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

**CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

-----🙞🙞🙞🙞🙞-----



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

**TÊN ĐỀ TÀI:**

**QUẢN LÝ DỰ ÁN CỦA CÔNG TY**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Lớp HP:** *DBMS330284\_22\_2\_03CLC*

**Nhóm thực hiện:** *Nhóm 02*

**Học kỳ:** *1*

**Năm học:** *2023 - 2024*

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

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2023-2024**

**Nhóm 02**

***Đề tài: Quản lý dự án của công ty***

|  |  |
| --- | --- |
| 21133060 | Bùi Đức Nhân (Nhóm trưởng) |
| 21133036 | Đặng Nguyễn Quan Huy |
| 21133062 | Huỳnh Thiện Nhân |
| 21133106 | Trần Văn Luân |

**NHẬN XÉT CỦA GV**

Tp. Hồ Chí Minh, ngày …., tháng 11, năm 2022

Giảng viên chấm điểm

**MỤC LỤC**

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 1

## 1. 1

1.1 . Mô tả bài toán 1

1.2 . Nghiệp vụ của bài toán 3

1.3 . Mô tả chức năng của bài toán 5

1.4. Tính lương nhân viên: 6

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 8

## 1. 8 2. 9

1. 10
2. 12
3. 17
4. 18

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 22

1. 22
2. 22
   1. Danh mục nhân viên 23
   2. Danh mục bảng phân ca 23
   3. Danh mục ca làm việc 23
   4. Danh mục hóa đơn 23
   5. Danh mục sản phẩm 24
   6. Danh mục phiếu chi 24

2.8. Danh mục nhà cung cấp 25

1. 25
   1. Hàm tìm kiếm khách hàng 26
   2. Tạo khách hàng mới 26
   3. Trigger bắt lỗi khi thêm khách hàng mới 27
2. 27
   1. Xuất hóa đơn 28
   2. Tìm kiếm hóa đơn 29
3. 29
   1. Tìm kiếm nguyên liệu 30
   2. Thêm nguyên liệu 30
4. 30
   1. Thêm sản phẩm 31
   2. Sửa sản phẩm 32
   3. Xóa sản phẩm 33
   4. Tìm kiếm sản phẩm 34
   5. Trigger bắt lỗi trùng tên khi thêm và chỉnh sửa sản phẩm 34 **7.** 34
   6. Doanh thu theo ngày, tháng, năm 35
   7. Chi phí theo tháng, năm 35
5. 36
   1. Thêm nhân viên 37
   2. Xóa nhân viên 38
   3. Tìm kiếm nhân viên 39
   4. Tính lương nhân viên 41
   5. Sửa thông tin nhân viên 41
   6. Trigger bắt lỗi khi thêm hoặc sửa thông tin nhân viên 42 **9.** 42

9.1. Thêm phiếu chi 44

1. 43

10.1. Thêm ca cho nhân viên 45

# CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 48 CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 50

1. 48
2. 48
3. 49
   1. Danh mục sản phẩm 52
   2. Danh mục kho 52
   3. Danh mục hóa đơn 54
   4. Danh mục hóa đơn ứng dụng 55
   5. Danh mục nhân viên 56
   6. Danh mục ca làm việc 57
   7. Danh mục bảng ca làm việc 58
   8. Danh mục phiếu chi 59
   9. Danh mục lương nhân viên 60

**LỜI CẢM ƠN**

Kính thưa thầy Nguyễn Thành Sơn,

Nhóm em xin gửi lời cảm ơn chân thành đến thầy vì đã hướng dẫn và giúp đỡ chúng em trong quá trình hoàn thiện đồ án môn Hệ quản trị cơ sở dữ liệu.

Nhờ vào sự chỉ dẫn và giúp đỡ của thầy, nhóm em đã có thể hoàn thành đồ án một cách hiệu quả và đạt được kết quả như mong đợi. Chúng em rất biết ơn thầy đã dành thời gian và tâm huyết để truyền đạt kiến thức và kinh nghiệm cho nhóm.

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

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

Trân trọng,

Bùi Quốc Khang,

Phạm Hùng Phong,

Đặng Minh Nhật, Đặng Phú Quý.

**LỜI NÓI ĐẦU**

Xin chào thầy và các bạn,

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

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

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

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

Cảm ơn quý thầy cô và các bạn sinh viên đã theo dõi và đóng góp ý kiến cho đồ án của chúng em.

**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

1. **Đặc tả đề tài**

**1.1 . Mô tả bài toán**

Một công ty phần mềm yêu cầu thiết kế một ứng dụng để quản lý Dự Án của công ty mình  với các yêu cầu như sau:

Công ty sẽ bao gồm  nhiều nhân viên làm việc. Nhân viên được phân thành bốn vai trò chính bao gồm: CEO (giám đốc), Project Manager (quản lý dự án), Team Leader (trưởng nhóm), thành viên. CEO, Project Manager, Team Leader được cấp tài khoản truy cập vào ứng dụng. Mỗi tài khoản được phân biệt bởi Username, bao gồm một mã Username và mật khẩu, để đăng nhập vào hệ thống. Mỗi nhân viên được phân biệt vớinhau bởi mã nhân viên; thông tin của một nhân viên bao gồm mã nhân viên, họ tên, email, SĐT, địa chỉ, level.

Dự án được quản lý theo hướng AGILE. Một dự án chỉ có thể được quản lý bởi một Project Manager, một Project Manager chỉ quản lý một dự án. Trong một dự án sẽ được chia thành nhiều Sprint khác nhau, phải hoàn thành một Sprint trước khi bắt đầu cái mới. Sprint là một khoản thời gian thực tế mà các team phối hợp để hoàn thành một phần công việc của dự án. PM là người quyết định thời gian cho các sprint để hoàn thành các phần của dự án. Mỗi Sprint sẽ được phân biệt với nhau bằng mã Sprint, bao gồm nội dung sẽ làm trong Sprint, ngày bắt đầu và ngày kết thúc.

Một Dự Án được xác định gồm các thông tin mã dự án, tên dự án, ngày bắt đầu, ngày kết thúc, chi phí, tiến độ, giai đoạn (bao gồm: thu thập, phân tích, thực hiện, kiểm thử và triển khai). Có các cấp bậc khác nhau đảm nhận các vị trí, chức năng nhất định trong dự án. Project Manager phân công lịch trình những công việc phải làm trong dự án cho các Team và Team Leader đại diện nhận quản lý các công việc đó.

Một Dự Án gồm có nhiều Team đảm nhận nhiều vai trò trong dự án và một Team có thể xuất hiện trong nhiều dự án khác nhau. Mỗi Team phụ thuộc vào một dự án. Để xác định một Team cần thông qua mã dự án và tên nhóm. Thông tin của team gồm tên nhóm. Trong một team thì chỉ duy nhất một người làm Team Leader quản lý nhóm đó.  Một Team có thể nhận nhiều công việc trong một dự án và một công việc chỉ được làm bởi một Team. Một Sprint gồm nhiều công việc và một công việc chỉ thuộc một Sprint. Công việc mà Team Leader nhận được từ Project Manager phân biệt với nhau bằng mã công việc, thông tin bao gồm mã công việc, tên công việc, ngày bắt đầu, ngày kết thúc, công việc tiên quyết, trạng thái, tiến độ (%).

Time Sprint là thời gian làm việc ước lượng của mỗi cá nhân hoàn thành một Sprint. Time Sprint sẽ dựa trên thời gian làm việc một ngày của mỗi cá nhân (capacity per day), khác nhau dựa trên trình độ, và ngày nghỉ (day off) của từng thành viên trong nhóm. Time Task là tổng thời gian ước lượng hoàn thành các nhiệm vụ của một nhân viên trong một Sprint. Có sự phụ thuộc vào nhân viên, Sprint và dự án; có thuộc tính số ngày nghỉ được tính dựa trên việc điểm danh của nhân viên xem nhân viên đó có nghỉ bao nhiêu ngày trong một Sprint,  Time Sprint và Time Task

Một nhân viên có thể xin nghỉ nhiều lần trong quá trình làm 1 dự án. Khi nghỉ, nhân viên sẽ xin phép nghỉ đến Team, cụ thể là Team Leader*.* Có thể chấm công cho nhân viên hôm đó sẽ vắng mặt. Mỗi lần chấm công của nhân viên khi vắng mặt sẽ được phân biệt phụ thuộc vào mã nhân viên xin nghỉ và ngày vắng, bao gồm ngày vắng mặt và nội dung xin nghỉ. Từ đó, quản lý thời gian làm việc của nhân viên xin nghỉ, và thiết lập lại Time Sprint cá nhân.

Team Leader sẽ gửi các đề xuất về nhân lực cần có cho nhóm của mình lên Project Manager. Sau đó, họ sẽ tổng hợp lại và gửi lên CEO chờ duyệt, nếu duyệt thành công, Project Manager sẽ:

* Tuyển nhân viên vào dự án. Môt nhân viên sẽ tham gia duy nhất một nhóm trong dự án dưới sự bố trí của Project Manager*.* Một nhân viên có thể tham gia nhiều nhóm ở những dự án khác nhau và có thêm thuộc tính capacity per day để xác định thời gian làm việc mỗi ngày của một người trong từng dự án*.* Trong một công việc nhóm gồm nhiều nhiệm vụ, một nhân viên trong một dự án có thể được phân công nhiều nhiệm vụ khác nhau (không trùng thời gian biểu với các nhiệm vụ khác) và mỗi nhiệm vụ chỉ có thể được giao cho một nhân viên. Mỗi nhiệm vụ của thành viên thuộc một công việc. Để xác định một nhiệm vụ cần thông qua mã nhiệm vụ. Thông tin của một nhiệm vụ bao gồm mã nhiệm vụ, tên nhiệm vụ, trạng thái nhiệm vụ, nhiệm vụ tiên quyết, thời gian thực tế, thời gian ước lượng.Biết rằng, nhiệm vụ có 2 loại là theo tuần tự và song song, Team Leader cần xác định rõ ràng để có những điều chỉnh hợp lý trong quá trình thực hiện. Nhiệm vụ tuần tự là phải làm theo độ thứ tự, kết quả của nhiệm vụ này là đầu vào của nhiệm vụ kia. Nhiệm vụ song song là những nhiệm vụ có thể làm song song với nhau không quan tâm thứ tự.
* Chỉ định tài nguyên. Tài nguyên có nhiều loại hình, và được phân biệt thông qua mã tài nguyên. Tài nguyên gồm các thông tin như mã tài nguyên, tên tài nguyên, loại tài nguyên. Một dự án được cấp nhiều loại hình tài nguyên, một loại hình tài nguyên có thể cấp cho nhiều dự án.

**1.2 . Nghiệp vụ của bài toán**

Bài toán quản lý dự án trong lĩnh vực phát triển phần mềm. Nó liên quan tới việc quản lý nhân lực dự án, phân công nhiệm vụ, nhận xét tiến độ. Khi CEO nhận dự án từ khách hàng và khởi tạo dự án, đội quản lý dự án cần làm rõ yêu cầu của dự án. Sau đó:

1. **CEO**

* Khởi tạo dự án và chỉ định Project Manager

Tạo dự án mới và cung cấp những thông tin liên quan: tên dự án, mô tả vắn tắt, deadline, tiến độ. Đồng thời chỉ định một Project Manager quản lý dự án.

* Giám sát tiến độ và chất lượng dự án

CEO sẽ không có quyền hạn trực tiếp quản lý hay điều hành một dự án, họ chỉ có thể đứng ngoài giám sát sự phát triển của dự án. Kiểm tra và phân tích các bản báo cáo của Project Manager trình lên, từ đó đưa ra những chỉ thị thích hợp cho Project Manager nếu cần thiết.

* Xem xét và duyệt cấp phát nguồn lực theo yêu cầu

Sau khi Project Manager tổng hợp đề xuất về nguồn lực và gửi lên CEO, họ sẽ có trách nhiệm xem xét và cân nhắc những đề xuất đó. Nếu không vấn đề gì, họ sẽ duyệt thông qua, và từ đó Project Manager cung cấp nguồn lực cho nhóm đã yêu cầu.

1. **Project Manager**

* Chỉ định Team Leader cho từng vị trí:

Project Manager phân mảnh các vị trí thành các Team, từ đó tiến hành chọn Team Leader để vận hành công việc nhóm một cách hiệu quả, người được chọn phải có kinh nghiệm trong việc quản lý nhân lực, khả năng điều phối hoạt động nhóm để đảm bảo công việc được thực hiện đúng tiến độ.

* Tổng hợp các đề xuất nguồn lực từ các nhóm:

Project Manager sẽ tổng hợp đề xuất nguồn nhân lực từ các nhóm và nguồn tài nguyên sẽ dùng, gửi lên CEO chờ duyệt. Nếu được thông qua, thì Project Manager sẽ tiến hành tuyển nhân lực theo yêu cầu các nhóm và chỉ định tài nguyên cần thiết.

* Tuyển nhân lực cho dự án:

Khi đã được CEO duyệt thông qua đề xuất nguồn lực, Project Manager tiến hành tuyển nhân lực tham gia vào dự án dựa trên quy mô và tính chất của dự án. Các thành viên được tuyển vào dự án sẽ được phân chia vào các Team dựa trên kỹ năng và kinh nghiệm, mỗi Team sẽ đảm nhận một vị trí trong dự án.

* Chỉ định tài nguyên cho các nhóm:

Khi đã được CEO duyệt thông qua đề xuất nguồn lực và phân tích triển khai dự án, PM đưa ra quyết định về các tài nguyên sử dụng trong suốt quá trình làm dự án. Cân nhắc chọn các nguồn lực phù hợp với quy mô và giá trị dự án. Đảm bảo tính nhất quán và bảo mật về công nghệ, công cụ, dữ liệu trong suốt quá trình triển khai dự án.

* Triển khai công việc:

Project Manager nhập khoảng thời gian thực tế mà các Team phối hợp để hoàn thành một phần của dự án và nhập lên hệ thống (hay gọi là Sprint)

Project Manager phân công công việc cho từng Team trong dự án. Trong quá trình làm Project, Team Leader nhận công việc từ phía PM và chia công việc cho các thành viên nhóm. Team Leader là người chịu trách nhiệm chính đối với công việc của nhóm, có nhiệm vụ báo cáo tiến độ nhóm cho PM.

* Giám sát và cập nhật tiến độ, giai đoạn của dự án:

Các Team Leader sẽ báo cáo lên cho Project Manager tiến độ công việc đã hoàn thành của cả nhóm. Dựa vào đó, Project Manager sẽ cập nhật tiến độ dự án tương ứng. Tạo các bản báo cáo thông minh để trình lên cho CEO, mô tả khái quát tiến độ của dự án dựa trên dữ liệu cập nhật từ các Team.

Ngoài ra, Project Manager cập nhật giai đoạn hiện tại dự án, bao gồm các giai đoạn sau: Thu thập và Phân tích → Thiết kế → Triển khai → Kiểm thử → Bàn giao

Một dự án sẽ trải qua các trạng thái như trên, nếu dự án đang trong quá trình thu thập và phân tích dữ liệu, yêu cầu của khách hàng thì Project Manager sẽ cập nhật trạng thái là “Thu thập và Phân tích”.

