

--	--	--

Đặc tả CSDL: Phần mềm Quản lý hợp đồng môi trường

I. ĐẶC TẢ

Phần mềm Quản lý hợp đồng môi trường là một phần mềm giúp quản lý các mẫu thử của các công ty đối tác, việc quản lý diễn ra như sau:

1. **Nhân viên** của công ty sẽ là người trực tiếp sử dụng phần mềm, các nhân viên của công ty sẽ có những thông tin sau: **Mã nhân viên** là mã định danh duy nhất cho mỗi nhân viên, **họ và tên, ngày sinh, chức vụ** của nhân viên (ví dụ: quan trắc viên, tư vấn viên,...), **số điện thoại** liên lạc của nhân viên, địa chỉ **email** dùng để liên lạc và thông báo hoạt động. Mỗi nhân viên trong công ty sẽ do một người **quản lý** có nhiệm vụ quản lý các hoạt động của họ, mỗi người quản lý sẽ quản lý nhiều nhân viên trong bộ phận của họ. Người quản lý có các thông tin sau: **Mã quản lý** là mã định danh của quản lý tách biệt với mã nhân viên, **họ và tên, mã nhân viên** là mã của nhân viên mà họ đang quản lý, **số điện thoại, email** để liên lạc dễ dàng..
2. **Khách hàng** của công ty là những công ty đối tác gửi mẫu thử đến để quan trắc. Khách hàng có thông tin như sau: Mỗi khách hàng khi ký kết với công ty sẽ có **Mã khách hàng** sẽ được ghi nhận lại và dùng cho cả những lần hợp tác tiếp theo, **tên công ty, địa chỉ, người đại diện, mã hợp đồng** để hợp đồng liên kết với khách hàng cho mỗi lần hợp tác, **số điện thoại, email** để liên lạc với khách hàng.
3. Khách hàng gửi mẫu quan trắc tới sẽ ký kết với công ty các bản **hợp đồng**, mỗi khách hàng có thể ký kết được nhiều hợp đồng cùng một thời điểm, một hợp đồng có hiệu lực khi chỉ một khách hàng đồng ý ký kết. Hợp đồng sẽ có các thông tin sau: **Mã hợp đồng** là mã định danh duy nhất của từng hợp đồng đã ký với mỗi khách hàng, **mã khách hàng** đã ký kết hợp đồng đó, **mã nhân viên** là mã của nhân viên đã được giao phụ trách, **ngày lập** ghi nhận ngày lập hợp đồng, **ngày trả** ghi lại ngày dự kiến hoàn thành, **việc cần làm** là danh sách các công việc cần thực hiện theo hợp đồng, **quý** cho biết hợp đồng thuộc quý nào trong năm và **trạng thái** mô tả tình trạng hiện tại của hợp đồng như đang xử lý, hoàn thành, hoặc trễ hạn. Một hợp đồng sẽ do một nhóm nhân viên phụ trách xử lý và một nhân viên có thể đảm nhiệm nhiều hợp đồng cùng một lúc.

	1	
--	---	--

--	--	--

4. Các hợp đồng sau khi ký kết sẽ được gửi về phía công ty quan trắc kèm với các **mẫu quan trắc**, trong một hợp đồng ký kết sẽ có thể có nhiều mẫu mẫu quan trắc và mỗi mẫu quan trắc chỉ thuộc về một hợp đồng duy nhất. Thông tin của mẫu quan trắc bao gồm: **Mã mẫu quan trắc** là mã định danh duy nhất của mỗi mẫu quan trắc cho dù có cùng loại hay cùng công ty cũng sẽ không bị trùng lặp, **tên mẫu quan trắc**, **mã nhân viên** trực tiếp phụ trách và quan trắc mẫu quan trắc, **mã hợp đồng** mà mẫu quan trắc thuộc về, **ngày lấy** là ngày mẫu quan trắc được gửi tới phòng và được bắt đầu quan trắc và **nội dung, kết quả** chi tiết của mẫu quan trắc, **ngày trả** là ngày mẫu quan trắc được kiểm nghiệm xong. Sau khi đã được gửi về thì nhân viên quan trắc sẽ phụ trách các mẫu thử đó, mỗi mẫu thử sẽ được nhiều nhân viên tham gia xử lý và một nhân viên có thể phụ trách nhiều mẫu thử cùng lúc.

5. Sau khi đã quan trắc thành công thì sẽ trả về một **phiếu kết quả**, một phiếu kết quả sẽ chỉ là kết quả của một mẫu thử. Phiếu kết quả có các thông tin sau: **Mã phiếu kết quả** sẽ được đánh số tự động bắt đầu từ 1, 2, 3, ... , **mã mẫu** sau khi quan trắc, **mã hợp đồng** mà mẫu thử thuộc về, **ngày lấy** là ngày mẫu được gửi tới, **ngày trả kết quả** sẽ ghi nhận lại ngày mà mẫu thử đã hoàn thành việc quan trắc và in phiếu kết quả, **trạng thái kết quả phân tích** là nội dung liên quan đến mẫu thử sau khi phân tích, **trạng thái xử lý** là trạng thái sau khi quan trắc của mẫu.

6. Chi tiết thông số của các mẫu quan trắc sau khi đã được xử lý sẽ được ghi nhận lại. Chi tiết thông số gồm các thông tin sau: **ID** của mẫu sẽ tự động ghi nhận số thứ tự của mẫu, **Mã phiếu kết quả** là mã của phiếu kết quả mà nó thuộc về, **Thông số** sẽ là các chỉ số thông tin liên qua tới mẫu và **đơn vị** của chúng, **quy chuẩn so sánh** là các quy chuẩn được áp dụng lên việc phân tích mẫu, **phương pháp phân tích** đã dùng cho việc phân tích mẫu. **Kết quả phân tích** sẽ ghi nhận lại quá trình phân tích mẫu và đưa ra nhận xét và đánh giá về mẫu. Chi tiết thông số thuộc về một phiếu kết quả cố định.

