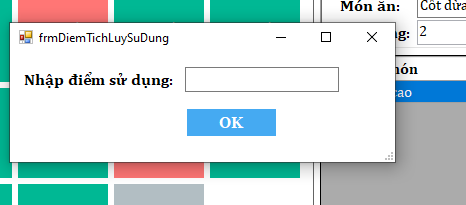




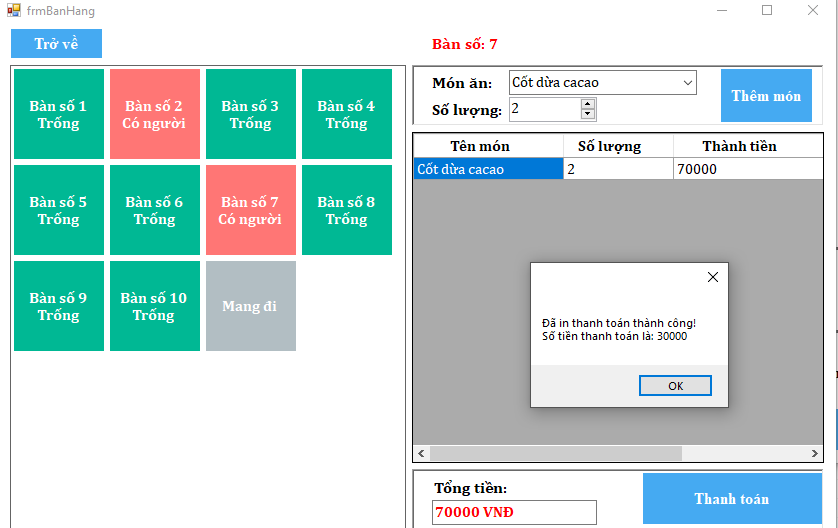
Khi chọn nút thanh toán, nếu khách hàng là hội viên đã được lựa chọn khi nãy, hệ thống sẽ hỏi có muốn sử dụng điểm để thanh toán không



Nếu chọn Yes, form “Nhập điểm sử dụng” sẽ xuất hiện



Giả sử nhập 40 điểm và chọn OK



Hệ thống sẽ thông báo số tiền mặt mà khách hàng phải trả và trả bàn về trạng thái trống