Sau khi đã hoàn thành giai đoạn trên, sẽ tiếp đến giai đoạn thiết kế, là việc áp dụng các công cụ, phương pháp, thủ tục để tạo ra mô hình hệ thống cần sử dụng. Sản phẩm cuối cùng của giai đoạn thiết kế là đặc tả hệ thống ở dạng tồn tại thực tế, sao cho chuyên viên kỹ thuật có thể dễ dàng chuyển thành chương trình và cấu trúc hệ thống. Đây là giai đoạn nhằm xây dựng hệ thống theo các thiết kế đã xác định.

Tiếp đến bước triển khai, giai đoạn này bao gồm các công việc sau: lựa chọn hệ quản trị cơ sở dữ liệu (SQL Server, Oracle, MySQL, …) và cài đặt cơ sở dữ liệu cho hệ thống; Lựa chọn công cụ lập trình để xây dựng các modules chương trình của hệ thống (Microsoft Visual Studio, PHP Designer,…). Lựa chọn công cụ để xây dựng giao diện hệ thống (DevExpress, Dot Net Bar,…). Phân chia nhiệm vụ cho từng nhóm và đảm bảo sự chính xác đáp ứng các yêu cầu được giao và đảm bảo đúng tiến độ. Viết tài liệu hướng dẫn sử dụng, tài liệu kỹ thuật hoặc clip hướng dẫn.

Tiếp sau đó là tới bước kiểm thử, trước hết phải lựa chọn công cụ kiểm thử. Kiểm chứng các modules chức năng của hệ thống thông tin, chuyển các thiết kế thành các chương trình (phần mềm). Thử nghiệm hệ thống thông tin. Cuối cùng là khắc phục các lỗi (nếu có). Viết test case theo yêu cầu.

1. **Team Leader**

* Đề xuất nguồn nhân lực cho một nhóm

Nhóm sẽ thảo luận đề xuất nguồn lực lên Project Manager, phải chọn những người có đủ trình độ, kiến thức thích hợp với yêu cầu công việc của nhóm. Ngoài ra, cần đảm bảo lịch trình làm việc không trùng với các dự án khác của thành viên.

* Xác định thời gian thực hiện nhiệm vụ trong 1 ngày của cá nhân (capacity per day)

Team Leader dựa trên trình độ, năng lực, vai trò của từng thành viên của team để xác định thời gian thực hiện nhiệm vụ trong 1 ngày (capacity per day) của cá nhân. Với nhân viên có trình độ cao, thời gian thực hiện sẽ thấp hơn mức tiêu chuẩn để dành thời gian hỗ trợ các thành viên mới, thiếu kinh nghiệm.

* Phân chia nhiệm vụ cho các thành viên

Team Leader giao nhiệm vụ cho các thành viên trong nhóm. Nội dung nhiệm vụ, thời gian và kết quả theo yêu cầu đều được truyền đạt rõ ràng tới các thành viên để đảm bảo chất lượng đầu ra cũng như tiến độ.

Nhiệm vụ gồm 2 loại: nhiệm vụ có thể làm song song và nhiệm vụ làm tuần tự. Team Leader cần soạn rõ lịch trình kế hoạch cho từng loại nhiệm vụ, để thống nhất thời gian, thứ tự làm việc cho thành viên.

* Giám sát nhiệm vụ của từng thành viên

Cuối ngày, các thành viên trong nhóm sẽ họp lại để bàn luận về những công việc đã làm trong hôm nay, và phải báo cáo nhiệm vụ đã làm cho Team Leader, từ đó Team Leader sẽ đưa ra nhận xét chỉnh sửa kịp thời cho các thành viên để họ có thể hoàn thành nhiệm vụ. Việc đó sẽ làm giảm việc chậm tiến độ dự án chung của cả dự án và giám sát ngày nghỉ.

* Quản lý ngày nghỉ của thành viên

Nhân viên nghỉ sẽ xin phép đến Team Leader trước một khoảng thời gian. Team Leader sẽ cập nhật và lưu trữ thông tin ngày nghỉ, dự án mà nhân viên ấy đang tham gia. Từ đó hệ thống có thể dựa vào thông tin đó để tính toán lại time sprint phù hợp.

* Điểm danh cho thành viên nhóm

Chịu trách nhiệm điểm danh cho các thành viên nhóm mình. Đánh dấu những thành viên vắng mặt để thiết lập lại Time Sprint tương ứng cho mỗi cá nhân

* **1.3 . Mô tả chức năng của bài toán** Đối Đối với CEO

**Khởi tạo dự án:** Cho phép CEO điền thông tin dự án để khởi tạo (vắn tắt mục tiêu và kết quả dự án). Cho phép chỉ định PM cho dự án.

**Giám sát dự án:** Cho phép CEO xem những bản báo cáo, thống kê thông minh đã được phân tích bởi Project Manager.

* Đối với Project Manager

**Chức năng tuyển chọn nhân viên cho từng nhóm công việc:** cho phép PM có thể  thêm, xóa nhân viên. (Bỏ)

**Chức năng chỉ định nhóm trưởng của mỗi team:** cho phép PM có thể chọn người phù hợp với vai trò trưởng nhóm dựa trên kinh nghiệm, kiến thức và khả năng lãnh đạo.

**Chức năng tạo sprint:** cho phép PM nhập thời gian mỗi sprint cho dự án.

Xây dựng lịch trình dự án: cho phép Project Manager nhập kế hoạch chi tiết cho dự án lên hệ thống. Từ đó sẽ phân công cho các Team Leader để triển khai cho các thành viên của họ.

**Chức năng kiểm tra lịch trình:** cho phép CEO có thể xem lịch trình, kết quả công việc của từng nhân viên, Team để tránh xung đột thời gian.

**Chỉ định tài nguyên cho dự án:** cho phép Project Manager CEO xác định các loại tài nguyên sử dụng cho dự án lên hệ thống.

**Phân chia công việc cho các team:** cho phép Project Manager xác định công việc cho từng team. Cho phép Team Leader xác định công việc cho mỗi cá nhân trong team. Cần chú ý các công việc tuần tự khi phân chia

**Project Manager giám sát tiến độ dự án:** cho phép PM theo dõi và phân tích tình hình thực tế để đưa ra thay đổi hợp lý về kế hoạch tiến độ. PM cập nhật các giai đoạn hiện tại của dự án để. Từ đó PM sẽ đưa ra các bản báo cáo thống kê chi tiết cho CEO, cập nhật các giai đoạn của dự án.

* Team Leader

**Cấp capacity per day cho thành viên:** Cho phép Team Leader nhập lên phần mềm thời gian thực hiện nhiệm vụ của từng thành viên trong 1 ngày.

**Cập nhật Time Sprint của mỗi thành viên**: cho phép Team Leader cập nhật lên phần mềm thời gian thực hiện các nhiệm vụ của từng thành viên trong 1 sprint.

**Phân công nhiệm vụ cho các thành viên:** phân công nhiệm vụ trên phần mềm cho các thành viên trong nhóm, kèm theo là các thông tin như nội dung nhiệm vụ, thời hạn hoàn thành, nhiệm vụ tiên quyết,...

**Quản lý và giám sát tiến độ công việc được giao:** thường xuyên cập nhật tiến độ nhiệm vụ của các thành viên, đảm bảo kịp tiến độ chung của nhóm cũng như cả dự án. Những ai hoàn thành trước hạn và thỏa mãn yêu cầu của Team Leader thì sẽ được đánh dấu là “đã hoàn thành”, còn quá hạn nhưng chưa hoàn thành thì là “quá hạn” nếu được Team Leader nhận xét chỉnh sửa thì sẽ là “đang chỉnh sửa”. Sắp xếp thứ tự phân công nhiệm vụ hợp lý, để tránh tình trạng phải chờ nhiệm vụ tiên quyết.

**Điểm danh:** quản lý thời gian đi làm của các thành viên,cho phép Team Leader cập nhập ngày nghỉ cho nhân viên và thiết lập lại Time Sprint tương ứng

**1.4.Tính tiến độ:**

**a) Hiệu suất một cá nhân:**

Ý tưởng: mỗi người sẽ có một hoặc các nhiệm vụ có thể thuộc các công việc khác nhau. Việc tính tiến độ cá nhân sẽ dựa trên tổng số thời gian nhiệm vụ đã hoàn thành cho đến thời điểm hiện tại so với khoản thời gian thực tế mà nhân viên làm việc trong một sprint.

* *Time Sprint*

Ta có, Time sprint của một thành viên = capacity per day \* (số ngày trong sprint - số ngày nghỉ)

* *Time Tasks*

Time Tasks = x=1n(thời gian ước tính); trong đó n: tổng số nhiệm vụ của một nhân viên

Cập nhật tiến độ nhiệm vụ: nếu nhiệm vụ hoàn thành thì Time Tasks = Time Tasks - thời gian ước tính

* *Ước lượng tiến độ cá nhân*

Công thức tính : Y =  Time sprintTime Tasks

* Nếu Y >= 1: Đang hoàn thành đúng tiến độ
* Nếu Y < 1: Đang trễ tiến độ.
* *Ví dụ:*

Trong một Sprint chia thành 3 nhiệm vụ dành cho cá nhân:

* Task 1: 8h
* Task 2: 16h
* Task 3: 24h

Minh họa:

\* Time Tasks

|  |  |  |
| --- | --- | --- |
| 24 | 18 | 6 |

Với mỗi công việc Done, thanh Time Tasks sẽ trừ đi thời gian của task đó.

\* Time Sprint

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

Mỗi sprint làm việc 8 ngày, mỗi ngày làm việc 6 tiếng.

Sprint = 8 ngày \* 6 giờ/ngày = 48 giờ.

Sau mỗi một ngày làm việc, thanh Time Sprint sẽ trừ đi 6 giờ.

**b) Tiến độ một công việc và nhiệm vụ trong một sprint :**

* **Tiến độ của nhiệm vụ:**

Ý tưởng: Chúng ta sẽ thiết lập khoảng thời gian là 4 ngày trước khi kết thúc Sprint để làm cột mốc cảnh báo cho quản lý biết nhiệm vụ đó sắp trễ tiến độ hay không, nếu tới thời gian kết thúc Sprint mà nhiệm vụ chưa hoàn thành ta xác định nhiệm vụ đang trễ tiến độ.

* Tiến độ của công việc:

Ý tưởng: tính bằng cách lấy số nhiệm vụ đã hoàn thành trong một công việc cho đến hiện tại chia tổng số lượng nhiệm vụ trong một Sprint hiện tại

Công thức: Tiến độ một công việc trong một Sprint =  (tổng số lượng nhiệm vụ đã hoàn thành / tổng số lượng nhiệm vụ) x 100 (%)

Ví dụ: nhân viên A và B đảm nhận nhiều nhiệm vụ, các nhiệm vụ có thể thuộc 1 công việc ở Sprint hiện tại, suy ra:

* Nhân viên A đã hoàn thành cả 2 nhiệm vụ
* Nhân viên B có 3 nhiệm vụ, đã hoàn thành 1 nhiệm vụ còn 2 nhiệm vụ chưa hoàn thành thì tiến độ công việc No2 được tính như sau:

Tiến độ một công việc =  [2 (A) + 1 (B)] / 5 = 75%

**c) Tiến độ một dự án:**

Ý tưởng: được tính bằng cách lấy tổng các công việc đã hoàn thành trong một Sprint trong dự án chia tổng số công việc trong Sprint đó

Công thức: Tiến độ cả dự án =  (tổng công việc hoàn thành trong một Sprint) / số lượng công việc trong một Sprint

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

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

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

8

## 2. Thiết kế cơ sở dữ liệu mức logic

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

* CongViec(MaCV, TenCV, Luong)
* NhanVien(MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, NgayTuyenDung, SoCa, Thuong, MaCV, SDT)
* BangPhanCa(MaCa, MaNV, NgayLam)
* CaLamViec(MaCa, NgayLam, GioBatDau,GioKetThuc)
* HoaDon(MaHD, Ngay, MaKH, MaNV, TrangThai, TriGiaHD)
* ChiTietHoaDon(MaHD, MaSP, SoLuong, TongTien, DonGia)
* KhachHang(MaKH, TenKH, SDT, DiemTL)
* SanPham(MaSP, TenSP, DonGia, TinhTrang, MaLoaiSP)
* LoaiSanPham(MaLoaiSP, TenLoaiSP)
* ChiTietHoaDonUngDung (MaHD\_UD, MaSP, SoLuong, DonGia, TongTien)
* HoaDonUngDung (MaHD, Ngay, MaUngDung, TrangThai, MaNV, TriGiaHD)
* UngDung(MaUngDung, TenUngDung, ChietKhauPhanTram)
* CheBien(MaSP, MaNL, LieuLuong, DonVi)
* NguyenLieu(MaNL, TenNL, MaNCC, SoLuong, DonVi, TinhTrang)
* ChiTietDonNhapHang(MaNL, MaDonNH, DonGia, SL, TongTien)
* NhaCungCap(MaNCC, TenNCC, DiaChi, SDT)
* DonNhapHang(MaDonNH, NgayNH, TriGiaDonNH, MaNCC)
* PhieuChi(MaPhieuChi, MaDonNH, NgayXuatPhieu, SoTienChi)

## 3. Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| S  T  T | BẢNG | RÀNG BUỘC |
| 1 | UOCLUONG | Khóa chính (MaNV, MaDA, MaSprint) |
| 2 | TAINGUYEN | Khóa chính MaTN |
| 3 | NHANVIEN | Khoá chính MaNV |
| 4 | DUAN | Khoá chính MaDA,  Khóa ngoại FK\_DUAN\_NHANVIEN (MaPM) đến NHANVIEN (MaNV) |
| 5 | CAP | Khoá chính (MaDA, MaTN),  Khóa ngoại FK\_CAP\_DUAN (MaDA) đến DUAN (MaDA)  Khóa ngoại FK\_CAP\_TAINGUYEN (MaTN) đến TAINGUYEN (MaTN) |
| 6 | SPRINT | Khoá chính MaSprint,  Khóa ngoại FK\_SPRINT\_DUAN (MaDA) đến DUAN (MaDA) |
| 7 | DIEMDANH | Khóa chính (Ngay, MaNV),  Khóa ngoại FK\_DIEMDANH\_NHANVIEN (MaNV) đến NHANVIEN (MaNV) |
| 8 | TAIKHOAN | Khoá chính MaNV,  Khóa ngoại FK\_TAIKHOAN\_NHANVIEN (MaNV) đến NHANVIEN (MaNV) |
| 9 | TEAMLEADER | Khoá chính (TenNhom, MaDA),  Khóa ngoại FK\_TEAM\_DUAN (MaDA) đến DUAN (MaDA),  Khóa ngoại FK\_TEAMLEADER\_NHANVIEN (MaNV) đến NHANVIEN (MaNV) |
| 10 | CONGVIEC | Khoá chính MaCV,  Khóa ngoại FK\_CONGVIEC\_TEAMLEADER (TenNhom, MaDA) đến TEAMLEADER (TenNhom, MaDA),  Khóa ngoại FK\_CONGVIEC\_SPRINT (MaSprint) đến SPRINT (MaSprint) |
| 11 | NHIEMVU | Khoá chính MaNhiemVu,  Khóa ngoại FK\_NHIEMVU\_NHANVIEN (MaNV) đến NHANVIEN (MaNV) ,  Khóa ngoại FK\_NHIEMVU\_CONGVIEC (MaCV) đến CONGVIEC (MaCV) |
| 12 | TEAM | Khoá chính (TenNhom, MaDA, MaNV),  Khóa ngoại FK\_TEAM\_TEAMLEADER (TenNhom, MaDA) đến TEAMLEADER (TenNhom, MaDA) ,  Khóa ngoại FK\_TEAM\_NHANVIEN (MaNV) đến NHANVIEN (MaNV) |

