**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 CỬA HÀNG ĐIỆN THOẠI**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Lớp HP:** *DBMS330284\_23\_1\_05*

**Nhóm thực hiện:** *Nhóm 10*

**Học kỳ:** *1*

**Năm học:** *2023 - 2024*

*Thành phố Hồ Chí Minh, tháng …, năm 2023***DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2023-2024**

**Nhóm 10**

***Đề tài: Quản lý cửa hàng điện thoại***

|  |  |
| --- | --- |
| 21133056 | Võ Công Minh |
| 21133016 | Lê Hoàng Đức Duy |
| 21133105 | Nguyễn Tấn Khang |
| 21133102 | Huỳnh Trung Hậu |

**NHẬN XÉT CỦA GV**

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

Giảng viên chấm điểm

MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 6](#_Toc151512717)

[1. Đặc tả đề tài 6](#_Toc151512718)

[2. Nghiệp vụ của bài toán 7](#_Toc151512719)

[3. Mô tả chức năng của bài toán 9](#_Toc151512720)

[4. Tính lương nhân viên: 9](#_Toc151512721)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 12](#_Toc151512722)

[1. Thiết kế cơ sở dữ liệu mức quan niệm 12](#_Toc151512723)

[2. Thiết kế cơ sở dữ liệu mức logic 13](#_Toc151512724)

[3. Các ràng buộc cần có 13](#_Toc151512725)

[4. Cài đặt các CSDL và các ràng buộc 15](#_Toc151512726)

[4.1. Bảng công việc 15](#_Toc151512727)

[4.2. Bảng nhân viên 15](#_Toc151512728)

[4.3. Bảng ca làm việc 15](#_Toc151512729)

[4.4. Bảng phân ca 16](#_Toc151512730)

[4.5. Bảng khách hàng 16](#_Toc151512731)

[4.6. Bảng điện thoại 16](#_Toc151512732)

[4.7. Bảng hãng điện thoại 16](#_Toc151512733)

[4.8. Bảng hóa đơn 16](#_Toc151512734)

[4.9. Bảng chi tiết đơn hàng 17](#_Toc151512735)

[4.10. Bảng nhà cung cấp 17](#_Toc151512736)

[4.11. Bảng đơn nhập 17](#_Toc151512737)

[4.12. Bảng chi tiết đơn nhập 17](#_Toc151512738)

[4.13. Bảng tài khoản 18](#_Toc151512739)

[5. Các view 20](#_Toc151512740)

[5.1. View xem danh sách điện thoại 20](#_Toc151512741)

[5.2. View xem danh sách hóa đơn 20](#_Toc151512742)

[5.3. View xem danh sách khách hàng 20](#_Toc151512743)

[5.4. View xem danh sách nhân viên 20](#_Toc151512744)

[5.5. View xem bảng phân ca 20](#_Toc151512745)

[6. Các trigger 21](#_Toc151512746)

[6.1. Trigger cập nhập lại số lượng điện thoại khi xóa hóa đơn 21](#_Toc151512747)

[6.2. Trigger kiểm tra thông tin tên và số điện thoại của khách hàng có trùng không 21](#_Toc151512748)

[6.3. Trigger kiểm tra tuổi và số điện thoại của nhân viên 22](#_Toc151512749)

[6.4. Trigger cập nhập trạng thái của điện thoại 22](#_Toc151512750)

[6.5. Trigger cập nhập trạng thái và mật khẩu tài khoản của nhân viên 23](#_Toc151512751)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 23](#_Toc151512752)

[1. Kết nối cơ sở dữ liệu 23](#_Toc151512753)

[2. Xem thông tin danh mục 24](#_Toc151512754)

[2.1. Danh mục điện thoại 24](#_Toc151512755)

[2.2. Danh mục hóa đơn 25](#_Toc151512756)

[2.3. Danh mục khách hàng 25](#_Toc151512757)

[2.4. Danh mục nhân viên 26](#_Toc151512758)

[3. Quản lý hàng hóa (điện thoại) 27](#_Toc151512759)

[3.1. Procedure - Thêm hoặc sửa thông tin điện thoại 27](#_Toc151512760)

[3.2. Procedure - Xóa thông tin điện thoại 28](#_Toc151512761)

[3.3. Procedure - Thêm vào giỏ hàng để thực hiện tạo hóa đơn 29](#_Toc151512762)

[3.4. Function - Tìm kiếm điện thoại theo hãng và giá 31](#_Toc151512763)

[3.5. Function - Tìm kiếm theo tên điện thoại 32](#_Toc151512764)

[3.6. Function – Lấy id lớn nhất của điện thoại để tạo điện thoại mới. 33](#_Toc151512765)

[4. Quản lý hóa đơn 34](#_Toc151512766)

[4.1. Procedure - Tạo hóa đơn mới 34](#_Toc151512767)

[4.2. Procedure - Xóa hóa đơn 35](#_Toc151512768)

[4.3. Procedure - Xem chi tiết hóa đơn 36](#_Toc151512769)

[4.4. Function - Tìm kiếm hóa đơn theo ngày và trạng thái 37](#_Toc151512770)

[4.5. Function – Lấy id lớn nhất của hóa đơn để tạo hóa đơn mới. 38](#_Toc151512771)

[4.6. Function – Tính tổng số lượng mặt hàng của một hóa đơn. 39](#_Toc151512772)

[5. Quản lý khách hàng 40](#_Toc151512773)

[5.1. Procedure - Thêm hoặc sửa thông tin khách hàng 40](#_Toc151512774)

[5.2. Procedure - Xóa khách hàng 41](#_Toc151512775)

[5.3. Procedure - Xem lịch sử mua hàng của khách hàng 42](#_Toc151512776)

[5.4. Procedure - Tìm kiếm khách hàng theo số điện thoại 43](#_Toc151512777)

[5.5. Function – Lấy id lớn nhất của khách hàng để tạo khách hàng mới. 44](#_Toc151512778)

[6. Quản lí nhân viên 44](#_Toc151512779)

[6.1. Procedure - Thêm hoặc sửa thông tin nhân viên 44](#_Toc151512780)

[6.2. Procedure - Xóa nhân viên 46](#_Toc151512781)

[6.3. Procedure - Tìm kiếm nhân viên theo sđt 46](#_Toc151512782)

[6.4. Procedure - Xem chi tiết thông tin của nhân viên 47](#_Toc151512783)

[6.5. Function – Lấy id lớn nhất của nhân viên để tạo nhân viên mới. 48](#_Toc151512784)

[6.6. Procedure – Đổi mật khẩu tài khoản đăng nhập 49](#_Toc151512785)

[6.7. Procedure – Nhân viên đăng nhập tài khoản 50](#_Toc151512786)

[6.8. Procedure – Nhân viên thay đổi thông tin cá nhân 51](#_Toc151512787)

[6.9. Procedure – Nhân viên chấm công 52](#_Toc151512788)

[6.10. Procedure – Nhân viên xem ca làm của mình 53](#_Toc151512789)

[7. Quản lí ca làm việc 54](#_Toc151512790)

[7.1. Procedure - Thêm ca làm việc cho nhân viên 54](#_Toc151512791)

[7.2. Function – Tìm kiếm ca làm theo ngày và ca 55](#_Toc151512792)

[8. Quản lí tài chính 57](#_Toc151512793)

[8.1. Procedure – Hiện thị thông tin cần thiết 57](#_Toc151512794)

[8.2. Function – Tính tổng số lượng các hóa đơn, đơn nhập, ca làm theo tháng 58](#_Toc151512795)

[8.3. Function – Tính thực hiện tính lương, tổng hóa đơn, tổng đơn nhập theo tháng 60](#_Toc151512796)

[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 63](#_Toc151512797)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN 67](#_Toc151512798)

[1. Giao diện đăng nhập và trang chủ: 67](#_Toc151512799)

[1.1. Giao diện login: 67](#_Toc151512800)

[1.2. Giao diện home: 67](#_Toc151512801)

[2. Giao diện Điện Thoại: 68](#_Toc151512802)

[2.1. Giao diện danh mục Điện Thoại 68](#_Toc151512803)

[2.2. Giao diện về thêm hoặc chỉnh sửa thông tin Điện Thoại: 69](#_Toc151512804)

[3. Giao diện Hóa Đơn: 69](#_Toc151512805)

[3.1. Giao diện về danh mục Hóa Đơn 69](#_Toc151512806)

[3.2. Giao diện tạo hóa đơn, khách hàng: 70](#_Toc151512807)

[4. Giao diện Khách Hàng: 70](#_Toc151512808)

[4.1. Giao diện về danh mục Khách Hàng: 70](#_Toc151512809)

[4.2. Giao diện thêm hoặc chỉnh sửa thông tin khách hàng: 71](#_Toc151512810)

[5. Giao diện Nhân Viên 72](#_Toc151512811)

[5.1. Giao diện về danh mục Nhân Viên: 72](#_Toc151512812)

[5.2. Giao diện thêm hoặc chỉnh sửa thông tin Nhân Viên: 73](#_Toc151512813)

[6. Giao diện Bảng Phân Ca 73](#_Toc151512814)

[7. Giao diện Tài Chính: 74](#_Toc151512815)

# LỜI CẢM ƠN

# LỜI NÓI ĐẦU

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

## Đặc tả đề tài

Một cửa hàng điện thoại cần có một hệ thống để quản lý. Chương trình quản lý được xây dựng cho hai đối tượng người dùng là người quản lý và nhân viên với các quyền khác nhau.

Đối với người dùng là nhân viên, họ sẽ có quyền truy cập vào các chức năng như bán hàng, xem doanh thu, xem ca làm việc. Còn với người dùng là người quản lý cửa hàng, họ sẽ có quyền truy cập vào các chức năng như quản lý nhân viên, quản lý kho, quản lý doanh thu, quản lý ca làm việc.

Mỗi nhân viên chỉ làm một công việc, một công việc có thể có nhiều nhân viên cùng làm. Các công việc phân biệt với nhau bằng mã công việc, mỗi công việc gồm các thông tin như mã công việc, tên công việc, lương. Các nhân viên phân biệt với nhau bằng mã nhân viên, mỗi nhân viên gồm các thông tin như mã nhân viên, họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, mã công việc.

Các nhân viên sẽ được chia ca để làm việc. Một nhân viên có thể làm nhiều ca, một ca có thể có nhiều nhân viên. Các ca làm việc được phân biệt với nhau qua mã ca, mỗi ca có các thông tin như mã ca, giờ bắt đầu, giờ kết thúc.

Khách hàng đến cửa hàng được nhân viên hỗ trợ, nếu quyết định mua sản phẩm và chưa có thông tin trước đó thì nhân viên sẽ tiến hành tạo thông tin cho khách hàng mới. Các khách hàng sẽ phân biệt với nhau bằng mã khách hàng. Mỗi khách hàng sẽ bao gồm các thông tin như mã khách hàng, họ tên, số điện thoại, địa chỉ.

Mỗi đơn hàng khi khách hàng mua sản phẩm sẽ được xuất hóa đơn bởi nhân viên. Mỗi hóa đơn được phân biệt với nhau bằng mã hóa đơn. Mỗi hóa đơn bao gồm mã hóa đơn, mã khách hàng, mã nhân viên xuất hóa đơn, ngày đặt hàng, trị giá hóa đơn, trạng thái.

Điện thoại trong cửa hàng gồm nhiều hãng, các hãng điện thoại phân biệt với nhau bằng tên hãng điện thoại. Một hãng có thể có nhiều loại điện thoại, mỗi loại điện thoại phân biệt với nhau bằng mã điện thoại. Mỗi hãng điện thoại có thể có nhiều loại, mỗi loại chỉ thuộc một hãng điện thoại. Các loại điện thoại có các thông tin như mã điện thoại, tên điện thoại, màu sắc, dung lượng, giá thành, số lượng, tên hãng điện thoại.

Để quản lý điện thoại cần có thông tin của nhà cung cấp cũng như lên đơn nhập hàng đảm bảo cung cấp đầy đủ các loại điện thoại để bán. Một nhà cung cấp có thể cung cấp nhiều loại điện thoại, một loại điện thoại cũng có thể được cung cấp từ nhiều nhà cung cấp. Các nhà cung cấp phân biệt với nhau bằng mã nhà cung cấp. Mỗi nhà cung cấp có các thông tin như mã nhà cung cấp, tên nhà cung cấp, địa chỉ, số điện thoại.

Đơn nhập hàng được phân biệt bởi mã đơn nhập hàng. Mỗi đơn nhập hàng có các thông tin như mã đơn nhập hàng, ngày, trị giá, mã nhà sản xuất, mã nhân viên nhập, trạng thái.

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

## Nghiệp vụ của bài toán

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

* Quản lý nhân viên:

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

* Quy trình thực hiện khi khách đến quán:

Đầu tiên, nhân viên sẽ chào khách khi khách bước vào quán. Sau đó nhân viên bán hàng sẽ hỗ trợ tư vấn theo nhu cầu của khách hàng. Sau khi khách hàng chọn được sản phẩm ưng ý. Nhân viên thu ngân sẽ hỏi khách từng mua sản phẩm chưa và bắt đầu thực hiện tạo hóa đơn theo thông tin khách hàng. Cuối cùng, nhân viên thu ngân xuất hóa đơn để đưa cho khách thanh toán, và sau đó chuyển qua bộ phận kỹ thuật xử lý. Sau khi kiểm tra và bổ sung theo yêu cầu của khách hàng. Sản phẩm sẽ được trao tận tay cho khách hàng.

* Quản lý hàng hóa, quản lý doanh thu và điều hành cửa hàng điện thoại chia thành các giai đoạn:

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

Tiếp theo, ở giai đoạn vận hành, chúng ta cần xây dựng chi phí của từng sản phẩm và lên giá thành cho các sản phẩm đó. Ngoài ra, chúng ta cũng cần xây dựng quy trình làm việc, nghiệp vụ cho các nhân viên thu ngân, bán hàng, kỹ thuật, kho hàng, bảo vệ.

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

## Mô tả chức năng của bài toán

Đối với nhân viên:

***Chức năng bán hàng:*** cho phép nhân viên xem thông tin các sản phẩm, tạo hóa đơn, thêm, xóa, sửa, xuất hóa đơn.

***Xem doanh thu:*** cho phép nhân viên có thể xem được doanh thu bán trong ca làm việc của mình.

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

Đối với người quản lý:

***Quản lý kho:*** cho phép người quản lý biết được số lượng của từng loại, từng hãng điện thoại còn trong kho hàng.

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

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

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

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

## Tính lương nhân viên:

Tính lương nhân viên trong cửa hàng điện thoại có thể bao gồm các yếu tố sau:

1. **Lương cơ bản:** đây là số tiền nhân viên được trả cố định hàng tháng cho việc

làm công việc cố định mà họ được giao. Còn lương của quản lý sẽ là lương cố định.

1. **Thời gian làm việc:** đây là số giờ mà nhân viên đã làm việc trong tháng.
2. **Lương theo giờ:** được tính theo công thức: Lương cơ bản (đồng/giờ) x Thời gian

làm việc.

1. **Tiền thưởng:**

Đây là khoản tiền thưởng dành cho những nhân viên làm việc vào những ca thuộc ngày lễ, những ca có nhân viên xin nghỉ đột xuất (mỗi ca làm việc này tương ứng với một ca thưởng) hoặc là khi đạt được chỉ tiêu nhất định trong tháng. Công thức:

Lương thưởng (đồng) = Lương cơ bản (đồng/giờ) x 1,5 x Số giờ tăng ca

Tổng lương = Lương theo giờ + Lương thưởng

Ví dụ:

Giả sử cửa hàng điện thoại có 2 loại nhân viên: nhân viên tư vấn (A) và nhân viên kỹ thuật (B). Cả 2 nhân viên ấy đều được tính lương theo giờ, cụ thể:

Nhân viên tư vấn A: 30.000 (đồng/giờ)

Nhân viên kỹ thuật B: 35.000 (đồng/giờ)

(1 ca = 6 tiếng)

Vào tháng này, nhân viên tư vấn A đã làm tổng cộng 30 ca, 12 tiếng tăng ca thêm – nhân viên kỹ thuật B đã làm tổng cộng 25 ca, 16 tiếng tăng ca thêm:

→ Lương của nhân viên A:

Lương theo giờ = Lương cơ bản x Thời gian làm việc

                      = 30.000 (đồng/giờ) x 6 (giờ) x 30 (ca) = 5.400.000 (đồng)

Lương thưởng = Lương cơ bản x 1,5 x số giờ tăng ca

                      = 30.000 (đồng/giờ) x 1,5 x 12 = 540.000 (đồng)

Vậy tổng lương của nhân viên A tháng này là:

Tổng lương = 5.400.000 + 540.000 = 5.980.000 (đồng)

→ Lương của nhân viên B:

Lương theo giờ = Lương cơ bản x Thời gian làm việc

                      = 35.000 (đồng/giờ) x 6 (giờ) x 25 (ca) =5.250.000 (đồng)

Lương thưởng = Ca thưởng x 1,5 x số giờ tăng ca

                      = 35.000 (đồng/giờ) x 1,5 x 16 = 840.000 (đồng)

