**Phần mở đầu**

1. **Lý do chọn đề tài**

Trong xã hội nhàu nay cuộc sống phát triển, cuộc sống con người mỗi lúc được cải thiện hơn do đó nhu cầu cuộc sống, làm việc và giải trí của con người ngày được nâng cao. Để đáp ứng được các nhu cầu của xã hội thì người dùng đã áp dụng công nghệ thông tin vào cuộc sống một cách hiệu quả nhất. Các phần mềm ứng dụng quản lý có nhiệm vụ tin học hoá các quy trình của doanh nghiệp. Hệ thống dùng để nạp, quản trị các thông số, dữ liệu của một doanh nghiệp. Các phần mềm quản lý được ứng dụng rộng rãi trong cuộc sống nhưu quản lý nhà hàng, khách sạn, quán ăn, ... Trong đó gara ô tô cũng là mô hình sử dụng phần mềm quản lý khá phổ biến.

Trong kinh doanh ở bất kỳ mô hình nào, dù đơn giản hay phức tạp thì đều sẽ có một hệ cơ sở dữ liệu riêng, nó được sinh ra trong quá trình hoạt động của doanh nghiệp. Do đó để có thể kinh doanh và vận hành tốt thì việc có một chương trình quản lý giúp hệ cơ sở dữ liệu được quản lý tốt và tương tác hiệu quả với con người là điều vô cùng cần thiết. Nên chúng em sẽ sử dụng Winform là một công cụ tốt có thể đáp ứng được nhiệm vụ tương tác với người dùng một cách tốt nhất, thuận tiện nhất để quản lý mô hình kinh doanh đơn giản của một Gara.

1. **Mục tiêu**

Tạo một chương trình có đầy đủ các chức năng phục vụ tốt cho việc kinh doanh của một Garage, đồng thời có thể giúp một người có thể dễ dàng làm quen và sử dụng làm việc hiệu quả. Chương trình còn phải tương tác tốt với dữ liệu, đảm bảo dữ liệu đưa vào hay lấy ra đều không làm rối hay vi phạm các quy định của bộ dữ liệu.

1. **Công cụ hỗ trợ**

* Môi trường: .net framework
* Ngôn ngữ: C#, SQL
* Hệ quản trị cơ sở dữ liệu: Microsoft SQL server management studio
* Thư viện giao diện chính: Guna frameword winform

1. **Phương pháp nghiên cứu**

* Thu thập các tài liệu, tìm hiểu về cách sử dụng windows form
* Hiểu rõ và vận dụng các kiến thực của hệ cơ sở dữ liệu các nội dung
  + Cách tạo cơ sở dữ liệu: thiết lập các bảng và tạo các ràng buộc
  + Sử dụng view và trigger
  + Sử dụng Function và Procedure để tương tác với các bảng trong cơ sở dữ liệu
  + Hiểu và vận dụng được Transaction
  + Phân quyền rõ ràng cho cơ sở dữ liệu
* Tìm hiểu các hoạt động quản lý của gara.
* Tổng hợp và phân tích thiết kế phần mềm quản lý gara bằng windows form.

**Phần nội dung**

1. **Đặc tả đề tài**
   1. **Dữ liệu của để tài**

Chương trình sử dụng tất cả 15 bảng để quản lí các dữ liệu cơ bản và cần thiết cho một Garage:

* **User(Username, Password, Chucvu, MaNV):** Bảng User lưu **Username**(tên tài khoản) và **Password**(mật khẩu), mỗi cá nhân người dùng chỉ có duy nhất một **Username** để đăng nhập vào chương trình. Bên cạnh đó, bảng User còn có thuộc tính **Chucvu** để phân biệt quyền đăng nhập vào chương trình đối với từng các nhân sử dụng.
* **TT\_Nguoi(NguoiID,Ten, DiaChi, DienThoai, NgaySinh, CCCD, GioiTinh):** **TT\_Nguoi** lưu trữ thông tin nhân viên và khách hàng, mỗi cá nhân sẽ có một ID riêng để phân biệt và bao gồm các thông tin cơ bản như tên, địa chỉ, số điện thoại, ngày sinh, cccd, giới tính.
* **Chucvu(MaCV, TenCV):** Bảng Chucvu sẽ bao gồm thuộc tính **MaCV** mỗi chức vụ có một mã riêng và **TenCV** để phân biệt vai trò của người dùng trong chương trình.
* **NhanVien(NV\_NguoiID, MaCV,Luong):** Mỗi nhân viên có một mã nhân viên được tham chiếu từ bảng **TTNguoi** và chỉ một mã chức vụ cùng một mức lương tương ứng.
* **KhachHang(KH\_NguoiID):** Mỗi khách hàng có một mã khách hàng (**KH\_NguoiID**) và các thông tin cá nhân được tham chiếu từ bảng **TT\_Nguoi** bằng **KH\_NguoiID.**
* **HoaDon(MaHoaDon, NgaylapHoaDon, MaHopDong, KH\_NguoiID, HoTen, SoTienThu):** Khách hàng(**KH\_NguoiID)** có thể thanh toán tiền của một hợp đồng (**MaHopDong**) làm nhiều lần trước hoặc sau khi nghiệm thu (trong cùng một ngày hoặc khác ngày). Mỗi lần thanh toán đều có mã hóa đơn (**MaHoaDon**) để phân biệt, thuộc tính **NgaylapHoaDon** thể hiện ngày lập hóa đơn. **SoTienThu** là số tiền thanh toán của lần thu đó. **HoTen** là họ tên của người mang tiền đến thanh toán (có thể khác với người đứng ra ký hợp đồng).
* **CongViec (MaCViec, NoiDungCViec,TienCong):** Một tiệm sửa xe sẽ chia công việc thành nhiều khâu khác nhau để thuận tiện cho việc tính toán chi phí với khách hàng. Mỗi công việc đều có một mã công việc(**MaCViec**) khác nhau, thuộc tính **NoiDungCV** mô tả nội dung chi tiết của công việc. **TienCong** là tiền mà khách hàng phải trả đối với từng công việc mà khách hàng đã yêu cầu.
* **VatLieu(MaVL, tenVL, SoLuong):** Mỗi vật liệu sẽ có một mã định danh riêng(**MaVL**) để phân biệt, **tenVL** là tên của vật liệu để biết rõ đó là vật liệu gì và thuộc tính **SoLuong** là số lượng của vật liệu còn tồn trong kho sẽ được cộng thêm khi nhập kho và trừ đi khi một công việc của khách hàng yêu cầu dùng tới vật liệu đó.
* **HopDong(SoHD, NgayHD, KH\_NguoiID, SoXe, TriGiaHD, NgayGiaoDK, NgayNgThu):** Mỗi hợp đồng sửa xe của Garage đều có một mã số hợp đồng khác nhau để phân biệt. **NgayHD** là ngày ký hợp đồng sửa xe với khách hàng là chủ xe(**KH\_NguoiID)**. **SoXe** là số đăng bộ của xe mà khách đem đến sửa. Một khách hàng có thể ký nhiều hợp đồng sửa chữa nhiều xe khác nhau hoặc sửa chữa nhiều lần cho cùng một xe trong cùng một ngày, những công việc sửa chữa cho một xe chỉ ký hợp đồng một lần. Thuộc tính **TriGiaHD** là tổng **TienCong** của các công việc được yêu cầu. **NgayGiaoDK** là ngày dự kiến phải giao xe cho khách. **NgayNgThu** là ngày nghiệm thu thật sự sau khi đã sửa chữa xong và khánh hàng thanh toán hết tiền của hợp đồng.
* **HopDong\_Backup(SoHD, NgayHD, KH\_NguoiID, SoXe, TriGiaHD, NgayGiaoDK, NgayNgThu):** bảng này để lưu trữ dữ liệu cho chức năng thống kê. Bảng có các cột giống như bảng HopDong với khóa chính là SoHD và không có khóa ngoại, sau khi một hợp đồng được nghiệm thu thì sẽ chuyển toàn bộ dữ liệu qua đây.
* **ChiTiet\_HD(SoHD, MaCViec, TriGiaCV, MaNV):** Mỗi hợp đồng sửa xe có thể gồm nhiều công việc, **MaCViec** là mã số của từng công việc, **TriGiaCV** là chi phí của công việc đã tính toán với khách, tổng **TriGiaCV** của một **SoHD** sẽ là **TriGiaHD** của một hợp đồng. Mỗi công việc của hợp đồng được giao cho một nhân viên (**MaNV**) phụ trách. Một người nhân viên có thể được giao một hoặc nhiều công việc của một hay nhiều hợp đồng khác nhau.
* **ChiTiet\_HD\_Backup(SoHD, MaCViec, TriGiaCV, MaNV):** cũng giống như bảng HopDong\_Backup thì bảng này cũng là bảng sao lưu của ChiTiet\_HD sau khi một hợp đồng được backup thì các chi tiết của nó cũng được chuyển đến đây.
* **NhapKho(MaNKho,MaVL, maNhaCC, SoLuong, Giatri, ngayNhap,MANV)**: Mỗi lần nhập kho cho một vật liệu sẽ có một mã định danh khác nhau(**MaNKho**), **MaVL** là mã của vật liệu được lấy ở bảng **VatLieu** dùng để định danh rõ ràng, mỗi vật liệu sẽ được cung cấp bởi một nhà cung cấp được xác định qua mã nhà cung cấp, **Giatri** là số tiền của vật liệu đó với số lượng là 1, **SoLuong** là tổng số lượng mặt hàng đó nhập trong một đợt kèm theo ngày nhập hàng, một ngày có thể nhập nhiều lần cho một vật liệu và cuối cùng cần có nhân viên xác nhận việc nhập hàng.
* **NhaCungCap(maNhaCC,TenNhaCC, DienThoai,DiaChi):** Mỗi nhà cung cấp phải có một **maNhaCC** khác nhau để thuận tiện cho việc quản lý vật liệu nhập vào kho. Thông tin nhà cung cấp bao gồm **TenNhaCC, DienThoai** và **DiaChi** để thuận tiện cho việc mua bán vật liệu.
  1. **Các chức năng của đề tài:**

