**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ NHÂN VIÊN CƠ SỞ SẢN XUẤT CÀ PHÊ**

**MÃ MÔN HỌC: DBMS330284  
HỌC KỲ II – NĂM HỌC 2022-2023  
Thực hiện: Nhóm 03  
Giảng viên hướng dẫn: TS. Nguyễn Thành Sơn**

***Thành phố Hồ Chí Minh, Tháng 2 năm 2023***

**DANH SÁCH THÀNH VIÊN THAM GIA**

HỌC KỲ II NĂM HỌC 2022 – 2023

Nhóm: **03**

*Tên đề tài: Xây dựng chương trình quản lý nhân viên*

|  |  |  |
| --- | --- | --- |
| STT | HỌ VÀ TÊN SINH VIÊN | MÃ SỐ SINH VIÊN |
| 1 | Phan Xuân Dũng | 20110452 |
| 2 | Nguyễn Xuân Kha | 19110130 |
| 3 | Huỳnh Thanh Tuấn | 20110120 |
| 4 | Nguyễn Minh Quang | 20143481 |

**LỜI NÓI ĐẦU**

Phần mềm Excel từ lâu đã được các bộ phận Nhân sự và các cấp quản lý sử dụng để quản lý nhân viên trong nhiều doanh nghiệp. Ngày nay với sự phát triển của công nghệ 4.0 đặc biệt là sự bùng nổ và phát triển của công nghệ thông tin thì việc quản lý nhân viên như chấm công, đánh giá hiệu suất làm việc, … cũng cần có sự nâng cấp và cải tiến. Vì vậy, để có thể hiện thực hoá đươc mục tiêu trên, nhóm em đã lên ý tưởng, ứng dụng những kiến thức học được trong môn học Hệ quản trị cơ sở dữ liệu vào việc “**Xây dựng chương trình quản lý nhân viên**” trên Windows. Chương trình này giúp các công việc trong công tác quản lý nhân viên chuyển từ thủ công sang tự động. Từ việc quản lý rời rạc trên các file excel thành quản lý tập trung giúp có cái nhìn toàn diện hỗ trợ việc ra quyết định.

**ĐẶC TẢ CHƯƠNG TRÌNH QUẢN LÝ NHÂN VIÊN**

1. **Mô tả bài toán**

Một công ty sản xuất cà phê cần xây dựng một hệ thống để quản lý nhân viên. Một hệ thống quản lý nhân viên cần lưu trữ nhưng dữ liệu như sau: Quản lý nhân viên, Quản lý trình độ, Quản lý phòng ban, Quản lý Bảo hiểm, Quản lý chức vụ, Quản lý hợp đồng lao động, Quản lý quá trình công tác, Quản lý chấm công, Quản lý lương nhân viên.

Quản lý thông tin nhân viên hiệu quả là vấn đề mà nhiều công ty, doanh nghiệp chú trọng hàng đầu. Quá trình đó không hề đơn giản khi thao tác công việc thủ công, nhầm lẫn hay sai sót là điều thường thấy. Thông tin nhân viên bao gồm: Họ tên, Giới tính, ngày sinh, địa chỉ, số CMND, quê quán, nơi ở hiện tại, điện thoại…

Mỗi nhân viên được cấp tên đăng nhập và mật khẩu để đăng nhập vào hệ thống. Mỗi nhân viên sẽ có chức vụ và quyền hạn khác nhau nên thuộc các nhóm người dùng và có phân quyền khác nhau.

Các nhân viên sẽ có trình độ khác nhau nên cần phân chia trình độ, Quản lý trình độ: Mã trình độ và tên trình độ.

Vì vậy, hệ thống quản lý nhân sự được triển khai theo hệ thống các phòng ban trong tổ chức. Trong đó, Các Phòng ban bao gồm:

Ban giám đốc: Giám đốc, 1 PGĐ Hành chính tổ chức, 1 PGĐ Kinh doanh.

Phòng ban gồm 3 đầu mối: Phòng Hành chính, Phòng Tài chính Kế toán, Phòng Kế hoạch Kinh doanh.

Các trạm thu mua ở huyện Đăk Hà: 02 trạm và Phân xưởng sản xuất.

Phòng nhân sự nắm vai trò cốt lõi trong hệ thống vì phòng ban này sẽ chịu trách nhiệm điều hành, triển khai và báo cáo hiệu quả hoạt động của hệ thống quản lý nhân sự. Quản lý phòng ban, bộ phận gồm các thông tin: Mã và tên của các phòng ban trong công ty.

Quản lý chức vụ: mã chức vụ, tên chức vụ…

Quản lý hợp đồng lao động: số hợp đồng, ngày ký, loại hợp đồng, thời hạn hợp đồng, lương căn bản, hệ số lương…

Bảo hiểm xã hội (BHXH); Bảo hiểm y tế (BHYT), trợ cấp hàng tháng…nên công khai minh bạch, chính xác tuyệt đối giữa nhân viên với công ty. Quản lý Bảo hiểm: số bảo hiểm, nơi cấp, nơi đăng ký khám chữa bệnh…

Quản lý công việc cũng là tính năng cơ bản của một phần mềm quản lý nhân sự. Nó báo cáo, theo dõi, quản lý, đánh giá khách quan tiến độ dự án, năng lực của nhân viên cũng như nguồn lực hiện tại của công ty. Quản lý quá trình làm việc gồm: ngày vào làm, bộ phận, chức vụ, quá trình nghỉ phép, khen thưởng, kỷ luật…

Hoạt động quản lý chấm công là công việc thường thấy của bộ phận Quản lý nhân sự. Họ có nhiệm vụ cập nhật, đối chiếu, điều chỉnh thời gian biểu chấm công. Cập nhật thông tin ngày làm việc của nhân viên, theo dõi, kiểm tra ngày đi làm thực tế, ngày tăng ca. Công việc quan trọng này giúp nhân viên cập nhật đúng bảng lương, thưởng phù hợp với công sức và năng lực làm việc. Thông tin Quản lý chấm công gồm: ngày công, phân ca, nghỉ phép, thêm giờ, tăng ca…

Đa số tất cả các phần mềm quản lý nhân viên sẽ tích hợp tính năng làm bảng lương, thanh toán lương thưởng, thống kế, báo cáo theo chế độ lao động. Quản lý lương của nhân viên gồm các thông tin: lương, thưởng, phụ cấp, quá trình nâng lương, ứng lương….

1. **Nghiệp vụ bài toán**

Các hoạt động quản lý nhân viên của bài toán được tiến hành như sau:

* 1. **Hoạt động quản lý hồ sơ nhân viên**

Bộ phận quản lý nhân sự chịu trách nhiệm thực hiện nhiệm vụ này bằng cách thực hiện việc lập và điều chỉnh hồ sơ nhân viên.

Khi công ty có nhu cầu tuyển dụng nhân viên mới và Ban Giám đốc đã đưa ra quyết định, bộ phận quản lý nhân sự sẽ tiến hành lập danh sách và cập nhật thông tin hồ sơ của nhân viên mới này. Ngoài ra, trong quá trình hoạt động, bộ phận quản lý nhân sự cũng thường xuyên cập nhật thông tin của nhân viên hiện tại để phát hiện và sửa đổi các sai sót nếu có. Các thay đổi này được cập nhật dựa trên thông tin mà nhân viên cung cấp.

Bộ phận quản lý nhân sự cũng có nhiệm vụ theo dõi các thông tin về mức lương của nhân viên. Nếu Giám đốc quyết định thay đổi mức lương của một nhân viên, bộ phận quản lý nhân sự sẽ tiến hành điều chỉnh thông tin trong hồ sơ của nhân viên đó. Các thay đổi về mức lương cũng được ghi vào hồ sơ để phục vụ cho việc tính lương của bộ phận kế toán.

Nếu một nhân viên rời khỏi công ty, sau khi Ban Giám đốc đã đưa ra quyết định, bộ phận quản lý nhân sự sẽ xóa thông tin về nhân viên đó khỏi hệ thống hồ sơ.

* 1. **Hoạt động quản lý chấm công**

Phòng nhân sự có trách nhiệm thực hiện hoạt động quản lý chấm công gồm cập nhật và điều chỉnh chấm công:

+ Mỗi ngày, phòng nhân sự cập nhật thông tin ngày công làm việc của nhân viên trong công ty để ghi vào bảng theo dõi chấm công.

+ Nếu công ty có nhu cầu làm thêm giờ, ban giám đốc sẽ ra quyết định cho phòng nhân sự có thể điều động nhân viên. Việc này phải dựa trên cơ sở cập nhật thông tin về ngày làm thực tế của nhân viên trong tháng để bố trí, sắp xếp sao cho phù hợp.

+ Khi nhân viên đã làm thêm giờ, phòng nhân sự sẽ cập nhật tiếp ngày công làm thêm vào bảng chấm công, thông tin vẫn lấy từ phía nhân viên.

+ Cuối tháng phòng nhân sự sẽ dựa trên bảng theo dõi chấm công để lập ra báo cáo chấm công trình lên Ban Giám Đốc kí duyệt.

* 1. **Quản lý lương, thưởng**

Phòng kế toán sau khi nhận được báo cáo chấm công đã được Giám đốc phê duyệt của phòng nhân sự sẽ thực hiện cập nhật các thông tin mức lương căn bản và các khoản phụ cấp, BHYT, BHXH, … để là cơ sở tính lương. Dựa vào các thông tin được cập nhật phòng kế toán thực hiện tính lương cho từng nhân viên với đầy đủ các khoản thưởng, phụ cấp, khoản phí phải nộp. Sau khi hoàn tất bảng lương thực tế nhận được của mỗi nhân viên sẽ được trình lên Giám đốc để kí duyệt. Sau khi bảng lương được duyệt nhân viên sẽ nhận được lương. Ngoài ra Giám đốc sẽ nhận được báo cáo về lương, thưởng, BHYT, BHXH phải đóng hàng tháng.

1. **Mô tả các chức năng của bài toán**

**+** Thêm, xóa, cập nhật thông tin nhân viên

+ Thêm, xóa, cập nhật phòng ban khi công ty phát sinh phòng ban mới.

+ Quản lý danh mục cho công ty như chức danh, hợp đồng.

+ Quản lý chế độ cho nhân viên giúp cho nhân viên được hưởng các chế độ ưu đãi cũng công ty

+ Quản lý chấm công

+ Lập bảng theo dõi chấm công và báo cáo chấm công trong tháng

+ Quản lý lương cho nhân viên

+ Lập phiếu lương cho từng nhân viên trong công ty

+ Quản lý luân chuyển nhân viên

+ Tìm kiếm, tra cứu thông tin nhân viên trong công ty

+ Thống kê báo cáo danh sách nhân viên

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

1. **Công thức tính lương**

- Tổng số ngày theo lịch dương trong năm là 365 ngày trong đó số ngày làm việc trong năm là 281 ngày, Số ngày nghỉ có hưởng lương là 30 ngày.

- Tiền lương, các khoản phụ cấp tính theo bậc công việc để xác định đơn giá.

+ Mức lương tối thiểu vùng được áp dụng là 3.250.000 đồng/tháng áp dụng theo nghị định 90/2019/NĐ-CP của Chính phủ ký ngày 15 tháng 11 năm 2019.

+ Phụ cấp khu vực: 0,4

- Đơn giá tiền lương/công áp dụng theo nghị định số 205/2004/ NĐ-CP ngày 14 tháng 12 năm 2004 về việc quy định hệ thống thang lương, bảng lương và chế độ phụ cấp lương trong các công ty nhà nước; Quyết định số 142/QĐ-TCT-TCCB ngày 13 tháng 4 năm 2016 của Tổng Công ty Cà phê Việt Nam về việc ban hành quy định tạm thời hệ thống bảng lương và chế độ phụ cấp lương Công ty mẹ - Tổng Công ty Cà phê Việt Nam.

Lương tháng theo bậc:

+ Bậc 1: 1,07 x 3.250.000

+ Bậc 2: 1,13 x 3.250.000

+ Bậc 3: 1,19 x 3.250.000

+ Bậc 4: 1,42 x 3.250.000

**Lương ngày** = Lương tháng/26

**Lương phụ cấp =** 0,4 x 1.390.000

Làm thêm vào ngày chủ nhật(Tăng ca): Lương ngày/8 \* 200% \* số giờ làm thêm

**Lương nhân được** = (Lương ngày\*Số ngày đi làm) + Lương tăng ca + Lương phụ cấp - Lương ứng trước

**Ví dụ:** Ông A ký hợp đồng với công ty B với mức lương cơ bản là 3.250.000/tháng, chưa được tăng lương(bậc 1) và phụ cấp vùng 556 nghìn. Ông A không làm tăng ca, nghỉ 6 ngày không phép trong tổng số 26 ngày làm của tháng và đã được ứng trước 500 nghìn đồng.

**Tóm tắt:**

Lương ngày = (3.250.000\*1.07)/26 = 133.75 nghìn đồng

Lương tăng ca = 0

Lương phụ cấp = 556 nghìn

Lương ứng trước = 500 nghìn

⇒ Lương ông A nhận được = 133.75\*20 + 0 + 556 - 500 = 2.731.000 đồng

1. **Bảo hiểm**

Mức đóng BHXH 26% trong đó: người lao động đóng **8%,** đơn vị đóng 18%. 18% đơn vị đóng bao gồm: 3% vào quỹ ốm đau và thai sản; 1% vào quỹ tai nạn lao động, bệnh nghề nghiệp; 14% vào quỹ hưu trí và tử tuất.

Mức đóng bảo hiểm y tế 4,5% trong đó: người lao động đóng **1,5%,** đơn vị đóng 3%.

Mức đóng bảo hiểm thất nghiệp 2% trong đó người lao động đóng **1%,** đơn vị đóng 1%

Theo đó, tổng số tiền mỗi người lao động phải đóng khi tham gia BHXH bắt buộc là 10,5%. Trách nhiệm của đơn vị là 22% và không phát sinh thêm bất cứ một khoản chi phí nào.