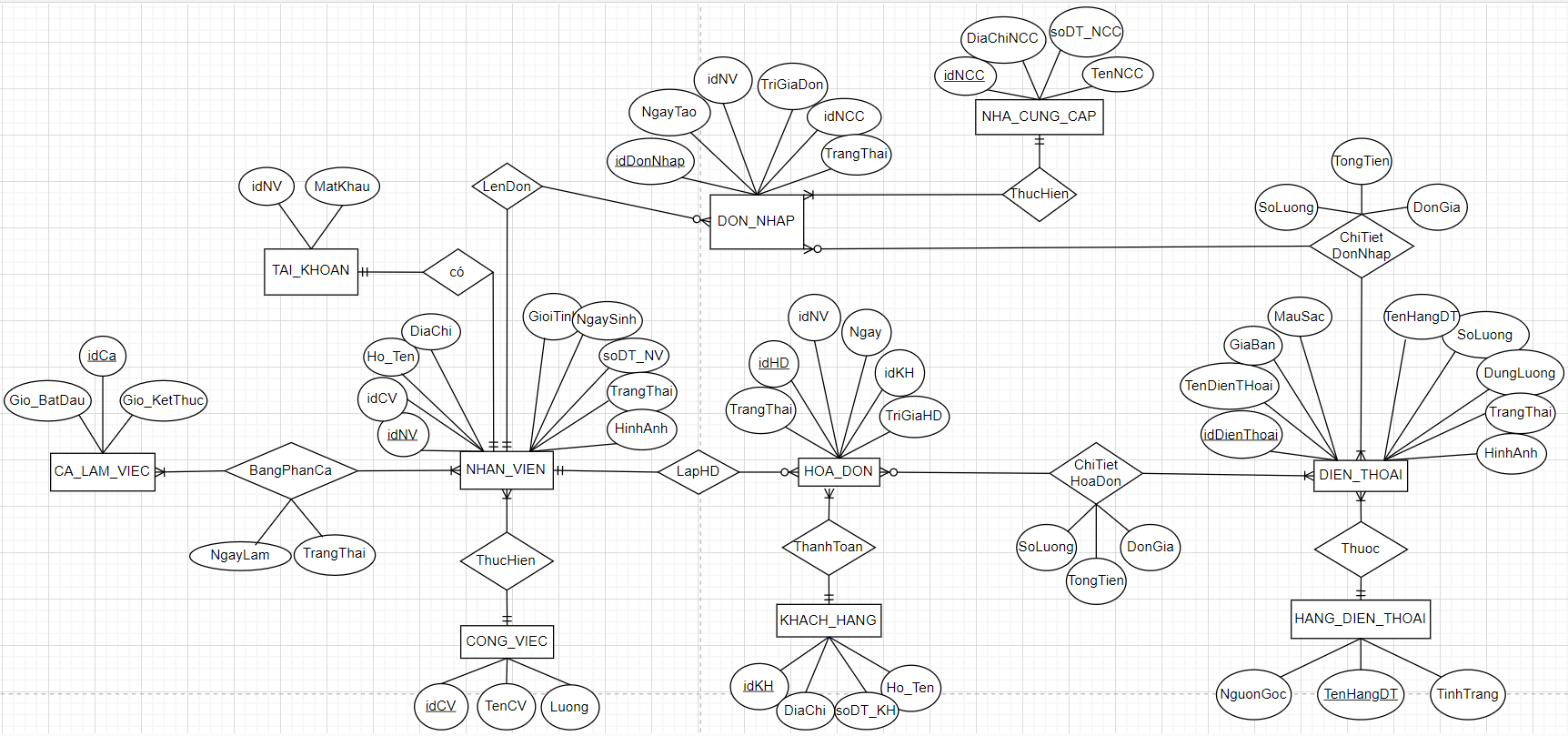
Vậy tổng lương của nhân viên B tháng này là:

Tổng lương = 5.250.000 + 840.000 = 6.090.000 (đồng)

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Thiết kế cơ sở dữ liệu mức quan niệm

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



## Thiết kế cơ sở dữ liệu mức logic

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

* CaLamViec (idCa, Gio\_BatDau, Gio\_KetThuc)
* CongViec (idCV, TenCV, LuongTheoGio)
* NhanVien (idNV, Ho\_Ten, idCV, DiaChi, GioiTinh, soDT\_NV, NgaySinh, TrangThai, HinhAnh)
* BangPhanCa (idCa, idNV, NgayLam, TrangThai)
* HoaDon (idHD, idNV, TriGiaHD, idKH, Ngay, TrangThai)
* ChiTietHoaDon (idHD, idDienThoai, SoLuong, DonGia, TongTien)
* KhachHang (idKH, TenKH, soDT\_KH, DiaChi)
* DienThoai (idDienThoai, TenDienThoai, TenHangDT, SoLuong, MauSac, DungLuong, GiaBan, TrangThai, HinhAnh)
* NhaCungCap (idNCC, DiaChiNCC, soDT\_NCC, TenNCC)
* HangDienThoai (TenHangDT, NguonGoc, TrangThai)
* DonNhap (idDonNhap, idNV, idNCC, TriGiaDon, NgayTao, TrangThai)
* ChiTietDonNhap (idDonNhap, idDienThoai, SoLuong, TongTien, DonGia, NgayNhap)
* TaiKhoan (idNV, MatKhau)

## Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| S T T | BẢNG | RÀNG BUỘC |
| 1 | CongViec | Khóa chính idCV,  NOT NULL TenCV, CHECK Luong > 0 |
| 2 | NhanVien | Khóa chính idNV, Khóa ngoại idCV đến bảng CongViec, NOT NULL HoTen, NOT NULL SDT, CHECK NgaySinh (>=18 tuổi) |
| 3 | CaLamViec | Khoá chính idCa,  CHECK Gio\_BatDau > Gio\_KetThuc |
| 4 | BangPhanCa | Khoá chính (idCa, idNV, NgayLam), Khóa ngoại idCa đến bảng CaLamViec,  Khóa ngoại idNV đến bảng NhanVien |
| 5 | HoaDon | Khoá chính idHD, Khoá ngoại idKH đến bảng KhachHang, Khoá ngoại idNV đến bảng NhanVien, NOT NULL TriGiaHD,  CHECK TriGiaHD > 0 |
| 6 | DienThoai | Khoá chính idDienThoai, Khoá ngoại idHangDT đến bảng HangDienThoai, NOT NULL TenDienThoai,  NOT NULL MauSac,  NOT NULL DungLuong,  NOT NULL GiaBan,  NOT NULL SoLuong |
| 7 | ChiTietHoaDon | Khoá chính (idHD, idDienThoai), Khoá ngoại idHD đến bảng HoaDon, Khoá ngoại idDienThoai đến bảng DienThoai,  NOT NULL SoLuong,  NOT NULL DonGia,  CHECK (SoLuong > 0 và DonGia > 0) |
| 8 | KhachHang | Khoá chính idKH, NOT NULL TenKH, NOT NULL soDT |
| 9 | HangDienThoai | Khoá chính TenHangDT, NOT NULL NguonGoc |
| 10 | NhaCungCap | Khoá chính idNCC, NOT NULL TenNCC,  NOT NULL soDT\_NCC |
| 11 | DonNhap | Khoá chính idDonNhap,  Khóa ngoại idNV đến bảng NhanVien,  Khóa ngoại idNCC đến bảng NhaCungCap, NOT NULL TriGiaDon |
| 12 | ChiTietDonNhap | Khoá chính (idDonNhap, idDienThoai),  Khóa ngoại idDonNhap đến bảng DonNhap,  Khóa ngoại idDienThoai đến bảng DienThoai, CHECK (SoLuong > và DonGia > 0) |
| 13 | TaiKhoan | Khóa ngoại idNV,  NOT NULL MatKhau |

## Cài đặt các CSDL và các ràng buộc

### Bảng công việc

|  |
| --- |
| CREATE TABLE CongViec(  idCV nchar(100) CONSTRAINT PK\_CongViec PRIMARY KEY,  TenCV nvarchar(100) NOT NULL,  LuongTheoGio float CONSTRAINT CK\_Luong\_CV CHECK (LuongTheoGio > 0)  ) |

### Bảng nhân viên

|  |
| --- |
| CREATE TABLE NhanVien(  idNV nchar(100) CONSTRAINT PK\_NhanVien PRIMARY KEY,  idCV nchar(100) CONSTRAINT FK\_NhanVien\_CV FOREIGN KEY REFERENCES CongViec(idCV),  Ho\_Ten nvarchar(100) NOT NULL,  NgaySinh date NOT NULL CONSTRAINT CK\_ngaysinh\_NV CHECK (DATEDIFF(YEAR, NgaySinh, GETDATE()) >= 18),  DiaChi nvarchar(100),  GioiTinh nvarchar(100),  soDT\_NV nchar(100) NOT NULL,  HinhAnh image,  TrangThai nvarchar(max)  ) |

### Bảng ca làm việc

|  |
| --- |
| CREATE TABLE CaLamViec(  idCa nchar(10),  Gio\_BatDau time NOT NULL,  Gio\_KetThuc time NOT NUll,  CONSTRAINT PK\_CaLamViec PRIMARY KEY (idCa),  CONSTRAINT CK\_gio\_CLV CHECK (Gio\_KetThuc > Gio\_BatDau)  ) |

### Bảng phân ca

|  |
| --- |
| CREATE TABLE BangPhanCa(  idCa nchar(100),  idNV nchar(100) CONSTRAINT FK\_BangPhanCa\_NV FOREIGN KEY REFERENCES NhanVien(idNV),  NgayLam date NOT NULL,  TrangThai nvarchar(max),  CONSTRAINT PK\_BangPhanCa PRIMARY KEY (idCa, idNV, NgayLam),  ) |

### Bảng khách hàng

|  |
| --- |
| CREATE TABLE KhachHang(  idKH nchar(100) CONSTRAINT PK\_KhachHang PRIMARY KEY,  TenKH nvarchar(100) NOT NULL,  soDT\_KH nchar(100) NOT NULL,  DiaChi nvarchar(100)  ) |

### Bảng điện thoại

|  |
| --- |
| CREATE TABLE DienThoai(  idDienThoai nchar(100) CONSTRAINT PK\_DienThoai PRIMARY KEY,  TenHangDT nvarchar(100) CONSTRAINT FK\_DienThoai\_HangDT FOREIGN KEY REFERENCES HangDienThoai(TenHangDT),  TenDienThoai nvarchar(100) NOT NULL,  SoLuong float NOT NULL,  MauSac nvarchar(100) NOT NULL,  DungLuong nchar(100) NOT NULL,  GiaBan float NOT NULL,  TrangThai nvarchar(100),  HinhAnh image  ) |

### Bảng hãng điện thoại

|  |
| --- |
| CREATE TABLE HangDienThoai(  TenHangDT nvarchar(100) CONSTRAINT PK\_HangDienThoai PRIMARY KEY,  NguonGoc nvarchar(100) NOT NULL,  TrangThai nvarchar(100)  ) |

### Bảng hóa đơn

|  |
| --- |
| CREATE TABLE HoaDon(  idHD nchar(100) CONSTRAINT PK\_HoaDon PRIMARY KEY,  idNV nchar(100) CONSTRAINT FK\_HoaDon\_NV FOREIGN KEY REFERENCES NhanVien(idNV),  idKH nchar(100) CONSTRAINT FK\_HoaDon\_KH FOREIGN KEY REFERENCES KhachHang(idKH),  Ngay date NOT NULL,  TriGiaHD float NOT NULL CONSTRAINT CK\_HoaDon CHECK (TriGiaHD > 0),  TrangThai nvarchar(100) NOT NULL  ) |

### Bảng chi tiết đơn hàng

|  |
| --- |
| CREATE TABLE ChiTietHoaDon(  idHD nchar(100) CONSTRAINT FK\_ChiTietHoaDon\_HD FOREIGN KEY REFERENCES HoaDon(idHD),  idDienThoai nchar(100) CONSTRAINT FK\_ChiTietHoaDon\_DT FOREIGN KEY REFERENCES DienThoai(idDienThoai),  SoLuong float NOT NULL,  DonGia float NOT NULL,  TongTien float NOT NULL,  CONSTRAINT PK\_ChiTietHoaDon PRIMARY KEY (idHD,idDienThoai)  ) |

### Bảng nhà cung cấp

|  |
| --- |
| CREATE TABLE NhaCungCap(  idNCC nchar(100) CONSTRAINT PK\_NhaCungCung PRIMARY KEY,  TenNCC nvarchar(100) NOT NULL,  soDT\_NCC nchar(100) NOT NULL,  DiaChiNCC nvarchar(100)  ) |

### Bảng đơn nhập

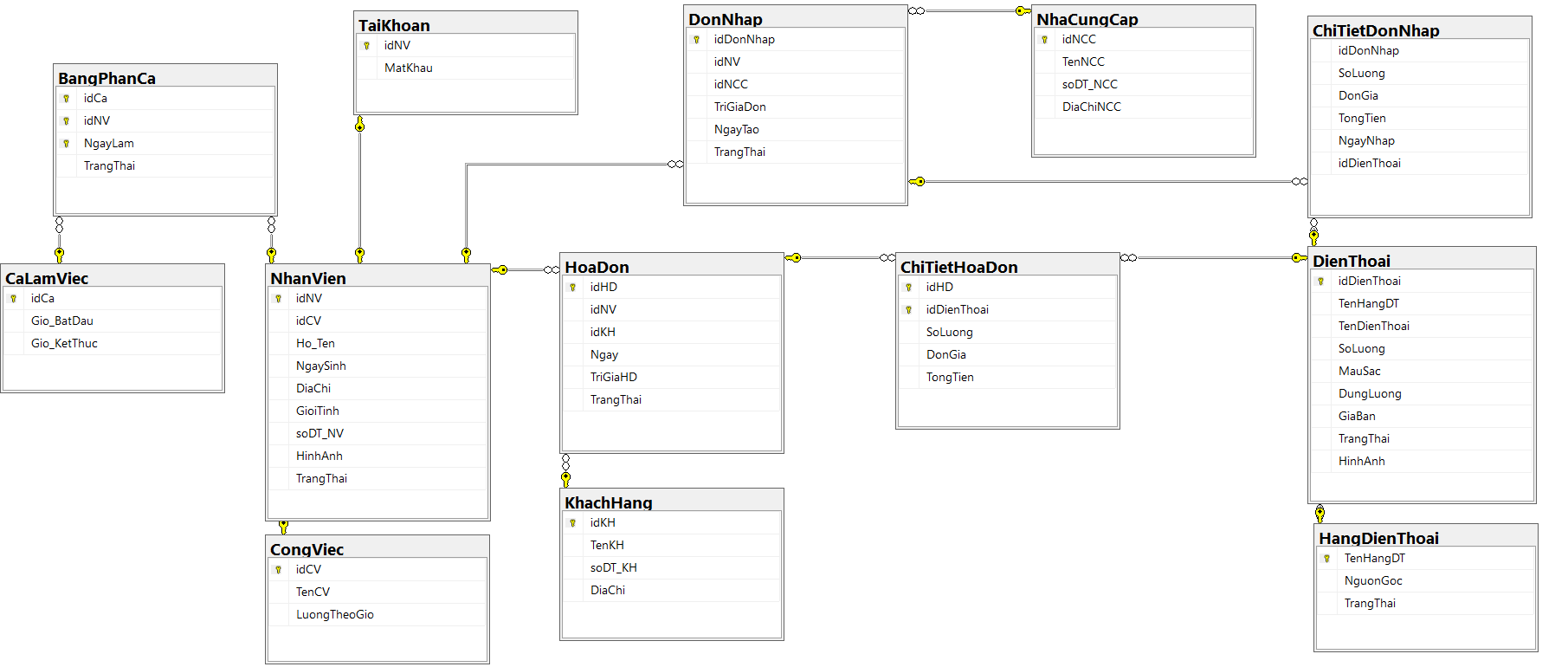
|  |
| --- |
| CREATE TABLE DonNhap(  idDonNhap nchar(100) CONSTRAINT PK\_DonNhap PRIMARY KEY,  idNV nchar(100) CONSTRAINT FK\_DonNhap\_NV FOREIGN KEY REFERENCES NhanVien(idNV),  idNCC nchar(100) CONSTRAINT FK\_DonNhap\_NCC FOREIGN KEY REFERENCES NhaCungCap(idNCC),  TriGiaDon float NOT NULL,  NgayTao date NOT NULL,  TrangThai NVARCHAR(100)  ) |

### Bảng chi tiết đơn nhập

|  |
| --- |
| CREATE TABLE ChiTietDonNhap(  idDonNhap nchar(100) CONSTRAINT FK\_ChiTietDonNhap\_DN FOREIGN KEY REFERENCES DonNhap(idDonNhap),  SoLuong float NOT NULL,  DonGia float NOT NULL,  TongTien float NOT NULL,  NgayNhap date,  idDienThoai nchar(100) CONSTRAINT FK\_ChiTietDonNhap\_DT FOREIGN KEY REFERENCES DienThoai(idDienThoai),  CONSTRAINT CK\_CTDN CHECK (SoLuong > 0 AND DonGia > 0)  ) |

### Bảng tài khoản

|  |
| --- |
| CREATE TABLE TaiKhoan(  idNV nchar(100) CONSTRAINT PK\_TK PRIMARY KEY,  MatKhau nchar(100) NOT NULL,  CONSTRAINT FK\_TK FOREIGN KEY (idNV) REFERENCES NhanVien(idNV)  ) |

******

## Các view

### 5.1. View xem danh sách điện thoại

|  |
| --- |
| CREATE or ALTER VIEW v\_dienthoai  AS  SELECT \* FROM DienThoai  GO |

### 5.2. View xem danh sách hóa đơn

|  |
| --- |
| CREATE or ALTER VIEW v\_hoadon  AS  SELECT \* FROM HoaDon  go |

### 5.3. View xem danh sách khách hàng

|  |
| --- |
| CREATE or ALTER VIEW v\_khachhang  AS  SELECT \* FROM KhachHang  go |

### 5.4. View xem danh sách nhân viên

|  |
| --- |
| CREATE or ALTER VIEW v\_nhanvien  AS  SELECT NhanVien.idNV, NhanVien.Ho\_Ten, NhanVien.soDT\_NV, CongViec.TenCV, NhanVien.TrangThai, NhanVien.HinhAnh  FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV  WHERE CongViec.TenCV <> N'Quản lý'  GO |

### 5.5. View xem bảng phân ca

|  |
| --- |
| CREATE or ALTER VIEW v\_phancong  AS  SELECT BangPhanCa.NgayLam ,NhanVien.idNV, NhanVien.Ho\_Ten ,BangPhanCa.idCa, BangPhanCa.TrangThai  FROM BangPhanCa join NhanVien ON BangPhanCa.idNV = NhanVien.idNV join CongViec ON NhanVien.idCV = CongViec.idCV join CaLamViec on BangPhanCa.idCa = CaLamViec.idCa  go |

## Các trigger

### 6.1. Trigger cập nhập lại số lượng điện thoại khi xóa hóa đơn

|  |
| --- |
| CREATE or ALTER TRIGGER HoaDon\_InsteadOfDelete --DONE  ON HoaDon  INSTEAD OF DELETE  AS  BEGIN  -- Tạo bảng tạm thời lưu danh sách các id của hóa đơn bị xóa  DECLARE @DeletedHD TABLE (idHD nchar(10));  -- Lấy danh sách các id của hóa đơn bị xóa và lưu vào bảng tạm thời  INSERT INTO @DeletedHD (idHD)  SELECT idHD FROM deleted;  -- Cập nhật lại số lượng bảng điện thoại bằng truy vấn lồng nhau  UPDATE DienThoai  SET SoLuong = DienThoai.SoLuong + D.SoLuong  FROM DienThoai  INNER JOIN (  SELECT idHD, idDienThoai, SUM(SoLuong) AS SoLuong  FROM ChiTietHoaDon  WHERE idHD IN (SELECT idHD FROM @DeletedHD)  GROUP BY idHD, idDienThoai  ) AS D ON DienThoai.idDienThoai = D.idDienThoai;  -- Tiếp theo, xóa chi tiết hóa đơn liên quan  DELETE FROM ChiTietHoaDon  WHERE idHD IN (SELECT idHD FROM @DeletedHD);  -- Tiếp theo, xóa hóa đơn  DELETE FROM HoaDon  WHERE idHD IN (SELECT idHD FROM @DeletedHD);  END;  GO |