Về phần chức năng, chương trình sẽ bao gồm 3 chức năng cơ bản: Lưu trữ thông tin(Thêm, xóa, chỉnh sửa), Tìm kiếm thông tin, Thống kê.

* Lưu trữ thông tin: **Nhân viên, Khách hàng, Vật liệu, Nhà cung cấp, Hợp đồng, Hóa đơn, Nhập kho** (Thêm, xóa, chỉnh sửa)
* Chức năng **Thêm** sẽ cho chúng ta tạo mới một trường thông tin bao gồm các thuộc tính tương ứng với mỗi bảng Nhân viên, Khách hàng, Vật liệu, Nhà cung cấp, Hợp đồng, Hóa đơn, Nhập kho.
* Chức năng **Xóa** sẽ xóa toàn bộ các thuộc tính trong trường thông tin dựa theo khóa chính tương ứng với mỗi bảng được người dùng nhập vào.
* Chức năng **Chỉnh sửa** cho phép chúng ta thay đổi, cập nhật lại thông tin về các thuộc tính trong trường thông tin tương ứng với khóa chính mà chúng ta nhập vào.
* Tìm kiếm thông tin:Cho phép người dùng xem lại dữ liệu được lưu trữ trong cơ sở dữ liệu
* Tìm kiếm **nhà cung cấp** bằng mã nhà cung cấp
* Tìm kiếm **nhân viên** bằng tên hoặc mã nhân viên
* Tìm kiếm **khách hàng** bằng tên hoặc mã khách hàng
* Tìm **vật liệu** theo mã vật liệu
* Tìm **hợp đồng** bằng mã số hợp đồng
* Tìm kiếm thông tin **nhập kho** bằng mã vật liệu và khoảng thời gian
* Thống kê: chức năng thống kê được phân quyền chỉ cho phép chủ mới được vào xem
* **Thống kê vật liệu nhập kho mỗi tháng**: sẽ cho biết tổng số lượng vật liệu đã nhập, số lượng chi tiết của từng vật liệu và tổng số tiền đã chi trả cho việc mua vật liệu trong tháng.
* **Doanh thu theo tháng**: là thống kê lại số lượng hợp đồng đã kết thúc trong tháng và tính tổng số tiền của các hợp đồng đó.

1. **Thiết kế cơ sở dữ liệu**
   1. **Thiết kế mức quan điểm**
   2. **Thiết kế ở mức logic**
   3. **Các ràng buộc của dữ liệu**

* Các ràng buộc trong bảng như:
  + NOT NULL: đảm bảo dữ liệu cột không có giá trị NULL
  + DEFAULT: giá trị mặc định của cột nếu khi chèn vào không đưa giá trị
  + CHECK: dữ liệu khi chèn vào cần thỏa mãn điều kiện nào đó
  + PRIMARY KEY: là sự kết hợp của NOT NULL và UNIQUE, giá trị này dùng để xác định các hàng khác nhau của bảng
  + FOREIGN KEY: là giá trị duy nhất tham khảo từ bảng khác
* Các ràng buộc giữa các bảng:
  + Khi bảng NHAPKHO được thêm hay chỉnh sửa vào thì số lượng của vật liệu phải thay đổi theo tương ứng
  + Khi một CHITIET\_HD được thêm hay xóa đi vào thì vật liệu trong kho và giá trị của hợp đồng phải được cập nhật
  + Khi thêm hóa đơn, giá trị của hóa đơn lớn hơn giá trị hợp đồng thì không cho phép thêm vào nhưng nếu bằng nhau thì cập nhật ngày nghiệm thu của hợp đồng và chuyển toàn bộ dữ liệu sang bảng backup
  1. **Cài đặt CSDL và các ràng buộc**
     1. **Tạo bảng và các ràng buộc đơn giản.**
* Bảng User

CREATE TABLE USERS

(

Username VARCHAR(20) CONSTRAINT PK\_USERS PRIMARY KEY,

MaNV char(6) FOREIGN KEY REFERENCES NHANVIEN(NV\_NguoiID),

Pass VARCHAR(20) NOT NULL,

ChucVu NVARCHAR(30) NOT NULL,

)

* Bảng TT\_NGUOI

CREATE TABLE TT\_NGUOI

(

NguoiID CHAR(6) CONSTRAINT PK\_TT\_NGUOI PRIMARY KEY,

Hoten NVARCHAR(30) NOT NULL,

DiaChi NVARCHAR(30),

DienThoai CHAR(11) NOT NULL,

NgaySinh DATE NOT NULL,

CCCD CHAR(11) NOT NULL,

GioiTinh BIT NOT NULL,

)

* Bảng CHUCVU

CREATE TABLE CHUCVU

(

MaCV CHAR(6) CONSTRAINT PK\_CHUCVU PRIMARY KEY,

TenCV NVARCHAR(20) NOT NULL,

)

* Bảng NHANVIEN

CREATE TABLE NHANVIEN

(

NV\_NguoiID CHAR(6) CONSTRAINT PK\_NHANVIEN PRIMARY KEY FOREIGN KEY REFERENCES TT\_NGUOI(NguoiID) ON DELETE CASCADE,

MaCV CHAR(6) NOT NULL FOREIGN KEY REFERENCES CHUCVU(MaCV),

Luong DECIMAL NOT NULL CHECK(Luong>0),

)

* Bảng KHACHHANG

CREATE TABLE KHACHHANG

(

KH\_NguoiID CHAR(6) CONSTRAINT PK\_KHACHHANG PRIMARY KEY FOREIGN KEY REFERENCES TT\_NGUOI(NguoiID) ON DELETE CASCADE,

)