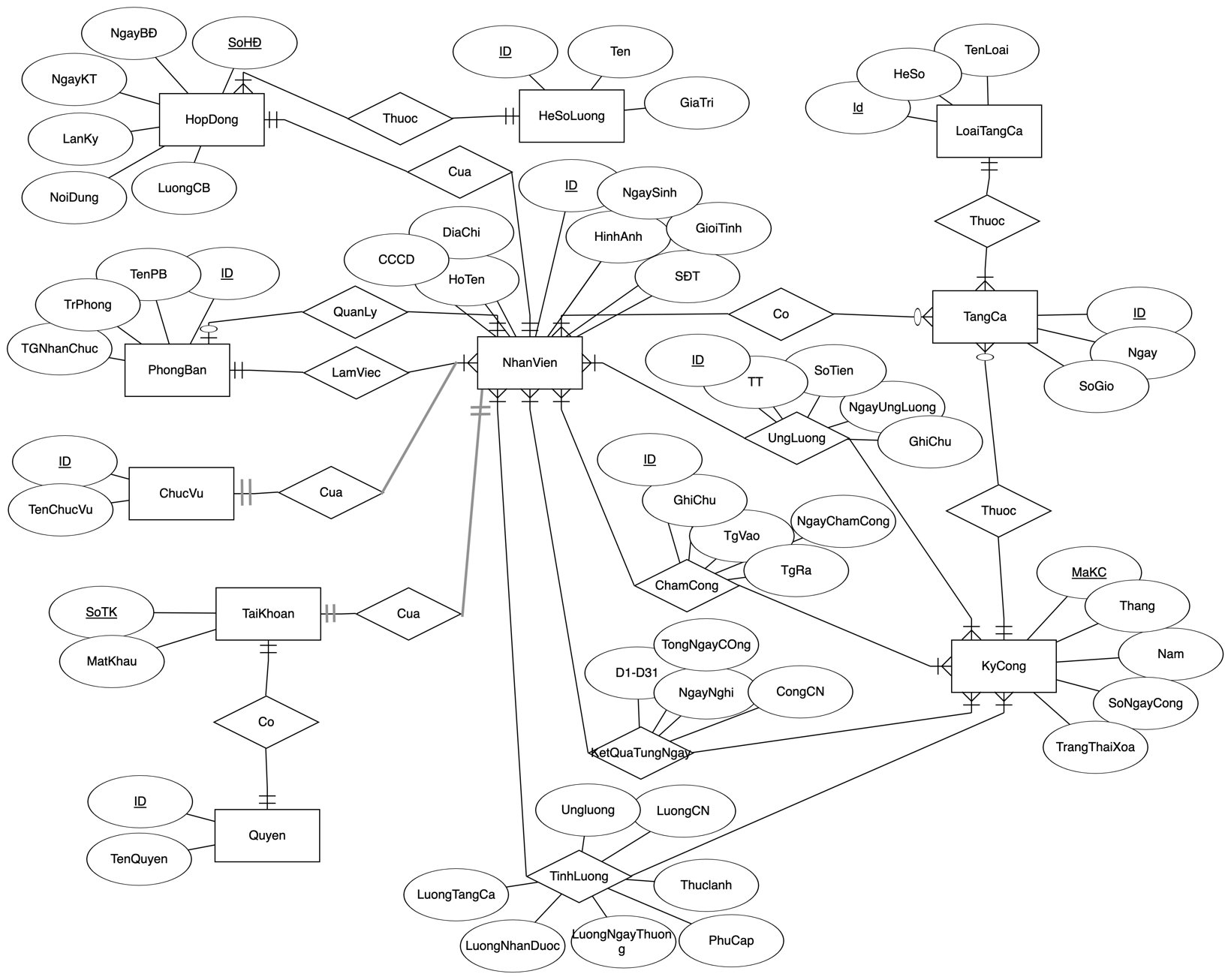
Vậy nếu theo công thức trên ta có thể tính được tiền lương tháng thực tế của nhân viên là:

**Lương thực tế nhận được** = Lương nhận được \* 89,5%

Theo như ví dụ ở phần trước thực tế ông A sẽ nhận được 2.444.245 đã bao gồm bảo hiểm.

**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

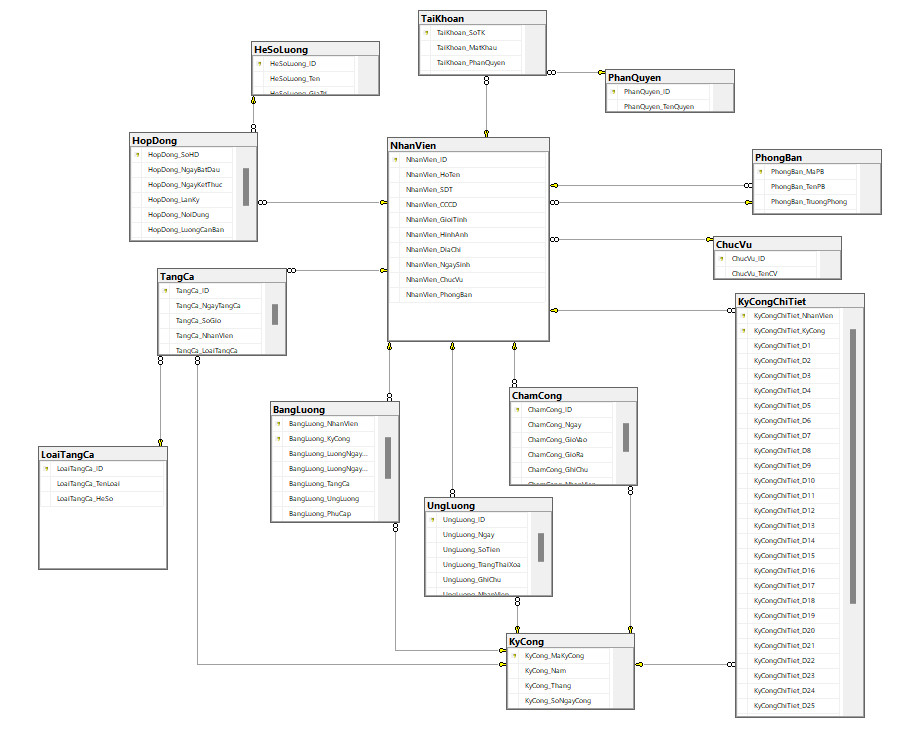
1. **Thiết kế cơ sở dữ liệu mức quan niêm**



1. **Thiết kế cơ sở dữ liệu mức logic**

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

* NhanVien(ID, HoTen, SĐT, CCCD, GioiTinh, HinhAnh, DiaChi, NgaySinh, ChucVu, PhongBan)
* UngLuong(ID, Ngay, SoTien, TrangThaiXoa, GhiChu, NhanVien, KyCong)
* KyCong(ID, Ngay, SoTien, TrangThaiXoa, GhiChu, NhanVien, KyCong)
* HopDong(SoHD, NgayBĐ, NgayKT, Lanky, LuongCanBan, HeSoLuong, NhanVien)
* HeSoLuong(ID, Ten, GiaTri)
* KyCong(MaKyCong, Nam, Thang, SoNgayCong, TrangThaiXoa)
* KyCongChiTiet(NhanVien, KyCong, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23, D24, D25, D26, D27, D28, D29, D30, D31, NgayNghi, CongChuNhat, TongNgayCong)
* ChamCong(ID, Ngay, GioVao, GioRa, GhiChu, NhanVien, KyCong)
* TinhLuong(NhanVien, KyCong, LuongNgayThuong, LuongNgayCN, TangCa, UngLuong, PhuCap, LuongNhanDuoc, ThucLanh)
* LoaiTangCa(ID, NgayTangCa, SoGio, NhanVien, LoaiTangCa, KyCong)
* PhongBan(ID, TenPB, TruongPhong ,TGNhanChuc)
* ChucVu(ID, TenChucVu)
* Quyen(ID, TenQuyen)
* TaiKhoan(TenTK, MatKhau)



**3. Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **TABLE** | **RÀNG BUỘC** |
| **1** | NhanVien | Ràng buộc khóa chính NhanVien\_ID Ràng buộc khóa ngoại NhanVien\_ChucVu tham chiếu đến bảng ChucVu. Ràng buộc khóa ngoại NhanVien\_PhongBan tham chiếu đến bảng PhongBan. |
| **2** | UngLuong | Rằng buộc khóa chính UngLuong\_ID  Ràng buộc khóa ngoại UngLuong\_NhanVien tham chiếu đến bảng NhanVien. Ràng buộc khóa ngoại UngLuong\_KyCong tham chiếu đến bảng KyCong. |
| **3** | HopDong | Rằng buộc khóa chính HopDong\_SoHD  Ràng buộc khóa ngoại HopDong\_HeSoLuong tham chiếu đến bảng HeSoLuong. Ràng buộc khóa ngoại HopDong\_NhanVien tham chiếu đến bảng NhanVien. |
| **4** | Hệ số lương | Rằng buộc khóa chính HeSoLuong\_ID |
| **5** | KyCong | Rằng buộc khóa chính KyCong\_MaKyCong |
| **6** | KyCongChiTiet | Rằng buộc khóa chính KyCongChiTiet\_NhanVien, KyCongChiTiet\_KyCong |
| **7** | ChamCong | Ràng buộc khóa ngoại ChamCong\_NhanVien tham chiếu đến bảng NhanVien. Ràng buộc khóa ngoại ChamCong\_KyCong tham chiếu đến bảng KyCong. |
| **8** | BangLuong | Ràng buộc khóa chính BangLuong\_KyCong, BangLuong\_NhanVien, Ràng buộc khóa ngoại BangLuong\_KyCong tham chiếu đến bảng KyCong, BangLuong\_NhanVien tham chiếu đến bảng NhanVien |
| **9** | LoaiTangCa | Ràng buộc khóa chính LoaiTangCa\_ID |
| **10** | TangCa | Ràng buộc khóa chính TangCa\_ID, Ràng buộc khóa ngoại TangCa\_LoaiTangCa Tham chiếu đến bảng LoaiTangCa và TangCa\_KyCong tham chiếu đến bảng KyCong |
| **11** | ChucVu | Ràng buộc khóa chính ChucVu\_ID |
| **12** | PhongBan | Ràng buộc khóa chính PhongBan\_MaPB và Ràng buộc khóa ngoại PhongBan\_TruongPhong tham chiếu đến bảng NhanVien |
| **13** | TaiKhoan | Ràng buộc khóa chính TaiKhoan\_SoTK và ràng buộc khóa ngoại TaiKhoan\_PhanQuyen tham chiếu đến bảng PhanQuyen, TaiKhoan\_NhanVien tham chiếu đến bảng NhanVien |
| **14** | PhanQuyen | Ràng buộc khóa chính PhanQuyen\_ID |

**4. Cài đặt CSDL và các ràng buộc**

**4.1. Tạo bảng và các ràng buộc đơn giản.**

**Bảng nhân viên**

--Bảng Nhân viên

CREATE TABLE NhanVien

(

NhanVien\_ID INT IDENTITY(1,1) PRIMARY KEY,

NhanVien\_HoTen NVARCHAR(50),

NhanVien\_SDT CHAR(10),

NhanVien\_CCCD CHAR(12),

NhanVien\_GioiTinh NVARCHAR(3),

NhanVien\_HinhAnh IMAGE,

NhanVien\_DiaChi NVARCHAR(50),

NhanVien\_NgaySinh DATE,

NhanVien\_ChucVu INT,

NhanVien\_PhongBan INT

);

GO

**Bảng ứng lương**

--Bảng Ứng lương

CREATE TABLE UngLuong

(

UngLuong\_ID INT PRIMARY KEY,

UngLuong\_Ngay INT,

UngLuong\_SoTien FLOAT,

UngLuong\_TrangThaiXoa BIT,

UngLuong\_GhiChu NVARCHAR(50),

UngLuong\_NhanVien INT,

UngLuong\_KyCong INT

);

GO

**Bảng hợp đồng**

-- Bảng Hợp đồng

CREATE TABLE HopDong

(

HopDong\_SoHD INT IDENTITY(1,1) PRIMARY KEY,

HopDong\_NgayBatDau DATE,

HopDong\_NgayKetThuc DATE,

HopDong\_LanKy INT,

HopDong\_NoiDung NVARCHAR(50),

HopDong\_LuongCanBan FLOAT,

HopDong\_HeSoLuong INT,

HopDong\_NhanVien INT

);

GO

**Bảng hệ số lương**

-- Bảng Hệ số lương

CREATE TABLE HeSoLuong

(

HeSoLuong\_ID INT PRIMARY KEY,

HeSoLuong\_Ten NVARCHAR(10) NOT NULL,

HeSoLuong\_GiaTri FLOAT

);

GO

**Bảng kỳ công**

CREATE TABLE KyCong

(

KyCong\_MaKyCong INT IDENTITY(1, 1) PRIMARY KEY,

KyCong\_Nam INT,

KyCong\_Thang INT,

KyCong\_SoNgayCong INT,

KyCong\_TrangThaiXoa BIT

);

GO

**Bảng kỳ công chi tiết**

CREATE TABLE KyCongChiTiet

(

KyCongChiTiet\_NhanVien INT,

KyCongChiTiet\_KyCong INT,

KyCongChiTiet\_D1 VARCHAR(10),

KyCongChiTiet\_D2 VARCHAR(10),

KyCongChiTiet\_D3 VARCHAR(10),

KyCongChiTiet\_D4 VARCHAR(10),

KyCongChiTiet\_D5 VARCHAR(10),

KyCongChiTiet\_D6 VARCHAR(10),

KyCongChiTiet\_D7 VARCHAR(10),

KyCongChiTiet\_D8 VARCHAR(10),

KyCongChiTiet\_D9 VARCHAR(10),

KyCongChiTiet\_D10 VARCHAR(10),

KyCongChiTiet\_D11 VARCHAR(10),

KyCongChiTiet\_D12 VARCHAR(10),

KyCongChiTiet\_D13 VARCHAR(10),

KyCongChiTiet\_D14 VARCHAR(10),

KyCongChiTiet\_D15 VARCHAR(10),

KyCongChiTiet\_D16 VARCHAR(10),

KyCongChiTiet\_D17 VARCHAR(10),

KyCongChiTiet\_D18 VARCHAR(10),

KyCongChiTiet\_D19 VARCHAR(10),

KyCongChiTiet\_D20 VARCHAR(10),

KyCongChiTiet\_D21 VARCHAR(10),

KyCongChiTiet\_D22 VARCHAR(10),

KyCongChiTiet\_D23 VARCHAR(10),

KyCongChiTiet\_D24 VARCHAR(10),

KyCongChiTiet\_D25 VARCHAR(10),

KyCongChiTiet\_D26 VARCHAR(10),

KyCongChiTiet\_D27 VARCHAR(10),

KyCongChiTiet\_D28 VARCHAR(10),

KyCongChiTiet\_D29 VARCHAR(10),

KyCongChiTiet\_D30 VARCHAR(10),

KyCongChiTiet\_D31 VARCHAR(10),

KyCongChiTiet\_NgayNghi INT,

KyCongChiTiet\_CongChuNhat INT,

KyCongChiTiet\_TongNgayCong INT,

CONSTRAINT PK\_KyCongChiTiet PRIMARY KEY(KyCongChiTiet\_NhanVien, KyCongChiTiet\_KyCong)