### 6.2. Trigger kiểm tra thông tin tên và số điện thoại của khách hàng có trùng không

|  |
| --- |
| CREATE or ALTER TRIGGER KhachHang\_KiemTra\_TenKH\_sdtKH --DONE  ON KhachHang  AFTER INSERT  AS  BEGIN  IF EXISTS ( SELECT \* FROM inserted i WHERE EXISTS (SELECT \*FROM KhachHang k WHERE k.TenKH = i.TenKH AND k.idKH <> i.idKH))  BEGIN  -- Nếu trùng thì rollback  ROLLBACK;  RAISERROR('Tên Khách Hàng đã tồn tại!', 16,1)  END  IF EXISTS ( SELECT \* FROM inserted i WHERE EXISTS (SELECT \*FROM KhachHang k WHERE k.soDT\_KH = i.soDT\_KH AND k.idKH <> i.idKH))  BEGIN  -- Nếu trùng thì rollback  ROLLBACK;  RAISERROR('Số Điện Thoại Khách Hàng đã tồn tại!', 16,1)  END  END  GO |

### 6.3. Trigger kiểm tra tuổi và số điện thoại của nhân viên

|  |
| --- |
| CREATE OR ALTER TRIGGER NhanVien\_KiemTra\_Tuoi  ON NhanVien  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS ( SELECT 1 FROM inserted i WHERE DATEDIFF(YEAR, CAST(i.NgaySinh AS DATE), GETDATE()) < 18)  BEGIN  -- Nếu tuổi nhân viên dưới 18, thực hiện ROLLBACK và hiển thị thông báo lỗi  ROLLBACK;  RAISERROR('Nhân Viên phải lớn hơn hoặc bằng 18 tuổi!', 16, 1);  END  -- Kiểm tra số điện thoại nhân viên  IF EXISTS ( SELECT 1 FROM inserted i WHERE LEN(i.soDT\_NV) <> 10 )  BEGIN  -- Nếu số điện thoại không có đúng 10 số, thực hiện ROLLBACK và hiển thị thông báo lỗi  ROLLBACK;  RAISERROR('Số Điện Thoại Nhân Viên sai định dạng!', 16, 1);  END  END;  GO |

### 6.4. Trigger cập nhập trạng thái của điện thoại

|  |
| --- |
| CREATE OR ALTER TRIGGER DienThoai\_KiemTra\_TrangThai --DONE  ON DienThoai  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra nếu có ít nhất một bản ghi có TrangThai là 'Ngừng Kinh Doanh'  IF EXISTS (SELECT 1 FROM inserted WHERE TrangThai = N'Ngừng Kinh Doanh')  BEGIN  UPDATE DienThoai  SET TrangThai = N'Ngừng Kinh Doanh'  FROM DienThoai  INNER JOIN inserted ON DienThoai.idDienThoai = inserted.idDienThoai  WHERE inserted.TrangThai = N'Ngừng Kinh Doanh';  END  ELSE  BEGIN  -- Kiểm tra nếu có ít nhất một bản ghi có SoLuong > 0  IF EXISTS (SELECT 1 FROM inserted WHERE SoLuong > 0)  BEGIN  UPDATE DienThoai  SET TrangThai = N'Còn hàng'  FROM DienThoai  INNER JOIN inserted ON DienThoai.idDienThoai = inserted.idDienThoai  WHERE inserted.SoLuong > 0;  END  ELSE  BEGIN  UPDATE DienThoai  SET TrangThai = N'Hết hàng'  FROM DienThoai  INNER JOIN inserted ON DienThoai.idDienThoai = inserted.idDienThoai;  END  END  END  GO |

### 6.5. Trigger cập nhập trạng thái và mật khẩu tài khoản của nhân viên

|  |
| --- |
| CREATE OR ALTER TRIGGER NhanVien\_CapNhap\_TrangThai\_MatKhau  ON NhanVien  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  IF EXISTS (SELECT \* FROM NhanVien INNER JOIN inserted i ON NhanVien.idNV = i.idNV WHERE i.TrangThai IS NULL OR NhanVien.TrangThai = N'Non-Active')  BEGIN  UPDATE NhanVien  SET TrangThai = N'Active', MatKhau = '123456'  FROM NhanVien  INNER JOIN inserted i ON NhanVien.idNV = i.idNV  END  ELSE IF EXISTS (SELECT \* FROM NhanVien INNER JOIN inserted i ON NhanVien.idNV = i.idNV WHERE NhanVien.MatKhau is not null)  BEGIN  UPDATE NhanVien  SET MatKhau = i.MatKhau  FROM NhanVien  INNER JOIN inserted i ON NhanVien.idNV = i.idNV;  END  ELSE  BEGIN  UPDATE NhanVien  SET TrangThai = N'Non-Active', MatKhau = NULL  FROM NhanVien  INNER JOIN inserted i ON NhanVien.idNV = i.idNV  DECLARE @userName nvarchar(10);  SET @userName = (SELECT idNV FROM inserted);  -- Drop user  EXEC('DROP USER ' + @userName);  -- Drop login  EXEC('DROP LOGIN ' + @userName);  END  END;  GO |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

## Kết nối cơ sở dữ liệu

|  |
| --- |
| public class DBConnection  {  public static SqlConnection conn;  public static string pq;  public static void connn( string a)  {  string ConnStr = @"Data Source=localhost;Initial Catalog=QuanLyCuaHangDienThoai;User ID=" + a + "; Password=123456";  conn = new SqlConnection(ConnStr);  pq = a;  }  public static SqlConnection open()  {  try  {  if (conn.State == ConnectionState.Closed)  {  conn.Open();  return conn;  }  else  {  return conn;  }  }  catch (SqlException ex)  {  MessageBox.Show("User đăng nhập không tồn tại: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  return conn;  }  }  public static SqlConnection close()  {  if (conn.State == ConnectionState.Open)  {  conn.Close();  return conn;  }  else { return conn; }  }  } |

## Xem thông tin danh mục

### 2.1. Danh mục điện thoại

|  |
| --- |
| CREATE or ALTER VIEW v\_dienthoai  AS  SELECT \* FROM DienThoai  GO |

Thực thi trên C#:

|  |
| --- |
| public void danhsach\_dienthoai()  {  try  {  string query = "SELECT \* FROM v\_dienthoai";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Product.DataSource = table;  dataGridView\_Product.Columns[0].HeaderText = "ID";  dataGridView\_Product.Columns[1].HeaderText = "Hãng";  dataGridView\_Product.Columns[2].HeaderText = "Tên";  dataGridView\_Product.Columns[3].HeaderText = "Màu sắc";  dataGridView\_Product.Columns[4].HeaderText = "Dung lượng";  dataGridView\_Product.Columns[5].HeaderText = "Giá";  dataGridView\_Product.Columns[6].HeaderText = "Số lượng";  dataGridView\_Product.Columns[7].HeaderText = "Trạng Thái";  dataGridView\_Product.Columns[8].HeaderText = "Hình ảnh";  }  catch(Exception ex)  {  MessageBox.Show(ex.Message);  }  finally  {  DBConnection.close();  }  } |

### 2.2. Danh mục hóa đơn

|  |
| --- |
| CREATE or ALTER VIEW v\_hoadon  AS  SELECT \* FROM HoaDon  go |

Thực thi trên C#:

|  |
| --- |
| public void danhsach\_hoadon()  {  try  {  string query = "SELECT \* FROM v\_hoadon";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Order.DataSource = table;  dataGridView\_Order.Columns[0].HeaderText = "ID Hóa Đơn";  dataGridView\_Order.Columns[1].HeaderText = "ID Nhân Viên";  dataGridView\_Order.Columns[2].HeaderText = "ID Khách Hàng";  dataGridView\_Order.Columns[3].HeaderText = "Ngày Tạo";  dataGridView\_Order.Columns[4].HeaderText = "Trị Giá";  dataGridView\_Order.Columns[5].HeaderText = "Tình Trạng";  }  catch(Exception ex)  {  MessageBox.Show(ex.Message);  }  finally { DBConnection.close(); }  } |

### 2.3. Danh mục khách hàng

|  |
| --- |
| CREATE or ALTER VIEW v\_khachhang  AS  SELECT \* FROM KhachHang  go |

Thực thi trên C#:

|  |
| --- |
| public void danhsach\_khachhang()  {  try  {  string query = "SELECT \* FROM v\_khachhang";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Customer.DataSource = table;  dataGridView\_Customer.Columns[0].HeaderText = "ID";  dataGridView\_Customer.Columns[1].HeaderText = "Họ và Tên";  dataGridView\_Customer.Columns[2].HeaderText = "Số Điện Thoại";  dataGridView\_Customer.Columns[3].HeaderText = "Địa Chỉ";  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  finally { DBConnection.close(); }  } |

### 2.4. Danh mục nhân viên

|  |
| --- |
| CREATE or ALTER VIEW v\_nhanvien  AS  SELECT NhanVien.idNV, NhanVien.Ho\_Ten, NhanVien.soDT\_NV, CongViec.TenCV, NhanVien.TrangThai, NhanVien.HinhAnh  FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV  WHERE CongViec.TenCV <> N'Quản lý'  GO |

Thực thi trên C#:

|  |
| --- |
| public void danhsach\_nhanvien()  {  string query = "SELECT \* FROM v\_nhanvien";  try  {  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Employee.DataSource = table;  dataGridView\_Employee.Columns[0].HeaderText = "ID";  dataGridView\_Employee.Columns[1].HeaderText = "Họ và Tên";  dataGridView\_Employee.Columns[2].HeaderText = "Số Điện Thoại";  dataGridView\_Employee.Columns[3].HeaderText = "Vị Trí";  dataGridView\_Employee.Columns[4].HeaderText = "Trạng Thái";  }  catch(SqlException ex)  {  MessageBox.Show(ex.Message);  this.Hide();  }  finally { DBConnection.close(); }  } |

## Quản lý hàng hóa (điện thoại)

### 3.1. Procedure - Thêm hoặc sửa thông tin điện thoại

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_DienThoai\_InsertOrUpdate -- DONE  @idDienThoai nchar(10),  @TenDienThoai nvarchar(255) = null,  @TenHangDT nvarchar(255) = null,  @MauSac nvarchar(255) = null,  @DungLuong nvarchar(255) = null,  @GiaBan float = null,  @SoLuong float = null,  @TinhTrang nvarchar(255) = null,  @HinhAnh varbinary(max) = NULL  AS  BEGIN  IF(@TenDienThoai = '' OR @TenHangDT = '' OR @MauSac = '' OR @DungLuong = '')  BEGIN  RAISERROR (N'Kiểm tra thông tin điện thoại! ',16,1);  RETURN  END  IF( @SoLuong < 0)  BEGIN  RAISERROR (N'Kiểm tra thông tin lại Số Lượng! ',16,1);  RETURN  END  IF(@GiaBan <= 0)  BEGIN  RAISERROR (N'Kiểm tra thông tin lại Giá! ',16,1);  RETURN  END  IF EXISTS (SELECT 1 FROM DienThoai WHERE idDienThoai = @idDienThoai)  BEGIN  -- Bản ghi đã tồn tại, thực hiện UPDATE  UPDATE DienThoai  SET  TenDienThoai = @TenDienThoai,  TenHangDT = @TenHangDT,  MauSac = @MauSac,  DungLuong = @DungLuong,  GiaBan = @GiaBan,  SoLuong = @SoLuong,  TrangThai = @TinhTrang,  HinhAnh = @HinhAnh  WHERE idDienThoai = @idDienThoai;  END  ELSE  BEGIN  -- Bản ghi chưa tồn tại, thực hiện INSERT  INSERT INTO DienThoai (idDienThoai, TenDienThoai, TenHangDT, MauSac, DungLuong, GiaBan, SoLuong, TrangThai, HinhAnh)  VALUES (@idDienThoai, @TenDienThoai, @TenHangDT, @MauSac, @DungLuong, @GiaBan, @SoLuong, @TinhTrang, @HinhAnh);  END;  END;  GO |

Thực thi trên C#

|  |
| --- |
| public void them\_sua\_dienthoai(Byte[] b)  {    try  {  SqlCommand cmd = new SqlCommand("proc\_DienThoai\_InsertOrUpdate", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idDienThoai", txb\_id.Text.Trim());  cmd.Parameters.AddWithValue("@TenDienThoai", txb\_ten.Text.Trim());  cmd.Parameters.AddWithValue("@TenHangDT", cbbHang.Text.Trim());  cmd.Parameters.AddWithValue("@MauSac", txb\_mausac.Text.Trim());  cmd.Parameters.AddWithValue("@DungLuong", cbbDungLuong.Text.Trim());  cmd.Parameters.AddWithValue("@GiaBan", txb\_gia.Text.Trim());  cmd.Parameters.AddWithValue("@SoLuong", txb\_soluong.Text.Trim());  cmd.Parameters.AddWithValue("@HinhAnh", b);  cmd.ExecuteNonQuery();  lb\_thongbao.Visible = true;  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  DBConnection.close();  }  } |

### 3.2. Procedure - Xóa thông tin điện thoại

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_DienThoai\_Delete --DONE  (  @idDienThoai NVARCHAR(max)  )  AS  BEGIN  IF EXISTS (SELECT \* FROM DienThoai WHERE idDienThoai = @idDienThoai AND SoLuong = 0)  BEGIN  IF NOT EXISTS (SELECT \* FROM ChiTietDonNhap join DonNhap ON ChiTietDonNhap.idDonNhap = DonNhap.idDonNhap WHERE idDienThoai = @idDienThoai AND TrangThai = N'Chưa nhận')  BEGIN  IF EXISTS (SELECT \* FROM ChiTietHoaDon join HoaDon ON ChiTietHoaDon.idHD = HoaDon.idHD WHERE ChiTietHoaDon.idDienThoai = @idDienThoai)  BEGIN  UPDATE DienThoai  SET TrangThai = N'Ngừng Kinh Doanh'  WHERE idDienThoai = @idDienThoai  END  ELSE  BEGIN  DELETE FROM ChiTietDonNhap WHERE idDienThoai = @idDienThoai  DELETE FROM DienThoai WHERE idDienThoai = @idDienThoai  END  END  ELSE  BEGIN  RAISERROR('Điện Thoại đang được nhập về!', 16, 1);  RETURN  END  END  ELSE  BEGIN  RAISERROR('Điện Thoại này vẫn còn hàng!', 16, 1);  RETURN  END  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void DienThoai\_Xoa()  {  DialogResult result = MessageBox.Show("Bạn có chắc muốn xóa sản phẩm", "thông báo", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  string err;  try  {  string a = dataGridView\_Product.Rows[mouseLocation.RowIndex].Cells["idDienThoai"].Value.ToString();  string sql = "exec proc\_DienThoai\_Delete '" + a.Trim() + "'";  SqlCommand cmd = new SqlCommand();  cmd.Connection = DBConnection.open();  cmd.CommandText = sql;  cmd.ExecuteNonQuery();  err = "Xóa thành công";  reset(dataGridView\_Product);  }  catch (SqlException ex)  {  err = ex.Message;  }  DBConnection.close();  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else  {  }  } |

### 3.3. Procedure - Thêm vào giỏ hàng để thực hiện tạo hóa đơn

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_DienThoai\_ThemGioHang -- DONE  (  @idDienThoai nvarchar(max) = null,  @SoLuong int = null  )  AS  BEGIN  IF EXISTS ( SELECT \* FROM DienThoai WHERE idDienThoai = @idDienThoai AND SoLuong = 0)  BEGIN  RAISERROR('Hết Hàng!', 16, 1);  RETURN  END;  IF @idDienThoai is not null and @SoLuong > 0  BEGIN  DECLARE @so\_luong\_ton int;  -- Lấy số lượng tồn kho của điện thoại  SELECT @so\_luong\_ton = DienThoai.SoLuong  FROM DienThoai  WHERE idDienThoai = @idDienThoai;  -- Kiểm tra nếu @so\_luong lớn hơn số lượng tồn kho  IF @SoLuong <= @so\_luong\_ton  BEGIN  -- Truy vấn thông tin sản phẩm nếu điều kiện hợp lệ  SELECT DienThoai.idDienThoai, DienThoai.TenDienThoai, DienThoai.GiaBan, @SoLuong AS SoLuong  FROM DienThoai  WHERE DienThoai.idDienThoai = @idDienThoai;  END;  ELSE  BEGIN  RAISERROR('Vượt quá số lượng cho phép!', 16, 1);  RETURN  END;  END;  ELSE  BEGIN  RAISERROR('None', 16, 1);  RETURN  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void DienThoai\_ThemGioHang(string soluong)  {  try  {  string a = dataGridView\_Product.Rows[mouseLocation.RowIndex].Cells["idDienThoai"].Value.ToString();  string b = soluong;  // Kiểm tra xem ID đã tồn tại trong DataTable chưa  DataRow existingRow = dt.Rows.Cast<DataRow>().FirstOrDefault(row => row["idDienThoai"].ToString() == a);  if (existingRow != null)  {  // Nếu ID đã tồn tại, cộng số lượng lại  int currentQuantity = Convert.ToInt32(existingRow["SoLuong"]);  int newQuantity = currentQuantity + Convert.ToInt32(b);  existingRow["SoLuong"] = newQuantity;  }  else  {  // Nếu ID chưa tồn tại, thêm vào DataTable  string query = "exec proc\_DienThoai\_ThemGioHang " + a + "," + b;  SqlDataAdapter adapter = new SqlDataAdapter(query, DBConnection.open());  DataTable dt\_giohang = new DataTable();  adapter.Fill(dt\_giohang);  dt.Merge(dt\_giohang);  }  // Hiển thị dữ liệu trên DataGridView  dataGridView\_GioHang.DataSource = dt;  dataGridView\_GioHang.Columns[0].HeaderText = "ID";  dataGridView\_GioHang.Columns[1].HeaderText = "Tên Điện Thoại";  dataGridView\_GioHang.Columns[2].HeaderText = "Giá Bán";  dataGridView\_GioHang.Columns[3].HeaderText = "Số Lượng";  }  catch (SqlException ex)  {  string err = ex.Message;  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  DBConnection.close();  }  } |