* Bảng VATLIEU

CREATE TABLE VATLIEU

(

MaVL CHAR(6) CONSTRAINT PK\_VATLIEU PRIMARY KEY,

TenVL NVARCHAR(20) NOT NULL,

SoLuong int NOT NULL

)

* Bảng CONGVIEC

CREATE TABLE CONGVIEC

(

MaCViec CHAR(6) CONSTRAINT PK\_CONGVIEC PRIMARY KEY,

NoiDungCV NVARCHAR(40) NOT NULL,

TienCong DECIMAL NOT NULL CHECK(TienCong>0),

VatLieu char(6) FOREIGN KEY REFERENCES VATLIEU(MaVL)

)

* Bảng HOPDONG

REATE TABLE HOPDONG

(

SoHD CHAR(15) CONSTRAINT PK\_HOPDONG PRIMARY KEY,

NgayHD DATE DEFAULT(GETDATE()),

KH\_NguoiID CHAR(6) NOT NULL FOREIGN KEY REFERENCES KHACHHANG(KH\_NguoiID),

SoXE CHAR(10) NOT NULL,

TriGiaHD INT DEFAULT(0),

NgayGiaoDuKien DATE,

NgayNghiemThu DATE

)

* Bảng HOPDONG\_BACKUP

CREATE TABLE HOPDONG\_BACKUP

(

SoHD CHAR(15) CONSTRAINT PK\_HOPDONG\_BU PRIMARY KEY,

NgayHD DATE,

KH\_NguoiID CHAR(6),

SoXE CHAR(10) NOT NULL,

TriGiaHD INT,

NgayGiaoDuKien DATE,

NgayNghiemThu DATE

)

* Bảng HOADON

CREATE TABLE HOADON

(

MaHoaDon CHAR(15) CONSTRAINT PK\_HOADON PRIMARY KEY,

NgayLapHoaDon DATE DEFAULT (GETDATE()),

MaHopDong CHAR(15) NOT NULL FOREIGN KEY REFERENCES HOPDONG(SoHD),

KH\_NguoiID CHAR(6) NOT NULL FOREIGN KEY REFERENCES KHACHHANG(KH\_NguoiID),

HoTen NVARCHAR(40) NOT NULL,

SoTienThu INT NOT NULL CHECK(SoTienThu>0),

CONSTRAINT fk\_HDon\_SoHD FOREIGN KEY (MaHopDong) REFERENCES HOPDONG (SoHD)ON DELETE CASCADE

)

* Bảng CHITIET\_HD

CREATE TABLE CHITIET\_HD

(

SoHD CHAR(15) FOREIGN KEY REFERENCES HOPDONG(SoHD),

MaCV CHAR(6) NOT NULL FOREIGN KEY REFERENCES CONGVIEC(MaCViec),

TriGiaCV INT NOT NULL CHECK(TriGiaCV>0),

MaNV CHAR(6) NOT NULL FOREIGN KEY REFERENCES NHANVIEN(NV\_NguoiID),

CONSTRAINT PK\_CHITIET\_HD PRIMARY KEY(SoHD,MaCV),

CONSTRAINT fk\_CTHD\_SoHD FOREIGN KEY (SoHD) REFERENCES HOPDONG (SoHD)ON DELETE CASCADE

)

* Bảng CHITIET\_HD\_BACKUP

CREATE TABLE CHITIET\_HD\_BACKUP

(

SoHD CHAR(15) FOREIGN KEY REFERENCES HOPDONG\_BACKUP(SoHD) ON DELETE CASCADE,

MaCV CHAR(6) NOT NULL,

TriGiaCV INT NOT NULL,

MaNV CHAR(6) NOT NULL,

CONSTRAINT PK\_CHITIET\_HD\_BU PRIMARY KEY(SoHD,MaCV)

)

* Bảng NHACUNGCAP

CREATE TABLE NHACUNGCAP

(

MaNhaCC CHAR(6) CONSTRAINT PK\_NHACUNGCAP PRIMARY KEY,

TenNhaCC NVARCHAR(30) NOT NULL,

DienThoai CHAR(11),

DiaChi NVARCHAR(30),

)

* Bảng NHAPKHO

CREATE TABLE NHAPKHO

(

MaNKho CHAR(15) CONSTRAINT PK\_NHAPKHO PRIMARY KEY,

MaVL CHAR(6) FOREIGN KEY REFERENCES VATLIEU(MaVL),

MaNhaCC CHAR(6) NOT NULL FOREIGN KEY REFERENCES NHACUNGCAP(MaNhaCC),

SoLuong INT,

GiaTri DECIMAL,

NgayNhap DATE DEFAULT(GETDATE()),

MaNV CHAR(6) FOREIGN KEY REFERENCES NHANVIEN(NV\_NguoiID)

)

* Bảng NHAPKHO\_BACKUP

CREATE TABLE NHAPKHO\_BACKUP

(

MaNKho CHAR(15) CONSTRAINT PK\_NHAPKHO\_BACKUP PRIMARY KEY,

MaVL CHAR(6),

MaNhaCC CHAR(6),

SoLuong INT,

GiaTri DECIMAL,

NgayNhap DATE,

MaNV CHAR(6)

)

* + 1. **Các VIEW**
* Cở sở dữ liệu có 4 view gồm: VIEW\_NHAPKHO, VIEW\_CVIEC, VIEW\_NV, VIEW\_KH
* Các view này kết hợp nhiều bảng với nhau lại để có thể hiện rõ ràng tên của đối tượng thay vì chỉ là mã số
* Code

-----View kết hợp bảng nhập kho với bảng nhà cung cấp và bảng vật liệu để lấy tên của cả hai

CREATE VIEW VIEW\_NHAPKHO AS

SELECT NHAPKHO.MaNKho, VATLIEU.TenVL, NHACUNGCAP.TenNhaCC, NHAPKHO.SoLuong, NHAPKHO.GiaTri, NHAPKHO.NgayNhap, NHAPKHO.MaNV

FROM NHAPKHO,VATLIEU, NHACUNGCAP

WHERE NHAPKHO.MaVL=VATLIEU.MaVL AND NHAPKHO.MaNhaCC=NHACUNGCAP.MaNhaCC

-----View kết hợp bảng công việc với bảng vật liệu để lấy tên của vật liệu có trong công việc

CREATE VIEW VIEW\_CVIEC AS

SELECT MaCViec, NoiDungCV,TienCong,TenVL

FROM CONGVIEC,VATLIEU

WHERE CONGVIEC.VatLieu=VATLIEU.MaVL

-----View kết hợp bảng thông tin người với bảng nhân viên để lấy toàn bộ thông tin của các nhân viên

CREATE VIEW VIEW\_NV AS

SELECT dbo.NHANVIEN.NV\_NguoiID, dbo.TT\_NGUOI.Hoten, dbo.TT\_NGUOI.DiaChi, dbo.TT\_NGUOI.DienThoai, dbo.TT\_NGUOI.NgaySinh, dbo.TT\_NGUOI.CCCD, dbo.TT\_NGUOI.GioiTinh, dbo.CHUCVU.TenCV, dbo.NHANVIEN.Luong

FROM dbo.NHANVIEN INNER JOIN dbo.TT\_NGUOI ON dbo.NHANVIEN.NV\_NguoiID = dbo.TT\_NGUOI.NguoiID INNER JOIN dbo.CHUCVU ON dbo.NHANVIEN.MaCV = dbo.CHUCVU.MaCV

-----View kết hợp bảng thông tin người với bảng khách hàng để lấy toàn bộ thông tin của các khách hàng

CREATE VIEW VIEW\_KH AS

SELECT dbo.KHACHHANG.KH\_NguoiID, dbo.TT\_NGUOI.Hoten, dbo.TT\_NGUOI.DiaChi, dbo.TT\_NGUOI.DienThoai, dbo.TT\_NGUOI.NgaySinh, dbo.TT\_NGUOI.CCCD, dbo.TT\_NGUOI.GioiTinh

FROM dbo.KHACHHANG INNER JOIN dbo.TT\_NGUOI ON dbo.KHACHHANG.KH\_NguoiID = dbo.TT\_NGUOI.NguoiID

* + 1. **Các trigger để đảm bảo các ràng buộc phức tạp**
* Trigger của bảng NHAPKHO tác động lên bảng VATLIEU
  + Khi thêm một hàng thì số lượng vật liệu phải tăng lên

CREATE TRIGGER TRG\_UPDATE\_VL ON NHAPKHO

AFTER INSERT AS

BEGIN

DECLARE @MaVL CHAR(6),@SL INT

SELECT @MaVL=MaVL,@SL=SoLuong FROM INSERTED

UPDATE VATLIEU

SET SoLuong = SoLuong + @SL

WHERE MaVL=@MaVL

END

* + Khi cập nhật một hàng nếu mã vật liệu không thay đổi thì chỉ cần cập nhật lại số lương của vật liệu đó còn nếu không bằng nhau thì cập nhật số lượng của cả hai vật liệu có liên quan.