4. **Cài đặt các CSDL và các ràng buộc**

## Bảng công việc

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

## Bảng nhân viên

CREATE TABLE NhanVien(

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

HoNV nvarchar(10) NOT NULL,

TenNV nvarchar(10) NOT NULL,

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

GioiTinh nvarchar(3),

DiaChi nvarchar(100),

SDT nchar(11) check (len(SDT)=10),

MaCV nchar(10) CONSTRAINT FK\_NhanVien\_CV FOREIGN KEY REFERENCES

CongViec(MaCV),

SoCa int,

Thuong int,

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

)

## Bảng khách hàng

CREATE TABLE KhachHang(

MaKH nchar(10) CONSTRAINT PK\_KhachHang PRIMARY KEY,

TenKH nvarchar(50) NOT NULL,

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

DiemTL int

)

## Bảng nhà cung cấp

CREATE TABLE NhaCungCap(

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

TenNCC nvarchar(50) NOT NULL,

DiaChi nchar(100),

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

)

## Bảng đơn nhập hàng

CREATE TABLE DonNhapHang(

MaDonNH nchar(10) CONSTRAINT PK\_DonNhapHang PRIMARY KEY,

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

TriGiaDonNH float NOT NULL,

MaNCC nchar(10) CONSTRAINT FK\_DonNhapHang\_NCC FOREIGN KEY REFERENCES

NhaCungCap(MaNCC),

TinhTrang nvarchar(50)

)

## Bảng nguyên liệu

|  |
| --- |
| CREATE TABLE NguyenLieu(  MaNL nchar(10) CONSTRAINT PK\_NguyenLieu PRIMARY KEY,  TenNL nvarchar(50) NOT NULL,  MaNCC nchar(10) CONSTRAINT FK\_NguyenLieu\_NCC FOREIGN KEY REFERENCES NhaCungCap(MaNCC),  SL int check (SL>0),  DonVi nchar(10), TinhTrang nchar(10)  ) |

## Bảng chi tiết nhập hàng

CREATE TABLE ChiTietNhapHang(

MaDonNH nchar(10) CONSTRAINT FK\_ChiTietNH\_DonNH FOREIGN KEY REFERENCES

DonNhapHang(MaDonNH),

MaNL nchar(10) CONSTRAINT FK\_ChiTietNH\_NL FOREIGN KEY REFERENCES

NguyenLieu(MaNL),

DonGia float,

SL int check (SL>0),

DonVi nchar(10),

TongTien float

CONSTRAINT PK\_ChiTietNhapHang PRIMARY KEY (MaDonNH, MaNL)

)

## Bảng loại sản phẩm

CREATE TABLE LoaiSanPham(

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

TenLoaiSP nvarchar(50) NOT NULL

)

## Bảng sản phẩm

CREATE TABLE SanPham(

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

TenSP nvarchar(50) NOT NULL,

DonGia float NOT NULL,

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

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

LoaiSanPham

)

## Bảng ứng dụng

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

## Bảng hoá đơn

CREATE TABLE HoaDon(

MaHD nchar(10) CONSTRAINT PK\_HoaDon PRIMARY KEY,

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

MaKH nchar(10) CONSTRAINT FK\_HoaDon\_KH FOREIGN KEY REFERENCES

KhachHang(MaKH),

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

TriGiaHD int NOT NULL

)

## Bảng chi tiết hoá đơn

CREATE TABLE ChiTietHoaDon(

MaHD nchar(10) CONSTRAINT FK\_ChiTietHD\_HD FOREIGN KEY REFERENCES

HoaDon(MaHD),

MaSP nchar(10) CONSTRAINT FK\_ChiTietHD\_SP FOREIGN KEY REFERENCES

SanPham(MaSP),

SL int check (SL > 0),

DonGia float,

TongTien float

CONSTRAINT PK\_ChiTietHD PRIMARY KEY (MaHD, MaSP)

)

## Bảng hóa đơn ứng dụng

CREATE TABLE HoaDonUngDung(

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

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

MaUngDung nchar(10) CONSTRAINT FK\_HoaDonUngDung\_UD FOREIGN KEY REFERENCES

UngDung(MaUngDung),

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

NhanVien(MaNV),

TriGiaDH int NOT NULL

)

## Bảng chi tiết hoá đơn qua ứng dụng

CREATE TABLE ChiTietHoaDonUngDung(

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

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

SL int check (SL>0), DonGia float,

TongTien float

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

)

## Bảng các ca làm việc

CREATE TABLE CaLamViec(

MaCa nchar(10),

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

GioBatDau nchar(10),

GioKetThuc nchar(10),

CONSTRAINT PK\_CaLamViec PRIMARY KEY (MaCa, NgayLam)

)

## Bảng phân ca làm cho nhân viên

CREATE TABLE BangPhanCa(

MaCa nchar(10),

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

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

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

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

NgayLam)

)

## Bảng phiếu chi

CREATE TABLE PhieuChi(

MaPhieuChi nchar(10) CONSTRAINT PK\_PHIEUCHI PRIMARY KEY,

MaDonNH nchar(10) CONSTRAINT FK\_PhieuChi\_DonNH FOREIGN KEY REFERENCES DonNhapHang(MaDonNH),

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

SoTienChi float

)

## Bảng chế biến

CREATE TABLE CheBien(

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

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

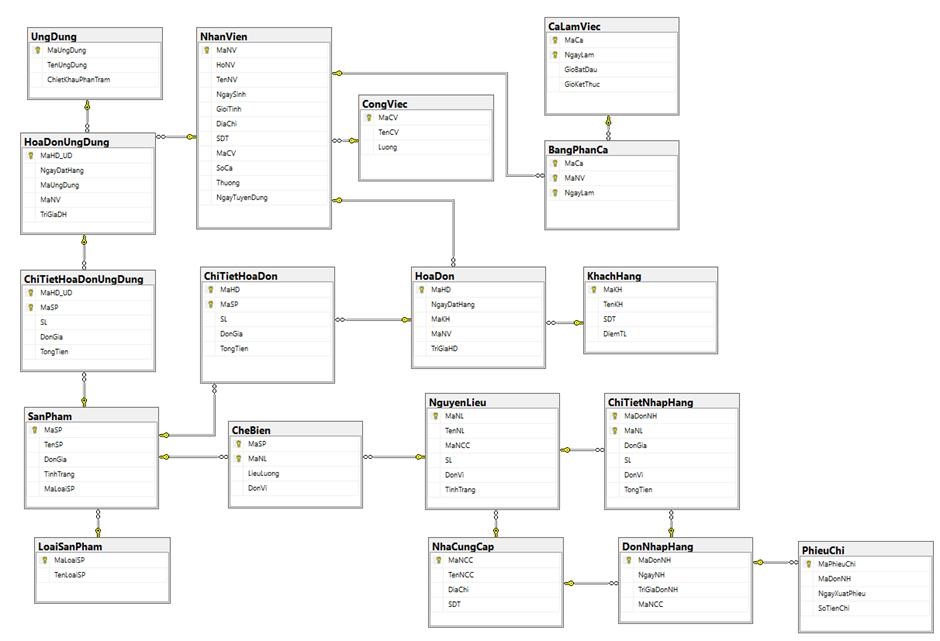
NguyenLieu(MaNL),

LieuLuong int,

DonVi nchar(10)

CONSTRAINT PK\_CheBien PRIMARY KEY (MaSP,MaNL)

)



16

5. **Các view**

## a. Xem ca làm việc của nhân viên trong ngày

CREATE VIEW V\_CaLamTrongNgay AS

SELECT clv.\*, nv.TenNV, nv.MaCV, cv.TenCV

FROM dbo.BangPhanCa pc

JOIN dbo.NhanVien nv ON pc.MaNV = nv.MaNV

JOIN dbo.CaLamViec clv ON pc.MaCa = clv.MaCa

JOIN dbo.CongViec cv ON nv.MaCV = cv.MaCV WHERE clv.NgayLam = CONVERT(DATE, GETDATE())

## b. Xem danh sách sản phẩm

--View danh sách sản phẩm

CREATE VIEW V\_DanhSachSanPham AS

SELECT MaSP, TenSP, DonGia, TinhTrang

FROM dbo.SanPham

GO

### c. Xem số lượng sản phẩm đã bán trong ngày

CREATE VIEW V\_SoLuongSanPhamDaBanTrongNgay AS

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

AS SoLuongDaBanQuaUD

FROM dbo.SanPham sp

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

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

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

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

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

GROUP BY sp.MaSP

### d. Xem thông tin nhân viên

CREATE VIEW V\_ThongTinNhanVien AS

SELECT nv.TenNV, nv.NgaySinh, nv.SDT, nv.SoCa, nv.NgayTuyenDung, nv.Thuong, cv.TenCV, cv.Luong

FROM dbo.NhanVien nv JOIN dbo.CongViec cv ON nv.MaCV = cv.MaCV

## e. Xem phiếu chi trong ngày

CREATE VIEW V\_PhieuChiTrongNgay AS

SELECT \*

FROM dbo.PhieuChi pc

WHERE pc.NgayXuatPhieu = CONVERT(DATE,GETDATE())

## f. Xem đơn nhập hàng chưa giao

CREATE VIEW vi\_DonNhapHang AS

SELECT \*

FROM dbo.DonNhapHang dnh

WHERE TinhTrang = N'Chưa giao'

6. **Các Trigger**

## a. Trigger đặt trạng thái hoá đơn

CREATE TRIGGER set\_HoaDon\_TrangThai

ON HoaDon

FOR INSERT

AS

BEGIN

UPDATE HoaDon

SET TrangThai = 'Chưa thanh toán'

WHERE MaHD IN (SELECT MaHD FROM INSERTED)

END;

**b. Trigger thay đổi số lượng của nguyên liệu mỗi khi bán được sản phẩm (được tính theo công thức).**

CREATE TRIGGER TG\_CheBienSanPham

ON ChiTietHoaDon

AFTER Insert

AS

Begin

-- Xử lý mã NL CF

Declare @newMaSP nchar(10), @newSL int

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

From inserted ne

IF (@newMaSP = 'SP01' OR @newMaSP = 'SP02' OR @newMaSP = 'SP03')

Begin

UPDATE NguyenLieu

SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong

from CheBien where MaSP=@newMaSP))/1000

WHERE MaNL = 'CF'

End

-- Xử lý mã NL DG

IF (@newMaSP = 'SP01' OR @newMaSP = 'SP02' OR @newMaSP = 'SP03' OR

@newMaSP = 'SP04' OR @newMaSP = 'SP05' OR @newMaSP = 'SP06' OR @newMaSP = 'SP07' OR

@newMaSP = 'SP08' OR @newMaSP = 'SP09' OR @newMaSP = 'SP10' OR @newMaSP = 'SP11')

Begin

UPDATE NguyenLieu

SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong from CheBien

where MaSP=@newMaSP))/1000

WHERE MaNL = 'DG'

End

-- Xử lý mã NL SD

IF (@newMaSP = 'SP02')

Begin

UPDATE NguyenLieu

SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong from CheBien

where MaSP=@newMaSP))/1000

WHERE MaNL = 'SD'

End

-- Xử lý mã NL ST

IF (@newMaSP = 'SP03')

Begin

UPDATE NguyenLieu

SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong

from CheBien

|  |  |  |
| --- | --- | --- |
|  | where MaSP=@newMaSP))/1000  WHERE MaNL = 'ST'  End  -- Xử lý mã NL SRD  IF (@newMaSP = 'SP04' OR @newMaSP = 'SP08')  Begin  UPDATE NguyenLieu  SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong  from CheBien  where MaSP=@newMaSP))/1000  WHERE MaNL = 'SRD' | |
|  | End  -- Xử lý mã N  IF (@newMaSP =  Begin | L SRD  'SP07' OR @newMaSP = 'SP11')  UPDATE NguyenLieu  SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong  from CheBien  where MaSP=@newMaSP))/1000  WHERE MaNL = 'SRT' |
|  | End  -- Xử lý mã N  IF (@newMaSP =  Begin | L SRD  'SP05' OR @newMaSP = 'SP09')  UPDATE NguyenLieu  SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong  from CheBien  where MaSP=@newMaSP))/1000  WHERE MaNL = 'SRV' |
|  | End  -- Xử lý mã N  IF (@newMaSP =  Begin | L SRD  'SP06' OR @newMaSP = 'SP10')  UPDATE NguyenLieu  SET SL = (SL\*1000 - @newSL\*(select TOP 1 LieuLuong  from CheBien  where MaSP=@newMaSP))/1000  WHERE MaNL = 'SRVQ' |
| End | End | |

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

|  |
| --- |
| CREATER TRIGGER TG\_SPHetHang  ON ChiTietHoaDon  AFTER INSERT  AS  BEGIN  Declare @newMaSP nchar(10), @newSL int, @MaNLMin nchar(10), @SLMin int, @LieuLuongCan float  Select @newMaSP = ne.MaSP, @newSL = ne.SL  From inserted ne  --  Select @SLMin=min(NguyenLieu.SL) From NguyenLieu, CheBien  Where NguyenLieu.MaNL = CheBien.MaNL and CheBien.MaSP = @newMaSP  -- |
| Select TOP 1 @MaNLMin = NguyenLieu.MaNL, @LieuLuongCan =  CheBien.LieuLuong  From NguyenLieu, CheBien  Where NguyenLieu.MaNL = CheBien.MaNL and CheBien.MaSP = @newMaSP  and NguyenLieu.SL = @SLMin  --  IF (@newSL\*@LieuLuongCan - @SLMin\*1000 > 0)  BEGIN  UPDATE SanPham  SET TinhTrang = 'Hết hàng'  WHERE MaSP=@newMaSP  ROLLBACK  END  END |

### d. Kiểm tra trùng lặp số điện thoại của khách hàng

CREATE TRIGGER TG\_TrungSDT

ON dbo.KhachHang

AFTER INSERT, UPDATE

AS

BEGIN

-- Kiểm tra số điện thoại vừa thêm có bị trùng lặp

IF EXISTS (

SELECT \*

FROM inserted i

WHERE EXISTS (

SELECT \*

FROM dbo.KhachHang k

WHERE k.SDT = i.SDT AND k.MaKH <> i.MaKH

)

)

BEGIN

-- Nếu trùng thì rollback

--PRINT N'Số điện thoại đã tồn tại'

ROLLBACK;

END END

## e. Kiểm tra tên sản phẩm có bị trùng không

|  |
| --- |
| CREATE TRIGGER TG\_TrungTenSP  ON dbo.SanPham  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra tên sản phẩm vừa thêm có bị trùng lặp IF EXISTS (  SELECT \*  FROM inserted i WHERE EXISTS (  SELECT \*  FROM dbo.SanPham sp  WHERE sp.TenSP = i.TenSP AND sp.MaSP <> i.MaSP ) )  BEGIN  -- Nếu trùng thì rollback  RAISERROR ('Tên sản phẩm bị trùng', 16, 1) |
| ROLLBACK;  END END |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