);

GO

**Bảng chấm công**

-- Bảng chấm công

CREATE TABLE ChamCong

(

ChamCong\_ID INT IDENTITY(1,1) PRIMARY KEY,

ChamCong\_Ngay INT,

ChamCong\_GioVao VARCHAR(50),

ChamCong\_GioRa VARCHAR(50),

ChamCong\_GhiChu NVARCHAR(255),

ChamCong\_NhanVien INT,

ChamCong\_KyCong INT

);

GO

**Bảng lương**

CREATE TABLE BangLuong

(

BangLuong\_NhanVien INT,

BangLuong\_KyCong INT,

BangLuong\_LuongNgayThuong FLOAT,

BangLuong\_LuongNgayCN FLOAT,

BangLuong\_TangCa FLOAT,

BangLuong\_UngLuong FLOAT,

BangLuong\_PhuCap FLOAT,

BangLuong\_LuongNhanDuoc FLOAT,

BangLuong\_ThucLanh FLOAT,

CONSTRAINT PK\_BangLuong PRIMARY KEY(BangLuong\_NhanVien, BangLuong\_KyCong)

);

GO

**Bảng phân loại tăng ca**

-- Bảng loại tăng ca

CREATE TABLE LoaiTangCa

(

LoaiTangCa\_ID INT PRIMARY KEY,

LoaiTangCa\_TenLoai NVARCHAR(20),

LoaiTangCa\_HeSo FLOAT

);

GO

**Bảng tăng ca**

-- Bảng tăng ca

CREATE TABLE TangCa

(

TangCa\_ID INT IDENTITY(1,1) PRIMARY KEY,

TangCa\_NgayTangCa INT,

TangCa\_SoGio FLOAT,

TangCa\_NhanVien INT,

TangCa\_LoaiTangCa INT,

TangCa\_KyCong INT

);

GO

**Bảng chức vụ**

-- Bảng chức vụ

CREATE TABLE ChucVu

(

ChucVu\_ID INT PRIMARY KEY,

ChucVu\_TenCV NVARCHAR(50)

);

GO

**Bảng phòng ban**

-- Bảng phòng ban

CREATE TABLE PhongBan

(

PhongBan\_MaPB INT PRIMARY KEY,

PhongBan\_TenPB NVARCHAR(50),

PhongBan\_TruongPhong INT,

PhongBan\_TG\_NhanChuc DATE,

);

GO

**Bảng tài khoản**

-- Bảng tài khoản

CREATE TABLE TaiKhoan

(

TaiKhoan\_SoTK VARCHAR(50) PRIMARY KEY,

TaiKhoan\_MatKhau VARCHAR(32),

TaiKhoan\_PhanQuyen INT,

TaiKhoan\_NhanVien INT

);

GO

**Bảng phân quyền**

-- Bảng Phân quyền

CREATE TABLE PhanQuyen

(

PhanQuyen\_ID INT PRIMARY KEY,

PhanQuyen\_TenQuyen NVARCHAR(20)

);

GO

1. **Trigger kiểm tra các ràng buộc**

**Trigger tăng ID tự động trong bảng Phòng Ban**

|  |
| --- |
| CREATE TRIGGER TGR\_PhongBan  On PhongBan  NSTEAD OF INSERT  AS  BEGIN  DECLARE @maxid int = 0  SELECT @maxid = MAX(PhongBan\_MaPB) FROM PhongBan  IF @maxid IS NULL SET @maxid = 1  ELSE SET @maxid = @maxid + 1  INSERT INTO PhongBan (PhongBan\_MaPB)  VALUES (@maxid)  UPDATE PhongBan  SET PhongBan\_TenPB = i.PhongBan\_TenPB, PhongBan\_TG\_NhanChuc = i.PhongBan\_TG\_NhanChuc, PhongBan\_TruongPhong = i.PhongBan\_TruongPhong  FROM inserted i  WHERE PhongBan.PhongBan\_MaPB = @maxid  PRINT N'Đã thêm phòng ban có mã là: '+ CAST(@maxid AS VARCHAR)  END; |

**Trigger tăng ID tự động trong bảng Phân Quyền**

|  |
| --- |
| CREATE TRIGGER TGR\_PhanQuyen  On PhanQuyen  INSTEAD OF INSERT  AS  BEGIN  DECLARE @maxid int = 0  SELECT @maxid = MAX(PhanQuyen\_ID) FROM PhanQuyen  IF @maxid IS NULL SET @maxid = 1  ELSE SET @maxid = @maxid + 1  INSERT INTO PhanQuyen (PhanQuyen\_ID)  VALUES(@maxid)  UPDATE PhanQuyen  SET PhanQuyen\_TenQuyen = i.PhanQuyen\_TenQuyen  FROM inserted i  WHERE PhanQuyen.PhanQuyen\_ID = @maxid  PRINT N'Đã thêm phân quyền có mã là: '+ CAST(@maxid AS VARCHAR)  END; |

**Trigger kiểm tra số tài khoản đã có tồn tại trên hệ thống**

|  |
| --- |
| CREATE TRIGGER TGR\_TaiKhoan  On TaiKhoan  INSTEAD OF INSERT  AS  BEGIN  DECLARE @checkid int  SELECT @checkid = i.TaiKhoan\_SoTK FROM inserted i  IF NOT EXISTS(SELECT \* FROM TaiKhoan WHERE TaiKhoan\_SoTK = @checkid)  INSERT INTO TaiKhoan (TaiKhoan\_SoTK, TaiKhoan\_MatKhau, TaiKhoan\_NhanVien, TaiKhoan\_PhanQuyen)  SELECT i.TaiKhoan\_SoTK, (CONVERT(VARCHAR(32), HashBytes('MD5', i.TaiKhoan\_MatKhau), 2)), i.TaiKhoan\_NhanVien, i.TaiKhoan\_PhanQuyen FROM inserted i  ELSE  PRINT N'Số tài khoản: '+ CAST(@checkid AS VARCHAR)+N' đã tồn tại.'  END; |

**Trigger không cho thêm dữ liệu vào bảng nhân viên tăng ca khi dữ liệu nhập vào không thỏa:**

|  |
| --- |
| CREATE TRIGGER ThemNVTangCa  ON dbo.TangCa  FOR INSERT  AS  BEGIN  DECLARE @day INT =0;  SELECT @day = TangCa\_NgayTangCa FROM dbo.TangCa  IF(@day <=0 or @day >31)  ROLLBACK TRAN  END |

**Trigger kiểm tra ngày nghỉ và tổng ngày công có hợp lệ.**

|  |
| --- |
| CREATE TRIGGER TR\_KyCongChiTiet ON KyCongChiTiet  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra các giá trị mới của KyCongChiTiet  IF EXISTS (  SELECT \*  FROM inserted i  INNER JOIN KyCong kc ON i.KyCongChiTiet\_KyCong = kc.KyCong\_MaKyCong  WHERE i.KyCongChiTiet\_NgayNghi >= 0  AND i.KyCongChiTiet\_TongNgayCong >= 0  AND i.KyCongChiTiet\_TongNgayCong < kc.KyCong\_SoNgayCong  )  BEGIN  -- Nếu có giá trị không hợp lệ, rollback transaction và hiển thị thông báo lỗi  RAISERROR('Giá trị không hợp lệ', 16, 1)  ROLLBACK TRANSACTION  RETURN  END  END |

**Trigger không cho thêm, cập nhật dữ liệu vào Kỳ Công khi dữ liệu nhập vào không thỏa:**

|  |
| --- |
| CREATE TRIGGER trg\_Insert\_KyCong  ON KyCong  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (SELECT 1 FROM inserted WHERE inserted.KyCong\_Thang < 1 OR inserted.KyCong\_Thang > 12)  BEGIN  RAISERROR('Giá trị của KyCong\_Thang phải nằm trong khoảng từ 1 đến 12', 16, 1)  ROLLBACK TRANSACTION  END  END |

**Trigger không cho thêm, cập nhật dữ liệu vào Chấm Công khi dữ liệu nhập vào không thỏa:**

|  |
| --- |
| CREATE TRIGGER [TenTrigger]  ON [dbo].[ChamCong]  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT \*  FROM inserted  WHERE ChamCong\_Ngay < 1 OR ChamCong\_Ngay > 31  )  BEGIN  RAISERROR('Ngày chấm công không hợp lệ', 16, 1)  ROLLBACK TRANSACTION  END  END |

**Trigger không cho thêm, cập nhật dữ liệu vào Kỳ Công Chi Tiết khi dữ liệu nhập vào không thỏa:**

|  |
| --- |
| CREATE TRIGGER Trigger\_KyCongChiTiet\_KiemTraGiaTri  ON KyCongChiTiet  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT \*  FROM inserted  WHERE NOT (  inserted.KyCongChiTiet\_D1 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D2 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D3 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D4 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D5 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D6 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D7 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D8 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D9 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D10 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D11 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D12 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D13 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D14 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D15 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D16 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D17 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D18 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D19 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D20 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D21 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D22 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D23 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D24 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D25 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D26 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D27 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D28 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D29 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D30 IN ('X', 'V', 'CN') AND  inserted.KyCongChiTiet\_D31 IN ('X', 'V', 'CN')  )  )  BEGIN  -- Nếu có giá trị không hợp lệ, rollback transaction và hiển thị thông báo lỗi  RAISERROR('Giá trị không hợp lệ', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END;  END; |

1. **Các View**

**a.View danh sách ứng lương của các nhân viên đã ứng**

|  |
| --- |
| CREATE VIEW DS\_UngLuong AS  SELECT UngLuong\_ID, UngLuong\_Ngay, UngLuong\_NhanVien, UngLuong\_SoTien, UngLuong\_GhiChu  FROM UngLuong  WHERE UngLuong\_TrangThaiXoa = 0 |

**b. View danh sách phòng ban**

|  |
| --- |
| CREATE VIEW DS\_Phongban AS  SELECT PhongBan\_MaPB, PhongBan\_TenPB  FROM PhongBan |

**c. View tên quyền trong bảng phân quyền**

|  |
| --- |
| CREATE VIEW DS\_Quyen AS  SELECT PhanQuyen\_TenQuyen  FROM PhanQuyen |

**d. View danh sách tài khoản**

|  |
| --- |
| CREATE VIEW DS\_TaiKhoan AS  SELECT TaiKhoan\_SoTK, TaiKhoan\_MatKhau, TaiKhoan\_NhanVien  FROM TaiKhoan |

**e. View các chức vụ hiện có**

|  |
| --- |
| CREATE VIEW v\_ChucVu AS  SELECT \* FROM dbo.ChucVu; |

f. **View thông tin về các nhân viên có tăng ca**

|  |
| --- |
| CREATE VIEW v\_NhanVienCoTangCa AS  SELECT \* FROM dbo.TangCa  WHERE TangCa\_NgayTangCa IS NOT NULL |

g. **View các loại tăng ca hiện có**

|  |
| --- |
| CREATE VIEW v\_LoaiTangCa AS  SELECT \* FROM dbo.LoaiTangCa; |

**h. View danh sách kì công chưa xóa**

|  |
| --- |
| CREATE VIEW DS\_KyCong AS  SELECT KyCong\_MaKyCong , KyCong\_Nam ,KyCong\_Thang ,KyCong\_SoNgayCong  FROM KyCong  WHERE KyCong\_TrangThaiXoa = 0 |

**i.View danh sách chấm công**

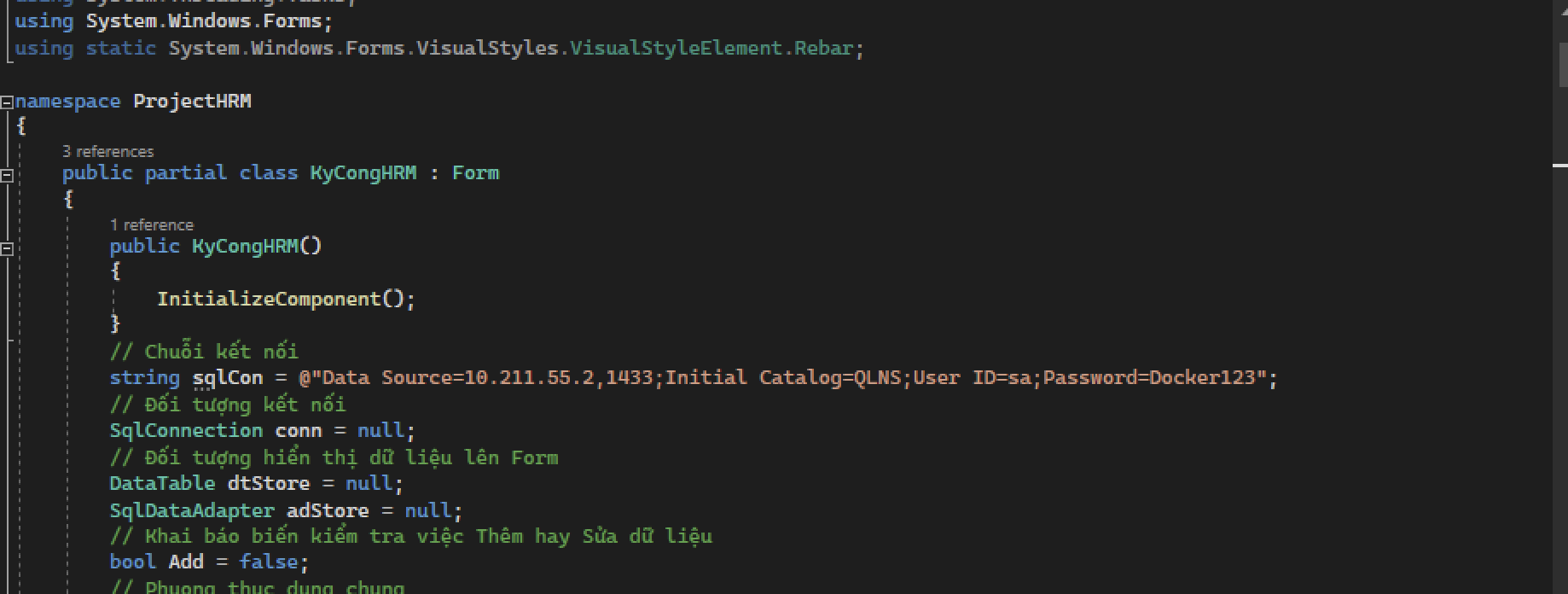
|  |
| --- |
| CREATE VIEW View\_ChamCong AS  SELECT ChamCong\_ID, ChamCong\_GioVao, ChamCong\_GioRa, ChamCong\_Ngay, ChamCong\_GhiChu ,ChamCong\_NhanVien,ChamCong\_KyCong  FROM ChamCong |