CREATE TRIGGER TRG\_UPDATE\_VL\_3 ON NHAPKHO

AFTER UPDATE AS

BEGIN

DECLARE @MaVL CHAR(6),@SL INT

SELECT @MaVL=MaVL,@SL=SoLuong FROM INSERTED

DECLARE @MaVL\_DEL CHAR(6),@SL\_DEL INT

SELECT @MaVL\_DEL=MaVL,@SL\_DEL=SoLuong FROM deleted

IF(@MaVL = @MaVL\_DEL)

BEGIN

UPDATE VATLIEU

SET SoLuong = SoLuong + @SL - @SL\_DEL

WHERE MaVL=@MaVL

END

ELSE

BEGIN

UPDATE VATLIEU

SET SoLuong = SoLuong - @SL\_DEL

WHERE MaVL=@MaVL\_DEL

UPDATE VATLIEU

SET SoLuong = SoLuong + @SL

WHERE MaVL=@MaVL

END

END

* Trigger của bảng CHITIET\_HD tác động lên bảng VATLIEU
  + Khi thêm thì số lượng của vật liệu phải giảm đi 1

CREATE TRIGGER TRG\_UPDATE\_VL\_2 ON CHITIET\_HD

AFTER INSERT AS

BEGIN

DECLARE @MaCV CHAR(6),@MaVL CHAR(6) SELECT @MaCV=MaCV FROM INSERTED SELECT @MAVL=VatLieu FROM CONGVIEC WHERE MaCViec=@MaCV

UPDATE VATLIEU

SET SoLuong=SoLuong-1

WHERE MaVL=@MaVL

END

* + Khi xóa một chi tiết của một hợp đồng chưa được nghiệm thu thì số lượng của vật liệu cần phải được hoàn lại

CREATE TRIGGER TRG\_UPDATE\_VL\_4 ON CHITIET\_HD

AFTER DELETE AS

BEGIN

DECLARE @SoHD CHAR(15),@MaCV CHAR(6),@MaVL CHAR(6),@NgayNghiemThu DATE

SELECT @SoHD=SoHD FROM DELETED

SELECT @NgayNghiemThu=NgayNghiemThu

FROM HOPDONG\_BACKUP WHERE SoHD=@SoHD

SELECT @MaCV=MaCV FROM DELETED

SELECT @MaVL=VatLieu FROM CONGVIEC WHERE MaCViec=@MaCV

IF(@NgayNghiemThu is NULL)

BEGIN

UPDATE VATLIEU

SET SoLuong=SoLuong+1

WHERE MaVL=@MaVL

END

END

* Trigger của bảng CHITIET\_HD tác động lên bảng HOPDONG

Khi thêm hay xóa một chi tiết thì giá trị hợp đồng sẽ thay đổi theo

CREATE TRIGGER TRG\_UPDATE\_HDONG ON CHITIET\_HD

AFTER INSERT, UPDATE AS

BEGIN

DECLARE @SoHD CHAR(15),@TriGiaCV int

SELECT @SoHD=SoHD,@TriGiaCV=TriGiaCV FROM INSERTED

SELECT @TriGiaCV=SUM(TriGiaCV) FROM CHITIET\_HD WHERE SoHD=@SoHD

UPDATE HOPDONG

SET TriGiaHD = @TriGiaCV

WHERE SoHD = @SoHD

END

CREATE TRIGGER TRG\_UPDATE\_HDONG\_1 ON CHITIET\_HD

AFTER DELETE AS

BEGIN

DECLARE @SoHD CHAR(15),@TriGiaCV int

SELECT @SoHD=SoHD,@TriGiaCV=TriGiaCV FROM DELETED

UPDATE HOPDONG

SET TriGiaHD = TriGiaHD - @TriGiaCV

WHERE SoHD = @SoHD

END

* Trigger của bảng HOADON tác động lên bảng HOPDONG

Khi thêm một hóa vào nếu tổng hóa đơn bằng giá trị hợp đồng thì chuyển toàn bọ dữ liệu của hợp đồng sang bảng backup

CREATE TRIGGER TRG\_INSERT\_HOADON ON HOADON

AFTER INSERT AS

DECLARE @SoHD CHAR(15),@TongTien INT,@TriGiaHD INT

SELECT @SoHD=MaHopDong FROM INSERTED

SELECT @TongTien=sum(SoTienThu) FROM HOADON WHERE MaHopDong=@SoHD

SELECT @TriGiaHD=TriGiaHD FROM HOPDONG WHERE SoHD=@SoHD

IF(@TongTien=@TriGiaHD)

BEGIN

UPDATE HOPDONG

SET NgayNghiemThu=GETDATE()

WHERE SoHD=@SoHD;

INSERT INTO HOPDONG\_BACKUP

SELECT \*

FROM HOPDONG

WHERE HOPDONG.SoHD=@SoHD;

INSERT INTO CHITIET\_HD\_BACKUP

SELECT \*

FROM CHITIET\_HD

WHERE CHITIET\_HD.SoHD=@SoHD;

DELETE HOPDONG WHERE SoHD=@SoHD

END

1. **Thiết kế chức năng**
   1. **Xuất dữ liêu ra fom**

* Xuất cho bảng nhân viên

-----Xuất thông tin của toàn bộ nhân viên thông qua view đã tạo trước

CREATE PROC XUAT\_NV

as

SELECT \* FROM VIEW\_NV

* Xuất cho bảng khách hàng

----- Xuất thông tin của toàn bộ khách hàng thông qua view đã tạo trước

CREATE PROC XUAT\_KH

as

* Xuất cho bảng vật liệu

-----Xuất toàn bộ vật liệu

CREATE PROC XUAT\_VL

AS

SELECT \* FROM VATLIEU

* Xuất cho bảng nhà cung cấp

-----Xuất toàn bộ nhà sản xuất

CREATE PROC XUAT\_NCC

as

SELECT \* FROM NHACUNGCAP

GO

* Xuất cho bảng công việc

CREATE FUNCTION XUAT\_CVIEC() RETURNS table

as

RETURN (SELECT \* FROM VIEW\_CVIEC)

* Xuất cho bảng hợp đồng

-----Xuất tất cả các bảng hợp đồng đang trong giai đoạn thực hiện

CREATE FUNCTION XUAT\_HDONG() RETURNS table

as

RETURN (SELECT \* FROM HOPDONG)

* Xuất cho bảng nhập kho chính

-----Xuất thông tin nhập kho có trong bảng chính thông qua VIEW\_NHAPKHO

CREATE FUNCTION XUAT\_NHAPKHO() RETURNS table

as

RETURN (SELECT \* FROM VIEW\_NHAPKHO)

GO

* Xuất cho bảng hóa đơn

---Tìm hóa đơn theo mã Hợp đồng

CREATE FUNCTION XUAT\_HOADON(@soHD CHAR(15)) RETURNS TABLE

AS

RETURN(SELECT \* FROM HOADON WHERE MaHopDong=@soHD)