7. Khi nhân viên muốn sử dụng phần mềm thì nhân viên sẽ phải thực hiện các bước **đăng nhập**, các kết quả đăng nhập sẽ được lưu trữ lại tại phần quản lý đăng nhập. Mỗi lần đăng nhập sẽ chỉ là của một nhân viên thực hiện và mỗi nhân viên có thể đăng nhập nhiều lần trong ngày. Thông tin của đăng nhập như sau: **Mã đăng nhập** sẽ được đánh số tự động bắt đầu từ 1, 2, 3, ... ghi nhận lại các lần đăng nhập, **tên tài khoản** cũng chính là mã định danh của nhân viên hoặc có thể là **Email** của nhân viên, **mật khẩu** được bảo mật riêng của từng nhân viên, **mốc thời gian** đăng nhập.

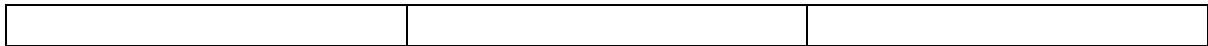
--	--	--

8. Sau khi đã đăng nhập thành công, nhân viên có thể sử dụng các chức năng của phần mềm, khi nhân viên muốn chỉnh sửa dữ liệu thì nhân viên phải gửi **yêu cầu chỉnh sửa**. Mỗi nhân viên có thể gửi nhiều yêu cầu chỉnh sửa thông tin lên chờ phê duyệt và mỗi yêu cầu sẽ ghi nhận lại chỉ một nhân viên thực hiện việc gửi nó. Yêu cầu chỉnh sửa sẽ có các thông tin sau: **Mã yêu cầu** là mã định danh cho từng yêu cầu, **mã nhân viên** đã gửi yêu cầu, **loại chỉnh sửa** mô tả loại chỉnh sửa (ví dụ: chỉnh sửa mẫu thử, hợp đồng, hay khách hàng), **ngày yêu cầu** và **ngày xử lý** lần lượt ghi lại ngày gửi yêu cầu và ngày xử lý yêu cầu. **trạng thái** cho biết tình trạng của yêu cầu (đang xử lý, đã xử lý, hoặc bị từ chối).

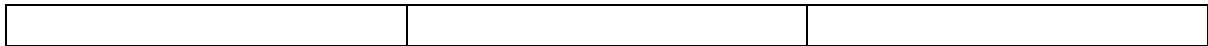
9. Sau khi đã được phê duyệt việc chỉnh sửa thì nhân viên có thể chỉnh sửa lại thông tin, mọi thông tin đã bị thay thế sẽ được **sao lưu** lại. Mỗi bản ghi của dữ liệu sao lưu sẽ do chính nhân viên đã chỉnh sửa chịu trách nhiệm và mỗi nhân viên sẽ phải có trách nhiệm với toàn bộ các bản ghi sao lưu của mình. Bản ghi của sao lưu dữ liệu sẽ có các thông tin sau: **Mã sao lưu** của từng lần chỉnh sửa, **tên bảng cần lưu** chỉ rõ bảng chứa dữ liệu đã thay đổi, **mã cần lưu** là mã của bản ghi bị chỉnh sửa (ví dụ: thay đổi thông tin khách hàng thì sẽ ghi nhận lại mã của khách hàng đó), **tên trường cần lưu** ghi lại trường cụ thể đã thay đổi, **mã người sửa** là mã của nhân viên đã gửi yêu cầu và được chấp nhận sửa thông tin, ngày chỉnh sửa, loại chỉnh sửa mô tả loại hành động (thêm, xóa, sửa), **giá trị ban đầu**, **giá trị thay đổi** lần lượt lưu giữ giá trị trước và sau khi chỉnh sửa. Thông tin sao lưu sẽ được ghi nhận từ các bảng tất cả thuộc tính kèm theo **hành động**, **ngày chỉnh sửa**, **bảng gốc**, **khóa chính**.

10. Ngoài ra, hệ thống cũng có **Chatbot** để nhắc nhở nhân viên về công việc cần làm hoặc các hợp đồng sắp đến hạn và trễ hạn. Chatbot này sẽ chỉ có một và được dung chung cho tất cả các nhân viên của công ty. Thông tin về Chatbot bao gồm: **Mã Chatbot** là mã định danh của chatbot, **tên Chatbot**, **mã nhân viên** nhận thông báo, **loại thông báo** mô tả loại thông báo (công việc hàng ngày, hợp đồng sắp hết hạn...), **thời gian thông báo** ghi lại thời gian gửi thông báo, **nội dung thông báo** chứa nội dung chi tiết của thông báo, **trạng thái thông báo** cho biết thông báo đã được gửi hay chưa, **lịch** là lịch trình gửi thông báo của chatbot.