**j.View danh sách Kỳ Công Chi Tiết**

|  |
| --- |
| CREATE VIEW View\_KyCongChiTiet AS  SELECT KyCongChiTiet\_NhanVien, KyCongChiTiet\_KyCong, KyCongChiTiet\_D1, KyCongChiTiet\_D2, KyCongChiTiet\_D3, KyCongChiTiet\_D4,  KyCongChiTiet\_D5, KyCongChiTiet\_D6, KyCongChiTiet\_D7 ,KyCongChiTiet\_D8 ,KyCongChiTiet\_D9,KyCongChiTiet\_D10,KyCongChiTiet\_D11 ,KyCongChiTiet\_D12,  KyCongChiTiet\_D13,KyCongChiTiet\_D14,KyCongChiTiet\_D15 ,KyCongChiTiet\_D16,KyCongChiTiet\_D17,KyCongChiTiet\_D18,KyCongChiTiet\_D19,KyCongChiTiet\_D20,  KyCongChiTiet\_D21, KyCongChiTiet\_D22,KyCongChiTiet\_D23, KyCongChiTiet\_D24, KyCongChiTiet\_D25,KyCongChiTiet\_D26,KyCongChiTiet\_D27,KyCongChiTiet\_D28,  KyCongChiTiet\_D29,KyCongChiTiet\_D30,KyCongChiTiet\_D31,KyCongChiTiet\_NgayNghi, KyCongChiTiet\_CongChuNhat, KyCongChiTiet\_TongNgayCong  FROM KyCongChiTiet |

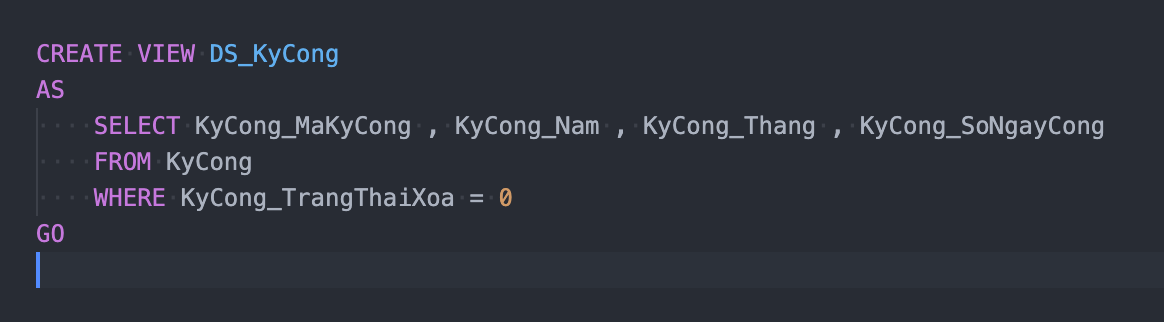
**CHƯƠNG 3: THIẾT KẾ CHỨC NĂNG**

1. **Kết nối cơ sở dữ liệu**



1. **Xem thông tin danh mục**
   1. **Danh mục kỳ công**

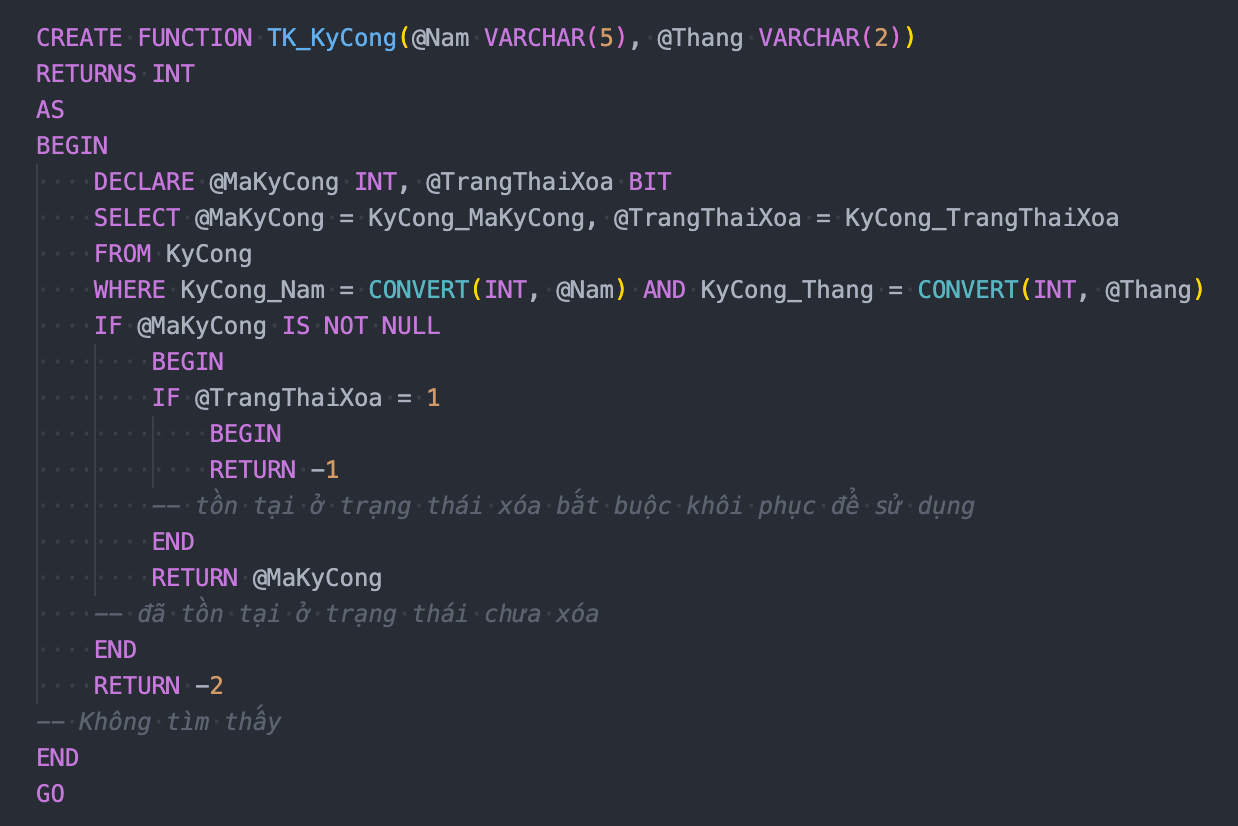
* VIEW: thực hiện lấy danh sách kỳ công có trạng thái xóa là false (0)



* 1. **Danh mục A**

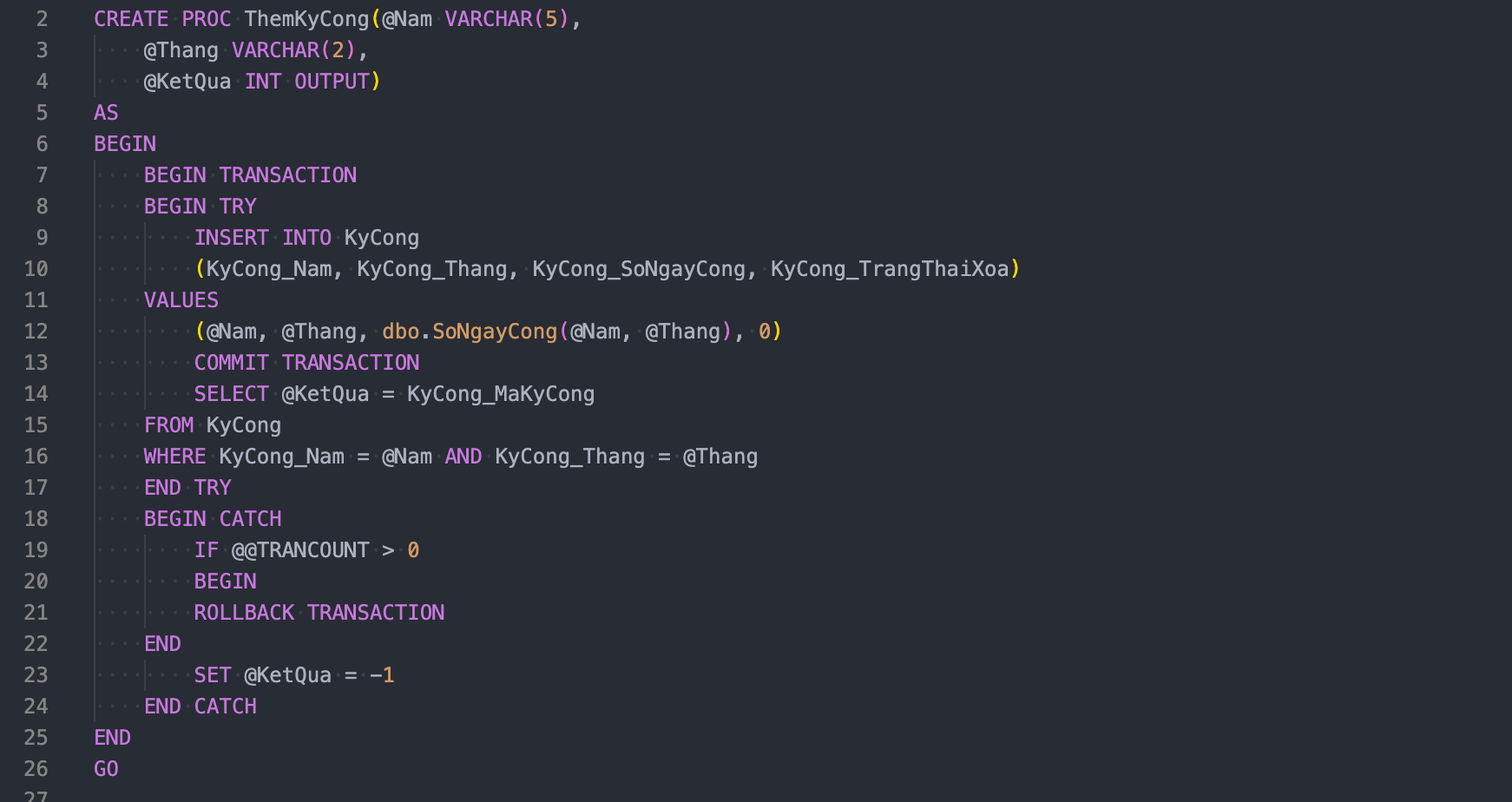
1. **Quản lý kỳ công**
   1. **Kiểm tra kỳ công**

Hàm kiểm tra kỳ công có tồn tại không dựa vào giá trị năm và tháng. Nếu tồn tại ở trạng thái xóa là false thì trả về mã của kỳ công đó. Ngược lại nếu là true thực hiện trả về -1. Nếu không tìm thấy trả về -2



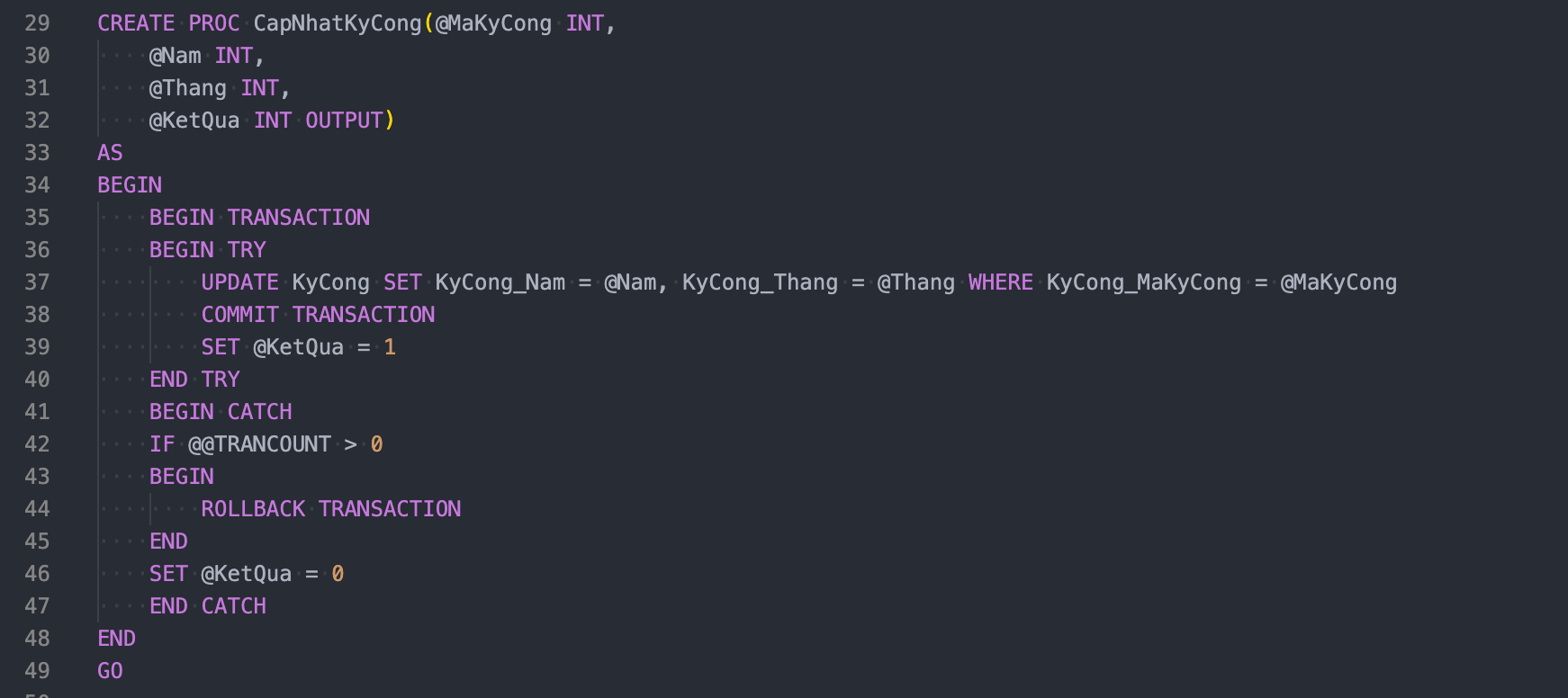
* 1. **Thêm kỳ công**

Thủ tục thực hiện thêm kỳ công dựa vào giá trị tháng năm truyền vào.