* 1. **Tìm kiếm trong các bảng**
* Bảng nhân viên

-----Tìm nhân viên theo tên

GO

CREATE FUNCTION TIM\_TEN\_NV(@name NVARCHAR(50)) RETURNS TABLE

AS

RETURN(SELECT \* FROM VIEW\_NV

WHERE Hoten LIKE N'%' + @name + '%')

GO

-----Tìm nhân viên theo mã số

CREATE FUNCTION TIM\_MS\_NV(@maNV CHAR(6)) RETURNS TABLE

AS

RETURN(SELECT \* FROM VIEW\_NV

WHERE NV\_NguoiID LIKE '%' + @maNV + '%')

GO

-----Lấy các nhân viên cùng một chức vụ

CREATE FUNCTION XUAT\_NV\_CHUCVU(@machucvu CHAR(6)) RETURNS table

as

RETURN (SELECT \* FROM VIEW\_NV WHERE NV\_NguoiID in (SELECT NV\_NguoiID FROM NHANVIEN WHERE MaCV=@machucvu))

* Bảng khách hàng

-----Tìm khách hàng theo tên

GO

CREATE FUNCTION TIM\_TEN\_KH(@name NVARCHAR(50)) RETURNS TABLE

AS

RETURN(SELECT \* FROM VIEW\_KH

WHERE Hoten LIKE N'%' + @name + '%')

GO

-----Tìm khách hàng theo mã số

CREATE FUNCTION TIM\_MS\_KH(@maKH CHAR(6)) RETURNS TABLE

AS

RETURN(SELECT \* FROM VIEW\_KH

WHERE KH\_NguoiID LIKE N'%' + @maKH + '%')

* Bảng vật liệu

-----Tìm vật liệu theo tên

GO

CREATE FUNCTION TIM\_TEN\_VL(@name NVARCHAR(50)) RETURNS TABLE

AS

RETURN(SELECT \* FROM VATLIEU

WHERE TenVL LIKE N'%' + @name + '%')

GO

-----Tìm vật liệu theo mã số

CREATE FUNCTION TIM\_MS\_VL(@maVL CHAR(6)) RETURNS TABLE

AS

RETURN(SELECT \* FROM VATLIEU

WHERE MaVL LIKE N'%' + @maVL + '%')

* Bảng nhà cung cấp

-----Tìm nhà cung cấp theo tên

CREATE FUNCTION TIM\_TEN\_NCC(@name NVARCHAR(50)) RETURNS TABLE

AS

RETURN(SELECT \* FROM NHACUNGCAP

WHERE TenNhaCC LIKE N'%' + @name + '%')

GO

-----Tìm nhà cung cấp theo mã số

CREATE FUNCTION TIM\_MS\_NCC(@maNCC CHAR(6)) RETURNS TABLE

AS

RETURN(SELECT \* FROM NHACUNGCAP

WHERE MaNhaCC LIKE N'%' + @maNCC + '%')

* Bảng hợp đồng

-----Tìm kiếm các hợp đồng đang thực hiện

CREATE FUNCTION TIMKIEM\_HDONG(@SoHD CHAR(15)) RETURNS table

as

RETURN (SELECT \* FROM HOPDONG WHERE SoHD = @SoHD)

* Bảng chi tiết hợp đồng

-----Lấy chi tiết công việc cho chỉnh sửa hợp đồng đang thực hiện

CREATE FUNCTION CONGVIEC\_CHITIET\_HD(@SoHD CHAR(15)) RETURNS

@congviec table(MaCV CHAR(6),TenCV NVARCHAR(40),MaTho CHAR(6),TenTho NVARCHAR(30))

AS

BEGIN

INSERT @congviec SELECT CHITIET\_HD.MaCV, CONGVIEC.NoiDungCV, CHITIET\_HD.MaNV, TT\_NGUOI.Hoten

FROM CHITIET\_HD, TT\_NGUOI, CONGVIEC

WHERE CHITIET\_HD.MaNV=TT\_NGUOI.NguoiID AND CHITIET\_HD.MaCV= CONGVIEC.MaCViec AND CHITIET\_HD.SoHD=@SoHD

return

END

* Bảng nhập kho

-----Xuất thông tin nhập kho trong một khoảng thời gian

CREATE FUNCTION TIM\_TG\_NHAPKHO(@ngaydautien date,@ngaycuoicung date) RETURNS table

as

RETURN (SELECT \* FROM VIEW\_NHAPKHO WHERE NgayNhap >= @ngaydautien AND NgayNhap <= @ngaycuoicung)

----- Xuất thông tin nhập kho theo vật liệu

CREATE FUNCTION TIM\_VL\_NHAPKHO(@tenVL NVARCHAR(20)) RETURNS table

as

RETURN (SELECT \* FROM VIEW\_NHAPKHO WHERE TenVL = @tenVL)

GO

* 1. **Thêm dữ liệu vào bảng**

Ở các chức năng thêm, xóa và sửa, nhóm dùng thủ tục có tham số để thêm vào, bên trong thủ tục có sử dụng transaction để xử lý và có thêm một biến output để có thể bắt kết quả trả lại thông báo cho người dùng.

* Bảng nhân viên

CREATE proc THEM\_NV

@nguoiid CHAR(6),@hoten nvarchar(30),@diachi nvarchar(30),@dienthoai CHAR(11),

@ngaysinh date,@cccd CHAR(11),@gioitinnh bit,@macv CHAR(6),@luong int,@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO dbo.TT\_NGUOI(NguoiID,Hoten,DiaChi,DienThoai,NgaySinh,CCCD,GioiTinh)

VALUES(@nguoiid, @hoten, @diachi, @dienthoai,@ngaysinh,@cccd,@gioitinnh)

INSERT INTO dbo.NHANVIEN(NV\_NguoiID,MaCV,Luong)

VALUES(@nguoiid,@macv,@luong)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng khách hàng

CREATE PROC THEM\_KH

@nguoiid CHAR(6),@hoten nvarchar(30),@diachi nvarchar(30),@dienthoai CHAR(11),

@ngaysinh date,@cccd CHAR(11),@gioitinh bit,@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO TT\_NGUOI VALUES (@nguoiid, @hoten, @diachi, @dienthoai, @ngaysinh, @cccd, @gioitinh)

INSERT INTO KHACHHANG VALUES(@nguoiid)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng vật liệu

CREATE PROC THEM\_VL

@mavl CHAR(6),@tenvl nvarchar(20),@soluong int,@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO VATLIEU VALUES (@mavl,@tenvl,@soluong)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng nhà cung cấp

CREATE PROC THEM\_NCC

@manhacc CHAR(6),@tennhacc nvarchar(30),@dienthoai CHAR(11),

@diachi nvarchar(30),@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO NHACUNGCAP VALUES (@manhacc, @tennhacc, @dienthoai, @diachi)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng công việc

CREATE PROC THEM\_CVIEC

@macv CHAR(6),@noidungcv nvarchar(40),@tiencong int,

@vatlieu char(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO CONGVIEC VALUES (@macv, @noidungcv, @tiencong, @vatlieu)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng hợp đồng: khi thêm hợp đồng cần đảm bảo ngày giao dự kiến lớn hoặc bằng với ngày lập hợp đồng nếu không sẽ không lập được

CREATE PROC THEM\_HDONG

@SoHD CHAR(15),@KH\_NguoiID CHAR(6),@SoXe CHAR(10),

@NgayGiaoDuKien DATE,@result int output

AS

BEGIN TRY

BEGIN TRAN

INSERT INTO HOPDONG VALUES(@SoHD, DEFAULT, @KH\_NguoiID, @SoXe, DEFAULT, @NgayGiaoDuKien,NULL)

IF(@NgayGiaoDuKien<GETDATE())

BEGIN

ROLLBACK TRAN

set @result=0

END

ELSE

BEGIN

set @result=1

COMMIT TRAN

END

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng chi tiết hợp đồng: một chi tiết hợp đồng khi thêm vào thì cần đảm bảo vật liệu cần cho chi tiết đó còn trong kho vì thế sau khi thêm vào trigger sẽ hoạt động và số lượng của vật liệu giảm xuống âm thì sẽ rollback lại thao tác. Ngoài ra, trị giá của chi tiết cũng sẽ không do người dùng nhập vào mà sẽ được lấy từ bảng công việc thông qua mã công việc.