### 3.4. Function - Tìm kiếm điện thoại theo hãng và giá

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_DienThoai\_TK\_Hang\_Gia -- DONE  (  @hang\_dt NVARCHAR(max) = null,  @giaban NVARCHAR(max) = null  )  RETURNS TABLE  AS  RETURN  (  WITH ResultTable AS  (  SELECT \*  FROM DienThoai  )  SELECT \*  FROM ResultTable  WHERE  (  (@hang\_dt IS NOT NULL AND @hang\_dt != '' AND TenHangDT LIKE '%' + @hang\_dt)  OR  @hang\_dt = ''  )  AND  (  (  @giaban = '< 10tr' AND GiaBan < 10000000  )  OR  (  @giaban = 'Từ 10tr - 25tr' AND GiaBan <= 25000000 AND GiaBan >= 10000000  )  OR  (  @giaban = '> 25tr' AND GiaBan > 25000000  )  OR  @giaban = ''  )  );  GO |

Thực thi trên C#:

|  |
| --- |
| public void DienThoai\_TK\_HangvaGia()  {  string value1 = btnHang.Tag.ToString();  string value2 = btnGia.Tag.ToString();  try  {  string query = "SELECT \* FROM func\_DienThoai\_TK\_Hang\_Gia('" + value1.Trim() + "','" + value2.Trim() + "')";  SqlDataAdapter dap = new SqlDataAdapter(query, DBconn.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Product.DataSource = table;  if(table.Rows.Count == 0 )  {  MessageBox.Show("Không tìm thấy", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  finally { DBconn.close(); }  } |

### 3.5. Function - Tìm kiếm theo tên điện thoại

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_DienThoai\_TK\_TenDT -- DONE  (  @ten\_dt NVARCHAR(max) = null  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM DienThoai  WHERE TenDienThoai LIKE '%' + @ten\_dt + '%'  );  GO |

Thực thi trên C#:

|  |
| --- |
| public void DienThoai\_TK\_Ten()  {  string value = txbSearch.Text;  btnGia.Text = "Giá";  btnHang.Text = "Hãng Điện Thoại";  try  {  string query = "SELECT \* FROM func\_DienThoai\_TK\_TenDT ('"+ value.Trim()+ "')";  SqlDataAdapter dap = new SqlDataAdapter(query, DBconn.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Product.DataSource = table;  if (table.Rows.Count == 0)  {  MessageBox.Show("Không tìm thấy", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  finally  {  DBconn.close();  }  } |

### 3.6. Function – Lấy id lớn nhất của điện thoại để tạo điện thoại mới.

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_DienThoai\_idMAX()  RETURNS NCHAR(10)  AS  BEGIN  DECLARE @maxId NCHAR(10)  SELECT @maxId = MAX(idDienThoai) FROM DienThoai  RETURN @maxId  END  GO |

Thực thi trên C#:

|  |
| --- |
| public string DienThoai\_idMAX()  {  string k = "";  try  {  string str = "SELECT dbo.func\_DienThoai\_idMAX()";  SqlCommand cmd = new SqlCommand(str, DBConnection.open());  SqlDataReader reader = cmd.ExecuteReader();  while (reader.Read())  k = reader.GetValue(0).ToString();  reader.Close();  return k;  }  catch(Exception ex)  {  MessageBox.Show(ex.Message);  return k;  }  finally  {  DBConnection.close();  }  } |

## Quản lý hóa đơn

### 4.1. Procedure - Tạo hóa đơn mới

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_HoaDon\_Insert -- DONE  @idHD nchar(10),  @idNV nvarchar(255) = null,  @idKH nvarchar(255) = null,  @idDienThoai nvarchar(255) = null,  @Ngay date,  @TriGiaHD float = null,  @TrangThai nvarchar(255) = N'Hoàn Thành',  @SoLuong float = null,  @DonGia float = null,  @TongTien float = null  AS  BEGIN  IF(@idNV = '')  BEGIN  RAISERROR (N'Bạn chưa nhập thông tin Nhân Viên! ',16,1);  RETURN  END  IF(@idKH = '')  BEGIN  RAISERROR (N'Bạn chưa nhập thông tin Khách Hàng! ',16,1);  RETURN  END  IF(@idDienThoai = '')  BEGIN  RAISERROR (N'Bạn chưa nhập thông tin Điện Thoại! ',16,1);  RETURN  END  IF NOT EXISTS (SELECT \* FROM HoaDon WHERE idHD = @idHD)  BEGIN  INSERT INTO HoaDon(idHD, idNV, idKH, Ngay, TriGiaHD, TrangThai)  VALUES (@idHD, @idNV, @idKH, @Ngay, @TriGiaHD, @TrangThai);  INSERT INTO ChiTietHoaDon(idHD, idDienThoai, SoLuong, DonGia, TongTien)  VALUES (@idHD, @idDienThoai, @SoLuong, @DonGia, @TongTien);  UPDATE DienThoai  SET SoLuong = DienThoai.SoLuong - @SoLuong  WHERE idDienThoai = @idDienThoai  END  ELSE  BEGIN  -- Bản ghi đã tồn tại, thực hiện UPDATE  UPDATE HoaDon  SET  idHD = @idHD,  idNV = @idNV,  idKH = @idKH,  Ngay = @Ngay,  TriGiaHD = @TriGiaHD,  TrangThai = @TrangThai  WHERE idHD = @idHD;  INSERT INTO ChiTietHoaDon(idHD, idDienThoai, SoLuong, DonGia, TongTien)  VALUES (@idHD, @idDienThoai, @SoLuong, @DonGia, @TongTien);  UPDATE DienThoai  SET SoLuong = DienThoai.SoLuong - @SoLuong  WHERE idDienThoai = @idDienThoai  END  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void tao\_hoadon\_cthd()  {  string err = "";    try  {  tong = Convert.ToDouble(txb\_ThanhTien.Text);  Tongmoi = tong + Tongmoi;  txb\_TriGiaHD.Text = Tongmoi.ToString();  SqlConnection connection = DBConnection.open();  SqlCommand cmd = new SqlCommand("proc\_HoaDon\_Insert", connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idHD", txb\_idHD.Text.Trim());  cmd.Parameters.AddWithValue("@idNV", cbo\_idNV.Text.Trim());  cmd.Parameters.AddWithValue("@idKH", txb\_idKH.Text.Trim());  cmd.Parameters.AddWithValue("@idDienThoai", cbo\_idDT.Text.Trim());  cmd.Parameters.AddWithValue("@Ngay", txb\_Ngay.Text.Trim());  cmd.Parameters.AddWithValue("@TriGiaHD", txb\_TriGiaHD.Text.Trim());  cmd.Parameters.AddWithValue("@SoLuong", txb\_SoLuong.Text.Trim());  cmd.Parameters.AddWithValue("@DonGia", txb\_DonGia.Text.Trim());  cmd.Parameters.AddWithValue("@TongTien", txb\_ThanhTien.Text.Trim());  cmd.ExecuteNonQuery();  lblBangChu.Text = "Bằng chữ: " + Functions.NumberToWords((int)Tongmoi) + " đồng";  err = "Lưu thông tin thành công";  ResetValuesHang();  }  catch (SqlException ex)  {  err = ex.Message;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  } |

### 4.2. Procedure - Xóa hóa đơn

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_HoaDon\_Delete --DONE  (  @idHD nchar(10),  @PhanQuyen int  )  AS  BEGIN  IF (@PhanQuyen = 1)  BEGIN  DELETE FROM HoaDon  WHERE idHD = @idHD  END  ELSE  BEGIN  UPDATE HoaDon  SET TrangThai = N'Đã Hủy'  WHERE idHD = @idHD  END  END  GO |

Thực Thi trên C#

|  |
| --- |
| public void HoaDon\_Xoa(int pq)  {  string value = dataGridView\_Order.Rows[mouseLocation.RowIndex].Cells["idHD"].Value.ToString();  string sql = "exec proc\_HoaDon\_Delete '" + value.Trim() + "'," + pq;  DialogResult result = MessageBox.Show("Bạn Có Chắc Chắn Muốn Xóa Hóa Đơn " + value, "Thông Báo", MessageBoxButtons.OK);  if (result == DialogResult.OK)  {  try  {  SqlCommand cmd = new SqlCommand(sql,DBConnection.open());  cmd.ExecuteNonQuery();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  finally { DBConnection.close(); }  }  } |

### 4.3. Procedure - Xem chi tiết hóa đơn

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_HoaDon\_ChiTietHoaDon --DONE  (  @id\_hoadon nvarchar(max)  )  AS  BEGIN  IF @id\_hoadon IS NOT NULL  BEGIN  -- Lấy thông tin Khách hàng  SELECT DISTINCT KhachHang.idKH, KhachHang.TenKH, KhachHang.soDT\_KH, KhachHang.DiaChi  FROM KhachHang, HoaDon  WHERE HoaDon.idKH = KhachHang.idKH AND HoaDon.idHD = @id\_hoadon;  -- Lấy thông tin Nhân viên  SELECT DISTINCT NhanVien.idNV, NhanVien.Ho\_Ten, CongViec.TenCV  FROM NhanVien  JOIN CongViec ON NhanVien.idCV = CongViec.idCV  JOIN HoaDon ON HoaDon.idNV = NhanVien.idNV  WHERE HoaDon.idHD = @id\_hoadon;  -- Lấy thông tin Điện thoại  SELECT DienThoai.idDienThoai, DienThoai.TenDienThoai, DienThoai.TenHangDT, DienThoai.MauSac, DienThoai.DungLuong, DienThoai.GiaBan, ChiTietHoaDon.SoLuong  FROM DienThoai, ChiTietHoaDon  WHERE ChiTietHoaDon.idDienThoai = DienThoai.idDienThoai AND ChiTietHoaDon.idHD = @id\_hoadon;  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void HoaDon\_ChiTietHoaDon(string value)  {  try  {  SqlCommand cmd = new SqlCommand("proc\_HoaDon\_ChiTietHoaDon", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@id\_hoadon", value);  DataTable khachHangTable = new DataTable();  DataTable nhanVienTable = new DataTable();  DataTable dienThoaiTable = new DataTable();  SqlDataReader reader = cmd.ExecuteReader();  // Đọc và lưu trữ bảng kết quả Khách hàng  khachHangTable.Load(reader);  // Chuyển sang bảng kết quả Nhân viên  nhanVienTable.Load(reader);  // Chuyển sang bảng kết quả Điện thoại  dienThoaiTable.Load(reader);  dataGridView\_CTHD\_kh.DataSource = khachHangTable;  dataGridView\_CTDH\_nv.DataSource = nhanVienTable;  dataGridView\_CTHD\_dt.DataSource = dienThoaiTable;  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  finally  {  DBConnection.close();  }  } |

### 4.4. Function - Tìm kiếm hóa đơn theo ngày và trạng thái

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_HoaDon\_TK\_DateVaTrangThai  (  @Ngay NVARCHAR(max) = NULL,  @Thang NVARCHAR(max) = NULL,  @Nam NVARCHAR(max) = NULL,  @TrangThai NVARCHAR(max) = NULL  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM HoaDon  WHERE  -- Kiểm tra theo Ngày  (DAY(Ngay) = TRY\_CAST(@Ngay AS INT) OR @Ngay = '')  AND  -- Kiểm tra theo Tháng  (MONTH(Ngay) = TRY\_CAST(@Thang AS INT) OR @Thang = '')  AND  -- Kiểm tra theo Năm  (YEAR(Ngay) = TRY\_CAST(@Nam AS INT) OR @Nam = '')  AND  -- Kiểm tra theo Trạng thái  (@TrangThai = N'Tất Cả' OR TrangThai = @TrangThai OR @TrangThai IS NULL)  );  GO |

Thực thi trên C#:

|  |
| --- |
| public void HoaDon\_TK\_NgayHoacTrangThai()  {  try  {  string query = "SELECT \* FROM func\_HoaDon\_TK\_DateVaTrangThai('" + txb\_NgayHD.Text.Trim() +"','"+ txb\_ThangHD.Text.Trim() +"','"+ txb\_NamHD.Text.Trim() + "', N'" + cbb\_timkiem\_tt.Text.Trim() + "')";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_Order.DataSource = table;  if (table.Rows.Count == 0)  {  MessageBox.Show("Không tìm thấy", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);    }  finally  {  DBConnection.close();  }  } |

### 4.5. Function – Lấy id lớn nhất của hóa đơn để tạo hóa đơn mới.

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_HoaDon\_idMAX()  RETURNS NCHAR(10)  AS  BEGIN  DECLARE @maxId NCHAR(10)  SELECT @maxId = MAX(idHD) FROM HoaDon  RETURN @maxId  END  GO |

Thực thi trên C#:

|  |
| --- |
| public string HoaDon\_idMAX()  {  string k = "";  try  {  string str = "SELECT dbo.func\_HoaDon\_idMAX()";  SqlCommand cmd = new SqlCommand(str, DBConnection.open());  SqlDataReader reader = cmd.ExecuteReader();  while (reader.Read())  k = reader.GetValue(0).ToString();  reader.Close();  return k;  }  catch(Exception ex)  {  MessageBox.Show(ex.Message);  return k;  }  finally { DBConnection.close(); }  } |

### 4.6. Function – Tính tổng số lượng mặt hàng của một hóa đơn.

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_HoaDon\_TongSL  (  @idHD nchar(10)  )  RETURNS FLOAT  AS  BEGIN  DECLARE @TongSL INT  SELECT @TongSL = SUM(SoLuong)  FROM ChiTietHoaDon WHERE idHD = @idHD  RETURN @TongSL  END  GO |

Thực thi trên C#

|  |
| --- |
| public string HoaDon\_TongSL(string value)  {  string sum="";  try  {  string str = "SELECT dbo.func\_HoaDon\_TongSL('" + value + "')";  SqlCommand cmd = new SqlCommand(str, DBConnection.open());  SqlDataReader reader = cmd.ExecuteReader();  while (reader.Read())  sum = reader.GetValue(0).ToString();  reader.Close();  return sum;  }  catch(Exception ex)  {  MessageBox.Show(ex.Message);  return sum;  }  finally  {  DBConnection.close();  }  } |

## 5. Quản lý khách hàng

### 5.1. Procedure - Thêm hoặc sửa thông tin khách hàng

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_KhachHang\_InsertOrUpdate --DONE  @idKH nchar(10),  @TenKH nvarchar(255),  @DiaChi nvarchar(255),  @soDT\_KH nvarchar(20)  AS  BEGIN  IF(@TenKH = '' OR @DiaChi = '' OR @soDT\_KH = '' )  BEGIN  RAISERROR (N'Bạn chưa nhập thông tin Khách Hàng! ',16,1);  RETURN  END  IF EXISTS (SELECT 1 FROM KhachHang WHERE idKH = @idKH)  BEGIN  -- Bản ghi khách hàng đã tồn tại, thực hiện UPDATE  UPDATE KhachHang  SET  TenKH = @TenKH,  DiaChi = @DiaChi,  soDT\_KH = @soDT\_KH  WHERE idKH = @idKH;  END  ELSE  BEGIN  -- Bản ghi khách hàng chưa tồn tại, thực hiện INSERT  INSERT INTO KhachHang (idKH, TenKH, DiaChi, soDT\_KH)  VALUES (@idKH, @TenKH, @DiaChi, @soDT\_KH);  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void KhachHang\_ThemHoacSua()  {  string err;  try  {  SqlConnection connection = DBConnection.open();  SqlCommand cmd = new SqlCommand("proc\_KhachHang\_InsertOrUpdate", connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idKH", txb\_idKH.Text.Trim());  cmd.Parameters.AddWithValue("@TenKH", txb\_TenKH.Text.Trim());  cmd.Parameters.AddWithValue("@soDT\_KH", txb\_sdtKH.Text.Trim());  cmd.Parameters.AddWithValue("@DiaChi", txb\_DiaChiKH.Text.Trim());  cmd.ExecuteNonQuery();  err = "Thao tác Thành Công!";  }  catch (SqlException ex)  {  err = ex.Message;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 5.2. Procedure - Xóa khách hàng

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_KhachHang\_XoaKH --DONE  @idKH NVARCHAR(255)  AS  BEGIN  -- Kiểm tra xem khách hàng có đơn hàng đang tồn tại không  IF EXISTS (SELECT 1 FROM HoaDon WHERE idKH = @idKH)  BEGIN  RAISERROR('Đây là Khách Hàng tiềm năng!', 16, 1)  END  ELSE  BEGIN  DELETE FROM KhachHang WHERE idKH = @idKH  END  END  GO |

Thực thi trên C#:

|  |
| --- |
| public void KhachHang\_Xoa()  {  string err;  try  {  string sql;  string a = dataGridView\_Customer.Rows[mouseLocation.RowIndex].Cells["idKH"].Value.ToString();  sql = "exec proc\_KhachHang\_XoaKH '" + a.Trim() + "'";  SqlCommand cmd = new SqlCommand();  cmd.Connection = DBConnection.open();  cmd.CommandText = sql;  cmd.ExecuteNonQuery();  err = "Xóa Thành Công";  }  catch (SqlException ex)  {  err = ex.Message;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 5.3. Procedure - Xem lịch sử mua hàng của khách hàng

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_KhachHang\_lsmh -- DONE  (  @idKH nvarchar(max)  )  AS  BEGIN  IF EXISTS (SELECT \* FROM HoaDon join ChiTietHoaDon ON HoaDon.idHD = ChiTietHoaDon.idHD WHERE HoaDon.idKH = @idKH)  BEGIN  SELECT HoaDon.idHD, HoaDon.Ngay,DienThoai.TenDienThoai,DienThoai.MauSac,DienThoai.DungLuong,ChiTietHoaDon.SoLuong,ChiTietHoaDon.TongTien  FROM KhachHang  join HoaDon ON KhachHang.idKH = HoaDon.idKH  join ChiTietHoaDon ON HoaDon.idHD = ChiTietHoaDon.idHD  join DienThoai ON DienThoai.idDienThoai = ChiTietHoaDon.idDienThoai  WHERE KhachHang.idKH = @idKH  END;  ELSE  BEGIN  RAISERROR('Khách vãng lai!',16,1);  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void KhachHang\_lsmh()  {  try  {  string value = dataGridView\_Customer.Rows[mouseLocation.RowIndex].Cells["idKH"].Value.ToString();  string query = "exec proc\_KhachHang\_lsmh N'" + value.Trim() + "'";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  DataTable dt = table;  dataGridView\_lsmh.DataSource = dt;  btn\_lsmh.Text = "ID Khách Hàng [ " + value + " ]";  dataGridView\_lsmh.Columns[0].HeaderText = "ID Hóa Đơn";  dataGridView\_lsmh.Columns[1].HeaderText = "Ngày Mua";  dataGridView\_lsmh.Columns[2].HeaderText = "Tên Điện Thoại";  dataGridView\_lsmh.Columns[3].HeaderText = "Màu Sắc";  dataGridView\_lsmh.Columns[4].HeaderText = "Dung Lượng";  dataGridView\_lsmh.Columns[5].HeaderText = "Số Lượng";  dataGridView\_lsmh.Columns[6].HeaderText = "Tổng Tiền";  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  btn\_lsmh.Text = "ID Khách Hàng [ ]";  dataGridView\_lsmh.DataSource = null;  }  finally  {  DBConnection.close();  }  } |