**1. Kết nối cơ sở dữ liệu**

|  |
| --- |
| public class MY\_DB  {  SqlConnection con = new SqlConnection(@"Data Source=LAPTOP-TSVFN4HJ;Initial  Catalog=QUANLYQUANCOFFEE\_Cur;User Id="+ GLOBAL.username + ";Password=" + GLOBAL.password + ";");    public SqlConnection getConnection  { get {  return con;  }  }  SqlConnection conAdmin = new SqlConnection(@"Data Source=LAPTOP-TSVFN4HJ;Initial  Catalog=QUANLYQUANCOFFEE\_Cur;Integrated Security=True"); public SqlConnection getConnectionAdmin  { get  {  return conAdmin;  }  }      // open the connection  public void openConnection()  {  if (con.State == ConnectionState.Closed)  {  con.Open();  }  }    public void openConnectionAdmin()  {  if (conAdmin.State == ConnectionState.Closed)  {  conAdmin.Open();  }  }      public void closeConnection()  {  if (con.State == ConnectionState.Open)  { con.Close();  }  }    public void closeConnectionAdmin()  {  if (conAdmin.State == ConnectionState.Open)  {  conAdmin.Close();  }  }  } |

## 2. Xem thông tin danh mục

### 2.1. Danh mục nhân viên

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

SELECT MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, CongViec.TenCV, SoCa,

Thuong, NgayTuyenDung, HinhAnh

FROM NhanVien, CongViec

WHERE NhanVien.MaCV = CongViec.MaCV

Thực thi trên C#:

|  |
| --- |
| public DataTable getAllEmployee()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_LoadEmployee", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); return dataTable;  } |

### 2.2. Danh mục bảng phân ca

CREATE VIEW [dbo].[v\_BangPhanCa]

AS

SELECT NhanVien.MaNV, NhanVien.HoNV, NhanVien.TenNV, BangPhanCa.MaCa,

BangPhanCa.NgayTrongTuan

FROM BangPhanCa, NhanVien

WHERE BangPhanCa.MaNV = NhanVien.MaNV

Thực thi trên C#:

|  |
| --- |
| private void displayDivision()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_BangPhanCa", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); current\_DataGridView = 2; dataGridView\_Division.DataSource = null; dataGridView\_Division.DataSource = dataTable; } |

### 2.3. Danh mục ca làm việc

CREATE VIEW [dbo].[v\_CaLamViec]

AS

SELECT \*

FROM CaLamViec

Thực thi trên C#:

|  |
| --- |
| private void displayShift()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_CaLamViec", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); current\_DataGridView = 1; dataGridView\_Division.DataSource = null; dataGridView\_Division.DataSource = dataTable; } |

### 2.4. Danh mục hóa đơn

CREATE VIEW [dbo].[vi\_HoaDonView] AS

SELECT MaHD, NgayDatHang, MaKH, MaNV, TriGiaHD FROM HoaDon;

Thực thi trên C#:

public void load\_tabPage\_Receipt()

{

SqlCommand cmd = new SqlCommand("SELECT \* FROM vi\_HoaDonView", db.getConnection);

|  |
| --- |
| SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable(); da.Fill(dt);  dataGridView3.DataSource = dt;    //Doi ten  dataGridView3.Columns["MaHD"].HeaderText = "Mã hoá đơn"; dataGridView3.Columns["NgayDatHang"].HeaderText = "Ngày đặt hàng"; dataGridView3.Columns["MaKH"].HeaderText = "Mã khách hàng"; dataGridView3.Columns["MaNV"].HeaderText = "Mã nhân viên"; dataGridView3.Columns["TriGiaHD"].HeaderText = "Trị giá hoá đơn";  } |

### 2.5. Danh mục sản phẩm

CREATE VIEW [dbo].[view\_danhSachSanPham] AS

SELECT \*

FROM SanPham

Thực thi trên C#:

|  |
| --- |
| public DataTable getDSSanPham()  {  SqlCommand command = new SqlCommand("Select \* from view\_danhSachSanPham", mydb.getConnection);  DataTable table = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(table); return table;  } |

### 2.6. Danh mục phiếu chi

CREATE VIEW [dbo].[DonNhapHangView]

AS

SELECT DNH.MaDonNH, DNH.NgayNH, DNH.MaNCC, DNH.TinhTrang, PC.MaPhieuChi,

PC.NgayXuatPhieu, CTH.MaNL, CTH.DonGia, CTH.SL, CTH.DonVi, PC.SoTienChi

FROM DonNhapHang DNH

LEFT JOIN PhieuChi PC ON DNH.MaDonNH = PC.MaDonNH

LEFT JOIN ChiTietNhapHang CTH ON DNH.MaDonNH = CTH.MaDonNH;

Thực thi trên C#:

|  |
| --- |
| private void load\_tabPage\_Payment()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM DonNhapHangView", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  dataGridViewPayment.DataSource = dt;  //Header text  dataGridViewPayment.Columns["MaDonNH"].HeaderText = "Mã Đơn Nhập Hàng"; dataGridViewPayment.Columns["NgayNH"].HeaderText = "Ngày nhập hàng"; dataGridViewPayment.Columns["NgayNH"].DefaultCellStyle.Format = "dd/MM/yyyy"; dataGridViewPayment.Columns["MaNCC"].HeaderText = "Mã Nhà Cung Cấp"; dataGridViewPayment.Columns["TinhTrang"].HeaderText = "Tình trạng"; dataGridViewPayment.Columns["MaPhieuChi"].HeaderText = "Mã phiếu chi"; dataGridViewPayment.Columns["NgayXuatPhieu"].HeaderText = "Ngày Xuất Phiếu"; dataGridViewPayment.Columns["NgayXuatPhieu"].DefaultCellStyle.Format = "dd/MM/yyyy"; dataGridViewPayment.Columns["MaNL"].HeaderText = "Mã nguyên liệu"; dataGridViewPayment.Columns["DonGia"].HeaderText = "Đơn giá"; dataGridViewPayment.Columns["SL"].HeaderText = "Số lượng"; dataGridViewPayment.Columns["DonVi"].HeaderText = "Đơn vị"; dataGridViewPayment.Columns["SoTienChi"].HeaderText = "Số tiền chi";  } |

### 2.7. Danh mục nguyên liệu

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

SELECT \*

FROM NguyenLieu

Thực thi trên C#:

|  |
| --- |
| public DataTable xemNguyenLieu()  {  SqlCommand command = new SqlCommand("select \* from view\_NguyenLieu", mydb.getConnection); DataTable dt = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(dt); return dt;  } |

### 2.8. Danh mục nhà cung cấp

CREATE VIEW [dbo].[view\_nhaCungCap] AS

SELECT \*

FROM dbo.NhaCungCap

Thực thi trên C#:

|  |
| --- |
| private void themnguyenlieuForm\_Load(object sender, EventArgs e)  {  DataTable tbNCC = new DataTable();  SqlCommand command = new SqlCommand("select \* from view\_nhaCungCap", mydb.getConnection);  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(tbNCC);    DataTable tbNguyenLieu = new DataTable(); tbNguyenLieu = nguyenLieu.getNguyenLieu();    cbNCC.DataSource = tbNCC; cbNCC.DisplayMember = "TenNCC"; cbNCC.Text = "";    cbTenNL.DataSource = tbNguyenLieu; cbTenNL.DisplayMember = "TenNL"; cbTenNL.Text = "";    cbTinhTrang.Items.Add("Còn hàng"); cbTinhTrang.Items.Add("Hết hàng");      tbMaNL.Enabled = false; cbTenNL.DropDownStyle = ComboBoxStyle.DropDownList; cbNCC.Enabled = false; tbDonVi.Enabled = false; } |

1. **Quản lý khách hàng**

**3.1. Hàm tìm kiếm khách hàng**

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

RETURNS nvarchar(50)

AS

BEGIN

DECLARE @TenKH nvarchar(50);

SELECT @TenKH = TenKH FROM KhachHang WHERE SDT = @SDT;

RETURN @TenKH;

END

Thực thi trên C#:

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

{

string sdt = textBoxcartsdtkh.Text.Trim();

|  |
| --- |
| db.openConnection();  SqlCommand cmd = new SqlCommand("SELECT dbo.SearchTenKHBySDT(@SDT)", db.getConnection); cmd.Parameters.AddWithValue("@SDT", sdt); object result = cmd.ExecuteScalar();    if (result != DBNull.Value && result != null)  {  string tenKH = (string)result; // Display the TenKH in the label textBoxcarttenkhachhang.Text = tenKH; db.closeConnection();  } else  {  // If the KhachHang doesn't exist, open the NewKhachHangForm to create a new one NewKhachHangForm newKhachHangForm = new NewKhachHangForm(sdt); newKhachHangForm.ShowDialog(); db.closeConnection();  }  } |

**3.2. Tạo khách hàng mới**

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

Thực thi trên C#:

|  |
| --- |
| private void saveButton\_Click(object sender, EventArgs e)  {  string MaKH = textBoxMaKH.Text; string TenKH = textBoxTenKH.Text; string SDT = textBoxSDT.Text;    if (string.IsNullOrEmpty(MaKH))  {  MessageBox.Show("Please enter a valid value for 'MaKH'."); return;  }    db.openConnection();  SqlCommand cmd = new SqlCommand("InsertNewKhachHang", db.getConnection); cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaKH", SqlDbType.NChar).Value = MaKH; cmd.Parameters.Add("@TenKH", SqlDbType.NVarChar).Value = TenKH; cmd.Parameters.Add("@SDT", SqlDbType.NChar).Value = SDT; |
| if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công!", "Add Customer", MessageBoxButtons.OK,  MessageBoxIcon.Information);  db.closeConnection();  } else  {  MessageBox.Show("Thêm thất bại", "Add Customer", MessageBoxButtons.OK,  MessageBoxIcon.Error);  db.closeConnection();  }  } |

**3.3. Trigger bắt lỗi khi thêm khách hàng mới**

|  |
| --- |
| CREATE TRIGGER trg\_InsertNewKhachHang  ON KhachHang  FOR INSERT, UPDATE  AS  BEGIN  -- check MaKH  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaKH) = ' ')  BEGIN  RAISERROR('Mã KH không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT \* FROM KhachHang WHERE MaKH IN (SELECT MaKH FROM inserted))  BEGIN  RAISERROR('Mã KH đã tồn tại', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check ten KH  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenKH) = ' ') BEGIN  RAISERROR('Tên KH không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check SDT  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(SDT) = ' ') BEGIN  RAISERROR('Mã KH không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT \* FROM KhachHang WHERE SDT IN (SELECT SDT FROM inserted))  BEGIN  RAISERROR('SDT đã tồn tại', 16, 1)  ROLLBACK TRANSACTION  RETURN  END END; |

1. **Quản lý hóa đơn**

**4.1. Xuất hóa đơn**

CREATE PROCEDURE [dbo].[proc\_AddHoaDon]

@MaHD nchar(10),

@NgayDatHang date,

@MaKH nchar(10),

@MaNV nchar(10),

@TriGiaHD int

AS

BEGIN

INSERT INTO HoaDon (MaHD, NgayDatHang, MaKH, MaNV, TriGiaHD)

VALUES (@MaHD, @NgayDatHang, @MaKH, @MaNV, @TriGiaHD); END;

Thực thi trên C#:

|  |
| --- |
| private void addButton\_Click(object sender, EventArgs e)  {  MY\_DB db = new MY\_DB();  // Get the parameter values from the text boxes string maHD = maHDTextBox.Text;  DateTime ngayDatHang = DateTime.Today;    string maKH = maKHTextBox.Text; string maNV = maNVTextBox.Text; int triGiaHD;  if (int.TryParse(triGiaHDTextBox.Text, out triGiaHD) == false)  {  MessageBox.Show("TriGiaHD must be a valid integer."); return; }  // Set up the stored procedure name and parameter names string storedProcedureName = "proc\_AddHoaDon";  string[] parameterNames = new string[] { "@MaHD", "@NgayDatHang", "@MaKH", "@MaNV",  "@TriGiaHD" };  string returnParameterName = "@ReturnValue";    using (SqlCommand command = new SqlCommand(storedProcedureName, db.getConnection))  {  command.CommandType = System.Data.CommandType.StoredProcedure;    // Add the parameters  command.Parameters.Add(parameterNames[0], System.Data.SqlDbType.NChar, 10).Value = maHD; command.Parameters.Add(parameterNames[1], System.Data.SqlDbType.Date).Value = ngayDatHang; command.Parameters.Add(parameterNames[2], System.Data.SqlDbType.NChar, 10).Value = maKH; command.Parameters.Add(parameterNames[3], System.Data.SqlDbType.NChar, 10).Value = maNV; command.Parameters.Add(parameterNames[4], System.Data.SqlDbType.Int).Value = triGiaHD;    // Open the connection db.openConnection();    // Execute the command  int rowsAffected = command.ExecuteNonQuery();    // Check the result if (rowsAffected > 0)  {  MessageBox.Show("Xuất thành công."); |
| } else {  MessageBox.Show("Xuất thất bại.");  }    }  } |

1. **2.Tìm kiếm hóa đơn**

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

Thực thi trên C#:

|  |
| --- |
| private void btnsearchReceipt\_Click(object sender, EventArgs e)  {  SqlCommand command = new SqlCommand("proc\_TimKiemHoaDon", db.getConnection); command.CommandType = CommandType.StoredProcedure; command.Parameters.AddWithValue("@NgayBatDau", dateTimePicker1receipt.Value); command.Parameters.AddWithValue("@NgayKetThuc", dateTimePicker2receipt.Value); db.openConnection();  DataTable table = new DataTable();  using (SqlDataAdapter adapter = new SqlDataAdapter(command))  { adapter.Fill(table);  }  dataGridView3.DataSource = table; db.closeConnection();  } |

1. **Quản lý kho nguyên liệu** 
   1. **Tìm kiếm nguyên liệu**

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_getIngreListByString] (@string NVARCHAR(50)) RETURNS @IngreList TABLE (MaNL VARCHAR(10), TenNL NVARCHAR(50), MaNCC VARCHAR(10), SoLuong INT, DonVi NVARCHAR(10), TinhTrang NVARCHAR(10)) AS  BEGIN  INSERT INTO @IngreList SELECT \*  FROM dbo.NguyenLieu  WHERE CONCAT(MaNL, TenNL, MaNCC, DonVi, TinhTrang) LIKE N'%' + @string + '%'    RETURN  END |

Thực thi trên C#:

public DataTable timNguyenLieu(string str)

{

string fnName = "func\_getIngreListByString";

|  |
| --- |
| SqlCommand command = new SqlCommand("Select \* from " + fnName + " (@string)", mydb.getConnection);  command.Parameters.Add("@string", SqlDbType.NChar).Value = str;  DataTable table = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(table); return table;  } |

* 1. **Thêm nguyên liệu**

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_themNguyenLieu]  @MaNL nchar(10),  @TenNL nvarchar(50),  @MaNCC nchar(10), @SoLuong int,  @DonVi nchar(10),  @TinhTrang nchar(10)  AS  BEGIN  BEGIN TRY  Thêm mới sản phẩm  INSERT INTO dbo.NguyenLieu (MaNL, TenNL, MaNCC, SL, DonVi, TinhTrang)  VALUES (@MaNL, @TenNL, @MaNCC, @SoLuong, @DonVi, @TinhTrang)  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE() RAISERROR(@err, 16, 1)  END CATCH END |