CREATE PROC THEM\_CHITIET\_HD

@SoHD CHAR(15),@MaCV CHAR(6),@MaNV CHAR(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

DECLARE @TriGiaCV INT

SELECT @TriGiaCV=TienCong FROM CONGVIEC WHERE

MaCViec=@MaCV

INSERT INTO CHITIET\_HD VALUES(@SoHD, @MaCV,

@TriGiaCV, @MaNV)

DECLARE @MaVL CHAR(6),@SoLuong INT

SELECT @MAVL=VatLieu FROM CONGVIEC WHERE

MaCViec=@MaCV

SELECT @SoLuong=SoLuong FROM VATLIEU WHERE

MaVL=@MaVL

IF(@SoLuong<0)

BEGIN

ROLLBACK TRAN

set @result=2

END

ELSE

BEGIN

set @result=1

COMMIT TRAN

END

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng nhập kho

CREATE PROC THEM\_NHAPKHO

@MaNKho CHAR(15),@MaVL CHAR(6),@MaNhaCC CHAR(6),

@SoLuong INT,@GiaTri DECIMAL,@MaNV CHAR(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO NHAPKHO VALUES(@MaNKho, @MaVL, @MaNhaCC, @SoLuong,@GiaTri,DEFAULT,@MaNV)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng hóa đơn: hóa đơn của một hợp đồng được thêm vào phải đảm bảo tổng hóa đơn của hợp đồng đó không lớn hơn giá trị hợp đồng.

CREATE PROC THEM\_HOADON

@mahoadon CHAR(15),@mahdong CHAR(15),@hoten NVARCHAR(40),

@sotienthu INT,@result int output

AS

BEGIN TRAN

BEGIN TRY

DECLARE @idkh CHAR(6),@tiendu int

SELECT @idkh=KH\_NguoiID FROM HOPDONG WHERE

SoHD=@mahdong

INSERT INTO HOADON VALUES(@mahoadon, DEFAULT, @mahdong,@idkh,@hoten,@sotienthu)

EXEC @tiendu=TIEN\_HOADON @mahdong

if(@tiendu < 0)

BEGIN

set @result=2

ROLLBACK TRAN

END

ELSE

BEGIN

set @result=1

COMMIT TRAN

END

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* 1. **Xóa dữ liệu**

Dữ liệu của các bảng được xóa thông qua các khóa chính

* Bảng nhân viên

CREATE PROC XOA\_NV

@nguoiid CHAR(6),

@result int output

as

BEGIN TRAN

BEGIN TRY

DELETE FROM dbo.TT\_NGUOI WHERE NguoiID = @nguoiid;

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng khách hàng

CREATE PROC XOA\_KH

@nguoiid CHAR(6),

@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM TT\_Nguoi

WHERE NguoiID=@nguoiid

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng vật liệu

CREATE PROC XOA\_VL

@mavl CHAR(6),

@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM VATLIEU

WHERE MaVL=@mavl

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng nhà cung cấp

CREATE PROC XOA\_NCC

@manhacc CHAR(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM NHACUNGCAP

WHERE MaNhaCC=@manhacc

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng công việc

CREATE PROC XOA\_CViec

@macv CHAR(6),

@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM DBO.CONGVIEC

WHERE MaCViec=@macv

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng hợp đồng

CREATE PROC XOA\_HDONG

@SoHD CHAR(15),@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM HOPDONG

WHERE SoHD=@SoHD

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng chi tiết hợp đồng

CREATE PROC XOA\_CHITIET\_HD(@SoHD CHAR(15),@MaCV CHAR(6))

AS

DELETE

FROM CHITIET\_HD

WHERE SoHD=@SoHD AND MaCV=

* Bảng nhập kho

-----Xóa nhập kho theo mã

CREATE PROC XOA\_NHAPKHO

@mankho CHAR(15),@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM NHAPKHO

WHERE MaNKho=@mankho

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

-----Xóa nhập kho theo tháng sẽ được chuyển sang bảng nhập kho backup

CREATE PROC XOA\_MONTH\_NHAPKHO

@ngaydautien date,@ngaycuoicung date,@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO NHAPKHO\_BACKUP

SELECT \*

FROM NHAPKHO

WHERE NgayNhap>=@ngaydautien AND NgayNhap <=

@ngaycuoicung

DELETE

FROM NHAPKHO

WHERE NgayNhap>=@ngaydautien AND NgayNhap<=

@ngaycuoicung

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng hóa đơn

CREATE PROC XOA\_HOADON

@mahoadon char(15) ,@result int output

AS

BEGIN TRAN

BEGIN TRY

DELETE

FROM HOADON

WHERE MaHoaDon = @mahoadon

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng backup

CREATE PROC XOA\_DOANHTHU

@ngaydautien date,@ngaycuoicung date

AS

BEGIN

DELETE

FROM HOPDONG\_BACKUP

WHERE NgayNghiemThu>= @ngaydautien AND NgayNghiemThu<=@ngaycuoicung

DELETE

FROM NHAPKHO\_BACKUP

WHERE NgayNhap>=@ngaydautien AND NgayNhap<=@ngaycuoicung

END

* 1. **Sửa dữ liệu trong bảng**
* Bảng nhân viên

CREATE PROC SUA\_NV

@nguoiid CHAR(6),@hoten nvarchar(30),@diachi nvarchar(30),@dienthoai CHAR(11),@ngaysinh date,

@cccd CHAR(11),@gioitinh bit,@macv CHAR(6),@luong INT,@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE dbo.TT\_NGUOI

SET Hoten=@hoten, DiaChi=@diachi, DienThoai=@dienthoai, NgaySinh=@ngaysinh, CCCD=@cccd, GioiTinh=@gioitinh

WHERE NguoiID = @nguoiid;

UPDATE dbo.NHANVIEN

SET Luong=@luong, MaCV=@macv

WHERE NV\_NguoiID=@nguoiid;

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng khách hàng

CREATE PROC SUA\_KH

@nguoiid CHAR(6),@hoten nvarchar(30),@diachi nvarchar(30),@dienthoai CHAR(11),

@ngaysinh date,@cccd CHAR(11),@gioitinh bit,@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE TT\_NGUOI

SET HoTen=@hoten, DiaChi=@DiaChi, DienThoai=@dienthoai, NgaySinh=@ngaysinh, CCCD=@cccd, GioiTinh=@gioitinh

WHERE NguoiID=@nguoiid

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng vật liệu

CREATE PROC SUA\_VL

@mavl CHAR(6),@tenvl nvarchar(20),@soluong int,@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE VATLIEU

SET TenVL=@tenvl,SoLuong=@soluong

WHERE MaVL=@mavl

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng nhà cung cấp

CREATE PROC SUA\_NCC

@manhacc CHAR(6),@tennhacc nvarchar(30),@dienthoai CHAR(11),

@diachi nvarchar(30),@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE NHACUNGCAP

SET TenNhaCC=@tennhacc, DienThoai=@dienthoai,

DiaChi=@diachi

WHERE MaNhaCC=@manhacc

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng công việc

CREATE PROC SUA\_CViec

@macv CHAR(6),@noidungcv nvarchar(40),@tiencong int,

@vatlieu char(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE CONGVIEC

SET NoiDungCV=@noidungcv, TienCong=@tiencong,

VatLieu=@vatlieu

WHERE MaCViec=@macv

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* Bảng nhập kho

CREATE PROC SUA\_NHAPKHO

@MaNKho CHAR(15),@MaVL CHAR(6),@MaNhaCC CHAR(6),

@SoLuong INT,@GiaTri DECIMAL,@MaNV CHAR(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE NHAPKHO