### 5.4. Procedure - Tìm kiếm khách hàng theo số điện thoại

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_KhachHang\_TK\_sdt  (  @sdt varchar(max) = null  )  AS  BEGIN  IF @sdt IS NOT NULL AND LEN(@sdt) = 10  BEGIN  IF EXISTS (SELECT \* FROM KhachHang WHERE KhachHang.soDT\_KH LIKE '%' + @sdt + '%')  BEGIN  SELECT \*  FROM KhachHang  WHERE KhachHang.soDT\_KH LIKE '%' + @sdt + '%';  END  ELSE  BEGIN  RAISERROR('Không tìm thấy! - Hãy tạo thông tin!', 16, 1);  END;  END  ELSE  BEGIN  RAISERROR('Số điện thoại không hợp lệ!', 16, 1);  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void KhachHang\_TKsdt()  {  try  {  string value = txb\_TimKiem\_sdt.Text;  string query = "exec proc\_KhachHang\_TK\_sdt '" + value + "'";  SqlDataAdapter adapter = new SqlDataAdapter(query, DBConnection.open());  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView\_Customer.DataSource = dataTable;  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load\_refresh(dataGridView\_Customer);  }  finally  {  DBConnection.close();  }  } |

### 5.5. Function – Lấy id lớn nhất của khách hàng để tạo khách hàng mới.

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_KhachHang\_idMAX()  RETURNS NCHAR(10)  AS  BEGIN  DECLARE @maxId NCHAR(10)  SELECT @maxId = MAX(idKH) FROM KhachHang  RETURN @maxId  END  GO |

Thực thi trên C#:

|  |
| --- |
| public string KhachHang\_idMAX()  {  string k = "";  try  {  string str = "SELECT dbo.func\_KhachHang\_idMAX()";  SqlCommand cmd = new SqlCommand(str, DBConnection.open());  SqlDataReader reader = cmd.ExecuteReader();  while (reader.Read())  k = reader.GetValue(0).ToString();  reader.Close();  return k;  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  return k;  }  finally { DBConnection.close(); }  } |

## 6. Quản lí nhân viên

### 6.1. Procedure - Thêm hoặc sửa thông tin nhân viên

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_NhanVien\_InsertOrUpdate  @idNV nchar(10),  @idCV nchar(10) = null,  @Ho\_Ten nvarchar(255),  @DiaChi nvarchar(255),  @soDT\_NV nvarchar(20),  @NgaySinh nvarchar(255),  @GioiTinh nvarchar(255)  AS  BEGIN  IF (@Ho\_Ten = '' OR @DiaChi = '' OR @soDT\_NV = '' OR @NgaySinh = '' OR @GioiTinh = '' OR @idCV is null)  BEGIN  RAISERROR (N'Bạn chưa nhập đầy đủ thông tin Nhân Viên!', 16, 1);  RETURN;  END  IF EXISTS (SELECT 1 FROM NhanVien WHERE idNV = @idNV)  BEGIN  -- Bản ghi nhân viên đã tồn tại, thực hiện UPDATE  UPDATE NhanVien  SET  idCV = @idCV,  Ho\_Ten = @Ho\_Ten,  DiaChi = @DiaChi,  soDT\_NV = @soDT\_NV,  NgaySinh = @NgaySinh,  GioiTinh = @GioiTinh  WHERE idNV = @idNV;  END  ELSE  BEGIN  -- Bản ghi nhân viên chưa tồn tại, thực hiện INSERT  INSERT INTO NhanVien (idNV, idCV, Ho\_Ten, DiaChi, soDT\_NV, NgaySinh, GioiTinh)  VALUES (@idNV, @idCV, @Ho\_Ten, @DiaChi, @soDT\_NV, @NgaySinh, @GioiTinh);  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void NhanVien\_ThemHoacSua()  {  string err;  try  {  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_InsertOrUpdate", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", txb\_idNV.Text.Trim());  cmd.Parameters.AddWithValue("@idCV", cbb\_idCV.SelectedValue);  cmd.Parameters.AddWithValue("@Ho\_Ten", txb\_tenNV.Text.Trim());  cmd.Parameters.AddWithValue("@DiaChi", txb\_diachi.Text.Trim());  cmd.Parameters.AddWithValue("@soDT\_NV", txb\_sdtNV.Text.Trim());  cmd.Parameters.AddWithValue("@NgaySinh", dateTimePicker\_ngsinhNV.Value.ToString("MM/dd/yyyy").Trim());  cmd.Parameters.AddWithValue("@GioiTinh", cbb\_Phai.Text.Trim());  cmd.ExecuteNonQuery();  lb\_thongbao.Visible = true;  }  catch (SqlException ex)  {  err = ex.Message;  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  finally  {  DBConnection.close();  }  } |

### 6.2. Procedure - Xóa nhân viên

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_NhanVien\_XoaNV --DONE  @idNV NVARCHAR(255)  AS  BEGIN  IF NOT EXISTS (SELECT \* FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV WHERE idNV = @idNV AND TenCV = N'Quản lý')  BEGIN  UPDATE NhanVien SET TrangThai='' WHERE idNV = @idNV;  END;  ELSE  BEGIN  RAISERROR('Không thể Xóa!',16,1);  END;  END  GO |

Thực thi trên C#:

|  |
| --- |
| public void NhanVien\_Xoa() // XÓA NHÂN VIÊN  {  string err;  try  {  string sql;  string a = dataGridView\_Employee.Rows[mouseLocation.RowIndex].Cells["idNV"].Value.ToString();  sql = "exec proc\_NhanVien\_XoaNV '" + a.Trim() + "'";  SqlCommand cmd = new SqlCommand();  cmd.Connection = DBConnection.open();  cmd.CommandText = sql;  cmd.ExecuteNonQuery();  err = "Xóa Thành Công";  }  catch (SqlException ex)  {  err = ex.Message;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 6.3. Procedure - Tìm kiếm nhân viên theo sđt

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_TK\_sdt  (  @sdt varchar(max) = null  )  AS  BEGIN  IF @sdt IS NOT NULL AND LEN(@sdt) = 10  BEGIN  IF EXISTS (SELECT \* FROM NhanVien WHERE NhanVien.soDT\_NV LIKE '%' + @sdt + '%')  BEGIN  SELECT \*  FROM NhanVien  WHERE NhanVien.soDT\_NV LIKE '%' + @sdt + '%';  END  ELSE  BEGIN  RAISERROR('Không tìm thấy! - Hãy tạo thông tin!', 16, 1);  END;  END  ELSE  BEGIN  RAISERROR('Số điện thoại không hợp lệ!', 16, 1);  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void NhanVien\_TKsdt()  {  try  {  string value = txb\_TimKiem\_sdt.Text;  string query = "exec proc\_NhanVien\_TK\_sdt '" + value + "'";  SqlDataAdapter adapter = new SqlDataAdapter(query, DBConnection.open());  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView\_Employee.DataSource = dataTable;  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  finally  {  DBConnection.close();  }  } |

### 6.4. Procedure - Xem chi tiết thông tin của nhân viên

|  |
| --- |
| CREATE or ALTER PROCEDURE proc\_NhanVien\_ttct -- DONE  (  @idNV nvarchar(max)  )  AS  BEGIN  IF EXISTS (SELECT \* FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV WHERE NhanVien.idNV = @idNV)  BEGIN  -- Lấy thông tin Nhân viên  SELECT NhanVien.Ho\_Ten, NhanVien.NgaySinh, NhanVien.GioiTinh, NhanVien.soDT\_NV, NhanVien.DiaChi, NhanVien.TrangThai  FROM NhanVien WHERE NhanVien.idNV = @idNV  -- Lấy thông tin công việc của nhân viên  SELECT CongViec.idCV , CongViec.TenCV  FROM CongViec join NhanVien ON CongViec.idCV = NhanVien.idCV WHERE NhanVien.idNV = @idNV  END;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public bool NhanVien\_ThongTinChiTiet(string value)  {  try  {  SqlConnection connection = DBConnection.open();  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_ttct", connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", value);  DataTable nhanVienTable = new DataTable();  DataTable congviecTable = new DataTable();  SqlDataReader reader = cmd.ExecuteReader();  // Chuyển sang bảng kết quả Nhân viên  nhanVienTable.Load(reader);  // Chuyển sang bảng kết quả Công Việc  congviecTable.Load(reader);  dataGridView\_ttnv.DataSource = nhanVienTable;  dataGridView\_ttcv.DataSource = congviecTable;  return true; // Success  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);  return false; // Failure  }  finally  {  DBConnection.close();  }  } |

### 6.5. Function – Lấy id lớn nhất của nhân viên để tạo nhân viên mới.

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_NhanVien\_idMAX()  RETURNS NCHAR(10)  AS  BEGIN  DECLARE @maxId NCHAR(10)  SELECT @maxId = MAX(idNV) FROM NhanVien  RETURN @maxId  END  GO |

Thực thi trên C#:

|  |
| --- |
| public string NhanVien\_idMAX()  {  string k = "";  try  {  string str = "SELECT dbo.func\_NhanVien\_idMAX()";  SqlCommand cmd = new SqlCommand(str, DBConnection.open());  SqlDataReader reader = cmd.ExecuteReader();  while (reader.Read())  k = reader.GetValue(0).ToString();  reader.Close();  return k;  }  catch(Exception ex)  {  MessageBox.Show(ex.Message);  return k;  }  finally  {  DBConnection.close();  }  } |

### 6.6. Procedure – Đổi mật khẩu tài khoản đăng nhập

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_DoiMK  (  @idNV NVARCHAR(255),  @mkCu NVARCHAR(255),  @mkMoi NVARCHAR(255)  )  AS  BEGIN  -- Kiểm tra xem mật khẩu cũ có đúng không  IF EXISTS (SELECT 1 FROM NhanVien WHERE idNV = @idNV AND MatKhau = @mkCu)  BEGIN  -- Nếu đúng, cập nhật mật khẩu mới  UPDATE NhanVien  SET MatKhau = @mkMoi  WHERE idNV = @idNV;  END  ELSE  BEGIN  -- Nếu mật khẩu cũ không đúng, thông báo lỗi  RAISERROR('Mật khẩu cũ không đúng.', 16,1);  END  END  GO |

Thực thi trong C#:

|  |
| --- |
| public void NhanVien\_DoiMK()  {  string err;  try  {  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_DoiMK", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", idNV);  cmd.Parameters.AddWithValue("@mkCu", txb\_mkCu.Text.Trim());  cmd.Parameters.AddWithValue("@mkMoi", txb\_mkMoi.Text.Trim());  cmd.ExecuteNonQuery();  err = "Đổi mật khẩu thành công";  }  catch (SqlException ex)  {  err = ex.Message;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 6.7. Procedure – Nhân viên đăng nhập tài khoản

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_DangNhap  (  @username nchar(10),  @password nchar(15)  )  AS  BEGIN  IF (@username = '' OR @password = '')  BEGIN  RAISERROR('Vui lòng nhập thông tin đăng nhập', 16,1)  END  ELSE  BEGIN  IF EXISTS ( SELECT 1 FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV WHERE idNV = @username AND TenCV = N'Quản Lý' AND MatKhau = @password)  BEGIN  RETURN 0;  END  ELSE IF EXISTS ( SELECT 1 FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV WHERE idNV = @username AND TenCV <> N'Quản Lý' AND MatKhau = @password)  BEGIN  RETURN 1;  END  ELSE  BEGIN  RAISERROR ('Mật Khẩu không chính xác',16,1)  END  END  END  GO |

Thực thi trên C#:

|  |
| --- |
| public string checklogin()  {  string err;  string rs = "";  try  {  DBConnection.connn(txb\_username.Text);  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_DangNhap", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@username", txb\_username.Text.Trim());  cmd.Parameters.AddWithValue("@password", txb\_password.Text.Trim());  SqlParameter returnParam = cmd.Parameters.Add("@ReturnValue", SqlDbType.Int);  returnParam.Direction = ParameterDirection.ReturnValue;  // Execute the stored procedure  if(DBConnection.conn.State != ConnectionState.Closed)  {  cmd.ExecuteNonQuery();  // Retrieve the return value  int returnValue = (int)returnParam.Value;  if (returnValue == 0)  {  MessageBox.Show("'Quản Lý' Đăng nhập thành công!");  }  else if (returnValue == 1)  {  MessageBox.Show("'Nhân Viên' Đăng nhập thành công");  }  rs = "login";  }  }  catch (SqlException ex)  {  err = ex.Message;  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  finally  {  DBConnection.close();  }  return rs;  } |

### 6.8. Procedure – Nhân viên thay đổi thông tin cá nhân

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_ChinhSuaThongTinCaNha  (  @idNV nchar(10),  @Ho\_Ten nvarchar(255),  @DiaChi nvarchar(255),  @soDT\_NV nvarchar(20),  @NgaySinh nvarchar(255),  @GioiTinh nvarchar(255)  )  AS  BEGIN  UPDATE NhanVien  SET  Ho\_Ten = @Ho\_Ten,  DiaChi = @DiaChi,  soDT\_NV = @soDT\_NV,  NgaySinh = @NgaySinh,  GioiTinh = @GioiTinh  WHERE idNV = @idNV;  END |

Thực thi trên C#:

|  |
| --- |
| public void NhanVien\_cstt()  {  string err;  try  {  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_ChinhSuaThongTinCaNha", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", txb\_idNV.Text.Trim());  cmd.Parameters.AddWithValue("@Ho\_Ten", txb\_nameNV.Text.Trim());  cmd.Parameters.AddWithValue("@DiaChi", txb\_diachi.Text.Trim());  cmd.Parameters.AddWithValue("@soDT\_NV", txb\_sdtNV.Text.Trim());  cmd.Parameters.AddWithValue("@NgaySinh", txb\_NgaySinh.Text.Trim());  cmd.Parameters.AddWithValue("@GioiTinh", txb\_Phai.Text.Trim());  cmd.ExecuteNonQuery();  err = "Thay đổi thành công!";  }  catch (SqlException ex)  {  err = "Thay đổi thất bại! " + ex.Message ;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 6.9. Procedure – Nhân viên chấm công

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_DiemDanh  (  @idNV nchar(10)  )  AS  BEGIN  DECLARE @ngay DATETIME;  SET @ngay = GETDATE();  IF EXISTS (SELECT \* FROM BangPhanCa WHERE NgayLam = @ngay)  BEGIN  UPDATE BangPhanCa  SET  TrangThai = N'Hoàn Thành'  WHERE idNV = @idNV AND NgayLam = @ngay;  END  ELSE  BEGIN  RAISERROR ('Bạn không có ca làm việc ngày hôm nay!',16,1)  END  END |

Thực thi trên C#:

|  |
| --- |
| public void NhanVien\_DiemDanh()  {  string err;  try  {  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_DiemDanh", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", txb\_idNV.Text.Trim());  cmd.ExecuteNonQuery();  err = "Điểm danh thành công!";  btn\_DiemDanh.Visible= false;  }  catch (SqlException ex)  {  err = "Điểm danh thất bại! " + ex.Message;  }  finally  {  DBConnection.close();  }  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

### 6.10. Procedure – Nhân viên xem ca làm của mình

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_CaLam  (  @idNV nchar(10)  )  AS  BEGIN  DECLARE @ngay DATETIME;  SET @ngay = CONVERT(DATE, GETDATE());  IF EXISTS (SELECT \* FROM BangPhanCa WHERE NgayLam > @ngay)  BEGIN  SELECT \* FROM v\_phancong  WHERE idNV = @idNV AND NgayLam > @ngay;  END  ELSE  BEGIN  RAISERROR ('Bạn không có ca làm việc tương lai!',16,1)  END  END  GO |

Thực thi trên C#:

|  |
| --- |
| public void NhanVien\_CaLamTuongLai()  {  string err;  try  {  string query = "exec proc\_NhanVien\_CaLam " + txb\_idNV.Text.Trim();  SqlDataAdapter adapter = new SqlDataAdapter(query, DBConnection.open());  DataTable dt = new DataTable();  adapter.Fill(dt);  dataGridView\_CaNV.DataSource = dt;  }  catch (SqlException ex)  {  err = ex.Message;  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  finally  {  DBConnection.close();  }  } |