--	--	--



--	--	--



--	--	--

--	--	--

--	--	--

--	--	--

- Procedure này dùng để thêm nhân viên vào trong công ty. Nó phục vụ cho chức năng quản lý nhân viên

```
CREATE PROCEDURE dbo.sp_AddEmployee
```

```
    @Manhanvien VARCHAR(10),
```

```
    @Hovaten NVARCHAR(100),
```

```
    @Ngaysinh DATE,
```

```
    @Chucvu NVARCHAR(50),
```

```
    @Sodienthoai VARCHAR(15),
```

```
    @Email VARCHAR(100)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Nhanvien (Manhanvien, Hovaten, Ngaysinh, Chucvu,
Sodienthoai, Email)
```

```
    VALUES (@Manhanvien, @Hovaten, @Ngaysinh, @Chucvu,
@Sodienthoai, @Email);
```

```
    PRINT 'Them nhan vien thanh cong';
```

```
END;
```

```
GO
```

1.2 Cập nhật thông tin nhân viên

- Procedure cập nhật thông tin nhân viên, dùng trong chức năng chỉnh sửa thông tin nhân viên, cho phép quản lý cập nhật thông tin chi tiết của một nhân viên đã tồn tại.

```
CREATE PROCEDURE dbo.sp_UpdateEmployee
```

```
    @Manhanvien VARCHAR(10),
```

```
    @Hovaten NVARCHAR(100),
```

```
    @Ngaysinh DATE,
```

```
    @Chucvu NVARCHAR(50),
```

```
    @Sodienthoai VARCHAR(15),
```

```
    @Email VARCHAR(100)
```

```
AS
```

	5	
--	---	--

--	--	--

BEGIN

UPDATE Nhanvien

SET Hovaten = @Hovaten,

Ngaysinh = @Ngaysinh,

Chucvu = @Chucvu,

Sodienthoai = @Sodienthoai,

Email = @Email

WHERE Manhanvien = @Manhanvien;

PRINT 'Cap nhat thong tin nhan vien thanh cong';

END;

GO

1.3 Thêm quản lý

- Procedure thêm một quản lý mới vào hệ thống, được sử dụng để thêm thông tin quản lý vào hệ thống khi có một quản lý mới được chỉ định.

CREATE PROCEDURE dbo.sp_AddManager

@Maquanly VARCHAR(10),

@Hovaten NVARCHAR(100),

@Sodienthoai VARCHAR(15),

@Email VARCHAR(100)

AS

BEGIN

INSERT INTO Quanly (Maquanly, Hovaten, Sodienthoai, Email)

VALUES (@Maquanly, @Hovaten, @Sodienthoai, @Email);

PRINT 'Them quan ly thanh cong';

END;

GO

	6	
--	---	--

--	--	--

1.4 Cập nhật thông tin quản lý

- Procedure thêm cập nhật thông tin quản lý đã có trong hệ thống, được sử dụng để chỉnh sửa thông tin quản lý khi cần thay đổi các thông tin về một quản lý.

```
CREATE PROCEDURE dbo.sp_UpdateManager
```

```
    @Maquanly VARCHAR(10),
```

```
    @Hovaten NVARCHAR(100),
```

```
    @Sodienthoai VARCHAR(15),
```

```
    @Email VARCHAR(100)
```

```
AS
```

```
BEGIN
```

```
    UPDATE Quanly
```

```
    SET Hovaten = @Hovaten,
```

```
        Sodienthoai = @Sodienthoai,
```

```
        Email = @Email
```

```
    WHERE Maquanly = @Maquanly;
```

```
    PRINT 'Cap nhat thong tin quan ly thanh cong';
```

```
END;
```

```
GO
```

1.5 Lấy danh sách nhân viên của một quản lý

- Procedure này cho phép lấy ra thông tin của tất cả thông tin những nhân viên thuộc sự quản lý của cùng 1 người.

```
CREATE PROCEDURE dbo.sp_GetEmployeesByManager
```

```
    @Maquanly VARCHAR(10)
```

```
AS
```

```
BEGIN
```

```
    SELECT NV.Manhanvien, NV.Hovaten, NV.Chucvu, NV.Sodienthoai,  
    NV.Email
```

```
    FROM Quanly_Nhanvien QN
```

	7	
--	---	--

--	--	--

```

INNER JOIN Nhanvien NV ON QN.Manhanvien = NV.Manhanvien
WHERE QN.Maquanly = @Maquanly;

PRINT 'Danh sach nhan vien duoc quan ly boi quan ly chi dinh';

END;

GO

```

1.6 Lấy thông tin chi tiết của nhân viên

- Procedure này cho phép lấy thông tin chi tiết của 1 nhân viên.

```

CREATE PROCEDURE dbo.sp_GetEmployeeDetails
    @Manhanvien VARCHAR(10)
AS
BEGIN
    SELECT Hovaten, Ngaysinh, Chucvu, Sodienthoai, Email
    FROM Nhanvien
    WHERE Manhanvien = @Manhanvien;

    PRINT 'Thong tin chi tiet cua nhan vien';

END;

GO

```

1.7 Thêm khách hàng

- Thêm một khách hàng mới vào hệ thống quản lý khách hàng. Procedure này được sử dụng trong chức năng thêm khách hàng khi doanh nghiệp hoặc cá nhân mới hợp tác với công ty.

```

CREATE PROCEDURE dbo.sp_AddCustomer
    @Makhachhang VARCHAR(10),
    @Tencongtty NVARCHAR(100),
    @Diachi NVARCHAR(255),
    @Nguoidaidien NVARCHAR(50),

```

	8	
--	---	--

--	--	--

```

        @Mahopdong VARCHAR(10),
        @Sodienthoai VARCHAR(15),
        @Email NVARCHAR(100)
AS
BEGIN
    INSERT INTO Quanlykhachhang (Makhachhang, Tencongtty, Diachi,
    Nguoidaidien, Mahopdong, Sodienthoai, Email)
        VALUES (@Makhachhang, @Tencongtty, @Diachi, @Nguoidaidien,
        @Mahopdong, @Sodienthoai, @Email);
    PRINT 'Them khach hang thanh cong';
END;
GO

```

1.8 Xóa khách hàng

- Xóa thông tin một khách hàng khỏi hệ thống. Procedure được sử dụng khi khách hàng không còn hợp tác với công ty.

```

CREATE PROCEDURE dbo.sp_DeleteCustomer
    @Makhachhang VARCHAR(10)
AS
BEGIN
    DELETE FROM Quanlykhachhang
    WHERE Makhachhang = @Makhachhang;
    PRINT 'Xoa khach hang thanh cong';
END;
GO

```

1.9 Thêm mẫu quan trắc

- Thêm một mẫu quan trắc mới vào hệ thống quản lý mẫu quan trắc. Procedure dùng khi công ty cần kiểm tra chất lượng môi trường theo hợp đồng.