Thực thi trên C#:

|  |
| --- |
| public bool themNguyenLieu(string maNl, string tenNl, string maNCC, int soLuong, string donVi, string tinhTrang)  {  SqlCommand command = new SqlCommand("proc\_themNguyenLieu", mydb.getConnection); command.CommandType = CommandType.StoredProcedure; command.Parameters.Add("@MaNl", SqlDbType.NChar).Value = maNl; command.Parameters.Add("@TenNl", SqlDbType.NVarChar).Value = tenNl; command.Parameters.Add("@MaNCC", SqlDbType.NChar).Value = maNCC; command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = soLuong; command.Parameters.Add("@DonVi", SqlDbType.NChar).Value = donVi; command.Parameters.Add("@TinhTrang", SqlDbType.NChar).Value = tinhTrang;    mydb.openConnection();    if (command.ExecuteNonQuery() == 1)  {  mydb.closeConnection();  return true;  } else  {  mydb.closeConnection();  return false;  }  } |

1. **Quản lý sản phẩm** 
   1. **Thêm sản phẩm**

CREATE PROCEDURE [dbo].[proc\_themSanPhamMoi]

@MaSP nchar(10),

@TenSP nvarchar(50),

@DonGia float,

@TinhTrang nchar(10),

@MaLoaiSP nchar(10),

@TenLoaiSP nvarchar(50),

@HinhAnh image

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Kiểm tra xem loại sản phẩm đã tồn tại hay chưa

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

@MaLoaiSP)

BEGIN

-- Nếu chưa tồn tại, thêm mới loại sản phẩm

INSERT INTO LoaiSanPham (MaLoaiSP, TenLoaiSP)

VALUES (@MaLoaiSP, @TenLoaiSP)

END

-- Thêm mới sản phẩm

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

HinhAnh)

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

@HinhAnh)

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @err NVARCHAR(MAX)

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

RAISERROR(@err, 16, 1)

END CATCH

END

Thực thi trên C#:

public bool themSanPham(string MaSP, string TenSP, float DonGia, string TinhTrang, string MaLoaiSP, string TenLoaiSP, MemoryStream HinhAnh)

{

SqlCommand command = new SqlCommand("proc\_themSanPhamMoi", mydb.getConnection); command.CommandType = CommandType.StoredProcedure; command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP; command.Parameters.Add("@TenSP", SqlDbType.NVarChar).Value = TenSP; command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia; command.Parameters.Add("@TinhTrang", SqlDbType.NChar).Value = TinhTrang; command.Parameters.Add("@MaLoaiSP", SqlDbType.NChar).Value = MaLoaiSP; command.Parameters.Add("@TenLoaiSP", SqlDbType.NVarChar).Value = TenLoaiSP; command.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = HinhAnh.ToArray();

mydb.openConnection();

if (command.ExecuteNonQuery() > 0)

{

mydb.closeConnection();

return true;

|  |
| --- |
| } else  {  mydb.closeConnection();  return false;  }  } |

**6.2. Sửa sản phẩm**

CREATE PROCEDURE [dbo].[proc\_suaSanPham]

@MaSP nchar(10),

@TenSP nvarchar(50),

@DonGia float,

@TinhTrang nchar(10),

@HinhAnh image

AS

BEGIN

BEGIN TRY

-- Thêm mới sản phẩm

UPDATE dbo.SanPham SET MaSP = @MaSP, TenSP = @TenSP, DonGia =

@DonGia,

TinhTrang = @TinhTrang, HinhAnh =

@HinhAnh

WHERE MaSP = @MaSP

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

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

RAISERROR(@err, 16, 1)

END CATCH

END

Thực thi trên C#:

|  |
| --- |
| public bool suaSanPham(string MaSP, string TenSP, float DonGia, string TinhTrang, MemoryStream HinhAnh)  {  SqlCommand command = new SqlCommand("proc\_suaSanPham", mydb.getConnection); command.CommandType = CommandType.StoredProcedure; command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = MaSP; command.Parameters.Add("@TenSP", SqlDbType.NVarChar).Value = TenSP; command.Parameters.Add("@DonGia", SqlDbType.Float).Value = DonGia; command.Parameters.Add("@TinhTrang", SqlDbType.NChar).Value = TinhTrang; command.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = HinhAnh.ToArray();    mydb.openConnection();    if (command.ExecuteNonQuery() > 0)  {  mydb.closeConnection();  return true;  } else  {  mydb.closeConnection();  return false;  }  } |

1. **3.Xóa sản phẩm**

CREATE PROCEDURE [dbo].[proc\_xoaSanPham] @MaSP nchar(10)

AS

BEGIN

|  |
| --- |
| BEGIN TRANSACTION BEGIN TRY  --Xoá nguyên liệu tiêu tốn theo @MaSP trong bảng CheBien  DELETE FROM dbo.CheBien WHERE CheBien.MaSP = @MaSP    --Xoá sản phẩm theo @MaSP trong bảng SanPham  DELETE FROM dbo.SanPham WHERE SanPham.MaSP = @MaSP COMMIT TRAN  END TRY    BEGIN CATCH ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1) END CATCH  END |

Thực thi trên C#:

|  |
| --- |
| public bool xoaSanPham(string maSP)  {  SqlCommand command = new SqlCommand("dbo.proc\_xoaSanPham", mydb.getConnection); command.CommandType = CommandType.StoredProcedure;  command.Parameters.Add("@MaSP", SqlDbType.NChar).Value = maSP;    mydb.openConnection();    if (command.ExecuteNonQuery() > 0)  {  mydb.closeConnection();  return true;  } else  {  mydb.closeConnection();  return false;  }  } |

* 1. **Tìm kiếm sản phẩm**

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_getProductListByString] (@string NVARCHAR(50))  RETURNS @ProductList TABLE (MaSP VARCHAR(10), TenSP NVARCHAR(50), DonGia INT,  TinhTrang NVARCHAR(10), MaLoaiSP VARCHAR(10), HinhAnh IMAGE) AS  BEGIN  INSERT INTO @ProductList  SELECT MaSP, TenSP, DonGia, TinhTrang, MaLoaiSP, HinhAnh  FROM dbo.SanPham  WHERE CONCAT(MaSP, TenSP, TinhTrang, MaLoaiSP) LIKE '%' + @string + '%'  RETURN END |

Thực thi trên C#:

public DataTable getDSSanPhamByString(string str)

{

string fnName = "func\_getProductListByString";

|  |
| --- |
| SqlCommand command = new SqlCommand("Select \* from " + fnName + " (@string)", mydb.getConnection);  command.Parameters.Add("@string", SqlDbType.NChar).Value = str;  DataTable table = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(table); return table;  } |

* 1. **Trigger bắt lỗi trùng tên khi thêm và chỉnh sửa sản phẩm**

|  |
| --- |
| CREATE TRIGGER TG\_TrungTenSP  ON dbo.SanPham  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra tên sản phẩm vừa thêm có bị trùng lặp  IF EXISTS (  SELECT \*  FROM inserted i  WHERE EXISTS (  SELECT \*  FROM dbo.SanPham sp  WHERE sp.TenSP = i.TenSP AND sp.MaSP <> i.MaSP  ) )  BEGIN  -- Nếu trùng thì rollback  RAISERROR ('Tên sản phẩm bị trùng', 16, 1)  ROLLBACK;  END  END |

1. **Thống kê doanh thu và chi phí** 
   1. **Doanh thu theo ngày, tháng, năm**

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

DECLARE @doanhThu float = 0;

SELECT @doanhthu = COALESCE(SUM(TriGiaHD), 0)

FROM HoaDon

WHERE YEAR(NgayDatHang) = @nam;

RETURN @doanhThu;

END;

* 1. **Chi phí theo tháng, năm**

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_tinhTongChiPhiThang](@thang INT, @nam INT) RETURNS float BEGIN  DECLARE @chiphi float;  SELECT @chiphi = SUM(SoTienChi)  FROM PhieuChi  WHERE MONTH(NgayXuatPhieu) = @thang AND YEAR(NgayXuatPhieu) = @nam;  RETURN @chiphi;  END;    CREATE FUNCTION [dbo].[func\_tinhTongChiPhiNam](@nam INT) RETURNS float BEGIN  DECLARE @chiphi float;  SELECT @chiphi = SUM(SoTienChi)  FROM PhieuChi  WHERE YEAR(NgayXuatPhieu) = @nam;  RETURN @chiphi;  END; |

Thực thi trên C# của doanh thu và chi phí:

|  |
| --- |
| public void load\_tabPage\_Statics()  {  db.openConnection();  SqlCommand cmd = new SqlCommand("SELECT dbo.func\_tinhDoanhThuNgay(@ngay, @thang,  @nam)", db.getConnection); cmd.Parameters.AddWithValue("@ngay", int.Parse(DateTime.Today.ToString("dd"))); cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM"))); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy"))); object result = cmd.ExecuteScalar(); double doanhthutheothang = (double)result; // change data type to double string doanhthuString = doanhthutheothang.ToString(); // convert to string  labeldoanhthutheongay.Text = "Doanh thu theo ngày: " + doanhthuString + " VND"; // set label text db.closeConnection();    db.openConnection();  cmd = new SqlCommand("SELECT dbo.func\_tinhDoanhThuThang(@thang, @nam)", db.getConnection); cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM"))); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy"))); result = cmd.ExecuteScalar();  doanhthutheothang = (double)result; // change data type to double doanhthuString = doanhthutheothang.ToString(); // convert to string labeldoanhthutheothang.Text = "Doanh thu theo tháng: " + doanhthuString + " VND"; // set label text db.closeConnection();    db.openConnection();  cmd = new SqlCommand("SELECT dbo.func\_tinhDoanhThuNam(@nam)", db.getConnection); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy"))); result = cmd.ExecuteScalar(); |
| doanhthutheothang = (double)result; // change data type to double doanhthuString = doanhthutheothang.ToString(); // convert to string  labeldoanhthutheonam.Text = "Doanh thu theo năm: " + doanhthuString + " VND"; // set label text db.closeConnection();    db.openConnection();  cmd = new SqlCommand("SELECT dbo.func\_tinhTongChiPhiThang(@thang, @nam)", db.getConnection); cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM"))); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy"))); result = cmd.ExecuteScalar(); doanhthutheothang = (double)result; // change data type to double doanhthuString = doanhthutheothang.ToString(); // convert to string  labelchiphithang.Text = "Chi phí theo tháng: " + doanhthuString + " VND"; // set label text db.closeConnection();    db.openConnection();  cmd = new SqlCommand("SELECT dbo.func\_tinhTongChiPhiNam(@nam)", db.getConnection); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy"))); result = cmd.ExecuteScalar(); doanhthutheothang = (double)result; // change data type to double doanhthuString = doanhthutheothang.ToString(); // convert to string labelchiphitheonam.Text = "Chi phí theo năm: " + doanhthuString + " VND"; // set label text db.closeConnection(); } |

1. **Quản lý nhân viên** 
   1. **Thêm nhân viên**

CREATE PROC [dbo].[proc\_AddEmployee]

@manv nchar(10), @ho nvarchar(10), @ten nvarchar(10), @ns date, @gt nvarchar(3), @dc nvarchar(100),

@sdt nchar(11), @macv nchar(10), @soca int, @thuong int, @ntd date, @ha image AS

INSERT INTO NhanVien (MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT,

MaCV, SoCa, Thuong, NgayTuyenDung, HinhAnh)

VALUES(@manv, @ho, @ten, @ns, @gt, @dc, @sdt, @macv, @soca, @thuong, @ntd, @ha)

Thực thi trên C#:

|  |
| --- |
| public bool Insert(string id, string ho, string ten, DateTime bd, string gd, string adr, string phn, ComboBox cv, int soca, int thuong, DateTime hd, MemoryStream pic)  { try  {  SqlCommand cmd = new SqlCommand("EXEC proc\_AddEmployee @manv, @ho, @ten, @ns, @gt, @dc, @sdt, @macv, @soca, @thuong, @ntd, @ha", db.getConnection);    cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = id; cmd.Parameters.Add("@ho", SqlDbType.NVarChar, 10).Value = ho; cmd.Parameters.Add("@ten", SqlDbType.NVarChar, 10).Value = ten; cmd.Parameters.Add("@ns", SqlDbType.Date).Value = bd; cmd.Parameters.Add("@gt", SqlDbType.NVarChar, 3).Value = gd; cmd.Parameters.Add("@dc", SqlDbType.NVarChar, 100).Value = adr; cmd.Parameters.Add("@sdt", SqlDbType.NVarChar, 11).Value = phn;  cmd.Parameters.Add("@macv", SqlDbType.NChar, 10).Value = cv.SelectedValue; cmd.Parameters.Add("@soca", SqlDbType.Int).Value = soca; cmd.Parameters.Add("@thuong", SqlDbType.Int).Value = thuong; cmd.Parameters.Add("@ntd", SqlDbType.Date).Value = hd;  cmd.Parameters.Add("@ha", SqlDbType.Image).Value = pic.ToArray(); |
| db.openConnection(); try  {  cmd.ExecuteNonQuery(); db.closeConnection();  return true;  } catch  {  db.closeConnection();  return false;  } } catch {  return false;  }  } |

* 1. **Xóa nhân viên**

CREATE PROCEDURE [dbo].[proc\_DeleteEmployee]

@MaNV nchar(10)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRANSACTION;

BEGIN TRY

-- Xóa dòng trong bảng OrderDetails có OrderID tương ứng

DELETE FROM BangPhanCa WHERE MaNV = @MaNV;

-- Xóa dòng trong bảng Orders có OrderID tương ứng

DELETE FROM NhanVien WHERE MaNV = @MaNV;

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

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

RAISERROR(@err, 16, 1)

ROLLBACK TRANSACTION;

THROW;

END CATCH

COMMIT TRANSACTION;

END

Thực thi trên C#:

|  |
| --- |
| private void remove\_Employee()  { try  {  DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn xóa nhân viên này không?", "Remove employee", MessageBoxButtons.YesNo, MessageBoxIcon.Error); if (result == DialogResult.Yes)  {  string ID = dataGridView\_Employees.CurrentRow.Cells[0].Value.ToString();    SqlCommand cmd = new SqlCommand("EXEC proc\_DeleteEmployee @manv", db.getConnection); cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = ID;    db.openConnection();  if (cmd.ExecuteNonQuery() == -1)  {  MessageBox.Show("Xóa thành công!", "Remove employee", MessageBoxButtons.OK, MessageBoxIcon.Information); |
| db.closeConnection();  reset\_Employee();  } else  {  MessageBox.Show("Xóa thất bại!", "Remove employee", MessageBoxButtons.OK,  MessageBoxIcon.Error);  db.closeConnection();  }  } } catch  {  MessageBox.Show("Xóa thất bại!", "Remove employee", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

**8.3. Tìm kiếm nhân viên**

CREATE PROC [dbo].[pro\_SearchByMaNV]

@MaNV nchar(10)

AS

BEGIN

SELECT \*

FROM v\_LoadEmployee

WHERE MaNV = @MaNV

END

CREATE PROC [dbo].[pro\_SearchBySDT]

@SDT nchar(11)