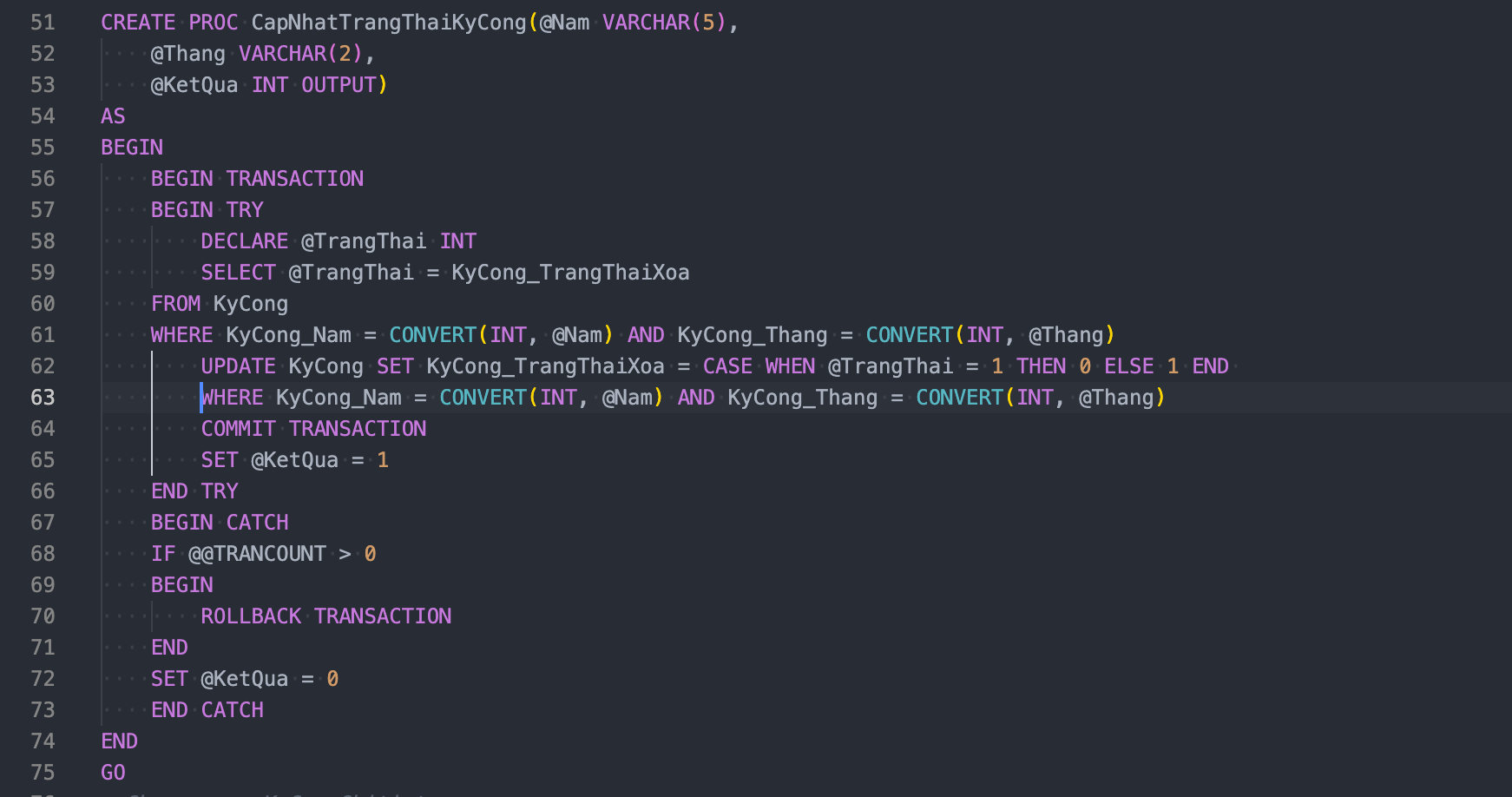
* 1. **Cập nhật kỳ công**

Thủ tục thực hiện cập nhật kỳ công với giá trị của tháng, năm truyền vào.



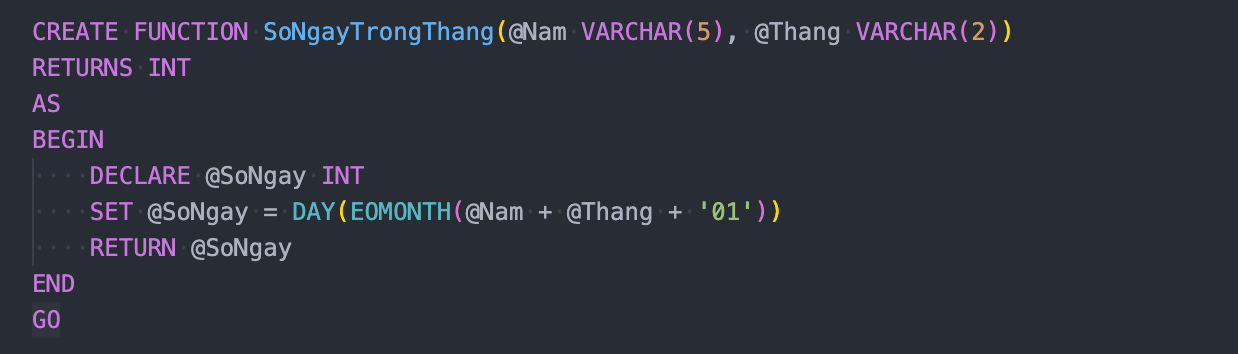
* 1. **Cập nhật trạng thái xóa kỳ công**

Thủ tục thực hiện cập nhật trạng thái xóa của kỳ công với giá trị cập nhật ngược với giá trị ban đầu



* 1. **Tính số ngày trong tháng**

Hàm tính số ngày trong tháng thực hiện trả về số ngày dựa vào giá trị tháng, năm truyền vào



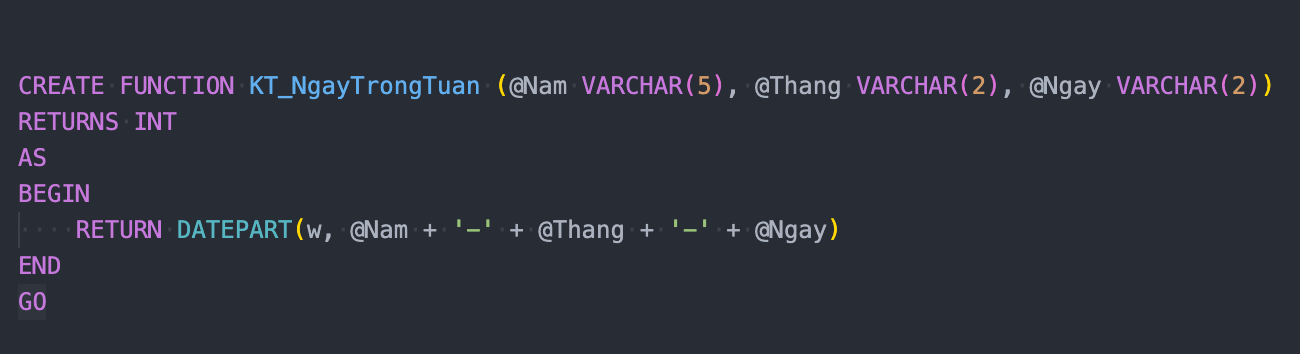
* 1. **Tính số ngày công của một kỳ công**

Hàm tính số ngày công thực hiện tính ngày làm việc dựa vào giá trị năm và tháng truyền vào thực hiện tính và trả về số ngày làm việc không bao gồm ngày chủ nhật



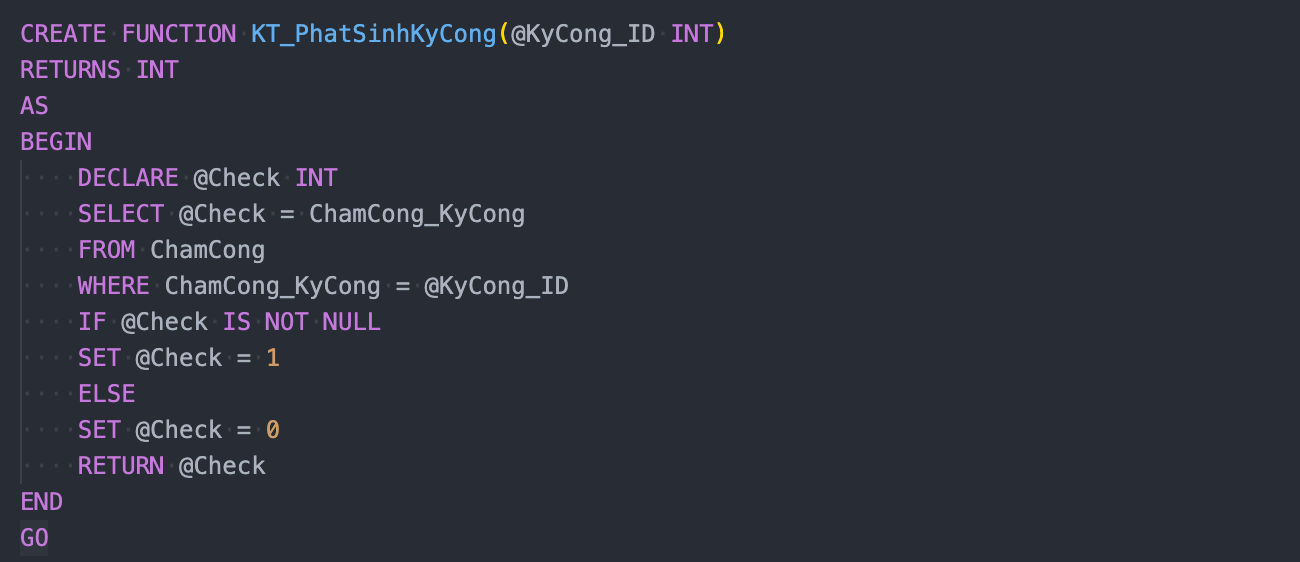
* 1. **Kiểm tra ngày trong tuần**

Hàm kiểm tra ngày trong tuần thực hiện trả về thứ trong tuần dựa vào giá trị ngày tháng năm truyền vào



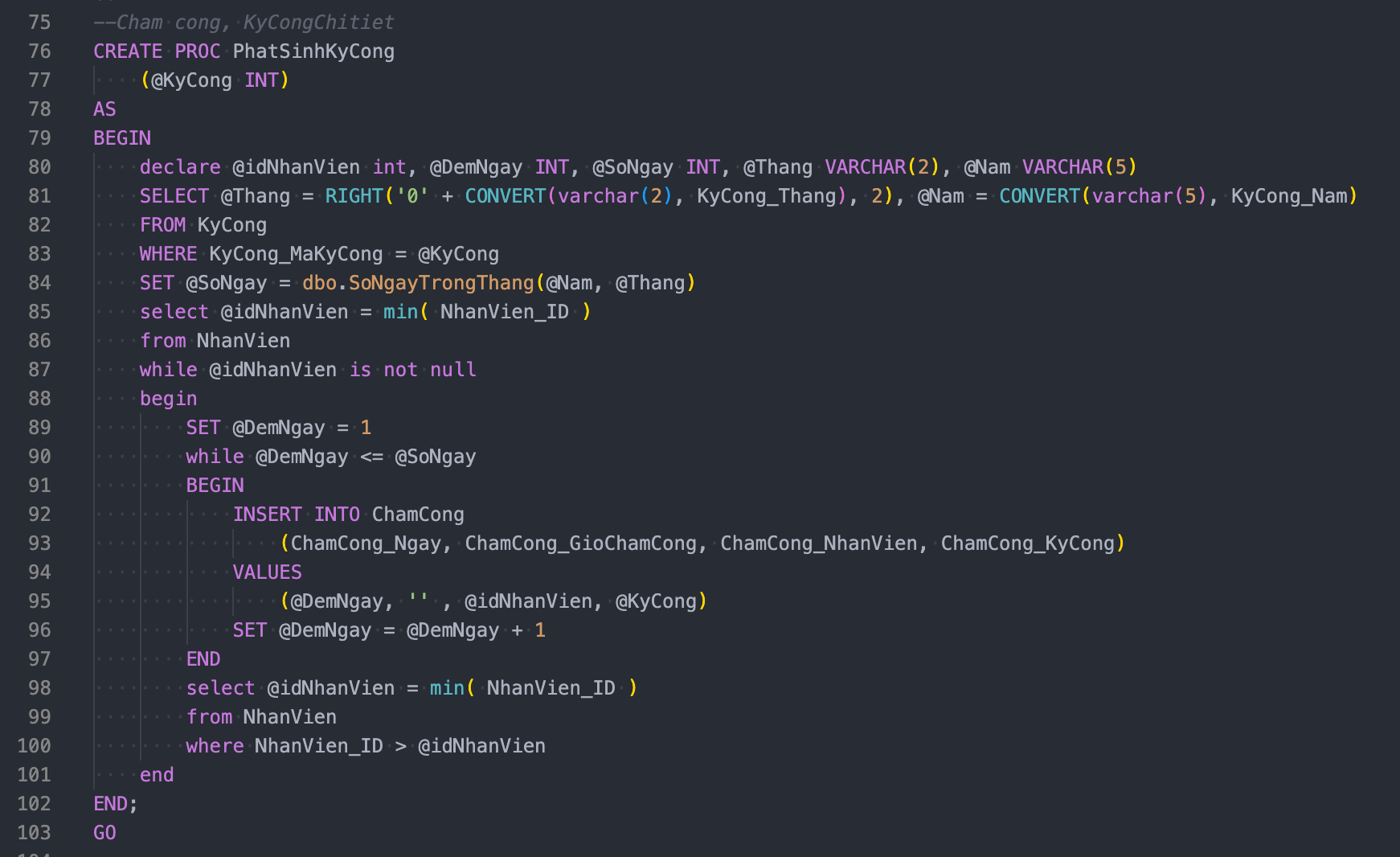
1. **Quản lý chấm công**
   1. **Kiểm tra phát sinh kỳ công**