## 7. Quản lí ca làm việc

### 7.1. Procedure - Thêm ca làm việc cho nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_PhanCa\_Insert  (  @NgayLam date,  @idCa NCHAR(10) = NULL,  @idNV NCHAR(10) = NULL  )  AS  BEGIN  -- Kiểm tra xem các tham số có giá trị không  IF @NgayLam IS NOT NULL AND @idCa IS NOT NULL AND @idNV IS NOT NULL  BEGIN  -- Kiểm tra xem NgayLam có ở tương lai không  IF CONVERT(DATE, @NgayLam, 103) > CONVERT(DATE, GETDATE(), 103)  BEGIN  -- Thực hiện chèn dữ liệu vào bảng  INSERT INTO BangPhanCa(NgayLam, idCa, idNV)  VALUES (@NgayLam, @idCa, @idNV);  -- Nếu bạn muốn trả về giá trị nào đó, bạn có thể thêm câu lệnh SELECT tại đây  -- SELECT @@IDENTITY AS 'InsertedID';  -- Hoặc có thể không cần câu lệnh SELECT nếu không cần trả về giá trị  END  ELSE  BEGIN  -- Xử lý trường hợp NgayLam không ở tương lai  RAISERROR('Không thể thêm ca vào ngày ở quá khứ! Vui lòng chọn lại!', 16, 1);  END  END  ELSE  BEGIN  -- Xử lý trường hợp các tham số không hợp lệ  RAISERROR('Các tham số không được để trống.', 16, 1);  END  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public void PhanCa\_ThemCa(int value)  {  try  {  string NgayLam = txb\_ThangCa01.Text.Trim() + "/" + txb\_NgayCa01.Text.Trim() + "/" + txb\_NamCa01.Text.Trim();  if (value == 1)  {  SqlConnection connection = DBConnection.open();  SqlCommand cmd = new SqlCommand("func\_PhanCa\_Insert", connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@NgayLam", NgayLam);  cmd.Parameters.AddWithValue("@idCa", txb\_ca01.Text.Trim());  cmd.Parameters.AddWithValue("@idNV", cbb\_idNVCa01.Text.Trim());  cmd.ExecuteNonQuery();  PhanCa\_TK\_Ngay\_Ca(1);  }  else  {  SqlConnection connection = DBConnection.open();  SqlCommand cmd = new SqlCommand("func\_PhanCa\_Insert", connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@NgayLam", NgayLam);  cmd.Parameters.AddWithValue("@idCa", txb\_ca02.Text.Trim());  cmd.Parameters.AddWithValue("@idNV", cbb\_idNVCa02.Text.Trim());  cmd.ExecuteNonQuery();  PhanCa\_TK\_Ngay\_Ca(2);  }  MessageBox.Show("Thêm Thành Công", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  DBConnection.close();  }  } |

### 7.2. Function – Tìm kiếm ca làm theo ngày và ca

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_PhanCa\_TK\_Date\_Ca  (  @Ngay NVARCHAR(max) = NULL,  @Thang NVARCHAR(max) = NULL,  @Nam NVARCHAR(max) = NULL,  @CaLam nchar(10) = NULL  )  RETURNS TABLE  AS  RETURN  (  SELECT v\_phancong.NgayLam, v\_phancong.Ho\_Ten,v\_phancong.idCa, v\_phancong.TrangThai  FROM v\_phancong  WHERE  -- Kiểm tra theo Ngày  (DAY(NgayLam) = TRY\_CAST(@Ngay AS INT) OR @Ngay = '')  AND  -- Kiểm tra theo Tháng  (MONTH(NgayLam) = TRY\_CAST(@Thang AS INT) OR @Thang = '')  AND  -- Kiểm tra theo Năm  (YEAR(NgayLam) = TRY\_CAST(@Nam AS INT) OR @Nam = '')  AND  (idCa = @CaLam)  );  GO |

Thực thi trên C#:

|  |
| --- |
| public void PhanCa\_TK\_Ngay\_Ca(int value)  {  try  {  if(value == 1)  {  string query = "SELECT \* FROM func\_PhanCa\_TK\_Date\_Ca('" + txb\_NgayCa01.Text.Trim() + "','" + txb\_ThangCa01.Text.Trim() + "','" + txb\_NamCa01.Text.Trim() + "','" + txb\_ca01.Text.Trim() + "')";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_CA\_01.DataSource = table;  if (table.Rows.Count == 0)  {  MessageBox.Show("Không tìm thấy", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  else  {  string query = "SELECT \* FROM func\_PhanCa\_TK\_Date\_Ca('" + txb\_NgayCa01.Text.Trim() + "','" + txb\_ThangCa01.Text.Trim() + "','" + txb\_NamCa01.Text.Trim() + "','" + txb\_ca02.Text.Trim() + "')";  SqlDataAdapter dap = new SqlDataAdapter(query, DBConnection.open());  DataTable table = new DataTable();  dap.Fill(table);  dataGridView\_CA\_02.DataSource = table;  if (table.Rows.Count == 0)  {  MessageBox.Show("Không tìm thấy", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }    }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  DBConnection.close();  }  } |

## 8. Quản lí tài chính

### 8.1. Procedure – Hiện thị thông tin cần thiết

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_HienThiThongTin  (  @idNV nvarchar(max) = null, -- ID của nhân viên  @thang int = null, -- Tháng cần tính lương (ví dụ: 11)  @thangHD int = null, -- Tháng cần tính tổng hóa đơn (ví dụ: 11)  @thangDN int = null -- Tháng cần tính tổng đơn nhập (ví dụ: 11)  )  AS  BEGIN  IF @thang != ''  BEGIN  SELECT NgayLam, Ho\_Ten, idCa, TrangThai  FROM v\_phancong  WHERE (MONTH(NgayLam) = TRY\_CAST(@Thang AS INT) OR @Thang = '') AND idNV = @idNV;  END  ELSE IF @thangHD != ''  BEGIN  SELECT \* FROM HoaDon WHERE (MONTH(Ngay) = TRY\_CAST(@thangHD AS INT) OR @thangHD = '');  END  ELSE IF @thangDN != ''  BEGIN  SELECT \* FROM DonNhap WHERE (MONTH(NgayTao) = TRY\_CAST(@thangDN AS INT) OR @thangDN = '');  END  ELSE  BEGIN  SELECT null;  END  END  GO |

Thực thi trên C#:

|  |
| --- |
| public DataTable HienThiThongTin(string a, string b, string c, string d)  {  try  {  SqlCommand cmd = new SqlCommand("proc\_HienThiThongTin", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", a);  cmd.Parameters.AddWithValue("@thang", b);  cmd.Parameters.AddWithValue("@thangHD", c);  cmd.Parameters.AddWithValue("@thangDN", d);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  return dataTable;  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return null;  }  finally  {  DBConnection.close();  }  } |

### 8.2. Function – Tính tổng số lượng các hóa đơn, đơn nhập, ca làm theo tháng

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_TongSo  (  @idNV nvarchar(max) = null, -- ID của nhân viên  @thangNV int = null, -- Tháng cần tính lương ví dụ 11  @thangHD int = null,  @thangDN int = null  )  RETURNS INT -- Điều này tùy thuộc vào định dạng lương của bạn  AS  BEGIN  DECLARE @Tong INT  DECLARE @thangDateNV date  DECLARE @thangDateHD date  DECLARE @thangDateDN date  IF (@thangNV != '')  BEGIN  SET @thangDateNV = CAST(CONVERT(varchar(4), YEAR(GETDATE())) + '-' + CAST(@thangNV AS nvarchar(2)) + '-01' AS date)  SELECT @Tong = count(idCa)  FROM CongViec  join NhanVien ON CongViec.idCV = NhanVien.idCV  join BangPhanCa On NhanVien.idNV = BangPhanCa.idNV  WHERE NhanVien.idNV = @idNV  AND BangPhanCa.NgayLam >= @thangDateNV  AND BangPhanCa.NgayLam < DATEADD(MONTH, 1, @thangDateNV);  RETURN @Tong;  END  ELSE IF (@thangHD != '')  BEGIN  SET @thangDateHD = CAST(CONVERT(varchar(4), YEAR(GETDATE())) + '-' + CAST(@thangHD AS nvarchar(2)) + '-01' AS date)  SELECT @Tong = count(idHD)  FROM HoaDon  WHERE  HoaDon.Ngay >= @thangDateHD AND  HoaDon.Ngay < DATEADD(MONTH, 1, @thangDateHD);  RETURN @Tong;  END  ELSE IF (@thangDN != '')  BEGIN  SET @thangDateDN = CAST(CONVERT(varchar(4), YEAR(GETDATE())) + '-' + CAST(@thangDN AS nvarchar(2)) + '-01' AS date)  SELECT @Tong = count(idDonNhap)  FROM DonNhap  WHERE  DonNhap.NgayTao >= @thangDateDN AND  DonNhap.NgayTao < DATEADD(MONTH, 1, @thangDateDN) AND DonNhap.TrangThai = N'Đã Nhận';  RETURN @Tong;  END  RETURN 0;  END  GO |

Thực thi trên C#:

|  |
| --- |
| public string SoLuong(string a, string b, string c, string d)  {  try  {  SqlCommand cmd = new SqlCommand("dbo.func\_TongSo", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", a);  cmd.Parameters.AddWithValue("@thangNV", b);  cmd.Parameters.AddWithValue("@thangHD", c);  cmd.Parameters.AddWithValue("@thangDN", d);  SqlParameter returnParameter = cmd.Parameters.Add("@SoLuong", SqlDbType.Decimal);  returnParameter.Direction = ParameterDirection.ReturnValue;  cmd.ExecuteNonQuery(); // Thực hiện hàm  int SoLuong = (int)returnParameter.Value;  return SoLuong.ToString();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return "ERROR";  }  finally  {  DBConnection.close();  }  } |

### 8.3. Function – Tính thực hiện tính lương, tổng hóa đơn, tổng đơn nhập theo tháng

|  |
| --- |
| CREATE OR ALTER FUNCTION func\_TinhLuong\_HD\_DN  (  @idNV nvarchar(max) = null, -- ID của nhân viên  @thang int = null, -- Tháng cần tính lương (ví dụ: 11)  @thangHD int = null, -- Tháng cần tính tổng hóa đơn (ví dụ: 11)  @thangDN int = null -- Tháng cần tính tổng đơn nhập (ví dụ: 11)  )  RETURNS decimal(18, 0) -- Điều này tùy thuộc vào định dạng lương của bạn  AS  BEGIN  DECLARE @result decimal(18, 0);  -- Check for the case of calculating salary  IF (@thang != '')  BEGIN  DECLARE @luong decimal(18, 2)  DECLARE @thangDate date  -- Chuyển đổi giá trị tháng (int) thành ngày đầu tháng  SET @thangDate = CAST(CONVERT(varchar(4), YEAR(GETDATE())) + '-' + CAST(@thang AS nvarchar(2)) + '-01' AS date)  SELECT @luong = COUNT(idCa) \* 6 \* (  SELECT LuongTheoGio  FROM CongViec  JOIN NhanVien ON CongViec.idCV = NhanVien.idCV  WHERE NhanVien.idNV = @idNV  )  FROM CongViec  JOIN NhanVien ON CongViec.idCV = NhanVien.idCV  JOIN BangPhanCa ON NhanVien.idNV = BangPhanCa.idNV  WHERE NhanVien.idNV = @idNV  AND BangPhanCa.NgayLam >= @thangDate  AND BangPhanCa.NgayLam < DATEADD(MONTH, 1, @thangDate);  SET @result = @luong;  END  -- Check for the case of calculating total from HoaDon  ELSE IF (@thangHD != '')  BEGIN  DECLARE @thangDateHD date  -- Chuyển đổi giá trị tháng (int) thành ngày đầu tháng  SET @thangDateHD = CAST(CONVERT(varchar(4), YEAR(GETDATE())) + '-' + CAST(@thangHD AS nvarchar(2)) + '-01' AS date)  SELECT @result = SUM(TriGiaHD)  FROM HoaDon  WHERE HoaDon.Ngay >= @thangDateHD  AND HoaDon.Ngay < DATEADD(MONTH, 1, @thangDateHD);  END  -- Check for the case of calculating total from DonNhap  ELSE IF (@thangDN != '')  BEGIN  DECLARE @thangDateDN date  -- Chuyển đổi giá trị tháng (int) thành ngày đầu tháng  SET @thangDateDN = CAST(CONVERT(varchar(4), YEAR(GETDATE())) + '-' + CAST(@thangDN AS nvarchar(2)) + '-01' AS date)  SELECT @result = SUM(TriGiaDon)  FROM DonNhap  WHERE DonNhap.NgayTao >= @thangDateDN  AND DonNhap.NgayTao < DATEADD(MONTH, 1, @thangDateDN)  AND DonNhap.TrangThai = N'Đã Nhận';  END  ELSE  BEGIN  SET @result = 0; -- Default value when no case matches  END  RETURN @result;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| public string NhanVien\_TinhLuong\_HD\_DN(string a, string b, string c, string d)  {  try  {  SqlCommand cmd = new SqlCommand("dbo.func\_TinhLuong\_HD\_DN", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@idNV", a);  cmd.Parameters.AddWithValue("@thang", b);  cmd.Parameters.AddWithValue("@thangHD", c);  cmd.Parameters.AddWithValue("@thangDN", d);  SqlParameter returnParameter = cmd.Parameters.Add("@result", SqlDbType.Decimal);  returnParameter.Direction = ParameterDirection.ReturnValue;  cmd.ExecuteNonQuery(); // Thực hiện hàm  if(returnParameter.Value == DBNull.Value)  {  decimal result = 0;  return result.ToString();  }  else  {  decimal result = (decimal)returnParameter.Value;  return result.ToString();  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return "ERROR";  }  finally  {  DBConnection.close();  }  } |

# CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN

Ứng với hai nhóm người dùng (Nhân viên và quản lí) là các role riêng

+ Đối với Role NhanVien (dành cho nhân viên)

|  |
| --- |
| CREATE ROLE NhanVien  -- Các quyền cơ bản cho NhanVien  GRANT SELECT, REFERENCES ON BangPhanCa TO NhanVien  GRANT SELECT, REFERENCES ON CaLamViec TO NhanVien  GRANT SELECT, REFERENCES ON ChiTietHoaDon TO NhanVien  GRANT SELECT, REFERENCES ON ChiTietDonNhap TO NhanVien  GRANT SELECT, REFERENCES ON CongViec TO NhanVien  GRANT SELECT, REFERENCES ON NhanVien TO NhanVien  GRANT SELECT, REFERENCES ON DienThoai TO NhanVien  GRANT SELECT, REFERENCES ON HangDienThoai TO NhanVien  GRANT SELECT, REFERENCES ON NhaCungCap TO NhanVien  GRANT SELECT, INSERT, REFERENCES ON DonNhap TO NhanVien  GRANT SELECT, INSERT, REFERENCES ON HoaDon TO NhanVien  GRANT SELECT, INSERT, REFERENCES ON KhachHang TO NhanVien  -- Các quyền thực thi proc, func của NhanVien  GRANT EXECUTE TO NhanVien  GRANT SELECT TO NhanVien  -- Bỏ bớt các quyền thêm, xóa, sửa, cập nhật từ các proc của NhanVien  DENY EXECUTE ON proc\_DienThoai\_Delete TO NhanVien  DENY EXECUTE ON proc\_DienThoai\_InsertOrUpdate TO NhanVien  DENY EXECUTE ON proc\_KhachHang\_XoaKH TO NhanVien  DENY EXECUTE ON proc\_NhanVien\_InsertOrUpdate TO NhanVien  DENY EXECUTE ON proc\_NhanVien\_XoaNV TO NhanVien  DENY EXECUTE ON proc\_PhanCa\_Insert TO NhanVien  DENY SELECT ON func\_PhanCa\_TK\_Date\_Ca TO NhanVien  DENY EXECUTE ON proc\_NhanVien\_ttct TO NhanVien |

+ Đối với Role quản lí ta sử dụng server role sysadmin.

* Trigger tạo tài khoản

|  |
| --- |
| CREATE OR ALTER TRIGGER NhanVien\_TaoTaiKhoanSQL ON NhanVien  AFTER INSERT  AS  BEGIN  SET NOCOUNT ON;  DECLARE @userName nchar(10), @passWord nchar(15), @idNV nchar(10), @idCV nchar(10);  DECLARE @sqlString nvarchar(2000);  -- Create a cursor to iterate over the inserted rows  DECLARE cursorNhanVien CURSOR FOR  SELECT idNV FROM inserted;  OPEN cursorNhanVien;  -- Fetch the first row from the cursor  FETCH NEXT FROM cursorNhanVien INTO @idNV;  -- Loop through the cursor  WHILE @@FETCH\_STATUS = 0  BEGIN  -- Generate unique username based on idNV  SET @userName = @idNV;  -- Create login  SET @sqlString = 'CREATE LOGIN [' + @userName + '] WITH PASSWORD=''123456'', DEFAULT\_DATABASE=[QuanLyCuaHangDienThoai], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF';  EXEC (@sqlString);  -- Create user  SET @sqlString = 'CREATE USER ' + @userName + ' FOR LOGIN ' + @userName;  EXEC (@sqlString);  -- Get idCV for role assignment  SELECT @idCV = idCV FROM NhanVien WHERE idNV = @idNV;  -- Assign role based on idCV  IF (@idCV = 'CV\_03')  SET @sqlString = 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @userName;  ELSE  SET @sqlString = 'ALTER ROLE NhanVien ADD MEMBER ' + @userName;  -- Execute role assignment  EXEC (@sqlString);  -- Fetch the next row from the cursor  FETCH NEXT FROM cursorNhanVien INTO @idNV;  END;  -- Close and deallocate the cursor  CLOSE cursorNhanVien;  DEALLOCATE cursorNhanVien;  END; |

* Trigger xóa nhân viên sẽ xóa tài khoản

|  |
| --- |
| CREATE OR ALTER TRIGGER NhanVien\_CapNhap\_TrangThai\_MatKhau  ON NhanVien  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  IF EXISTS (SELECT \* FROM NhanVien INNER JOIN inserted i ON NhanVien.idNV = i.idNV WHERE i.TrangThai IS NULL OR NhanVien.TrangThai = N'Non-Active')  BEGIN  UPDATE NhanVien  SET TrangThai = N'Active', MatKhau = '123456'  FROM NhanVien  INNER JOIN inserted i ON NhanVien.idNV = i.idNV  END  ELSE  BEGIN  UPDATE NhanVien  SET TrangThai = N'Non-Active', MatKhau = NULL  FROM NhanVien  INNER JOIN inserted i ON NhanVien.idNV = i.idNV  DECLARE @userName nvarchar(10);  SET @userName = (SELECT idNV FROM inserted);  -- Drop user  EXEC('DROP USER ' + @userName);  -- Drop login  EXEC('DROP LOGIN ' + @userName);  END  END;  GO |

* Procedure kiểm tra đăng nhập

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_NhanVien\_DangNhap  (  @username nchar(10),  @password nchar(15)  )  AS  BEGIN  IF (@username = '' OR @password = '')  BEGIN  RAISERROR('Vui lòng nhập thông tin đăng nhập', 16,1)  END  ELSE  BEGIN  IF EXISTS ( SELECT 1 FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV WHERE idNV = @username AND TenCV = N'Quản Lý' AND MatKhau = @password)  BEGIN  RETURN 0;  END  ELSE IF EXISTS ( SELECT 1 FROM NhanVien join CongViec ON NhanVien.idCV = CongViec.idCV WHERE idNV = @username AND TenCV <> N'Quản Lý' AND MatKhau = @password)  BEGIN  RETURN 1;  END  ELSE  BEGIN  RAISERROR ('Mật Khẩu không chính xác',16,1)  END  END  END  GO |