	9	
--	---	--

--	--	--

```
CREATE PROCEDURE sp_ThemQuanlymauquantrac
```

```
@Mamau VARCHAR(10),
```

```
@Mahopdong VARCHAR(10),
```

```
@Tenmau NVARCHAR(100),
```

```
@Noidung NVARCHAR(MAX),
```

```
@Ngaylay DATE,
```

```
@Ngaytra DATE,
```

```
@Ketqua NVARCHAR(50),
```

```
@Manhanvien VARCHAR(10)
```

```
AS
```

```
BEGIN
```

```
INSERT INTO Quanlymauquantrac (Mamau, Mahopdong, Tenmau,
Noidung, Ngaylay, Ngaytra, Ketqua, Manhanvien)
```

```
VALUES (@Mamau, @Mahopdong, @Tenmau, @Noidung, @Ngaylay,
@Ngaytra, @Ketqua, @Manhanvien);
```

```
END;
```

```
GO
```

1.10 Cập nhật mẫu quan trắc

- Cập nhật thông tin một mẫu quan trắc hiện có. Được sử dụng khi có sự thay đổi về thông tin mẫu (tên, nội dung, ngày lấy, ngày trả, kết quả, nhân viên thực hiện).

```
CREATE PROCEDURE sp_CapNhatQuanlymauquantrac
```

```
@Mamau VARCHAR(10),
```

```
@Tenmau NVARCHAR(100),
```

```
@Noidung NVARCHAR(MAX),
```

```
@Ngaylay DATE,
```

```
@Ngaytra DATE,
```

```
@Ketqua NVARCHAR(50),
```

```
@Manhanvien VARCHAR(10)
```

	10	
--	----	--

--	--	--

```

AS
BEGIN
    UPDATE Quanlymauquantrac
    SET
        Tenmau = @Tenmau,
        Noidung = @Noidung,
        Ngaylay = @Ngaylay,
        Ngaytra = @Ngaytra,
        Ketqua = @Ketqua,
        Manhanvien = @Manhanvien
    WHERE Mamau = @Mamau;
END;
GO

```

1.11 Xóa mẫu quan trắc

- Xóa thông tin một mẫu quan trắc khỏi hệ thống. Được dùng khi mẫu không còn giá trị hoặc đã bị hủy.

```

CREATE PROCEDURE sp_XoaQuanlymauquantrac
    @Mamau VARCHAR(10)
AS
BEGIN
    DELETE FROM Quanlymauquantrac
    WHERE Mamau = @Mamau;
END;
GO

```

1.12 Thêm phiếu kết quả

- Thêm một phiếu kết quả quan trắc mới vào hệ thống. Sử dụng khi hoàn thành kiểm tra mẫu và cần lưu kết quả vào cơ sở dữ liệu.

	11	
--	----	--

--	--	--

```

CREATE PROCEDURE sp_ThemPhieuketqua
    @Mamauthu VARCHAR(10),
    @Mahopdong VARCHAR(10),
    @Ngaylaymau DATE,
    @Ngaytraketqua DATE,
    @Trangthaiketquaphantich NVARCHAR(MAX),
    @Trangthaixuly NVARCHAR(30)
AS
BEGIN
    INSERT INTO Phieuketqua (Mamauthu, Mahopdong, Ngaylaymau,
    Ngaytraketqua, Trangthaiketquaphantich, Trangthaixuly)
    VALUES (@Mamauthu, @Mahopdong, @Ngaylaymau, @Ngaytraketqua,
    @Trangthaiketquaphantich, @Trangthaixuly);
END;
GO

```

1.13 Cập nhật phiếu kết quả

- Cập nhật thông tin của một phiếu kết quả quan trắc. Sử dụng khi có thay đổi về trạng thái phân tích, trạng thái xử lý, hoặc ngày trả kết quả.

```

CREATE PROCEDURE sp_CapNhatPhieuketqua
    @Maphieuketqua INT,
    @Ngaytraketqua DATE,
    @Trangthaiketquaphantich NVARCHAR(MAX),
    @Trangthaixuly NVARCHAR(30)
AS
BEGIN
    UPDATE Phieuketqua
    SET
        Ngaytraketqua = @Ngaytraketqua,

```

	12	
--	----	--

--	--	--

```

Trangthaiketquaphantich = @Trangthaiketquaphantich,
Trangthaixuly = @Trangthaixuly
WHERE Maphieuketqua = @Maphieuketqua;
END;
GO

```

1.14 Xóa phiếu kết quả

- Xóa một phiếu kết quả quan trắc khỏi cơ sở dữ liệu. Được dùng khi phiếu kết quả không còn cần thiết hoặc bị nhập sai.

```

CREATE PROCEDURE sp_XoaPhieuketqua
    @Maphieuketqua INT
AS
BEGIN
    DELETE FROM Phieuketqua
    WHERE Maphieuketqua = @Maphieuketqua;
END;
GO