Hàm kiểm tra phát sinh kỳ công thực hiện kiểm tra đã thực hiện chấm công cho kỳ công truyền vào không



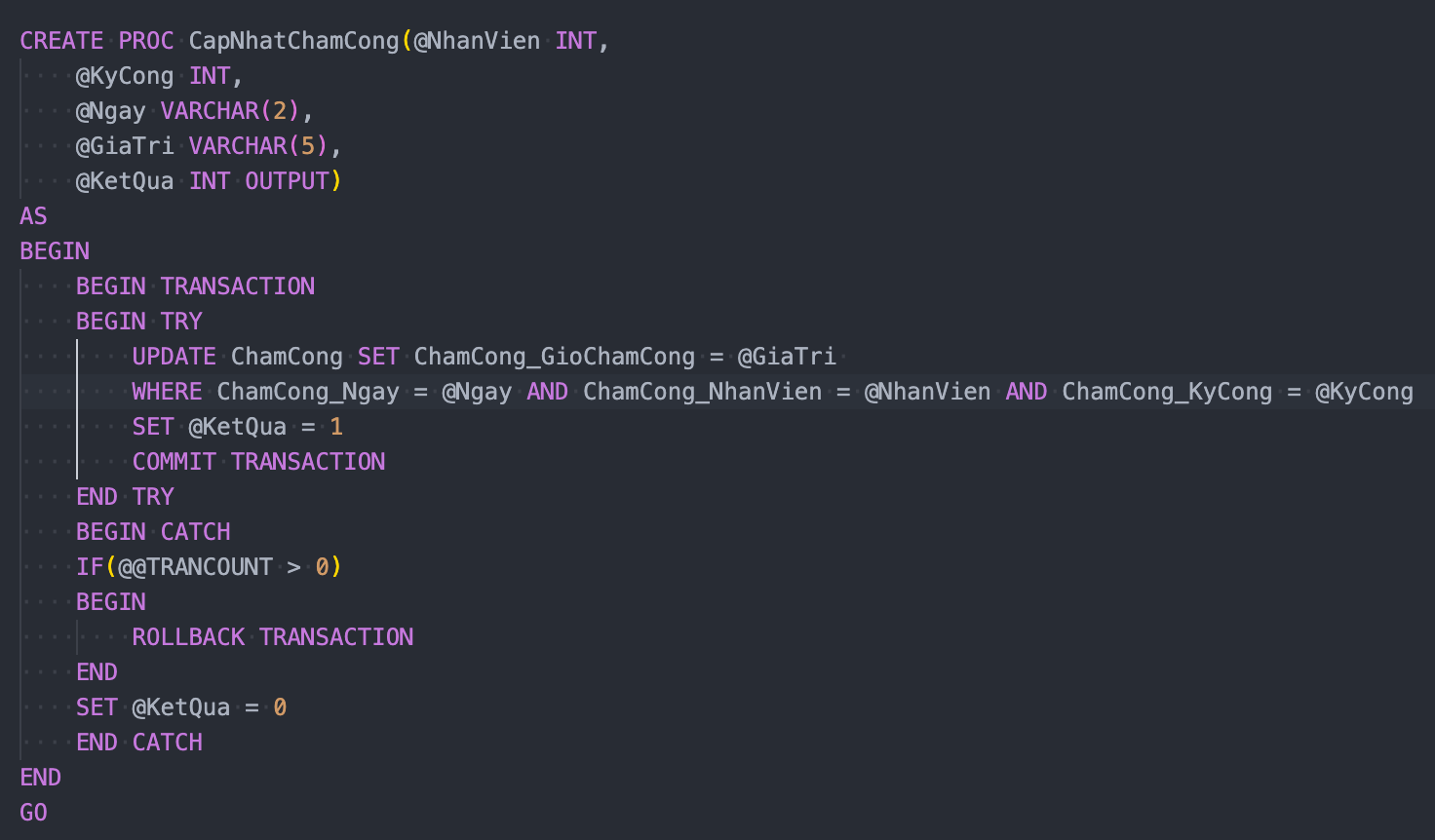
* 1. **Thêm dữ liệu chấm công**

Thủ tục phát sinh kỳ công thực hiện thêm tất cả các ngày công cho từng nhận viên trong kỳ công nhận vào với giá trị giờ chấm công ban đầu rỗng



* 1. **Cập nhật chấm công**

Thủ tục cập nhật chấm công thực hiện cập nhật thời gian chấm công của nhân viên trong ngày ở một kỳ công.



* 1. **Xóa chấm công**

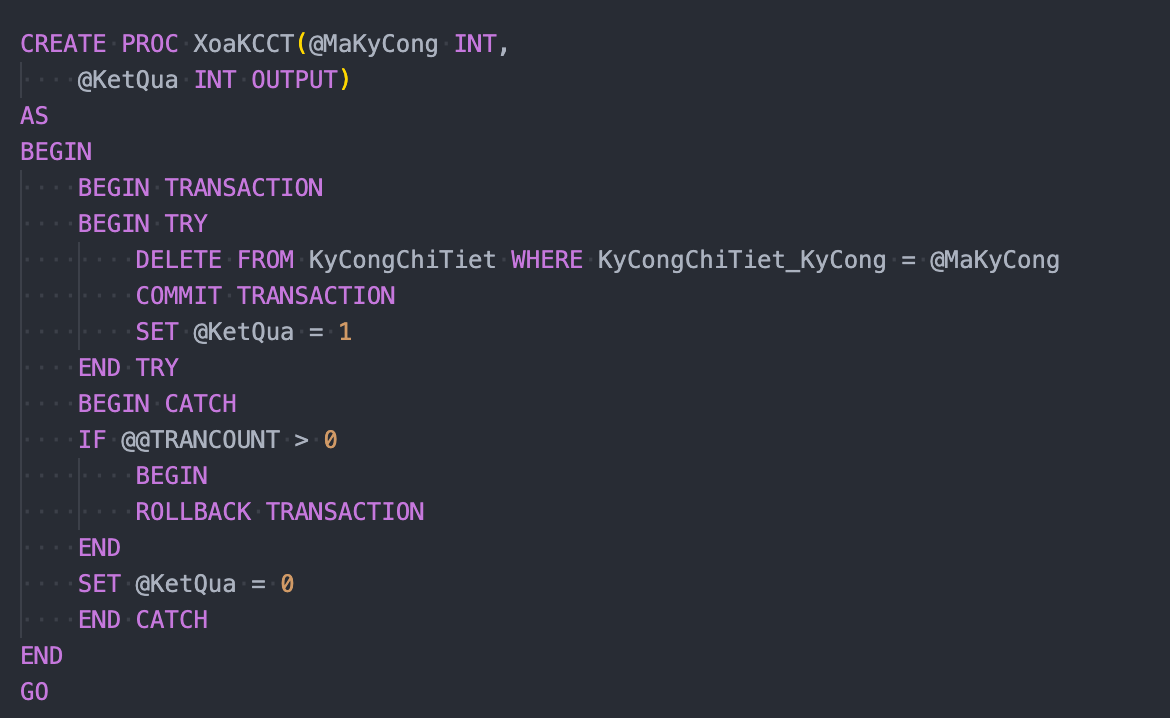
Thủ tục thực hiện xóa kết quả chấm công của từng nhân viên trong tất cả các ngày của kỳ công





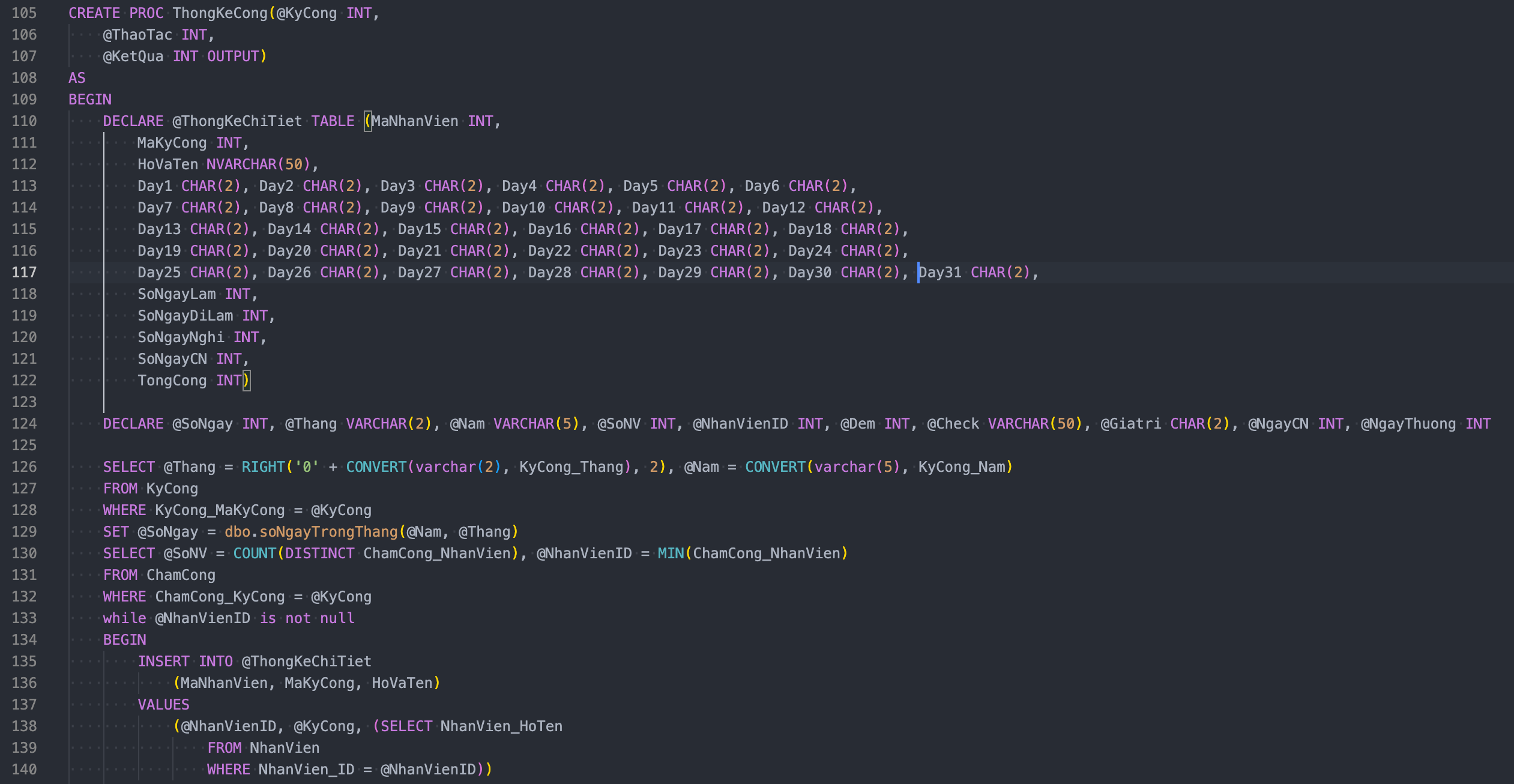
1. **Quản lý kỳ công chi tiết**
   1. **Xóa kỳ công chi tiết của kỳ công**

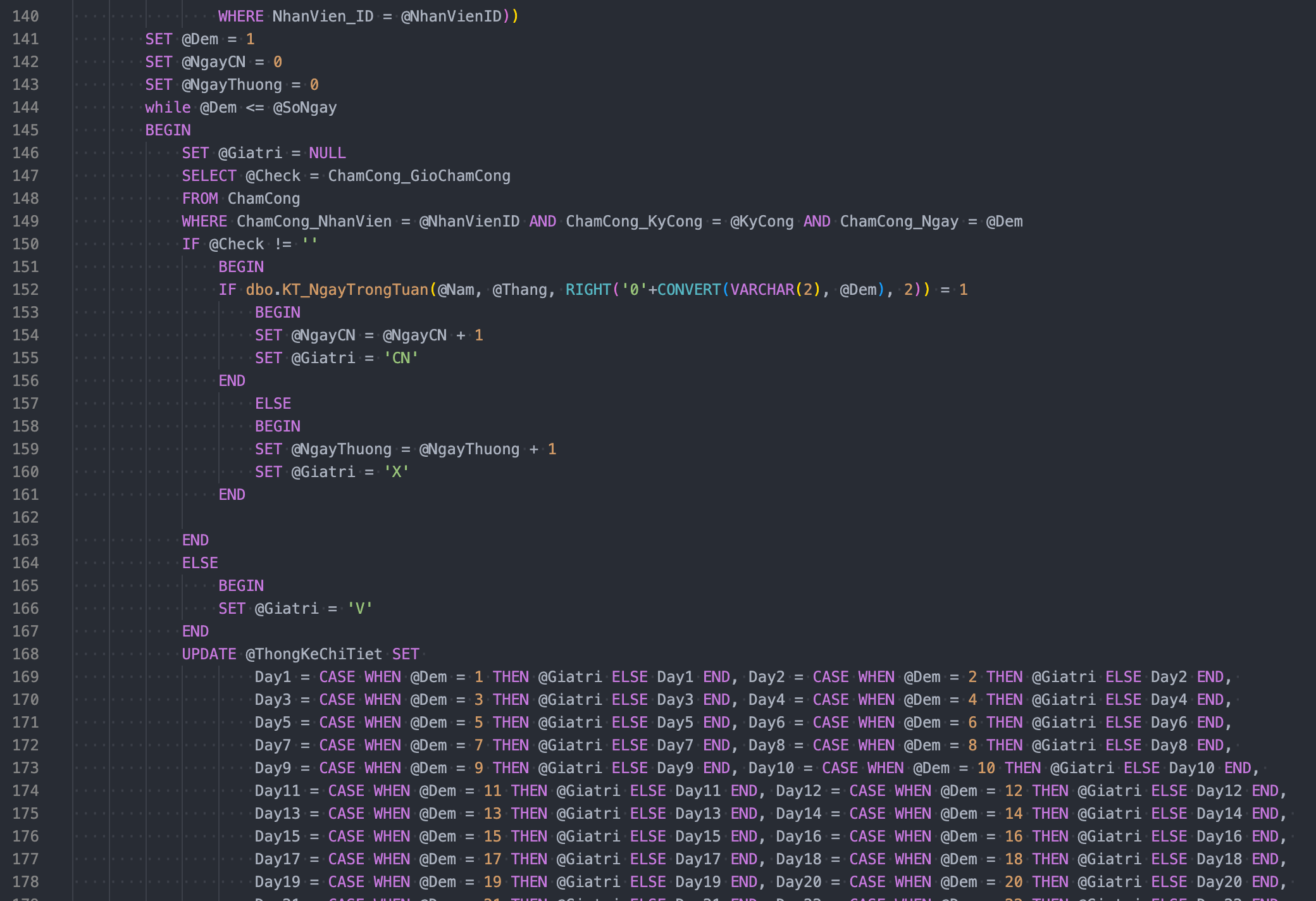
Thủ tục xóa kỳ công chi tiết thực hiện xóa danh sách kỳ công chi tiết của từng nhân viên dựa vào mã kỳ công truyền vào