SET MaVL=@MaVL, MaNhaCC=@MaNhaCC, SoLuong=@SoLuong,

GiaTri=@GiaTri, MaNV=@MaNV

WHERE MaNKho=@MaNKho

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* 1. **Xem lại hợp đồng của cả bảng chính và bảng backup**

Chức năng này giúp cho việc kiểm tra một hợp đồng đang thực hiện hay đã hoàn thành. Xuất ra toàn bộ thông tin của hợp đồng và chi tiết hợp đồng

-----Tìm hợp đồng trong cả hai bảng hợp đồng và hợp đồng backup

CREATE FUNCTION XUAT\_HOPDONG\_BACKUP(@soHD CHAR(15)) RETURNS

@table TABLE(SoHD CHAR(15),NgayHD DATE,KH\_NguoiID CHAR(6),SoXE CHAR(10),TriGiaHD INT,NgayGiaoDuKien DATE, NgayNghiemThu DATE)

AS

BEGIN

IF EXISTS(SELECT 1 FROM HOPDONG WHERE SoHD= @soHD)

BEGIN

INSERT @table SELECT \* FROM HOPDONG WHERE SoHD= @soHD

END

ELSE

BEGIN

INSERT @table SELECT \* FROM HOPDONG\_BACKUP WHERE SoHD=

@soHD

END

RETURN

END

-----Lấy chi tiết công việc cho xem hợp đồng kiểm tra ở cả hai bảng chính và bảng backup

CREATE FUNCTION CONGVIEC\_HD(@SoHD CHAR(15)) RETURNS

@congviec table(TenCV NVARCHAR(40),TenTho NVARCHAR(30),TriGiaCV int)

AS

BEGIN

IF EXISTS(SELECT 1 FROM HOPDONG WHERE SoHD= @soHD)

BEGIN

INSERT @congviec SELECT CONGVIEC.NoiDungCV,TT\_NGUOI.Hoten,CHITIET\_HD.TriGiaCV

FROM CHITIET\_HD, TT\_NGUOI, CONGVIEC

WHERE CHITIET\_HD.MaNV=TT\_NGUOI.NguoiID AND CHITIET\_HD.MaCV=CONGVIEC.MaCViec AND CHITIET\_HD.SoHD=@SoHD

END

ELSE

BEGIN

INSERT @congviec SELECT CONGVIEC.NoiDungCV,TT\_NGUOI.Hoten,CHITIET\_HD\_BACKUP.TriGiaCV

FROM CHITIET\_HD\_BACKUP, TT\_NGUOI, CONGVIEC

WHERE CHITIET\_HD\_BACKUP.MaNV=TT\_NGUOI.NguoiID AND CHITIET\_HD\_BACKUP.MaCV=CONGVIEC.MaCViec AND CHITIET\_HD\_BACKUP.SoHD=@SoHD

END

return

END

* 1. **Tính số tiền một hợp đồng phải hoàn thành**

Chức năng này giúp người dùng có thể tạo hóa đơn chính xác hơn khi chương trình đã đưa ra đề xuất cho người dùng về số tiền cần hoàn thành, từ đó sẽ không nhập số tiền quá quy định. Chức năng này nhóm dùng function trả về một giá trị.

CREATE FUNCTION TIEN\_HOADON(@soHD CHAR(15)) RETURNS int

AS

BEGIN

DECLARE @giatriHD int, @tiendathu int

SELECT @giatriHD=TriGiaHD FROM HOPDONG WHERE SoHD = @soHD

SELECT @tiendathu = SUM(SoTienThu) FROM HOADON WHERE MaHopDong=@soHD

RETURN @giatriHD - @tiendathu

END

* 1. **Thống kế doang thu và số lượng nhập kho của một tháng**

Chức năng này giúp người dùng có thể thống kê xem lại các hợp đồng đã hoàn thành và số lượng nhập kho trong một khoảng thời gian nhất định.

* Xuất ra dữ liệu

-----Xuất nhập kho backup theo tháng

CREATE FUNCTION XUAT\_NHAPKHO\_BACKUP(@ngaydautien date,@ngaycuoicung date) RETURNS

@table TABLE(MaNKho CHAR(15),MaVL CHAR(6),MaNhaCC CHAR(6),SoLuong INT,GiaTri DECIMAL,NgayNhap DATE,MaNV CHAR(6))

AS

BEGIN

INSERT @table SELECT \*

FROM NHAPKHO\_BACKUP

WHERE NgayNhap>=@ngaydautien AND NgayNhap<=@ngaycuoicung

RETURN

END

-----Xuất các hợp đồng có trong bảng backup theo tháng

CREATE FUNCTION XUAT\_DOANHTHU(@ngaydautien date,@ngaycuoicung date)

RETURNS

@doanhthu table(sohd CHAR(15),ngayHD date,ngaynghiemthu date, giatriHD int)

AS

BEGIN

INSERT @doanhthu SELECT SoHD,NgayHD,NgayNghiemThu,TriGiaHD

FROM HOPDONG\_BACKUP

WHERE NgayNghiemThu>=@ngaydautien AND

NgayNghiemThu<=@ngaycuoicung

return

END

* Tính tổng tiền của các hợp đồng

CREATE FUNCTION DOANHTHU(@ngaydautien date,@ngaycuoicung date) RETURNS INT

AS

BEGIN

DECLARE @doanhthu int

SELECT @doanhthu=sum(TriGiaHD) FROM HOPDONG\_BACKUP WHERE NgayNghiemThu>=@ngaydautien AND NgayNghiemThu<=@ngaycuoicung

return @doanhthu

END

* Tính tổng tiền đã nhập kho

CREATE FUNCTION TONGTIEN\_NHAPKHO(@ngaydautien date,@ngaycuoicung date) RETURNS INT

AS

BEGIN

DECLARE @doanhthu int

SELECT @doanhthu=sum(GiaTri\*SoLuong) FROM NHAPKHO\_BACKUP WHERE NgayNhap>=@ngaydautien AND NgayNhap<=@ngaycuoicung

return @doanhthu

END

1. **Tạo user và phân quyền**

* Đầu tiên ta sẽ tạo hai role là Managers và Member

Exec sp\_addrole 'Managers'

Exec sp\_addrole 'Members'

* Gán các quyền có thể thực hiện cho cả hai role, trong đó các thao tác tương tác với các bảng trong cơ sở dữ liệu chủ yếu do được thực hiện thông qua hàm và thủ tục nên chúng ta sẽ gán quyền thực thi của hàm và thủ tục cho từng role.
  + Role Managers được gán cho tất cả các hàm và thủ tục có trong cơ sở dữ liệu có nghĩa là Managers sẽ được thực hiện hoàn toàn các chức năng có trong ứng dụng.