Thực thi trong C#

|  |
| --- |
| public string checklogin()  {  string err;  string rs = "";  try  {  DBConnection.connn(txb\_username.Text);  SqlCommand cmd = new SqlCommand("proc\_NhanVien\_DangNhap", DBConnection.open());  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@username", txb\_username.Text.Trim());  cmd.Parameters.AddWithValue("@password", txb\_password.Text.Trim());  SqlParameter returnParam = cmd.Parameters.Add("@ReturnValue", SqlDbType.Int);  returnParam.Direction = ParameterDirection.ReturnValue;  // Execute the stored procedure  if(DBConnection.conn.State != ConnectionState.Closed)  {  cmd.ExecuteNonQuery();  // Retrieve the return value  int returnValue = (int)returnParam.Value;  if (returnValue == 0)  {  MessageBox.Show("'Quản Lý' Đăng nhập thành công!");  }  else if (returnValue == 1)  {  MessageBox.Show("'Nhân Viên' Đăng nhập thành công");  }  rs = "login";  }  }  catch (SqlException ex)  {  err = ex.Message;  MessageBox.Show(err, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  finally  {  DBConnection.close();  }  return rs;  } |

# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN

**Môi trường cài đặt và công nghệ sử dụng**- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 18.  
- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền  
tảng  
ASP.NET Web Application (.NET Framework) C#  
**Giao diện hệ thống**

## Giao diện đăng nhập và trang chủ:

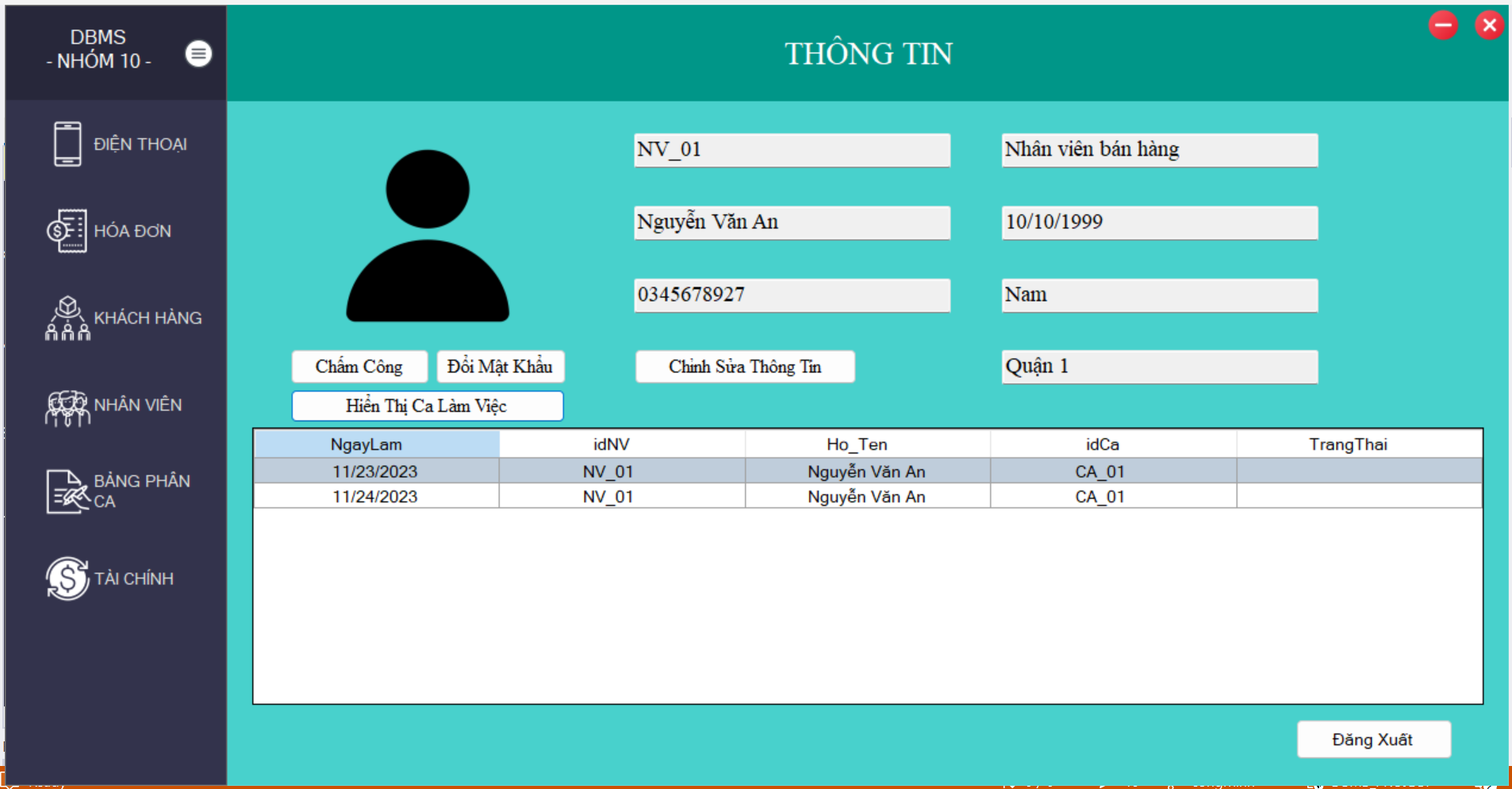
### Giao diện login:

* Thực hiện đăng nhập với Tài Khoản (nhân viên hoặc quản lí) đã được phân chia quyền.
* Các tài khoản sẽ được tự động tạo khi quản lí thực hiện thêm nhân viên.



### Giao diện home:

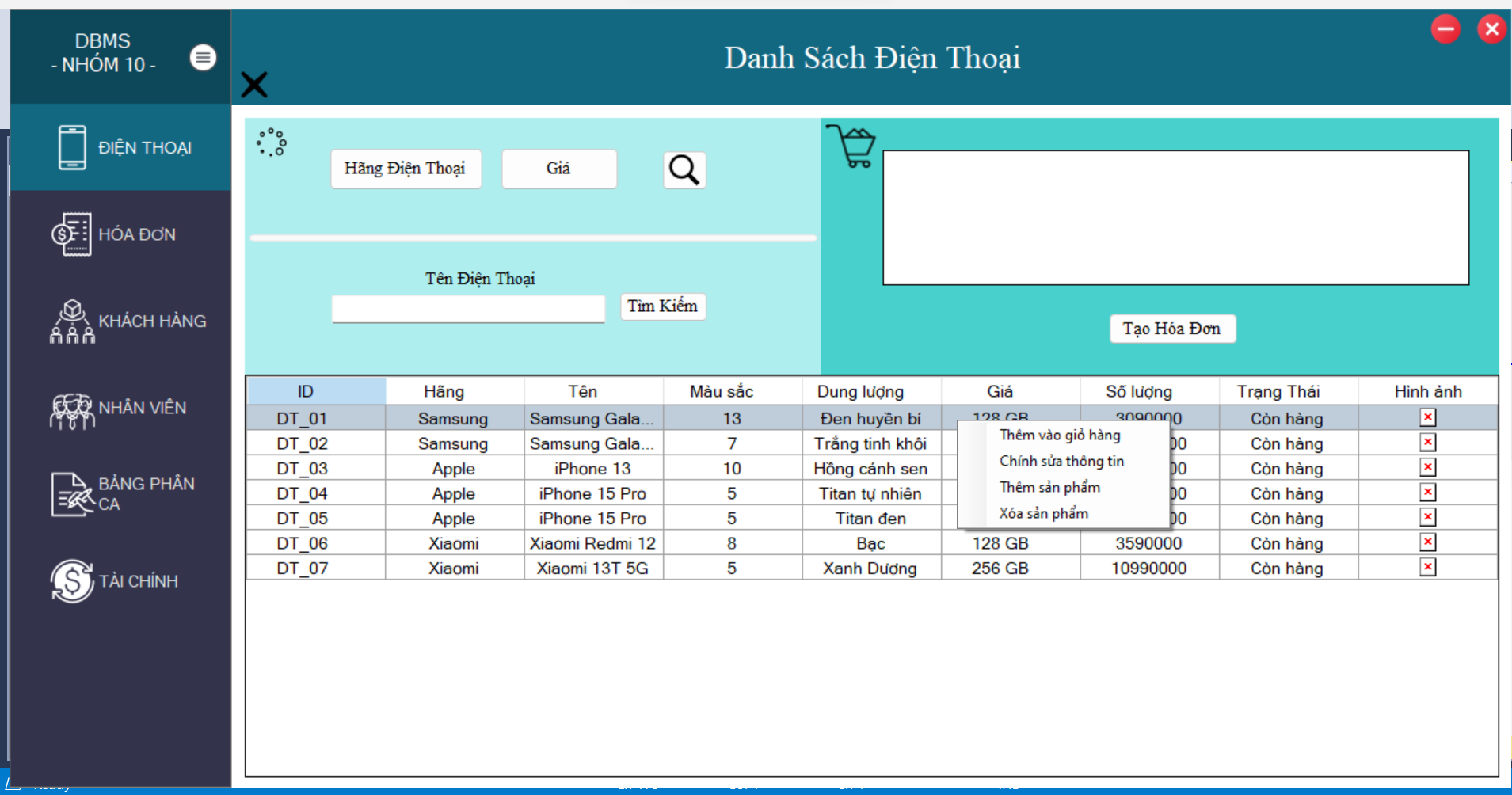
* Giao diện này hiển thị các thông tin cá nhân của Nhân Viên đó.
* Ở giao diện này nhân viên sẽ thể thực hiện chấm công, xem ca của mình trong tương lai, có thể đổi mật khẩu của tài khoản đăng nhập và chỉnh sửa thông tin cá nhân của bản thân.



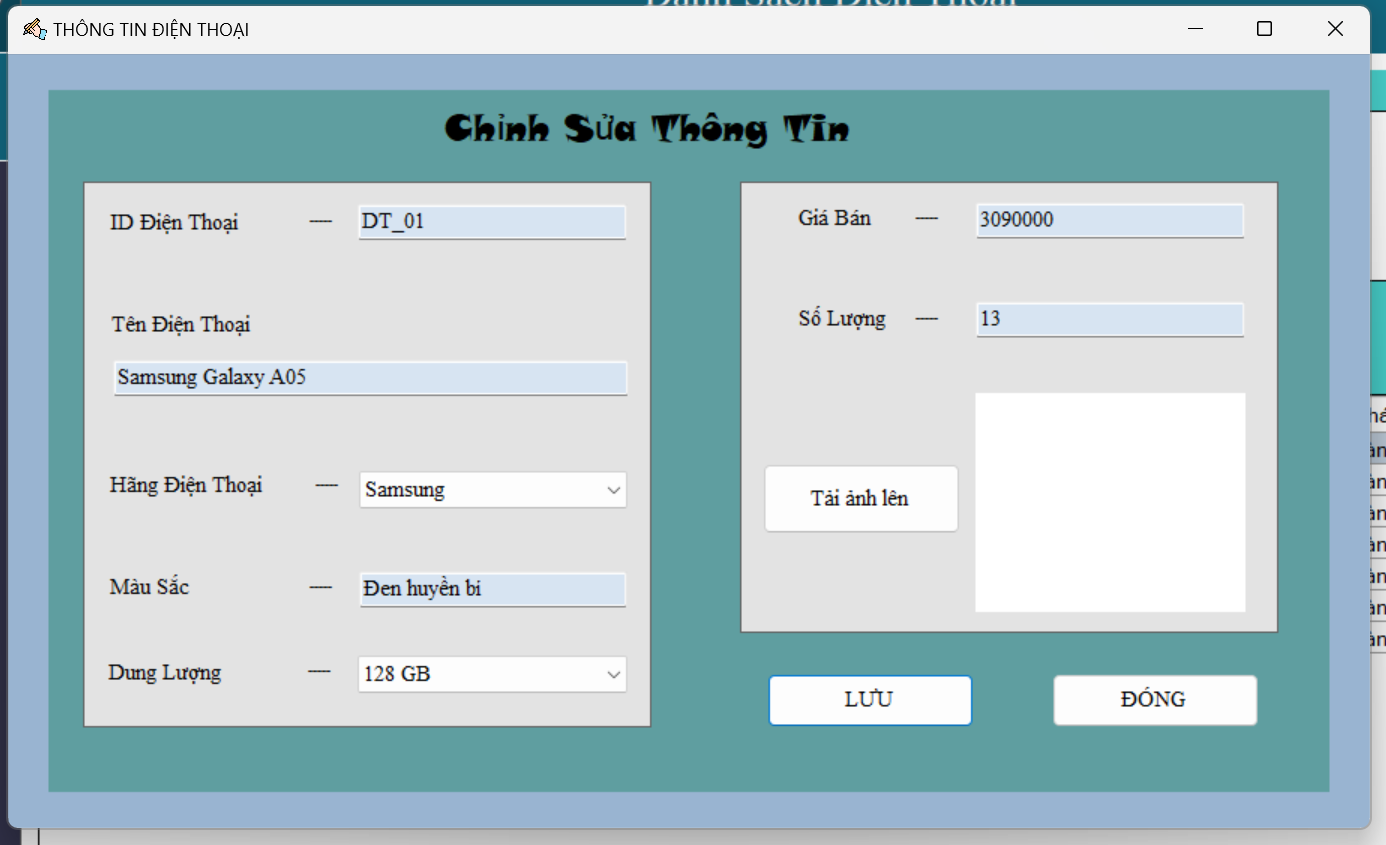
## Giao diện Điện Thoại:

### Giao diện danh mục Điện Thoại

* Trong giao diện này, Nhân Viên có thể tìm kiếm điện thoại theo hãng, giá hoặc tên điện thoại
* Có 4 chức năng chính sẽ được hiện thị khi chuột phải vào điện thoại.
  + Thêm giỏ hàng: Chức năng thực hiện khi thêm điện thoại vào giỏ hàng để thực hiện tạo hóa đơn
  + Các chức năng cơ bản: thêm, xóa, sửa



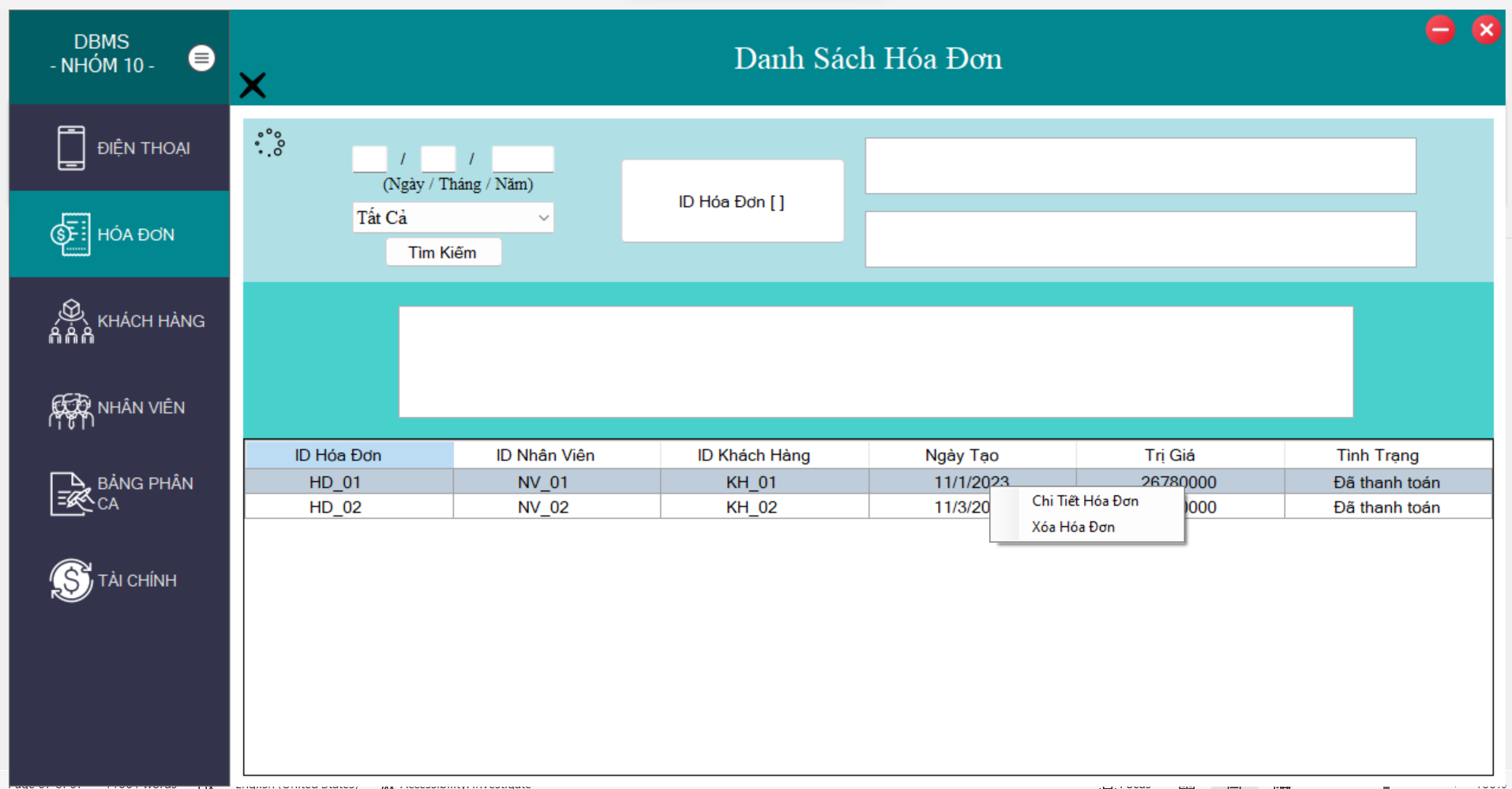
### Giao diện về thêm hoặc chỉnh sửa thông tin Điện Thoại:



## Giao diện Hóa Đơn:

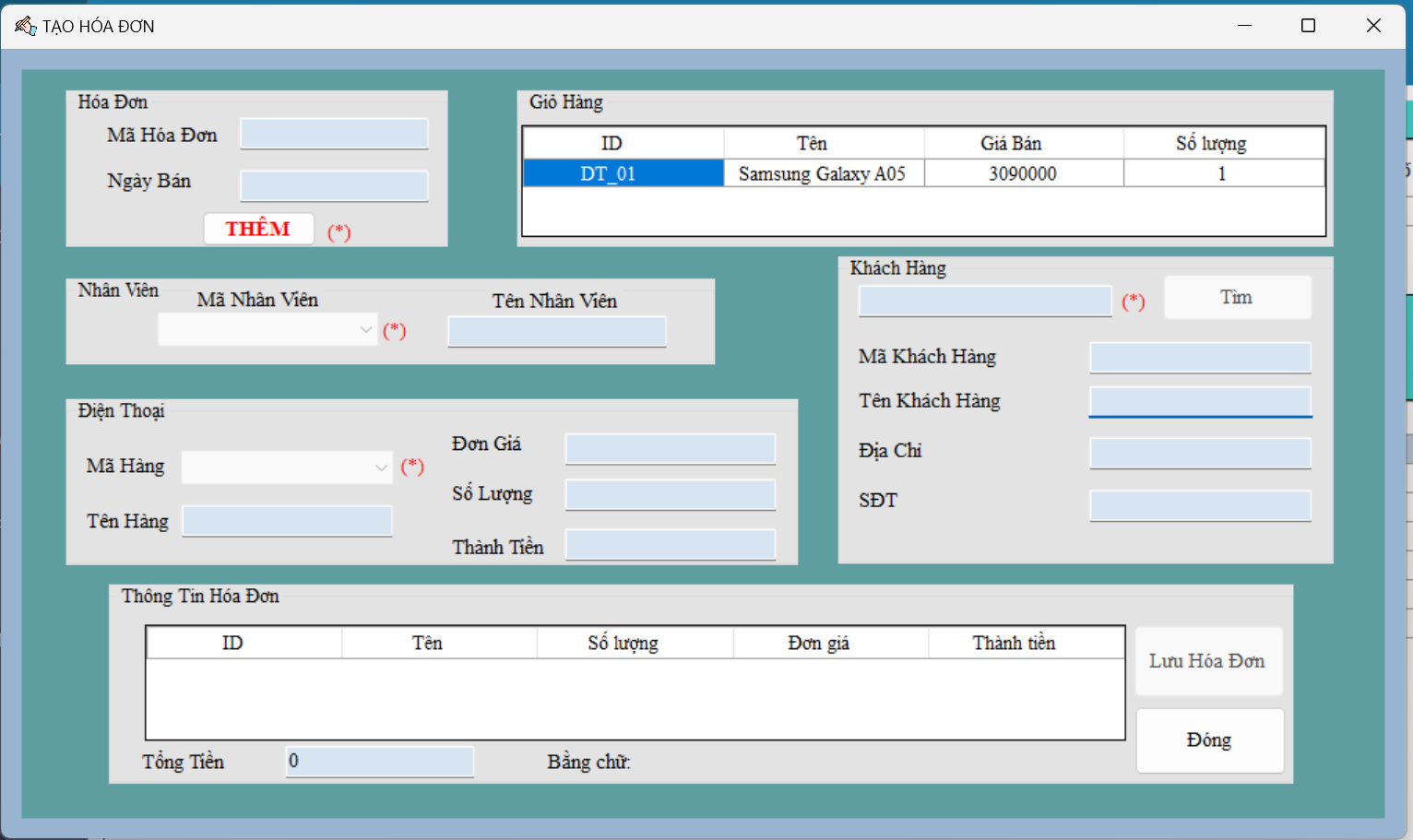
### Giao diện về danh mục Hóa Đơn

* Trong giao diện này, Nhân Viên có thể tìm kiếm hóa đơn theo ngày, tháng, năm, trạng thái.
* Có 2 chức năng chính sẽ được hiện thị khi chuột phải vào điện thoại.
  + Chi tiết hóa đơn: Xem thông tin chi tiết của hóa đơn đó
  + Chức năng xóa hóa đơn



### Giao diện tạo hóa đơn, khách hàng:

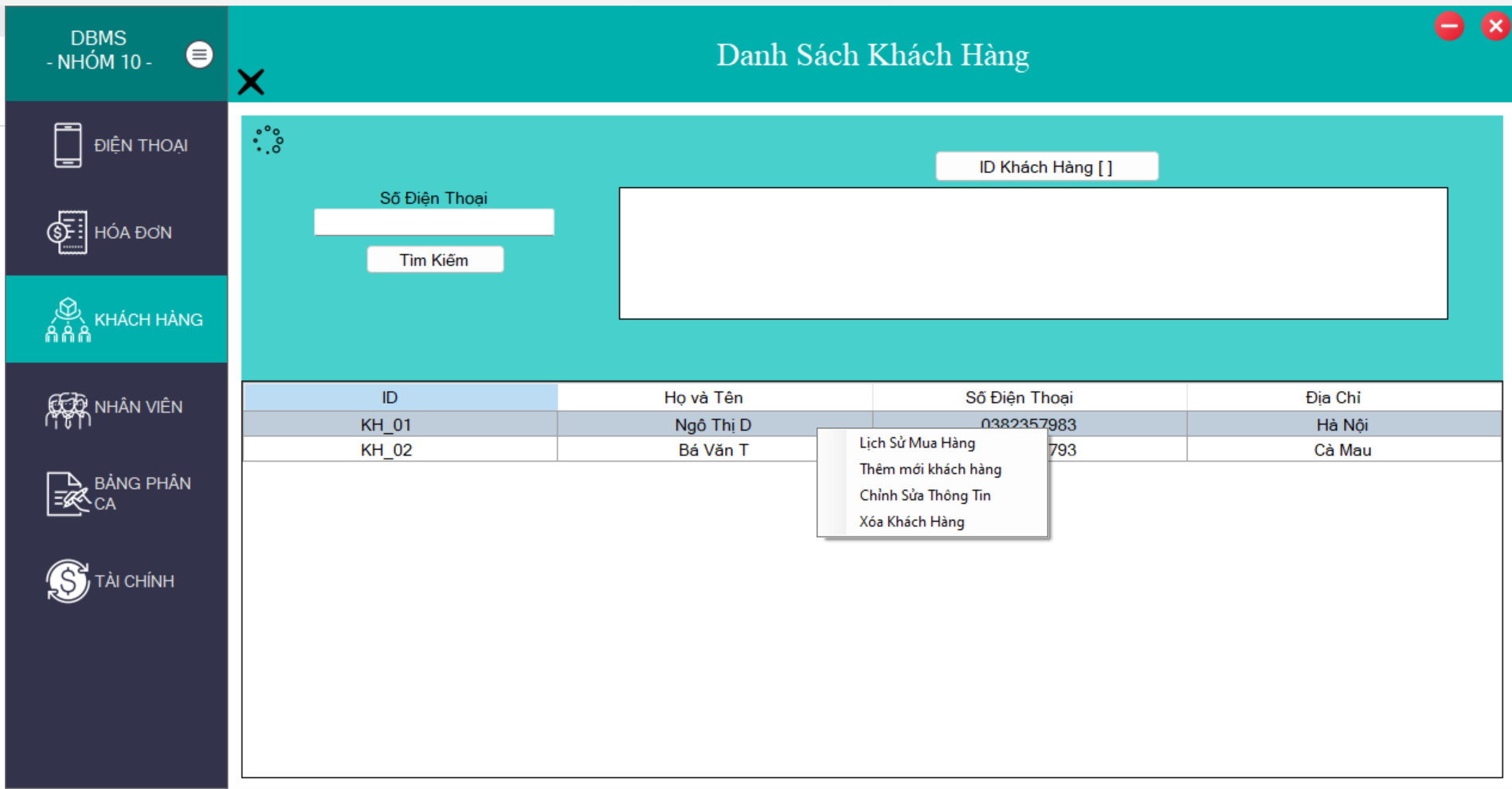
* Tạo khách hàng mua hóa đơn nếu như chưa có thông tin



## Giao diện Khách Hàng:

### Giao diện về danh mục Khách Hàng:

* Trong giao diện này, Nhân Viên có thể tìm kiếm khách hàng theo số điện thoại của khách hàng.
* Có 4 chức năng chính sẽ được hiện thị khi chuột phải vào điện thoại.
  + Lịch sử mua hàng: Xem thông tin các đơn hàng của khách hàng đó.
  + Các chức năng cơ bản: thêm, xóa, sửa



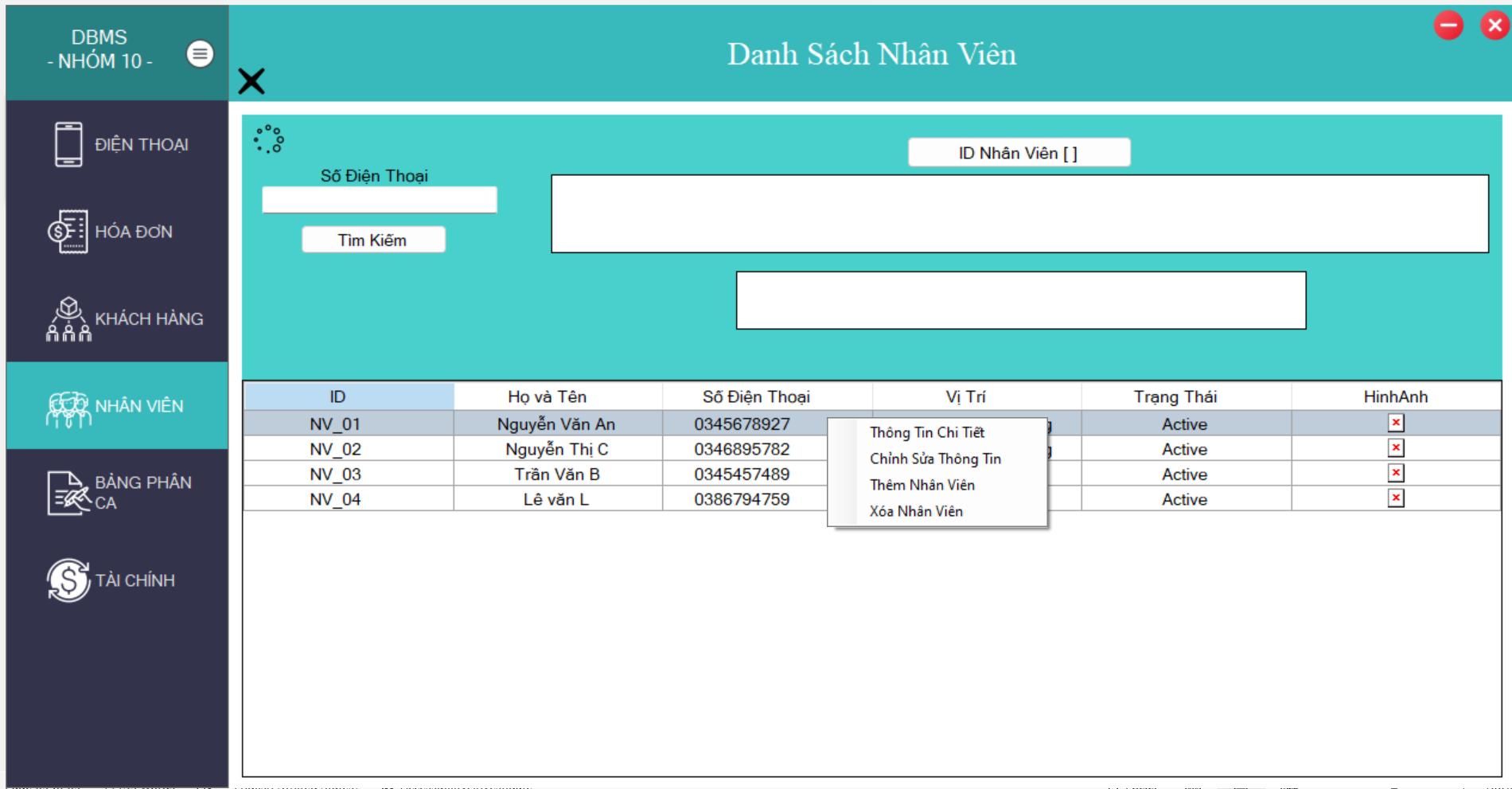
### Giao diện thêm hoặc chỉnh sửa thông tin khách hàng:



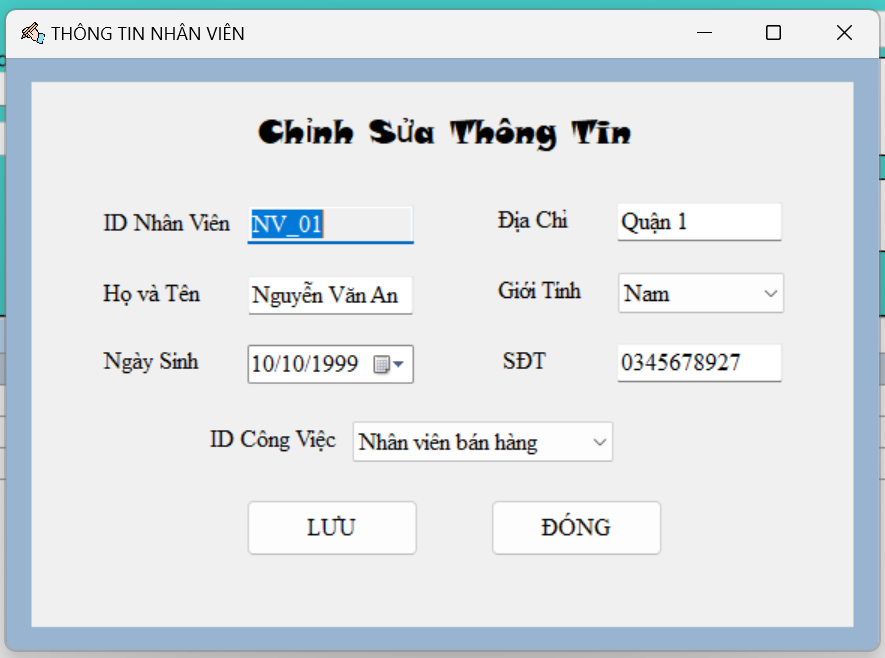
## Giao diện Nhân Viên

### Giao diện về danh mục Nhân Viên:

* Trong giao diện này, Quản Lý có thể tìm kiếm nhân viên theo hãng, giá hoặc tên điện thoại
* Có 4 chức năng chính sẽ được hiện thị khi chuột phải vào điện thoại.
  + Thông tin chi tiết: Xem các thông tin về nhân viên đó
  + Các chức năng cơ bản: thêm, xóa, sửa

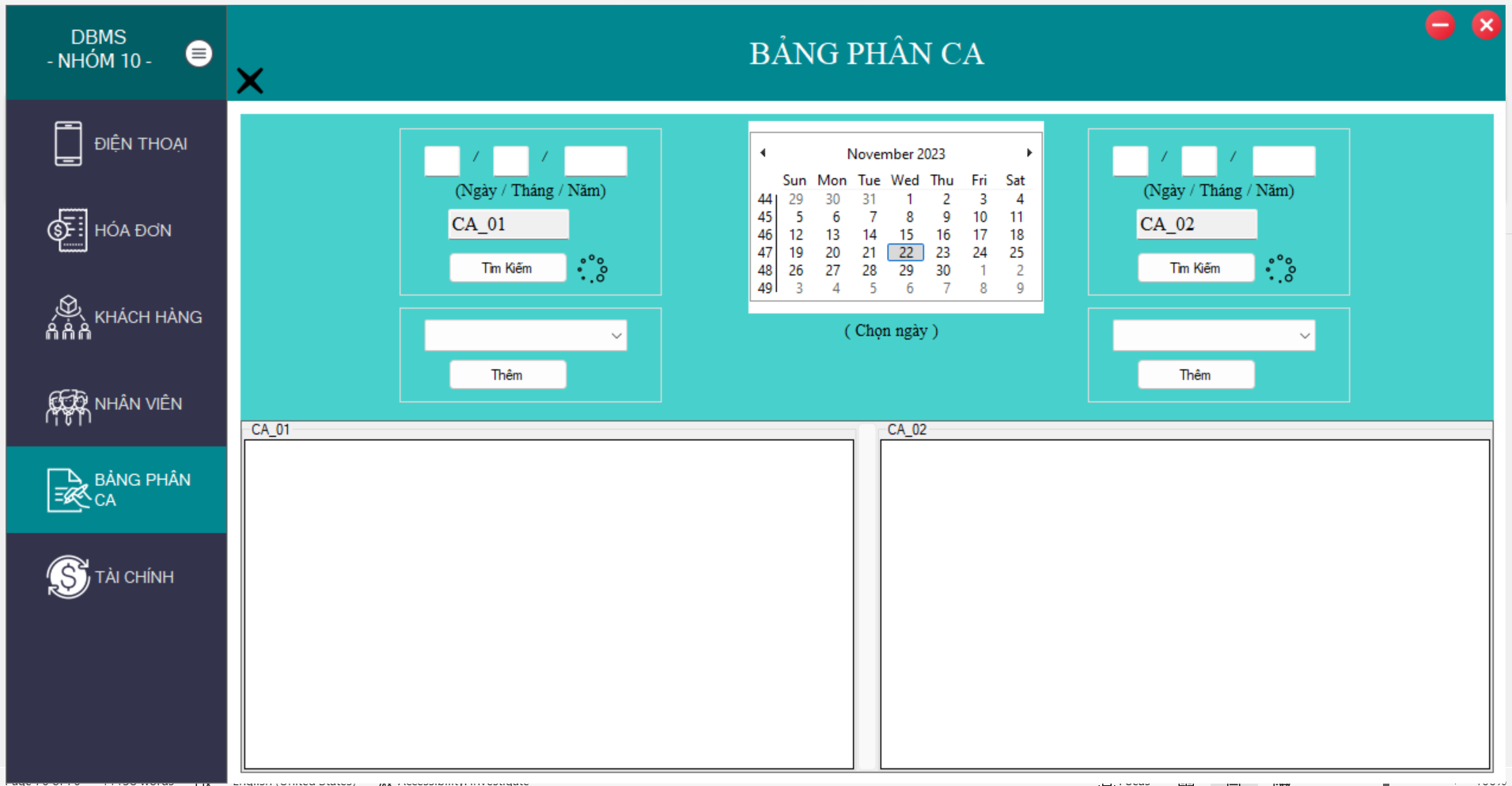


### Giao diện thêm hoặc chỉnh sửa thông tin Nhân Viên:



## Giao diện Bảng Phân Ca

* Trong giao diện này, Quản Lý có thể tìm kiếm ca làm việc theo ngày, tháng, năm và có thể thực hiện thêm ca làm cho nhân viên



## Giao diện Tài Chính:

* Trong giao diện này, Quản Lý có thể thực hiện tính lương cho nhân viên theo tháng, thực hiện tính tổng các hóa đơn theo tháng, thực hiện tính tổng đơn nhập theo tháng.