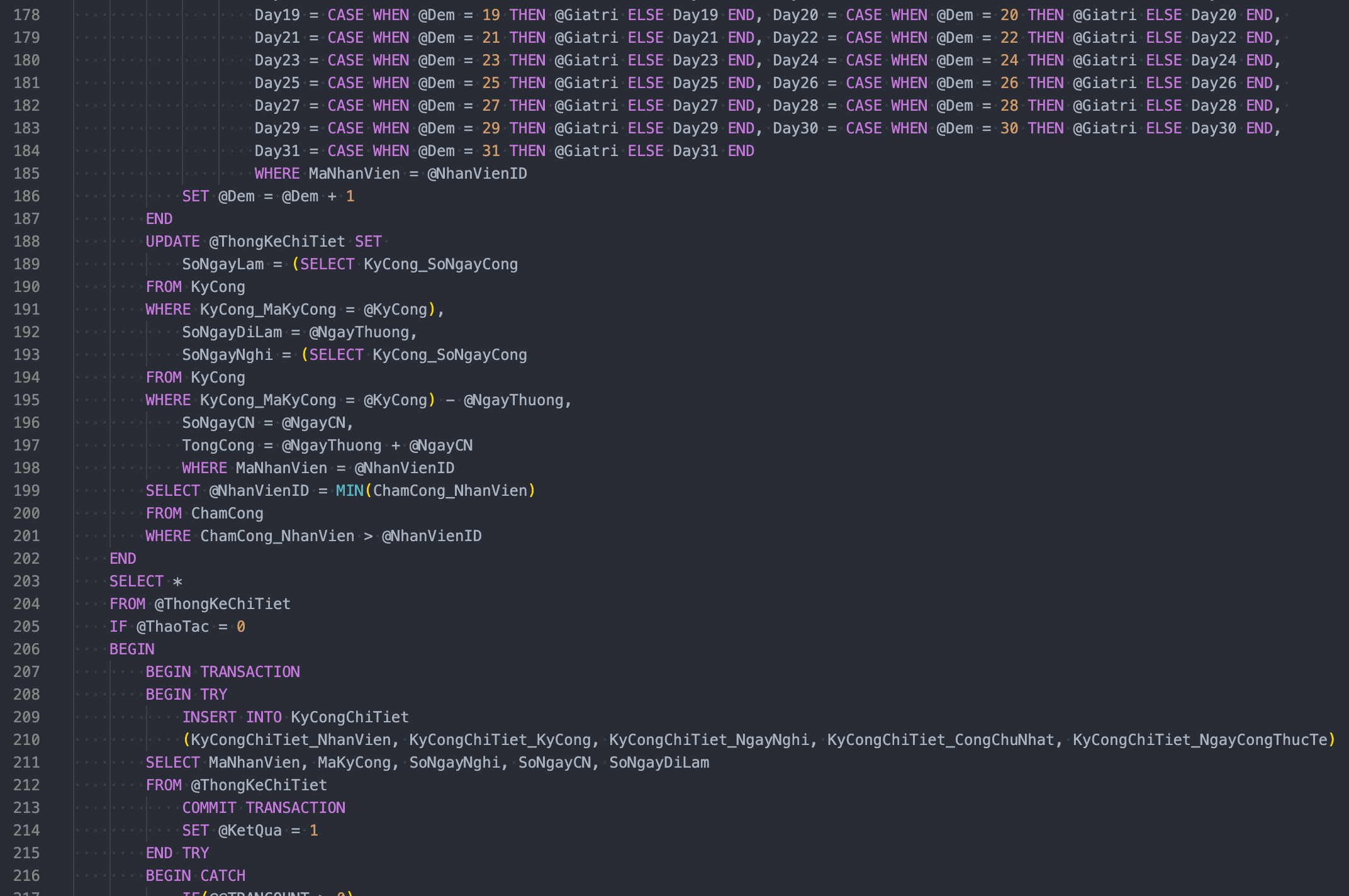


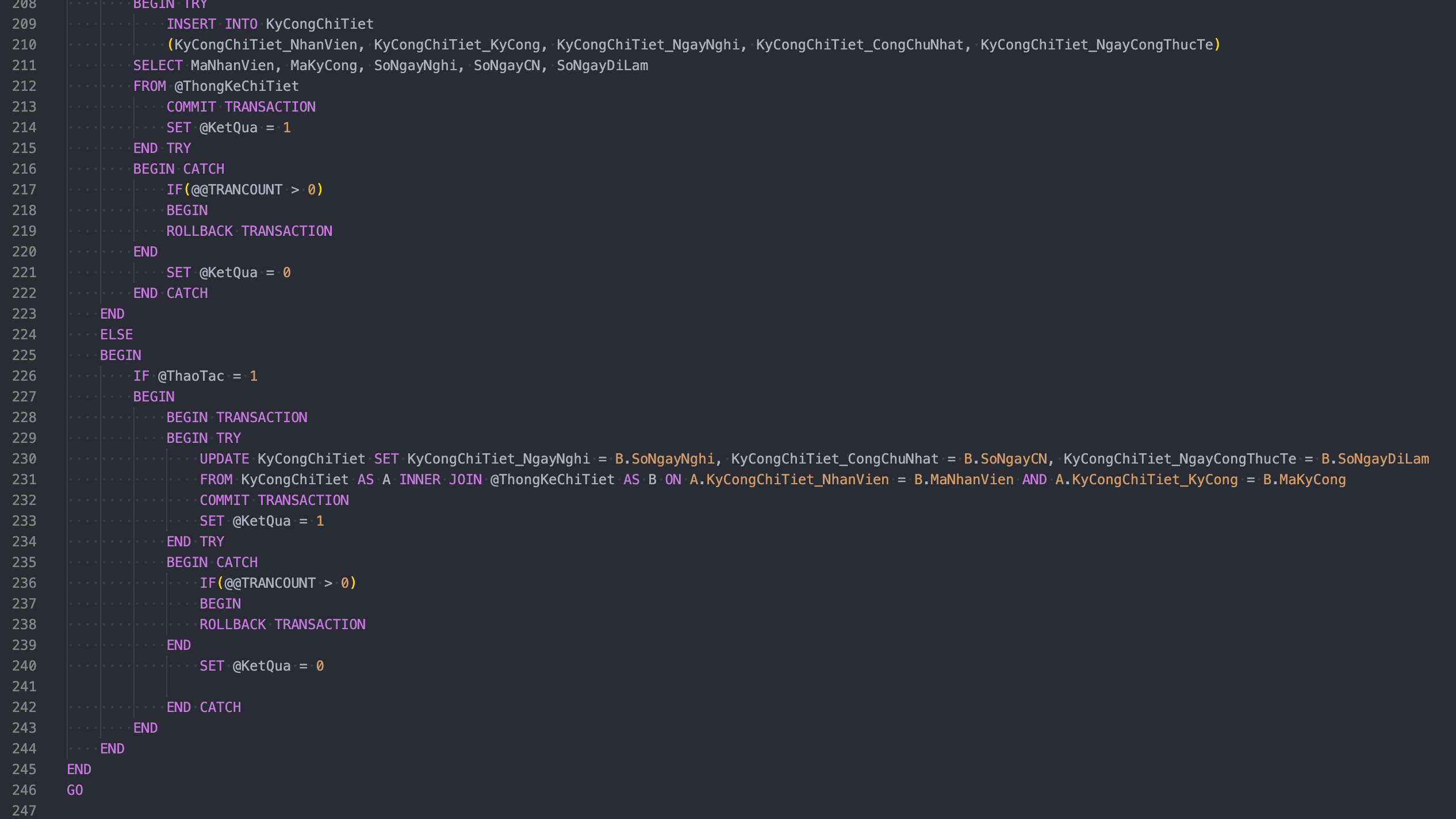
* 1. **Thống kê và thêm kỳ công chi tiết cho kỳ công**

Thủ tục thống kê chấm công thực hiện trả về bảng thống kê ngày đi làm của từng nhân viên dựa vào mã kỳ công truyền vào. Và thực hiện thêm hoặc cập nhật bảng kỳ công chi tiết dựa vào giá trị thao tác nhận vào.









1. **Quản lý ứng lương**
   1. **Tạo hàm hiển thị và gọi hàm lấy thông tin ứng lương**

CREATE FUNCTION UngLuong\_HienThi()

RETURNS TABLE

AS

RETURN (

SELECT ul.UngLuong\_ID, ul.UngLuong\_Ngay, ul.UngLuong\_SoTien, ul.UngLuong\_TrangThaiXoa, ul.UngLuong\_GhiChu, nv.NhanVien\_HoTen,kc.KyCong\_Nam, kc.KyCong\_Thang

FROM UngLuong ul

INNER JOIN KyCong kc ON kc.KyCong\_MaKyCong = ul.UngLuong\_KyCong

INNER JOIN NhanVien nv ON nv.NhanVien\_ID = ul.UngLuong\_NhanVien

)

GO

CREATE PROC [LayThongtinUngLuong]

AS

SELECT \* FROM dbo.UngLuong\_HienThi()

GO

6.2. **Xem thông tin chi tiết từng nhân viên ứng lương thông qua giá trị id truyền vào**

CREATE PROC [ungluongtheoid]

@ma\_id INT

AS

SELECT \* FROM UngLuong\_HienThi() ul WHERE ul.UngLuong\_ID=@ma\_id

GO

**6.3. Thêm dữ liệu ứng lương, nếu dữ liệu đã trùng rollback lại procedure , nếu dữ liệu mới sẽ được thêm vào**

CREATE PROCEDURE [ThemUngLuong]

@ngay INT,

@tien FLOAT,

@ghichu NVARCHAR(50),

@nhanvien NVARCHAR(50),

@nam INT,

@thang INT,

@ketqua INT OUTPUT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @nhanvien\_id INT, @kycong\_makycon INT

SELECT @nhanvien\_id = NhanVien\_ID FROM NhanVien WHERE NhanVien\_HoTen = @nhanvien

SELECT @kycong\_makycon = kc.KyCong\_MaKyCong FROM KyCong kc WHERE kc.KyCong\_Nam = @nam AND kc.KyCong\_Thang = @thang

IF NOT EXISTS(SELECT \* FROM UngLuong WHERE UngLuong\_Ngay = @ngay AND UngLuong\_NhanVien = @nhanvien\_id AND UngLuong\_KyCong = @kycong\_makycon)

BEGIN

BEGIN TRANSACTION;

BEGIN TRY

INSERT INTO UngLuong (UngLuong\_Ngay, UngLuong\_SoTien, UngLuong\_TrangThaiXoa, UngLuong\_GhiChu, UngLuong\_NhanVien, UngLuong\_KyCong)

VALUES

(

@ngay,

@tien,

0,

@ghichu,

@nhanvien\_id,

@kycong\_makycon

);

SET @ketqua=1;

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION;

SET @ketqua=0;

END CATCH

END

END

**6.4 Sửa dữ liệu ứng lương**

CREATE PROC [SuaUngLuong]

@id INT,

@ngay INT,

@tien FLOAT,

@ghichu NVARCHAR(50),

@nhanvien NVARCHAR(50),

@nam INT,

@thang INT

AS

UPDATE UngLuong

SET UngLuong\_Ngay = @ngay,

UngLuong\_SoTien=@tien,

UngLuong\_GhiChu = @ghichu,

UngLuong\_NhanVien = (SELECT NhanVien\_ID FROM NhanVien WHERE NhanVien\_HoTen = @nhanvien),

UngLuong\_KyCong = (SELECT kc.KyCong\_MaKyCong FROM KyCong kc WHERE kc.KyCong\_Nam = @nam AND kc.KyCong\_Thang = @thang)

WHERE UngLuong\_ID = @id

GO

**6.4 Set trạng thái trước khi xóa dữ liệu ứng lương**

CREATE PROC [XoaUngLuong]

@id INT

AS

UPDATE UngLuong

SET UngLuong\_TrangThaiXoa = 1 WHERE UngLuong\_ID = @id

GO

**6.5 Lưu hành động khi vẫn muốn xóa , trạng thái đã =1**

CREATE PROC [LuuXoaUngLuong]

@id INT

AS

IF EXISTS(SELECT \* FROM UngLuong WHERE UngLuong\_ID = @id AND UngLuong\_TrangThaiXoa = 1)

BEGIN

DELETE FROM UngLuong WHERE UngLuong\_ID = @id AND UngLuong\_TrangThaiXoa = 1

END

GO

**6.5 Hủy hành động xóa set lại trạng thái =0**

CREATE PROC [HuyXoaUngLuong]

@id INT

AS

UPDATE UngLuong

SET UngLuong\_TrangThaiXoa = 0 WHERE UngLuong\_ID = @id

GO

1. **Quản Lý Tăng Ca**
   1. **Xem thông tin tăng ca, hiển thị thêm về ngày,tháng ,năm tăng ca**

CREATE FUNCTION TangCa\_HienThi()