```

1.15 Thêm chi tiết thông số về mẫu

- Thêm thông tin chi tiết thông số quan trắc vào cơ sở dữ liệu. Sử dụng khi cần lưu trữ các thông số đo lường cụ thể của từng mẫu.

```

CREATE PROCEDURE sp_ThemChitietthongso
    @Maphieuketqua INT,
    @Thongso NVARCHAR(100),
    @Donvi NVARCHAR(50),
    @Phuongphapphantich NVARCHAR(200),
    @Ketquaphantich NVARCHAR(MAX),
    @Quychuansosanh NVARCHAR(MAX)
AS

```

	13	
--	----	--

--	--	--

BEGIN

INSERT INTO Chitietthongso (Maphieuketqua, Thongso, Donvi, Phuongphapphantich, Ketquaphantich, Quychuansosanh)

VALUES (@Maphieuketqua, @Thongso, @Donvi, @Phuongphapphantich, @Ketquaphantich, @Quychuansosanh);

END;

GO

1.16 Cập nhật chi tiết thông số cho từng mẫu

- Cập nhật thông tin chi tiết của một thông số quan trắc. Dùng khi có sự thay đổi về kết quả phân tích, quy chuẩn so sánh hoặc phương pháp phân tích.

CREATE PROCEDURE sp_CapNhatChitietthongso

@ID INT,

@Thongso NVARCHAR(100),

@Donvi NVARCHAR(50),

@Phuongphapphantich NVARCHAR(200),

@Ketquaphantich NVARCHAR(MAX),

@Quychuansosanh NVARCHAR(MAX)

AS

BEGIN

UPDATE Chitietthongso

SET

Thongso = @Thongso,

Donvi = @Donvi,

Phuongphapphantich = @Phuongphapphantich,

Ketquaphantich = @Ketquaphantich,

Quychuansosanh = @Quychuansosanh

WHERE ID = @ID;

END;

	14	
--	----	--

--	--	--

GO

1.17 Xóa chi tiết thông số

- Xóa thông tin chi tiết của một thông số quan trắc khỏi hệ thống. Được dùng khi thông số không còn giá trị hoặc bị nhập sai.

```
CREATE PROCEDURE sp_XoaChitietthongso
```

```
    @ID INT
```

```
AS
```

```
BEGIN
```

```
    DELETE FROM Chitietthongso
```

```
    WHERE ID = @ID;
```

```
END;
```

```
GO
```

2. Function

2.1 Function chuyển đổi từ hệ tam thập lục phân(36) sang hệ thập phân(10)

- Function này dùng để chuyển đổi kiểu dữ liệu từ kiểu chuỗi thuộc hệ tam thập lục phân sang số thập phân. Function này hỗ trợ trong việc tạo mã tự động cho một đối tượng nếu id đó thuộc hệ tam thập lục phân.

```
CREATE FUNCTION FUNC_B36TO10 (@B36 VARCHAR(10))
```

```
RETURNS INT AS
```

```
BEGIN
```

```
    DECLARE @I TINYINT
```

```
    DECLARE @TEMP INT
```

```
    DECLARE @B10 INT
```

```
    DECLARE @POW TINYINT
```

```
    SET @I = LEN(@B36)
```

```
    SET @B10 = 0
```

	15	
--	----	--

--	--	--

```

SET @POW = 0

WHILE( @I > 0 )

BEGIN

    SET @TEMP = ASCII(SUBSTRING(@B36, @I, 1))

    SET @B10 = @B10 + POWER(36, @POW) * IIF(@TEMP > 60,
@TEMP - 54, @TEMP - 48)

    SET @I = @I - 1

    SET @POW = @POW + 1

END

RETURN @B10

END;

GO

```

2.2 Function chuyển đổi từ hệ thập phân(10) sang tam thập lục phân(36)

- Dùng để chuyển đổi kiểu dữ liệu từ số thập phân sang một chuỗi thuộc hệ tam thập lục phân. Function này hỗ trợ trong việc tạo mã tự động cho một đối tượng nếu mã đó thuộc hệ tam thập lục phân.

```

CREATE FUNCTION FUNC_B10TO36 (@B10 INT, @SIZE TINYINT)
RETURNS VARCHAR(10) AS
BEGIN

    DECLARE @TEMP INT

    DECLARE @BASE10 INT

    DECLARE @POW INT

    DECLARE @RES VARCHAR(10)

    SET @RES = ""

    SET @POW = 0

    WHILE(POWER(36, @POW) <= @B10)

        SET @POW = @POW + 1

```

	16	
--	----	--

--	--	--

```

SET @POW = @POW - 1

WHILE( @POW >= 0 )

BEGIN

    SET @TEMP = ROUND(@B10 / POWER(36, @POW), 0)

    SET @RES = CONCAT(@RES, IIF(@TEMP < 10, CHAR(@TEMP +
        48), CHAR(@TEMP + 55)))

    SET @B10 = @B10 - ROUND(@B10 / POWER(36, @POW), 0) *
        POWER(36, @POW)

    SET @POW = @POW - 1

END

RETURN RIGHT(REPLICATE(0, 10) + @RES, @SIZE)

END;

GO

```

2.3 Function tăng mã thuộc hệ tam thập lục phân (36)

- Function này dùng để tạo mã thuộc hệ tam thập lục phân tự động cho một đối tượng.

```

CREATE FUNCTION FUNC_TANGID_B36 (@IdType VARCHAR(10),
    @CURRENT INT, @SIZE INT)

RETURNS VARCHAR(10) AS

BEGIN

    RETURN @IdType + DBO.FUNC_B10TO36(@CURRENT + 1, @SIZE -
        LEN(@IdType))

END;

GO