AS

BEGIN

SELECT \*

FROM v\_LoadEmployee

WHERE SDT = @SDT

END

CREATE PROC [dbo].[pro\_SearchByTenCV]

@TenCV nvarchar(50)

AS

BEGIN

SELECT \*

FROM v\_LoadEmployee

WHERE TenCV = @TenCV

END

CREATE PROC [dbo].[pro\_SearchByTenNV]

@TenNV nvarchar(10)

AS

BEGIN

SELECT \*

FROM v\_LoadEmployee

WHERE TenNV = @TenNV

END

Thực thi trên C#:

|  |
| --- |
| private void search\_Employee()  {  if (txt\_Search\_Employee.Text == "")  {  load\_tabPage\_Employees();  } else |
| {  SqlCommand cmd = new SqlCommand("", db.getConnection);    if (comBox\_Search\_Employee.Text == "Tìm theo Mã NV")  {  cmd.CommandText = "EXEC pro\_SearchByMaNV @manv";  cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = txt\_Search\_Employee.Text;  }  else if (comBox\_Search\_Employee.Text == "Tìm theo Tên")  {  cmd.CommandText = "EXEC pro\_SearchByTenNV @tennv";  cmd.Parameters.Add("@tennv", SqlDbType.NChar, 10).Value = txt\_Search\_Employee.Text;  }  else if (comBox\_Search\_Employee.Text == "Tìm theo SDT")  {  cmd.CommandText = "EXEC pro\_SearchBySDT @sdt";  cmd.Parameters.Add("@sdt", SqlDbType.NChar, 11).Value = txt\_Search\_Employee.Text;  }  else if (comBox\_Search\_Employee.Text == "Tìm theo Công Việc")  {  cmd.CommandText = "EXEC pro\_SearchByTenCV @tencv";  cmd.Parameters.Add("@tencv", SqlDbType.NVarChar, 50).Value = txt\_Search\_Employee.Text;  }  DataTable dt = employee.getEmployeeWithCommand(cmd); if (dt.Rows.Count > 0)  {  dataGridView\_Employees.DataSource = dt;  } else  {  MessageBox.Show("Không tìm thấy!", "Search employee", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  } |

**8.4. Tính lương nhân viên**

|  |
| --- |
| CREATE FUNCTION func\_tinhLuongNV(@thang INT, @nam INT) RETURNS TABLE AS RETURN (  SELECT nv.MaNV,nv.HoNV,nv.TenNV ,nv.SoCa,  (nv.SoCa\*cv.Luong\*4 + nv.Thuong\*(cv.Luong\*2)) +  CASE WHEN dbo.func\_tinhDoanhThuThang(@thang, @nam) > 100000000 THEN dbo.func\_tinhDoanhThuThang(@thang, @nam)\*0.01 ELSE 0  END AS Luong  FROM NhanVien nv  INNER JOIN CongViec cv ON nv.MaCV = cv.MaCV ) |

Code C# gọi danh mục:

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

{

MY\_DB db = new MY\_DB(); db.openConnection();

SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_tinhLuongNV(@thang, @nam)", db.getConnection);

cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM"))); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy")));

|  |
| --- |
| // Create a data adapter to fill a dataset with the results  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataSet ds = new DataSet(); da.Fill(ds);    dataGridViewluong.DataSource = ds.Tables[0]; db.closeConnection(); } |

**8.5. Sửa thông tin nhân viên**

|  |
| --- |
| CREATE PROC [dbo].[proc\_EditEmployee]  @manv nchar(10), @ho nvarchar(10), @ten nvarchar(10), @ngaysinh date, @gioitinh nvarchar(3), @diachi nvarchar(100), @sdt nchar(11), @macv nchar(10),  @soca int, @thuong int, @ngaytuyendung date, @hinhanh image AS  BEGIN  BEGIN TRY  UPDATE NhanVien  SET  HoNV = @ho, TenNV = @ten, NgaySinh = @ngaysinh,  GioiTinh = @gioitinh, DiaChi = @diachi, SDT = @sdt, MaCV = @macv,  SoCa = @soca, Thuong = @thuong, NgayTuyenDung = @ngaytuyendung, HinhAnh = @hinhanh  WHERE MaNV = @manv  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END |

Code C# gọi danh mục:

|  |
| --- |
| public bool Update(string id, string ho, string ten, DateTime bd, string gd, string adr, string phn, ComboBox cv, int soca, int thuong, DateTime hd, MemoryStream pic)  { try  {  SqlCommand cmd = new SqlCommand("EXEC proc\_EditEmployee @manv, @ho, @ten, @ns, @gt, @dc, @sdt, @macv, @soca, @thuong, @ntd, @ha", db.getConnection);    cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = id; cmd.Parameters.Add("@ho", SqlDbType.NVarChar, 10).Value = ho; cmd.Parameters.Add("@ten", SqlDbType.NVarChar, 10).Value = ten; cmd.Parameters.Add("@ns", SqlDbType.Date).Value = bd; cmd.Parameters.Add("@gt", SqlDbType.NVarChar, 3).Value = gd; cmd.Parameters.Add("@dc", SqlDbType.NVarChar, 100).Value = adr; cmd.Parameters.Add("@sdt", SqlDbType.NVarChar, 11).Value = phn;  cmd.Parameters.Add("@macv", SqlDbType.NChar, 10).Value = cv.SelectedValue; cmd.Parameters.Add("@soca", SqlDbType.Int).Value = soca; cmd.Parameters.Add("@thuong", SqlDbType.Int).Value = thuong; cmd.Parameters.Add("@ntd", SqlDbType.Date).Value = hd;  cmd.Parameters.Add("@ha", SqlDbType.Image).Value = pic.ToArray(); db.openConnection(); try  {  cmd.ExecuteNonQuery(); db.closeConnection();  return true;  } catch |
| {  db.closeConnection();  return false;  } } catch {  return false;  }  } |

**8.6. Trigger bắt lỗi khi thêm hoặc sửa thông tin nhân viên**

|  |
| --- |
| CREATE TRIGGER trg\_CheckNhanVien  ON NhanVien  FOR INSERT, UPDATE  AS  BEGIN  -- check MaNV  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(MaNV) = ' ') BEGIN  RAISERROR('Mã nhân viên không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT \* FROM NhanVien WHERE MaNV IN (SELECT MaNV FROM inserted))  BEGIN  RAISERROR('Mã nhân viên đã tồn tại', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check ho ten nhan vien  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(TenNV) = '' ) BEGIN  RAISERROR('Tên nhân viên không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(HoNV) = '') BEGIN  RAISERROR('Họ nhân viên không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NhanVien n ON i.MaNV != n.MaNV AND TRIM(i.HoNV) = TRIM(n.HoNV) AND TRIM(i.TenNV) = TRIM(n.TenNV))  BEGIN  RAISERROR('Họ tên nhân viên đã tồn tại', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check ngay sinh  IF EXISTS (  SELECT \* FROM inserted WHERE  datediff(year,inserted.NgaySinh,getdate())<(18) )  BEGIN  RAISERROR ('Nhân viên phải trên 18 tuổi', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check dia chi |
| IF EXISTS (SELECT \* FROM inserted WHERE TRIM(DiaChi) = '' )  BEGIN  RAISERROR('Địa chỉ không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check so dien thoai  IF EXISTS (SELECT \* FROM inserted WHERE TRIM(SDT) = '' ) BEGIN  RAISERROR('Số điện thoại không được để trống', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT \* FROM inserted WHERE len(TRIM(SDT)) = (10))  BEGIN  RAISERROR('Số điện thoại gồm 10 chữ số', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NhanVien n ON i.MaNV != n.MaNV AND TRIM(i.SDT) = TRIM(n.SDT))  BEGIN  RAISERROR('Số điện thoại đã tồn tại', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  -- check ngay tuyen dung  IF NOT EXISTS (  SELECT \* FROM inserted WHERE  (datediff(day,[NgayTuyenDung],getdate())>=(0))  )  BEGIN  RAISERROR ('Ngày tuyển dụng không thể là trong tương lai', 16, 1) ROLLBACK TRANSACTION  RETURN  END  END |

1. **Quản lý phiếu chi** 
   1. **Thêm phiếu chi**

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

Code C# thực thi:

|  |
| --- |
| private void btnaddPayment\_Click(object sender, EventArgs e)  {  MY\_DB db = new MY\_DB();  SqlCommand command = new  SqlCommand("proc\_InsertDonNhapHangAndPhieuChiAndChiTietNhapHang", db.getConnection); command.CommandType = CommandType.StoredProcedure;  float TongTien = float.Parse(textBoxDonGia.Text) \* float.Parse(textBoxSL.Text);  // Add parameters for the DonNhapHang table  command.Parameters.AddWithValue("@MaDonNH", textBoxMaDonNH.Text); command.Parameters.AddWithValue("@NgayNH", DateTime.Today); command.Parameters.AddWithValue("@TriGiaDonNH", TongTien); command.Parameters.AddWithValue("@MaNCC", textBoxMaNCC.Text); command.Parameters.AddWithValue("@TinhTrang", "Đang giao");    // Add parameters for the PhieuChi table  command.Parameters.AddWithValue("@MaPhieuChi", textBoxMaPhieuChi.Text); command.Parameters.AddWithValue("@NgayXuatPhieu", DateTime.Today); command.Parameters.AddWithValue("@SoTienChi", TongTien);    // Add parameters for the ChiTietNhapHang table  command.Parameters.AddWithValue("@MaNL", textBoxMaNL.Text);  command.Parameters.AddWithValue("@DonGia", float.Parse(textBoxDonGia.Text)); command.Parameters.AddWithValue("@SL", int.Parse(textBoxSL.Text)); command.Parameters.AddWithValue("@DonVi", textBoxDonVi.Text); command.Parameters.AddWithValue("@TongTien", TongTien);        db.openConnection(); int rowsAffected = command.ExecuteNonQuery();    // Check the result  if (rowsAffected > 0)  {  MessageBox.Show("Xuất thành công.");  } else {  MessageBox.Show("Xuất thất bại.");  }  db.closeConnection(); } |

1. **Quản lý phân ca** 
   1. **Thêm ca cho nhân viên**

CREATE PROC [dbo].[proc\_PhanCa]

@maca nchar(10),

@manv nchar(10),

@dayOfWeek nchar(10)

AS

BEGIN

BEGIN TRY

INSERT INTO BangPhanCa

VALUES(@maca, @manv, @dayOfWeek)

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

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

RAISERROR(@err, 16, 1)

END CATCH

END

Code C# thực thi:

|  |
| --- |
| private void shift()  {  SqlCommand cmd = new SqlCommand("EXEC proc\_PhanCa @maca, @manv, @dayOfWeek", db.getConnection);  cmd.Parameters.Add("@maca", SqlDbType.NChar, 10).Value = comBox\_shift.Text; cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = txt\_ID.Text;  cmd.Parameters.Add("@dayOfWeek", SqlDbType.NChar, 10).Value = comBox\_dayOfWeek.Text; db.openConnection(); try  {  cmd.ExecuteNonQuery(); db.closeConnection();  MessageBox.Show("Phân ca thành công!", "Phân ca", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (SqlException sqlEx)  {  db.closeConnection();  MessageBox.Show("Phân ca thất bại!\n\nLỗi: " + sqlEx.Message, "Phân ca", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);  }  catch (Exception ex)  {  db.closeConnection();  MessageBox.Show("Phân ca thất bại!\n\nLỗi: " + ex.Message, "Phân ca", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);  }  } |

**10.2. Xóa ca làm của nhân viên**

CREATE PROC [dbo].[proc\_XoaPhanCa]

@manv nchar(10),

@maca nchar(10),

@ngay nchar(10)

AS

BEGIN

BEGIN TRY

DELETE FROM BangPhanCa

WHERE MaNV = @manv and

MaCa = @maca and

NgayTrongTuan = @ngay

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

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

RAISERROR(@err, 16, 1)

END CATCH

END

Code C# thực thi:

|  |
| --- |
| private void btn\_remove\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("EXEC proc\_XoaPhanCa @manv, @maca, @ngay", db.getConnection); cmd.Parameters.Add("@manv", SqlDbType.NChar, 10).Value = txt\_IdEmployee.Text; cmd.Parameters.Add("@maca", SqlDbType.NChar, 10).Value = comBox\_IdShift.Text; cmd.Parameters.Add("@ngay", SqlDbType.NChar, 10).Value = comBox\_dayOfWeek.Text; db.openConnection(); try  {  cmd.ExecuteNonQuery(); db.closeConnection();  MessageBox.Show("Xóa thành công!", "Xóa phân ca", MessageBoxButtons.OK,  MessageBoxIcon.Information);  this.Close();  }  catch (SqlException sqlEx)  {  db.closeConnection();  MessageBox.Show("Xóa thất bại!\n\nLỗi: " + sqlEx.Message, "Xóa phân ca", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  db.closeConnection();  MessageBox.Show("Xóa thất bại!\n\nLỗi:" + ex.Message, "Xóa phân ca", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

# CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN

Ứng với hai nhóm người dung (Nhân viên và quản lý) là các role riêng + Đối với Role Staff (dành cho nhân viên):

|  |
| --- |
| CREATE ROLE Staff  --Gán các quyền trên table cho role Staff  GRANT SELECT, REFERENCES ON BangPhanCa TO Staff  GRANT SELECT, REFERENCES ON CaLamViec TO Staff  GRANT SELECT, REFERENCES ON CheBien TO Staff  GRANT SELECT, REFERENCES ON ChiTietHoaDon TO Staff  GRANT SELECT, REFERENCES ON ChiTietHoaDonUngDung TO Staff  GRANT SELECT, REFERENCES ON ChiTietNhapHang TO Staff  GRANT SELECT, REFERENCES ON CongViec TO Staff  GRANT SELECT, INSERT, REFERENCES ON DonNhapHang TO Staff  GRANT SELECT, INSERT, REFERENCES ON HoaDon TO Staff  GRANT SELECT, INSERT, REFERENCES ON HoaDonUngDung TO Staff  GRANT SELECT, INSERT, REFERENCES ON KhachHang TO Staff  GRANT SELECT, REFERENCES ON LoaiSanPham TO Staff  GRANT SELECT, REFERENCES ON NguyenLieu TO Staff  GRANT SELECT, REFERENCES ON NhaCungCap TO Staff  GRANT SELECT, REFERENCES ON NhanVien TO Staff  GRANT SELECT, INSERT, REFERENCES ON PhieuChi TO Staff  GRANT SELECT, REFERENCES ON SanPham TO Staff  GRANT SELECT, REFERENCES ON UngDung TO Staff  Gán quyền thực thi trên các procedure, function cho role Staff  GRANT EXECUTE TO Staff  GRANT SELECT TO Staff    DENY EXECUTE ON proc\_AddEmployee to Staff;  DENY EXECUTE ON proc\_DeleteEmployee to Staff;  DENY EXECUTE ON proc\_EditEmployee to Staff;  DENY EXECUTE ON proc\_suaCheBien to Staff;  DENY EXECUTE ON proc\_suaNguyenLieu to Staff;  DENY EXECUTE ON proc\_suaSanPham to Staff;  DENY EXECUTE ON proc\_xoaSanPham to Staff;  DENY EXECUTE ON proc\_xoaPhanCa to Staff; |