GRANT EXEC ON XUAT\_NV to Managers

GRANT SELECT ON XUAT\_NV\_CHUCVU to Managers

GRANT SELECT ON TIM\_TEN\_NV to Managers

GRANT SELECT ON TIM\_MS\_NV to Managers

GRANT EXEC ON XUAT\_KH to Managers

GRANT SELECT ON TIM\_TEN\_KH to Managers

GRANT EXEC ON XUAT\_CHUCVU to Managers

GRANT EXEC ON XUAT\_VL to Managers

GRANT SELECT ON TIM\_TEN\_VL to Managers

GRANT SELECT ON TIM\_MS\_VL to Managers

GRANT EXEC ON XUAT\_NCC to Managers

GRANT SELECT ON TIM\_TEN\_NCC to Managers

GRANT SELECT ON TIM\_MS\_NCC to Managers

GRANT SELECT ON XUAT\_CVIEC to Managers

GRANT SELECT ON XUAT\_HDONG to Managers

GRANT SELECT ON XUAT\_NHAPKHO to Managers

GRANT SELECT ON XUAT\_NHAPKHO\_BACKUP to Managers

GRANT SELECT ON TIM\_TG\_NHAPKHO to Managers

GRANT SELECT ON TIM\_VL\_NHAPKHO to Managers

GRANT SELECT ON XUAT\_DOANHTHU to Managers

GRANT SELECT ON XUAT\_HOADON to Managers

GRANT SELECT ON XUAT\_HOPDONG\_BACKUP to Managers

GRANT SELECT ON CONGVIEC\_CHITIET\_HD to Managers

GRANT SELECT ON CONGVIEC\_HD to Managers

GRANT SELECT ON TIMKIEM\_HDONG to Managers

GO

GRANT EXEC ON THEM\_NV to Managers

GRANT EXEC ON THEM\_KH to Managers

GRANT EXEC ON THEM\_VL to Managers

GRANT EXEC ON THEM\_NCC to Managers

GRANT EXEC ON THEM\_CVIEC to Managers

GRANT EXEC ON THEM\_USER to Managers

GRANT EXEC ON THEM\_HDONG to Managers

GRANT EXEC ON THEM\_CHITIET\_HD to Managers

GRANT EXEC ON THEM\_NHAPKHO to Managers

GRANT EXEC ON THEM\_HOADON to Managers

GO

GRANT EXEC ON XOA\_NV to Managers

GRANT EXEC ON XOA\_KH to Managers

GRANT EXEC ON XOA\_VL to Managers

GRANT EXEC ON XOA\_NCC to Managers

GRANT EXEC ON XOA\_CViec to Managers

GRANT EXEC ON XOA\_HDONG to Managers

GRANT EXEC ON XOA\_CHITIET\_HD to Managers

GRANT EXEC ON XOA\_NHAPKHO to Managers

GRANT EXEC ON XOA\_MONTH\_NHAPKHO to Managers

GRANT EXEC ON XOA\_DOANHTHU to Managers

GRANT EXEC ON XOA\_HOADON to Managers

GO

GRANT EXEC ON SUA\_NV to Managers

GRANT EXEC ON SUA\_KH to Managers

GRANT EXEC ON SUA\_VL to Managers

GRANT EXEC ON SUA\_NCC to Managers

GRANT EXEC ON SUA\_CViec to Managers

GRANT EXEC ON SUA\_USERS to Managers

GRANT EXEC ON SUA\_NHAPKHO to Managers

GO

GRANT EXEC ON TIEN\_HOADON to Managers

GO

GRANT EXEC ON DOANHTHU to Managers

GRANT EXEC ON TONGTIEN\_NHAPKHO to Managers

GRANT SELECT ON XUAT\_DOANHTHU to Managers

* + Role Members sẽ bị giới hạn quyền thực hiện các chức năng trên ứng dụng, trong đó Members được xem các dữ liệu cơ bảng nhưng ko có quyền xóa hay sửa trên bảng đó. Members chỉ được toàn quyền trên một số bảng.

GRANT EXEC ON XUAT\_NV to Members

GRANT SELECT ON XUAT\_NV\_CHUCVU to Members

GRANT SELECT ON TIM\_TEN\_NV to Members

GRANT SELECT ON TIM\_MS\_NV to Members

GRANT EXEC ON XUAT\_KH to Members

GRANT SELECT ON TIM\_TEN\_KH to Members

GRANT SELECT ON TIM\_MS\_KH to Members

GRANT EXEC ON XUAT\_CHUCVU to Members

GRANT EXEC ON XUAT\_VL to Members

GRANT SELECT ON TIM\_TEN\_VL to Members

GRANT SELECT ON TIM\_MS\_VL to Members

GRANT EXEC ON XUAT\_NCC to Members

GRANT SELECT ON TIM\_TEN\_NCC to Members

GRANT SELECT ON TIM\_MS\_NCC to Members

GRANT SELECT ON XUAT\_CVIEC to Members

GRANT SELECT ON XUAT\_HDONG to Members

GRANT SELECT ON XUAT\_NHAPKHO to Members

GRANT SELECT ON XUAT\_NHAPKHO\_BACKUP to Members

GRANT SELECT ON TIM\_TG\_NHAPKHO to Members

GRANT SELECT ON TIM\_VL\_NHAPKHO to Members

GRANT SELECT ON XUAT\_HOADON to Members

GRANT SELECT ON XUAT\_HOPDONG\_BACKUP to Members

GRANT SELECT ON CONGVIEC\_HD to Members

GRANT SELECT ON CONGVIEC\_CHITIET\_HD to Managers

GO

GRANT EXEC ON THEM\_KH to Members

GRANT EXEC ON THEM\_HDONG to Members

GRANT EXEC ON THEM\_CHITIET\_HD to Members

GRANT EXEC ON THEM\_NHAPKHO to Members

GRANT EXEC ON THEM\_HOADON to Members

GO

GRANT EXEC ON XOA\_KH to Members

GRANT EXEC ON XOA\_HDONG to Members

GRANT EXEC ON XOA\_CHITIET\_HD to Members

GRANT EXEC ON XOA\_NHAPKHO to Members

GRANT EXEC ON XOA\_MONTH\_NHAPKHO to Members

GRANT EXEC ON XOA\_HOADON to Members

GO

GRANT EXEC ON SUA\_KH to Members

GRANT EXEC ON SUA\_NHAPKHO to Members

GO

GRANT SELECT ON TIMKIEM\_HDONG to Members

GRANT EXEC ON TIEN\_HOADON to Members

* Thêm user và tạo login:
  + Ở bảng user sau khi thêm user vào bảng ta sẽ tạo một tài khoản trong cơ sở dữ liệu và phần quyền ngay cho user đó theo chức vụ của người đó bằng cách gán user mới vừa tạo cho một đang.

CREATE PROC THEM\_USER

@Username VARCHAR(20),@Pass VARCHAR(20),@Chucvu NVARCHAR(30),@manv CHAR(6),@result int output

AS

BEGIN TRAN

BEGIN TRY

INSERT INTO USERS

VALUES(@Username,@Pass,@Chucvu,@result)

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCHc

* + Khi một user được thêm vào bảng USER thì trigger trên bảng sẽ hoạt động để tạo ra một login tương ứng

CREATE TRIGGER CREATR\_LOGIN ON USERS

AFTER INSERT AS

DECLARE @username VARCHAR(20),@pass VARCHAR(20), @chucvu NVARCHAR(30)

SELECT @username = Username, @pass=Pass, @chucvu=ChucVu FROM INSERTED

DECLARE @t NVARCHAR(100)

SET @t = N'CREATE LOGIN ' + QUOTENAME(@Username) + ' WITH PASSWORD = ''' + @Pass +''''

EXEC(@t)

SET @t = N'CREATE USER' + QUOTENAME(@Username) + 'FOR LOGIN' + QUOTENAME(@Username)

EXEC(@t)

IF(@Chucvu=N'Quản lý')

BEGIN

EXEC sp\_addrolemember 'Managers', @Username

END

ELSE

BEGIN

EXEC sp\_addrolemember 'Members', @Username

END

* Sửa user:
  + Thủ tục giúp ập nhật lại mật khẩu cho tài khoản

CREATE PROC SUA\_USERS

@Username VARCHAR(20),@Pass VARCHAR(20),@result int output

AS

BEGIN TRAN

BEGIN TRY

UPDATE USERS

SET Pass=@Pass

WHERE Username=@Username

set @result=1

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

set @result=0

END CATCH

* + Sau khi cập nhật thì trigger update trên bảng user sẽ hoạt động và thay đổi mật khẩu của login của user đã cập nhật

CREATE TRIGGER ALTER\_LOGIN ON USERS

AFTER UPDATE AS

DECLARE @username VARCHAR(20),@pass VARCHAR(20)

SELECT @username = Username, @pass=Pass FROM INSERTED

DECLARE @t NVARCHAR(100)

SET @t = N'ALTER LOGIN ' + QUOTENAME(@Username) + ' WITH PASSWORD = ''' + @Pass +''''

EXEC(@t)