```

2.4 Function tăng mã thuộc hệ thập phân

- Function này dùng để tạo mã thuộc hệ thập phân tự động cho một đối tượng.

```

CREATE FUNCTION FUNC_TANGID_B10 (@IdType VARCHAR(10),
    @CURRENT INT, @SIZE INT)

```

--	--	--

RETURNS VARCHAR(10) AS

BEGIN

 RETURN @IdType + RIGHT(REPLICATE(0, 10) + CAST(@CURRENT +
1 AS VARCHAR(10)), @SIZE - LEN(@IdType))

END;

GO

2.5 Function kiểm tra đăng nhập

- Function này dùng để kiểm tra thông tin đăng nhập của người dùng (tài khoản hoặc email và mật khẩu) có hợp lệ hay không.

CREATE FUNCTION dbo.fn_KiemTraDangNhap

(

 @TaiKhoanHoacEmail NVARCHAR(100),

 @MatKhau NVARCHAR(100)

)

RETURNS BIT

AS

BEGIN

 DECLARE @IsValid BIT;

 SELECT @IsValid = CASE

 WHEN COUNT(1) > 0 THEN 1

 ELSE 0

 END

 FROM Quanlydangnhap

 WHERE (Taikhoan = @TaiKhoanHoacEmail OR Email =
@TaiKhoanHoacEmail)

 AND Matkhau = @MatKhau;

	18	
--	----	--

--	--	--

RETURN @IsValid;

END;

GO

2.6 Function lấy vai trò

- Function này dùng để xác định vai trò của người sử dụng là nhân viên hay quản lý sau khi đăng nhập vào.

CREATE FUNCTION dbo.fn_LayVaiTro

(

@TaiKhoanHoacEmail NVARCHAR(100)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @VaiTro NVARCHAR(50);

-- Kiểm tra vai trò Nhân viên

IF EXISTS (

SELECT 1

FROM Nhanvien nv

JOIN Quanlydangnhap qldn ON nv.Manhanvien = qldn.Taikhoan

WHERE qldn.Taikhoan = @TaiKhoanHoacEmail OR qldn.Email =
@TaiKhoanHoacEmail

)

BEGIN

SET @VaiTro = 'Nhanvien';

RETURN @VaiTro;

END

	19	
--	----	--

--	--	--

```
-- Kiểm tra vai trò Quản lý
IF EXISTS (
    SELECT 1
    FROM Quanly ql
    JOIN Quanlydangnhap qldn ON ql.Maquanly = qldn.Taikhoan
    WHERE qldn.Taikhoan = @TaiKhoanHoacEmail OR qldn.Email =
@TaiKhoanHoacEmail
)
BEGIN
    SET @VaiTro = 'QuanLy';
    RETURN @VaiTro;
END

-- Không xác định vai trò
SET @VaiTro = "";
RETURN @VaiTro;
END;
GO
```

3. Trigger

3.1 Trigger lưu trữ cập nhật hợp đồng

- Lưu trữ bản ghi lịch sử khi thực hiện cập nhật dữ liệu trong hợp đồng

```
CREATE TRIGGER trg_Update_QuanLyHopDong
ON QuanLyHopDong
AFTER UPDATE
AS
BEGIN
```

	20	
--	----	--

--	--	--

```
INSERT INTO QuanLyHopDong_History (Mahopdong, Makhachhang,
Manhanvien, Quy, Trangthai, Ngaylap, Ngaytra, Vieccanlam, Hanhdong,
Ngaychinh sua, Banggoc, Khoachinh)
```

```
SELECT
```

```
    d.Mahopdong, d.Makhachhang, d.Manhanvien, d.Quy, d.Trangthai,
    d.Ngaylap, d.Ngaytra, d.Vieccanlam,
```

```
    'UPDATE' AS Hanhdong,
```

```
    GETDATE() AS Ngaychinh sua,
```

```
    'QuanLyHopDong' AS Banggoc,
```

```
    CAST(d.Mahopdong AS NVARCHAR(100)) AS Khoachinh
```

```
FROM deleted d;
```

```
END;
```

```
go
```

3.2 Trigger lưu trữ xóa hợp đồng

- Lưu trữ bản ghi lịch sử khi thực hiện xóa dữ liệu trong hợp đồng

```
CREATE TRIGGER trg_Delete_QuanLyHopDong
```

```
ON QuanLyHopDong
```

```
AFTER DELETE
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO QuanLyHopDong_History (Mahopdong, Makhachhang,
    Manhanvien, Quy, Trangthai, Ngaylap, Ngaytra, Vieccanlam, Hanhdong,
    Ngaychinh sua, Banggoc, Khoachinh)
```

```
SELECT
```

```
    d.Mahopdong, d.Makhachhang, d.Manhanvien, d.Quy, d.Trangthai,
    d.Ngaylap, d.Ngaytra, d.Vieccanlam,
```

```
    'DELETE' AS Hanhdong,
```

```
    GETDATE() AS Ngaychinh sua,
```

```
    'QuanLyHopDong' AS Banggoc,
```

	21	
--	----	--

--	--	--

```

CAST(d.Mahopdong AS NVARCHAR(100)) AS Khoachinh
FROM deleted d;
END;
go