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

* Trigger tạo tài khoản

|  |
| --- |
| CREATE TRIGGER [dbo].[Trigger\_CreateSQLAccount] ON [dbo].[DANGNHAP] AFTER INSERT AS  DECLARE @userName nvarchar(30), @passWord nvarchar(10), @manv nvarchar(10) SELECT @userName=nl.TenDangNhap, @passWord=nl.MatKhau, @manv=nl.maNV FROM inserted nl  BEGIN  DECLARE @sqlString nvarchar(2000), @macv nvarchar(10)  ----  SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+ @passWord  +''', DEFAULT\_DATABASE=[QUANLYQUANCOFFEE\_Cur], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'  EXEC (@sqlString)  ----  SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName EXEC (@sqlString)  ----  SELECT @macv = MaCV |
| FROM NhanVien WHERE MaNV = @manv if (@macv = 'cv01')  SET @sqlString = 'ALTER SERVER ROLE sysadmin' + ' ADD MEMBER '  + @userName; else  SET @sqlString = 'ALTER ROLE Staff ADD MEMBER ' + @userName; EXEC (@sqlString)  END |

* Procedure xóa nhân viên

|  |
| --- |
| CREATE PROCEDURE [dbo].[proc\_DeleteEmployee]  @MaNV nchar(10)  AS  BEGIN  SET NOCOUNT ON;  DECLARE @username varchar(15);  SELECT @username=TenDangNhap FROM DANGNHAP WHERE maNV=@maNV  DECLARE @sql varchar(100)  DECLARE @SessionID INT;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username;  IF @SessionID IS NOT NULL  BEGIN  SET @sql = 'kill ' + Convert(NVARCHAR(20), @SessionID)  exec(@sql) END  BEGIN TRANSACTION;  BEGIN TRY  DELETE FROM HoaDon WHERE MaNV = @MaNV;  DELETE FROM BangPhanCa WHERE MaNV = @MaNV;  DELETE FROM NhanVien WHERE MaNV = @MaNV;  --  SET @sql = 'DROP USER '+ @username  exec (@sql)  --  SET @sql = 'DROP LOGIN '+ @username  exec (@sql)  --  DELETE FROM DANGNHAP WHERE maNV=@maNV;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  ROLLBACK TRANSACTION;  THROW;  END CATCH  COMMIT TRANSACTION;  END |

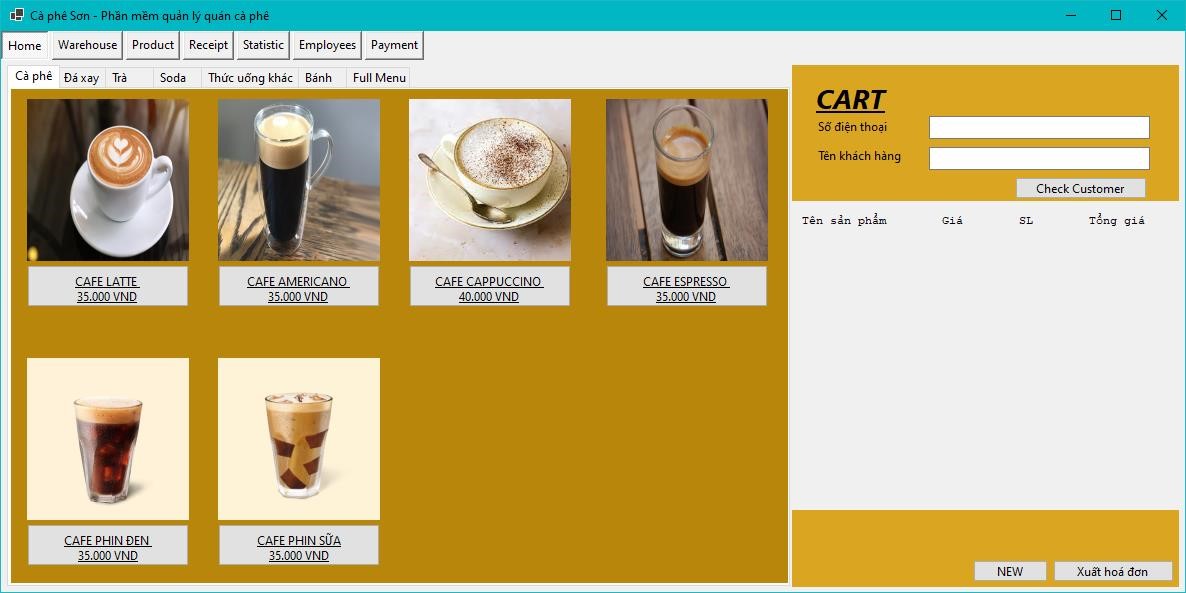
# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG

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

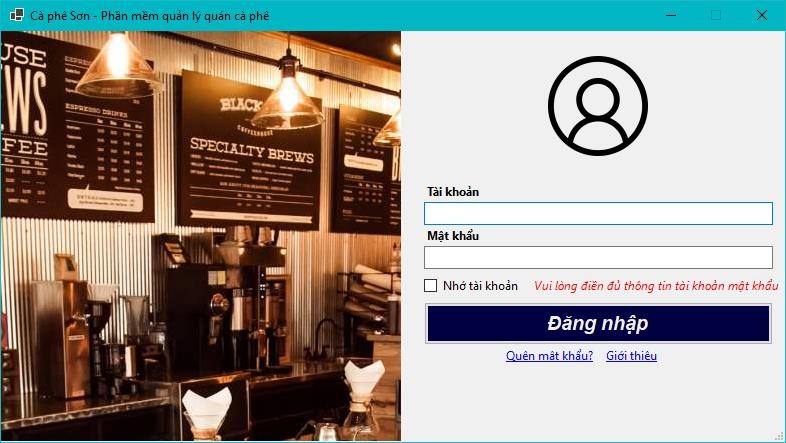
* Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.
* Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.

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

* 1. **Form Home**



* 1. **Form đăng nhập**



* Tạo tài khoản: Code C#:

|  |
| --- |
| private void buttonCreate\_Click(object sender, EventArgs e)  {  mydb.openConnectionAdmin();  SqlCommand cmd = new SqlCommand("proc\_ThemTaiKhoan", mydb.getConnectionAdmin); cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@manv", comboBoxMaNV.SelectedValue.ToString()); cmd.Parameters.AddWithValue("@tk", textBoxUsername.Text); cmd.Parameters.AddWithValue("@mk", textBoxPassword.Text);    if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Tạo tài khoản thành công"); mydb.closeConnectionAdmin();  } else  {  mydb.closeConnectionAdmin();  }  } |

Code SQL:

|  |
| --- |
| CREATE procedure [dbo].[proc\_ThemTaiKhoan] @tk NCHAR(10), @mk NCHAR(10), @manv NCHAR(10)  As  BEGIN  INSERT INTO DANGNHAP VALUES (@tk, @mk, @manv) END |

* Kiểm tra đăng nhập:

Code C#:

GLOBAL.username = textBoxUser.Text;

GLOBAL.password = textBoxPassword.Text;

SqlCommand cmd = new SqlCommand("SELECT dbo.checkLogin(@user, @pass)", mydb.getConnectionAdmin);

cmd.Parameters.AddWithValue("@user", textBoxUser.Text); cmd.Parameters.AddWithValue("@pass", textBoxPassword.Text); mydb.openConnectionAdmin();

bool count = (bool)cmd.ExecuteScalar();

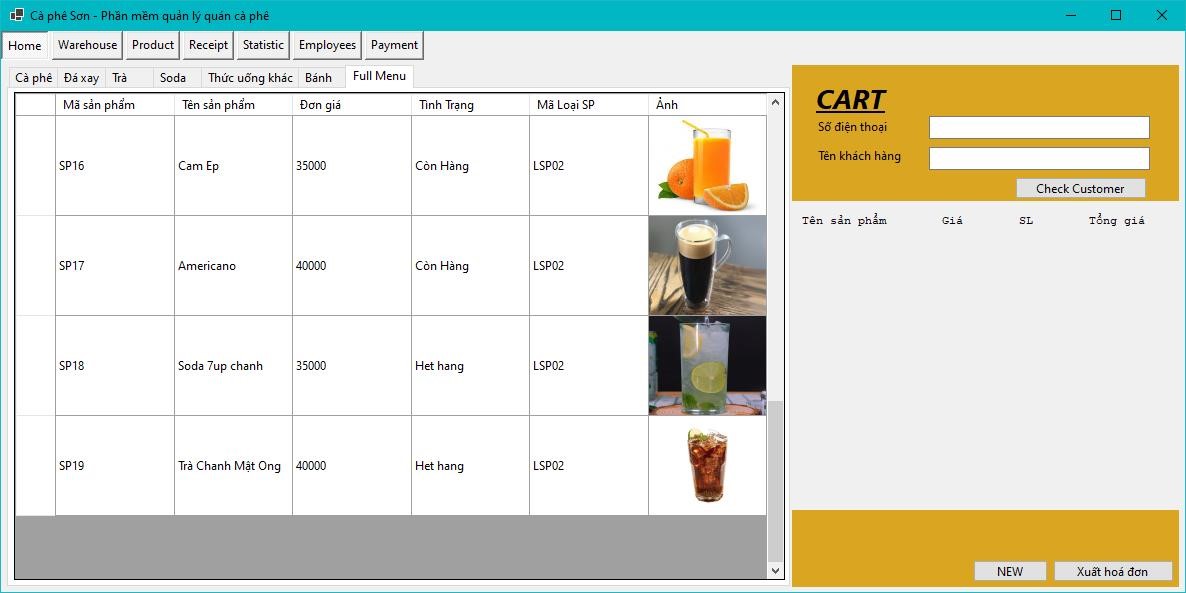
mydb.closeConnectionAdmin();

Code SQL:

|  |
| --- |
| CREATE FUNCTION [dbo].[checkLogin] (@username NVARCHAR(MAX), @password NVARCHAR(MAX))  RETURNS BIT  AS  BEGIN  DECLARE @result BIT;    SELECT @result = CAST(COUNT(\*) AS BIT)  FROM DANGNHAP  WHERE TenDangNhap = @username AND MatKhau = @password;  RETURN @result;  END; |

1. **Các Form Danh mục**

**3.1. Danh mục sản phẩm**



Code C# gọi danh mục:

|  |
| --- |
| public DataTable getDSSanPham()  {  //Lấy danh mục sản phẩm từ view  SqlCommand command = new SqlCommand("Select \* from view\_danhSachSanPham", mydb.getConnection);  DataTable table = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(table); return table;  } |

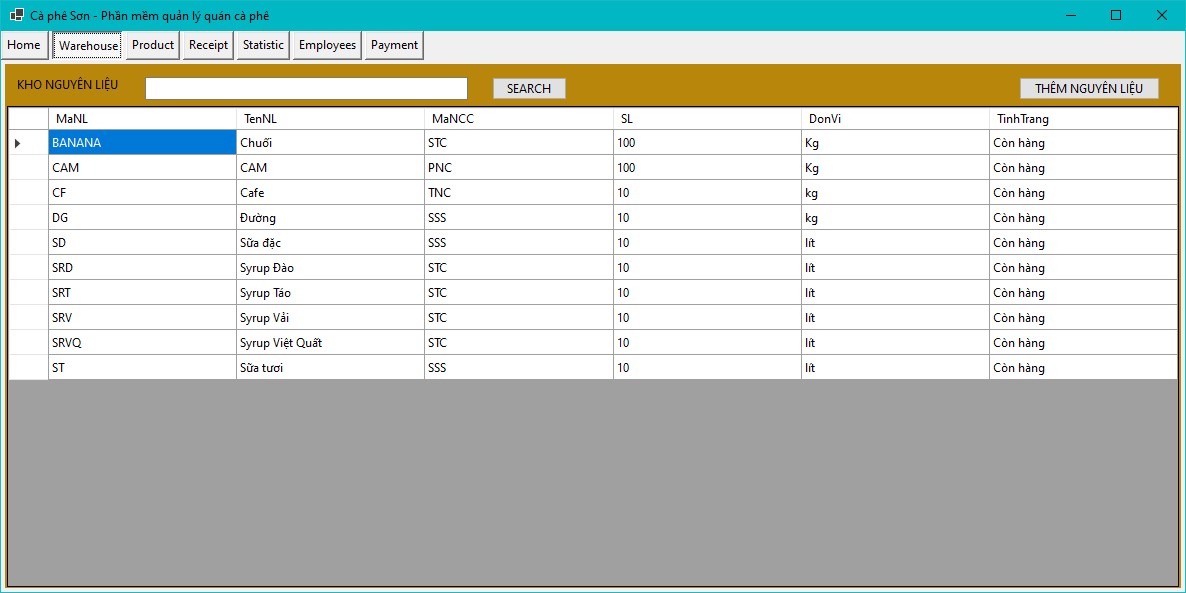
Code tạo view trên SQL:

CREATE VIEW [dbo].[view\_danhSachSanPham] AS

SELECT \*

FROM SanPham

## 3.2. Danh mục kho



Code C# gọi danh mục:

|  |
| --- |
| public DataTable xemNguyenLieu()  {  //Lấy danh sách nguyên liệu trong kho  SqlCommand command = new SqlCommand("select \* from view\_NguyenLieu", mydb.getConnection); DataTable dt = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter(command); adapter.Fill(dt); return dt;  } |

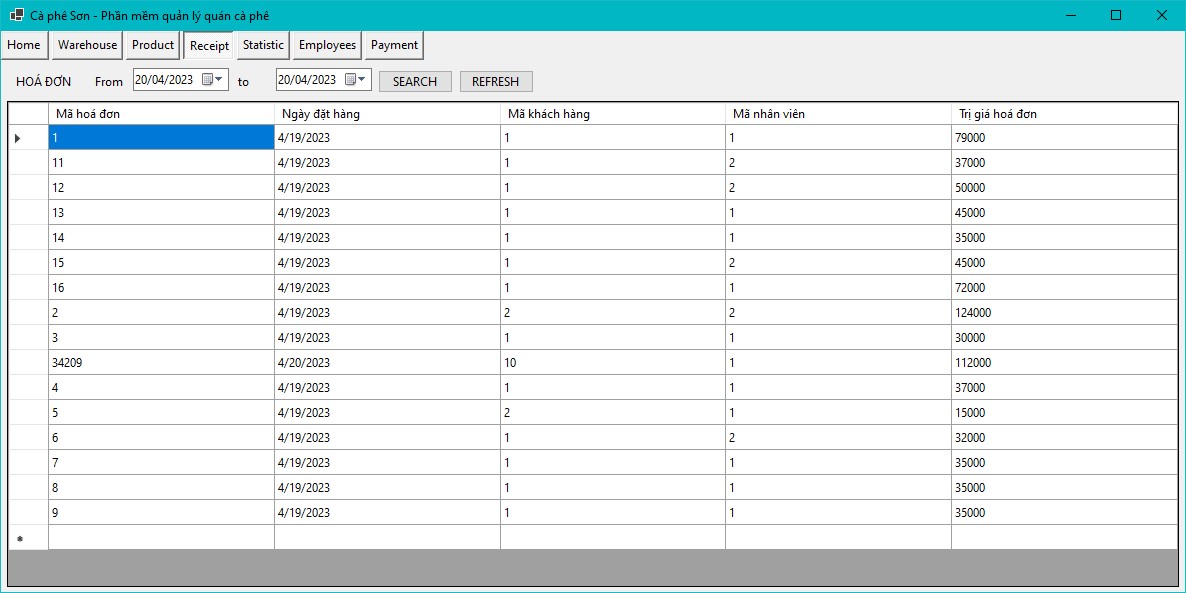
Code tạo view trên SQL:

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

SELECT \*

FROM NguyenLieu

## 3.3.Danh mục hóa đơn



Code C# gọi danh mục:

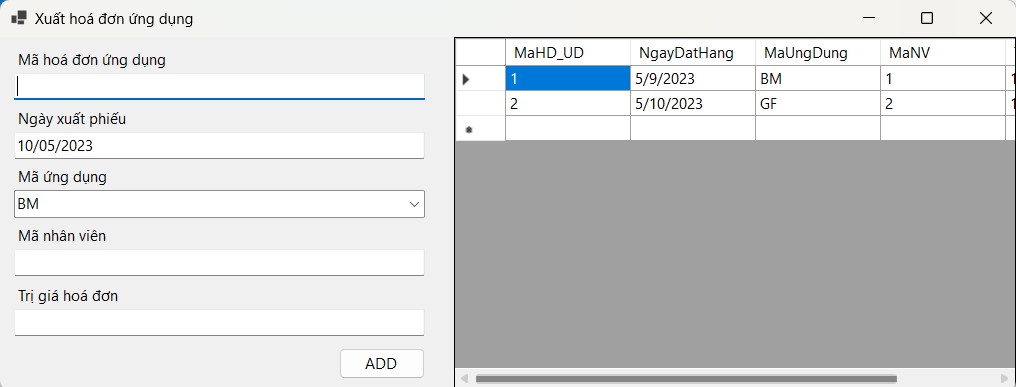
|  |
| --- |
| public void load\_tabPage\_Receipt()  {  //Lấy danh mục hóa đơn  SqlCommand cmd = new SqlCommand("SELECT \* FROM vi\_HoaDonView", db.getConnection); SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable(); da.Fill(dt);  dataGridView3.DataSource = dt;    //Doi ten  dataGridView3.Columns["MaHD"].HeaderText = "Mã hoá đơn"; dataGridView3.Columns["NgayDatHang"].HeaderText = "Ngày đặt hàng"; dataGridView3.Columns["MaKH"].HeaderText = "Mã khách hàng"; dataGridView3.Columns["MaNV"].HeaderText = "Mã nhân viên"; dataGridView3.Columns["TriGiaHD"].HeaderText = "Trị giá hoá đơn";  } |

Code tạo view trên SQL:

CREATE VIEW [dbo].[vi\_HoaDonView] AS

SELECT MaHD, NgayDatHang, MaKH, MaNV, TriGiaHD FROM HoaDon;

## 3.4.Danh mục hóa đơn ứng dụng



Code C# gọi danh mục:

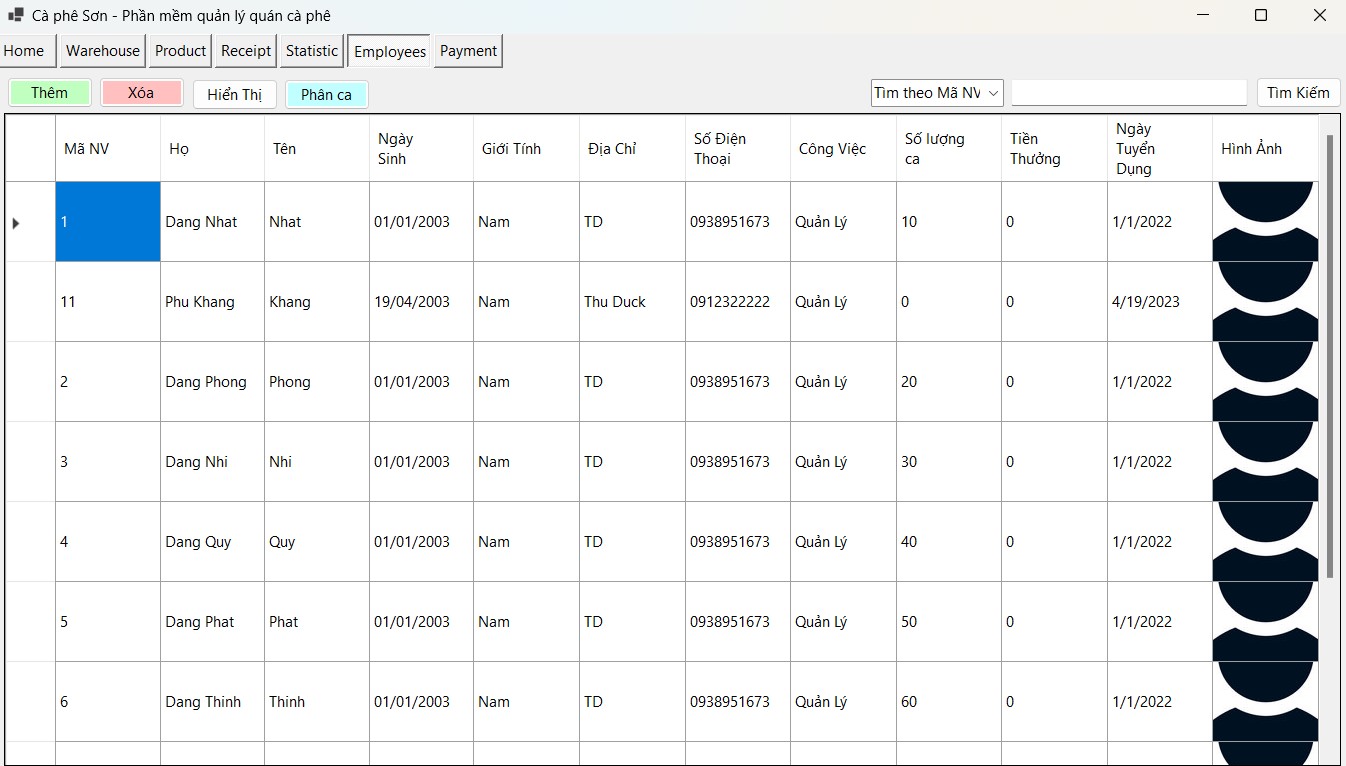
|  |
| --- |
| db.openConnection();  SqlDataAdapter da1 = new SqlDataAdapter("SELECT \* FROM vi\_HoaDonUngDung", db.getConnection);  DataTable dt1 = new DataTable();  da1.Fill(dt1);  dgvHDUD.DataSource = dt1; db.closeConnection(); |

Code tạo view trên SQL:

CREATE VIEW vi\_HoaDonUngDung AS

SELECT MaHD\_UD, NgayDatHang, MaUngDung, MaNV, TriGiaHD FROM HoaDonUngDung;

## 3.5.Danh mục nhân viên



Code C# gọi danh mục:

|  |
| --- |
| public DataTable getAllEmployee()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_LoadEmployee", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); return dataTable;  } |

Code tạo view trên SQL:

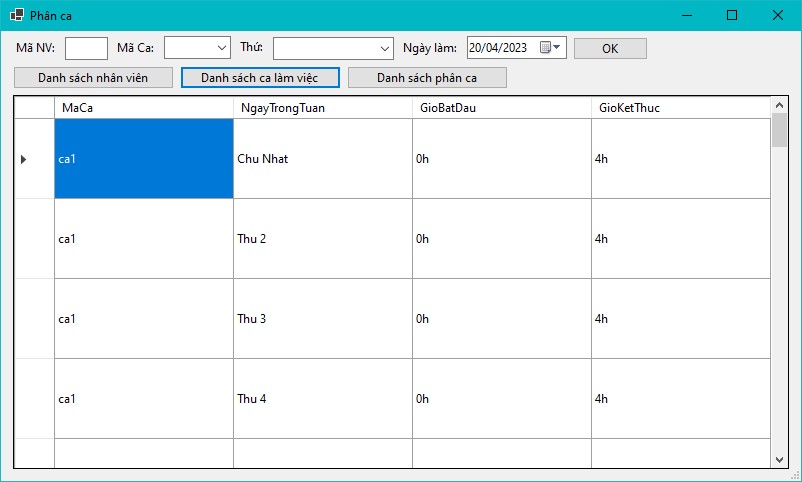
CREATE VIEW [dbo].[v\_LoadEmployee] AS

SELECT MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, CongViec.TenCV, SoCa, Thuong, NgayTuyenDung, HinhAnh

FROM NhanVien, CongViec

WHERE NhanVien.MaCV = CongViec.MaCV

## 3.6.Danh mục ca làm việc



Code C# gọi danh mục:

|  |
| --- |
| private void displayShift()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_CaLamViec", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); current\_DataGridView = 1;  dataGridView\_Division.DataSource = dataTable;  } |

Code tạo view trên SQL:

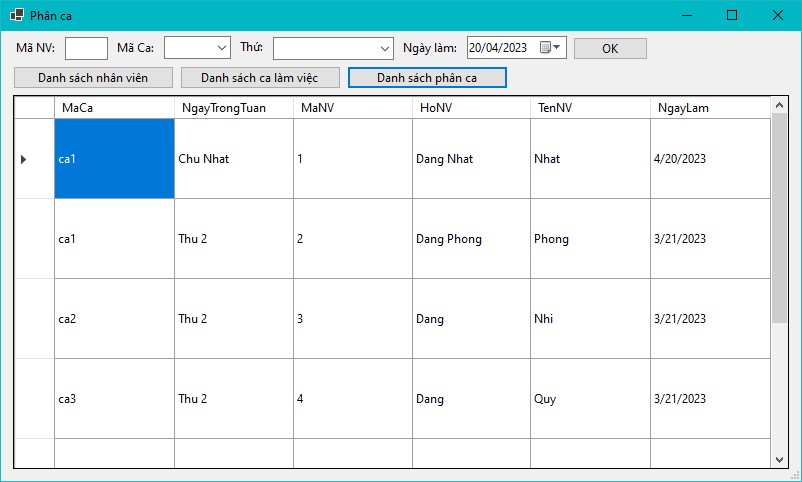
CREATE VIEW [dbo].[v\_CaLamViec]

AS

SELECT \*

FROM CaLamViec

## 3.7.Danh mục bảng ca làm việc



Code C# gọi danh mục:

|  |
| --- |
| private void displayDivision()  {  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_BangPhanCa", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); current\_DataGridView = 2;  dataGridView\_Division.DataSource = dataTable;  } |

Code tạo view trên SQL:

CREATE VIEW [dbo].[v\_BangPhanCa]

AS

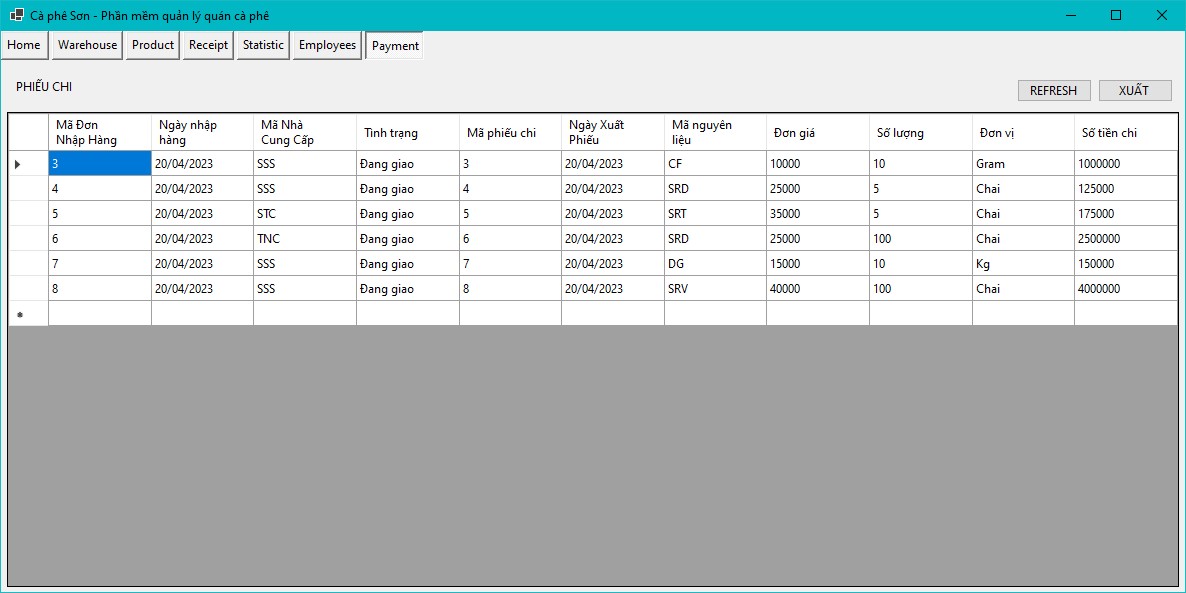
SELECT NhanVien.MaNV, NhanVien.HoNV, NhanVien.TenNV, BangPhanCa.MaCa,

BangPhanCa.NgayTrongTuan, BangPhanCa.NgayLam

FROM BangPhanCa, NhanVien

WHERE BangPhanCa.MaNV = NhanVien.MaNV

## 3.8.Danh mục phiếu chi



Code C# gọi danh mục:

private void load\_tabPage\_Payment()

{

SqlCommand cmd = new SqlCommand("SELECT \* FROM DonNhapHangView", db.getConnection); SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

dataGridViewPayment.DataSource = dt;

}

Code tạo view trên SQL:

CREATE VIEW [dbo].[DonNhapHangView]

AS

SELECT DNH.MaDonNH, DNH.NgayNH, DNH.MaNCC, DNH.TinhTrang, PC.MaPhieuChi,

PC.NgayXuatPhieu, CTH.MaNL, CTH.DonGia, CTH.SL, CTH.DonVi, PC.SoTienChi

FROM DonNhapHang DNH

LEFT JOIN PhieuChi PC ON DNH.MaDonNH = PC.MaDonNH

LEFT JOIN ChiTietNhapHang CTH ON DNH.MaDonNH = CTH.MaDonNH;

## 3.9.Danh mục lương nhân viên



Code C# gọi hàm:

|  |
| --- |
| private void tinhluongForm\_Load(object sender, EventArgs e)  {  MY\_DB db = new MY\_DB();  db.openConnection();      SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_tinhLuongNV(@thang, @nam)", db.getConnection);      cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM"))); cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy")));    // Create a data adapter to fill a dataset with the results  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataSet ds = new DataSet(); da.Fill(ds); |

dataGridViewluong.DataSource = ds.Tables[0];

db.closeConnection();

}

Code function trên SQL:

|  |
| --- |
| CREATE FUNCTION func\_tinhLuongNV(@thang INT, @nam INT) RETURNS TABLE AS RETURN (  SELECT nv.MaNV,nv.HoNV,nv.TenNV ,nv.SoCa,  (nv.SoCa\*cv.Luong\*4 + nv.Thuong\*(cv.Luong\*2)) +  CASE WHEN dbo.func\_tinhDoanhThuThang(@thang, @nam) > 100000000 THEN dbo.func\_tinhDoanhThuThang(@thang, @nam)\*0.01  ELSE 0  END AS Luong  FROM NhanVien nv  INNER JOIN CongViec cv ON nv.MaCV = cv.MaCV ) |