RETURNS TABLE

AS

RETURN (

SELECT TC.TangCa\_ID, TC.TangCa\_NgayTangCa, TC.TangCa\_SoGio, NV.NhanVien\_HoTen, LTC.LoaiTangCa\_TenLoai, kc.KyCong\_Thang, kc.KyCong\_Nam

FROM TangCa TC

INNER JOIN NhanVien NV ON TC.TangCa\_NhanVien = NV.NhanVien\_ID

INNER JOIN LoaiTangCa LTC ON TC.TangCa\_LoaiTangCa = LTC.LoaiTangCa\_ID

INNER JOIN KyCong kc ON kc.KyCong\_MaKyCong = TC.TangCa\_KyCong

)

GO

CREATE PROC LayThongtinTangCa

AS

SELECT \* FROM dbo.TangCa\_HienThi() tc

GO

**7.2 Lấy dữ liệu cụ thể thông qua id**

CREATE PROC [tangcatheoid]

@ma\_id INT

AS

SELECT \* FROM dbo.TangCa\_HienThi() tc WHERE tc.TangCa\_ID=@ma\_id

GO

**7.3 Thêm dữ liệu vào tăng ca nếu dữ liệu đã trùng =, tên loại tăng ca không hợp lệ báo lỗi và rollback . Nếu tất cả dữ liệu đều thỏa tiến hành thêm dữ liệu**

CREATE PROCEDURE [ThemTangCa]

@ngay INT,

@gio FLOAT,

@tennv NVARCHAR(50),

@tenloaitc NVARCHAR(50),

@thangkycong INT,

@namkycong INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @nhanvien\_id INT;

DECLARE @loaitc\_id INT;

-- Kiểm tra xem dữ liệu đã tồn tại hay chưa

IF EXISTS (SELECT \* FROM TangCa

WHERE TangCa\_NgayTangCa = @ngay

AND TangCa\_NhanVien = (SELECT NhanVien\_ID FROM NhanVien WHERE NhanVien\_HoTen = @tennv)

AND TangCa\_KyCong = (SELECT kc.KyCong\_MaKyCong FROM KyCong kc WHERE kc.KyCong\_Thang = @thangkycong AND kc.KyCong\_Nam = @namkycong))

BEGIN

RAISERROR('Dữ liệu đã tồn tại', 16, 1);

RETURN;

END

-- Lấy ID của loại tăng ca

SELECT @loaitc\_id = LoaiTangCa\_ID FROM LoaiTangCa WHERE LoaiTangCa\_TenLoai = @tenloaitc;

IF(@loaitc\_id IS NULL)

THROW 50000, 'Không tìm thấy loại tăng ca', 1;

BEGIN TRANSACTION;

BEGIN TRY

INSERT INTO TangCa (TangCa\_NgayTangCa, TangCa\_SoGio, TangCa\_NhanVien, TangCa\_LoaiTangCa, TangCa\_KyCong)

VALUES

(

@ngay,

@gio,

(SELECT NhanVien\_ID FROM NhanVien WHERE NhanVien\_HoTen = @tennv),

@loaitc\_id,

(SELECT kc.KyCong\_MaKyCong FROM KyCong kc WHERE kc.KyCong\_Thang = @thangkycong AND kc.KyCong\_Nam = @namkycong)

);

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION;

THROW;

END CATCH

END

GO

**7.4 Xóa tăng ca khi truyền dữ liệu id vào**

CREATE PROC [XoaTangCa]

@id INT

AS

DELETE FROM TangCa WHERE TangCa\_ID = @id

GO

1. **Quản Lý Lương**

**8.1. Xem lương được kết hợp từ 6 bảng : Nhân viên, Kỳ Công,Kỳ Công Chi Tiết, Tăng Ca, Loại Tăng Ca, Ứng Lương, Hợp Đồng, Hệ Số Lương ,mỗi nhân viên trong 1 kỳ công nhất định**

INSERT INTO BangLuong(BangLuong\_NhanVien, BangLuong\_KyCong, BangLuong\_LuongNgayThuong, BangLuong\_LuongNgayCN,

BangLuong\_TangCa, BangLuong\_UngLuong, BangLuong\_PhuCap, BangLuong\_LuongNhanDuoc, BangLuong\_ThucLanh)

SELECT nv.NhanVien\_ID,kc.KyCong\_MaKyCong,kcct.KyCongChiTiet\_NgayCongThucTe\*((3250000\*hsl.HeSoLuong\_GiaTri)/26) , kcct.KyCongChiTiet\_CongChuNhat\*((3250000\*hsl.HeSoLuong\_GiaTri)/26)\*2 ,

tc.TangCa\_SoGio\*ltc.LoaiTangCa\_HeSo\*((3250000\*hsl.HeSoLuong\_GiaTri)/26),

ul.UngLuong\_SoTien ,556 ,kcct.KyCongChiTiet\_NgayCongThucTe\*((3250000\*hsl.HeSoLuong\_GiaTri)/26)+kcct.KyCongChiTiet\_CongChuNhat\*((3250000\*hsl.HeSoLuong\_GiaTri)/26)\*2 +tc.TangCa\_SoGio\*ltc.LoaiTangCa\_HeSo\*((3250000\*hsl.HeSoLuong\_GiaTri)/26) - ul.UngLuong\_SoTien +556 , (kcct.KyCongChiTiet\_NgayCongThucTe\*((3250000\*hsl.HeSoLuong\_GiaTri)/26)+kcct.KyCongChiTiet\_CongChuNhat\*((3250000\*hsl.HeSoLuong\_GiaTri)/26)\*2 +tc.TangCa\_SoGio\*ltc.LoaiTangCa\_HeSo\*((3250000\*hsl.HeSoLuong\_GiaTri)/26) - ul.UngLuong\_SoTien +556 )\*0.9

FROM NhanVien nv,KyCong kc,KyCongChiTiet kcct,TangCa tc,LoaiTangCa ltc,UngLuong ul,HopDong hd,HeSoLuong hsl

WHERE kc.KyCong\_MaKyCong = kcct.KyCongChiTiet\_KyCong AND tc.TangCa\_LoaiTangCa = ltc.LoaiTangCa\_ID AND ul.UngLuong\_NhanVien = nv.NhanVien\_ID

AND kcct.KyCongChiTiet\_NhanVien = nv.NhanVien\_ID and tc.TangCa\_NhanVien = nv.NhanVien\_ID AND ul.UngLuong\_KyCong = kc.KyCong\_MaKyCong

AND hd.HopDong\_NhanVien = nv.NhanVien\_ID AND hsl.HeSoLuong\_ID = hd.HopDong\_HeSoLuong

**8.1.Tạo hàm kết hợp từ bảng lương và nhân viên kỳ công xem Họ tên và tháng năm lương cụ thể, sau đó procedure thực hiện xem**

CREATE FUNCTION Luong\_HienThi()

RETURNS TABLE

AS

RETURN (

SELECT nv.NhanVien\_HoTen,kc.KyCong\_Thang,kc.KyCong\_Nam,bl.BangLuong\_LuongNgayThuong, bl.BangLuong\_LuongNgayCN, bl.BangLuong\_TangCa, bl.BangLuong\_UngLuong, bl.BangLuong\_PhuCap, bl.BangLuong\_LuongNhanDuoc, bl.BangLuong\_ThucLanh

FROM BangLuong bl

INNER JOIN NhanVien nv ON bl.BangLuong\_NhanVien = nv.NhanVien\_ID

INNER JOIN KyCong kc ON kc.KyCong\_MaKyCong = bl.BangLuong\_KyCong

)

GO

CREATE PROC hienthiluong

AS

SELECT \* FROM Luong\_HienThi()

EXEC hienthiluong

GO

1. **Quản lý loại tăng ca**

-- LoaiTangCa

CREATE PROCEDURE sp\_ThemMoiLoaiTangCa

@LoaiTangCa\_ID INT,

@LoaiTangCa\_TenLoai NVARCHAR(20),

@LoaiTangCa\_HeSo FLOAT

AS

BEGIN

INSERT INTO LoaiTangCa (LoaiTangCa\_ID, LoaiTangCa\_TenLoai, LoaiTangCa\_HeSo)

VALUES (@LoaiTangCa\_ID, @LoaiTangCa\_TenLoai, @LoaiTangCa\_HeSo)

END

GO

CREATE PROCEDURE sp\_XoaLoaiTangCa

@LoaiTangCa\_ID INT

AS

BEGIN

DELETE FROM LoaiTangCa

WHERE LoaiTangCa\_ID = @LoaiTangCa\_ID

END

GO

CREATE PROCEDURE sp\_SuaLoaiTangCa

@LoaiTangCa\_ID INT,

@LoaiTangCa\_TenLoai NVARCHAR(20),

@LoaiTangCa\_HeSo FLOAT

AS

BEGIN

UPDATE LoaiTangCa

SET LoaiTangCa\_TenLoai = @LoaiTangCa\_TenLoai, LoaiTangCa\_HeSo = @LoaiTangCa\_HeSo

WHERE LoaiTangCa\_ID = @LoaiTangCa\_ID

END

GO

1. **Quản lý Chức Vụ**

-- ChucVu

CREATE PROCEDURE sp\_ThemMoiChucVu

@ChucVu\_ID INT,

@ChucVu\_TenCV NVARCHAR(50)

AS

BEGIN

INSERT INTO ChucVu (ChucVu\_ID, ChucVu\_TenCV)

VALUES (@ChucVu\_ID, @ChucVu\_TenCV)

END

GO

CREATE PROCEDURE sp\_XoaChucVu

@ChucVu\_ID INT

AS

BEGIN

DELETE FROM ChucVu

WHERE ChucVu\_ID = @ChucVu\_ID

END

GO

CREATE PROCEDURE sp\_SuaChucVu

@ChucVu\_ID INT,

@ChucVu\_TenCV NVARCHAR(50)

AS

BEGIN

UPDATE ChucVu

SET ChucVu\_TenCV = @ChucVu\_TenCV

WHERE ChucVu\_ID = @ChucVu\_ID

END

GO

1. **Quản lý phòng ban**

-- PhongBan

CREATE PROCEDURE sp\_ThemPhongBan

(

@MaPB INT,

@TenPB NVARCHAR(50),

@TruongPhong INT,

@TG\_NhanChuc DATE

)

AS

BEGIN

INSERT INTO PhongBan (PhongBan\_MaPB, PhongBan\_TenPB, PhongBan\_TruongPhong, PhongBan\_TG\_NhanChuc)

VALUES (@MaPB, @TenPB, @TruongPhong, @TG\_NhanChuc);

END

GO

CREATE PROCEDURE sp\_XoaPhongBan

(

@MaPB INT

)

AS

BEGIN

DELETE FROM PhongBan WHERE PhongBan\_MaPB = @MaPB;

END

GO

ALTER PROCEDURE sp\_ThemPhongBan

(

@TenPB NVARCHAR(50),

@TruongPhong INT,

@TG\_NhanChuc DATE

)

AS

BEGIN

INSERT INTO PhongBan (PhongBan\_TenPB, PhongBan\_TruongPhong, PhongBan\_TG\_NhanChuc)

VALUES (@TenPB, @TruongPhong, @TG\_NhanChuc);

END

GO

CREATE PROCEDURE sp\_CapNhatPhongBan

(

@MaPB INT,

@TenPB NVARCHAR(50),

@TruongPhong INT,

@TG\_NhanChuc DATE

)

AS

BEGIN

UPDATE PhongBan SET

PhongBan\_TenPB = @TenPB,

PhongBan\_TruongPhong = @TruongPhong,

PhongBan\_TG\_NhanChuc = @TG\_NhanChuc

WHERE PhongBan\_MaPB = @MaPB;

END

GO

1. **Quản lý tài khoản**

CREATE PROCEDURE ThemTaiKhoan

@SoTK VARCHAR(50),

@MatKhau VARCHAR(32),

@PhanQuyen INT,

@NhanVien INT

AS

BEGIN

INSERT INTO TaiKhoan (TaiKhoan\_SoTK, TaiKhoan\_MatKhau, TaiKhoan\_PhanQuyen, TaiKhoan\_NhanVien)

VALUES (@SoTK, @MatKhau, @PhanQuyen, @NhanVien);

END

GO

CREATE PROCEDURE XoaTaiKhoan

@SoTK VARCHAR(50)

AS

BEGIN

DELETE FROM TaiKhoan WHERE TaiKhoan\_SoTK = @SoTK;

END

GO

CREATE PROCEDURE SuaThongTinTaiKhoan

@SoTK VARCHAR(50),

@MatKhau VARCHAR(32) = NULL,

@PhanQuyen INT = NULL,

@NhanVien INT = NULL

AS

BEGIN

UPDATE TaiKhoan SET

TaiKhoan\_MatKhau = COALESCE(@MatKhau, TaiKhoan\_MatKhau),

TaiKhoan\_PhanQuyen = COALESCE(@PhanQuyen, TaiKhoan\_PhanQuyen),

TaiKhoan\_NhanVien = COALESCE(@NhanVien, TaiKhoan\_NhanVien)

WHERE TaiKhoan\_SoTK = @SoTK;

END

GO

1. **Phân quyền**

-- PhanQuyen

CREATE PROCEDURE ThemMoiPhanQuyen

@TenQuyen NVARCHAR(20)

AS

BEGIN

INSERT INTO PhanQuyen(PhanQuyen\_TenQuyen)

VALUES(@TenQuyen)

END;

GO

--Xóa một bản ghi khỏi bảng PhanQuyen dựa trên PhanQuyen\_ID:

CREATE PROCEDURE XoaPhanQuyen

@PhanQuyen\_ID INT

AS

BEGIN

DELETE FROM PhanQuyen

WHERE PhanQuyen\_ID = @PhanQuyen\_ID

END;

GO

--Cập nhật một bản ghi trong bảng PhanQuyen dựa trên PhanQuyen\_ID:

CREATE PROCEDURE CapNhatPhanQuyen

@PhanQuyen\_ID INT,

@TenQuyen NVARCHAR(20)

AS

BEGIN

UPDATE PhanQuyen

SET PhanQuyen\_TenQuyen = @TenQuyen

WHERE PhanQuyen\_ID = @PhanQuyen\_ID

END;

GO

CREATE VIEW vw\_PhanQuyen

AS

SELECT PhanQuyen\_ID, PhanQuyen\_TenQuyen

FROM PhanQuyen

GO