```

3.3 Trigger lưu trữ cập nhật khách hàng

- Lưu trữ bản ghi lịch sử khi thực hiện cập nhật dữ liệu khách hàng.

```

CREATE TRIGGER trg_Update_QuanLyKhachHang
ON QuanLyKhachHang
AFTER UPDATE
AS
BEGIN
    INSERT INTO QuanLyKhachHang_History (Makhachhang, Tencongtty,
Diachi, Nguoidaidien, Mahopdong, Sodienthoai, Email, Hanhdong,
Ngaychinhsua, Banggoc, Khoachinh)
    SELECT
        d.Makhachhang, d.Tencongtty, d.Diachi, d.Nguoidaidien, d.Mahopdong,
d.Sodienthoai, d.Email,
        'UPDATE' AS Hanhdong,
        GETDATE() AS Ngaychinhsua,
        'QuanLyKhachHang' AS Banggoc,
        CAST(d.Makhachhang AS NVARCHAR(100)) AS Khoachinh
    FROM deleted d;
END;
go

```

3.4 Trigger lưu trữ cập xóa thông tin khách hàng

- Lưu trữ bản ghi lịch sử khi thực hiện xóa dữ liệu của khách hàng.

```

CREATE TRIGGER trg_Delete_QuanLyKhachHang

```

	22	
--	----	--

--	--	--

ON QuanLyKhachHang

AFTER DELETE

AS

BEGIN

INSERT INTO QuanLyKhachHang_History (Makhachhang, Tencongtty, Diachi, Nguoidaidien, Mahopdong, Sodienthoai, Email, Hanhdong, Ngaychinh sua, Banggoc, Khoachinh)

SELECT

d.Makhachhang, d.Tencongtty, d.Diachi, d.Nguoidaidien, d.Mahopdong, d.Sodienthoai, d.Email,

'DELETE' AS Hanhdong,

GETDATE() AS Ngaychinh sua,

'QuanLyKhachHang' AS Banggoc,

CAST(d.Makhachhang AS NVARCHAR(100)) AS Khoachinh

FROM deleted d;

END;

go

3.5 Trigger lưu trữ cập nhật mẫu quan trắc

- Lưu trữ bản ghi lịch sử khi thực hiện cập nhật dữ liệu mẫu quan trắc

CREATE TRIGGER trg_Update_QuanLyMauQuanTrac

ON Quanlymauquantrac

AFTER UPDATE

AS

BEGIN

INSERT INTO QuanLyMauQuanTrac_History (Mamau, Mahopdong, Tenmau, Noidung, Ngaylay, Ngaytra, Ketqua, Manhanvien, Hanhdong, Ngaychinh sua, Banggoc, Khoachinh)

SELECT

	23	
--	----	--

--	--	--

```

d.Mamau, d.Mahopdong, d.Tenmau, d.Noidung,d.Ngaylay,
d.Ngaytra,d.Ketqua, d.Manhanvien,
'UPDATE' AS Hanhdong,
GETDATE() AS Ngaychinh sua,
'QuanLyMauQuanTrac' AS Banggoc,
CAST(d.Mamau AS NVARCHAR(100)) AS Khoachinh
FROM deleted d;
END;
GO

```

3.6 Trigger lưu trữ xóa mẫu quan trắc

- Lưu trữ bản ghi lịch sử khi thực hiện xóa dữ liệu của mẫu quan trắc.

```

CREATE TRIGGER trg_Delete_QuanLyMauQuanTrac
ON Quanlymauquantrac
AFTER DELETE
AS
BEGIN
    INSERT INTO QuanLyMauQuanTrac_History (Mamau, Mahopdong,
    Tenmau, Noidung,Ngaylay, Ngaytra,Ketqua, Manhanvien, Hanhdong,
    Ngaychinh sua, Banggoc, Khoachinh)
    SELECT
        d.Mamau, d.Mahopdong, d.Tenmau, d.Noidung,d.Ngaylay,
        d.Ngaytra,d.Ketqua, d.Manhanvien,
        'DELETE' AS Hanhdong,
        GETDATE() AS Ngaychinh sua,
        'QuanLyMauQuanTrac' AS Banggoc,
        CAST(d.Mamau AS NVARCHAR(100)) AS Khoachinh
    FROM deleted d;
END;

```


--	--	--

go

3.7 Trigger lưu trữ cập nhật phiếu kết quả:

- Lưu trữ bản ghi lịch sử khi thực hiện cập nhật dữ liệu của phiếu kết quả.

```
CREATE TRIGGER trg_Update_PhieuKetQua
ON PhieuKetQua
AFTER UPDATE
AS
BEGIN
    INSERT INTO PhieuKetQua_History (Maphieuketqua, Mamauthu,
    Mahopdong, Ngaylaymau, Ngaytraketqua,
    Trangthaiketquaphantich, Trangthaixuly, Hanhdong, Ngaychinhsua, Banggoc,
    Khoachinh)
    SELECT
        d.Maphieuketqua, d.Mamauthu, d.Mahopdong, d.Ngaylaymau,
        d.Ngaytraketqua, d.Trangthaixuly, d.Trangthaiketquaphantich,
        'UPDATE' AS Hanhdong,
        GETDATE() AS Ngaychinhsua,
        'PhieuKetQua' AS Banggoc,
        CAST(d.Maphieuketqua AS NVARCHAR(100)) AS Khoachinh
    FROM deleted d;
END;
```

go

3.8 Trigger lưu trữ xóa phiếu kết quả

- Lưu trữ bản ghi lịch sử khi thực hiện xóa dữ liệu của phiếu kết quả.

```
CREATE TRIGGER trg_Delete_PhieuKetQua
ON PhieuKetQua
AFTER DELETE
AS
```

	25	
--	----	--

--	--	--

BEGIN

INSERT INTO PhieuKetQua_History (Maphieuketqua, Mamauthu, Mahopdong, Ngaylaymau, Ngaytraketqua, TrangthaiKETQUAPHANTICH, TrangthaiXULY, Hanhdong, NgaychinhSua, Banggoc, Khoachinh)

SELECT

d.Maphieuketqua, d.Mamauthu, d.Mahopdong, d.Ngaylaymau, d.Ngaytraketqua, d.TrangthaiXULY, d.TrangthaiKETQUAPHANTICH, 'DELETE' AS Hanhdong, GETDATE() AS NgaychinhSua, 'PhieuKetQua' AS Banggoc, CAST(d.Maphieuketqua AS NVARCHAR(100)) AS Khoachinh FROM deleted d;

END;

go

4. Các câu lệnh tạo Cơ sở dữ liệu

- Khởi tạo cơ sở dữ liệu :

CREATE DATABASE QuanLyMauKiemDinhMoiTruong
USE DATABASE QuanLyMauKiemDinhMoiTruong

- Bảng Quanlydangnhap

CREATE TABLE Quanlydangnhap (
 Madangnhap INT NOT NULL IDENTITY PRIMARY KEY,
 Taikhoan VARCHAR(10) NOT NULL,
 Email VARCHAR(100) UNIQUE NOT NULL,
 Mocthoigian DATETIME DEFAULT GETDATE(),
 Matkhau VARCHAR(255) NOT NULL
);
GO

--	--	--

- Bảng Nhanvien

```
CREATE TABLE Nhanvien (
    Manhanvien VARCHAR(10) NOT NULL PRIMARY KEY,
    Hovaten NVARCHAR(100) NOT NULL,
    Ngaysinh DATE NOT NULL,
    Sodienthoai VARCHAR(15) NOT NULL,
    Chucvu NVARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL
);
GO
```

- Bảng Quanly

```
CREATE TABLE Quanly (
    Maquanly VARCHAR(10) NOT NULL PRIMARY KEY,
    Hovaten NVARCHAR(100) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    Sodienthoai VARCHAR(15) NOT NULL
);
GO
```

- Bảng Quanly_Nhanvien (Quan hệ nhiều-nhiều giữa Quanly và Nhanvien)

```
CREATE TABLE Quanly_Nhanvien (
    Maquanly VARCHAR(10) NOT NULL,
    Manhanvien VARCHAR(10) NOT NULL,
    PRIMARY KEY (Maquanly, Manhanvien),
    FOREIGN KEY (Maquanly) REFERENCES Quanly(Maquanly),
    FOREIGN KEY (Manhanvien) REFERENCES Nhanvien(Manhanvien)
);
```

	27	
--	----	--

--	--	--

GO

- Bảng Saoluudulieu

```
CREATE TABLE Saoluudulieu (
    Masoluu INT NOT NULL IDENTITY PRIMARY KEY,
    Tenbangcanluu VARCHAR(100) NOT NULL,
    Macanluu VARCHAR(10) NOT NULL,
    Tentruongcanluu VARCHAR(100) NOT NULL,
    Giatribandau VARCHAR(255) NOT NULL,
    Giatrithaydoi VARCHAR(255) NOT NULL,
    Loachinh sua VARCHAR(50) NOT NULL,
    Ngaychinh sua DATE NOT NULL,
    Manguoisua VARCHAR(10) NOT NULL,
    FOREIGN KEY (Manguoisua) REFERENCES Nhanvien(Manhanvien)
);
GO
```

- Bảng Yeucauchinh sua

```
CREATE TABLE Yeucauchinh sua (
    Mayeucau INT NOT NULL IDENTITY PRIMARY KEY,
    Loaiyeucau VARCHAR(100) NOT NULL,
    Manhanvien VARCHAR(10) NOT NULL,
    Ngayyeucau DATE NOT NULL,
    Ngayxuly DATE,
    Trangthai VARCHAR(50) NOT NULL,
    FOREIGN KEY (Manhanvien) REFERENCES Nhanvien(Manhanvien)
);
GO
```

	28	
--	----	--

--	--	--

- Bảng Chatbot

```
CREATE TABLE Chatbot (
    Machatbot INT NOT NULL IDENTITY PRIMARY KEY,
    Tenchatbot VARCHAR(100) NOT NULL UNIQUE,
    Manhanvien VARCHAR(10) NOT NULL,
    LoaiThongbao VARCHAR(255) NOT NULL,
    ThoiGianThongBao DATETIME NOT NULL,
    Lich DATETIME NOT NULL,
    Noidung NTEXT NOT NULL,
    Trangthai NVARCHAR(50) NOT NULL,
    FOREIGN KEY (Manhanvien) REFERENCES Nhanvien(Manhanvien)
);
GO
```

- Bảng Quanlykhachhang

```
CREATE TABLE Quanlykhachhang (
    Makhachhang VARCHAR(10) NOT NULL PRIMARY KEY,
    Tencongty NVARCHAR(100) NOT NULL,
    Nguoidaidien NVARCHAR(100) NOT NULL,
    Diachi NVARCHAR(255) NOT NULL,
    Mahopdong VARCHAR(10) NOT NULL,
    Sodienthoai VARCHAR(15) NOT NULL,
    Email VARCHAR(100) NOT NULL
);
GO
```

- Bảng Quanlyhopdong

	29	
--	----	--

--	--	--

```

CREATE TABLE Quanlyhopdong (
    Mahopdong VARCHAR(10) NOT NULL PRIMARY KEY,
    Makhachhang VARCHAR(10) NOT NULL,
    Manhanvien VARCHAR(10) NOT NULL,
    Quy FLOAT NOT NULL,
    Trangthai NVARCHAR(50) NOT NULL,
    Ngaylap DATE NOT NULL,
    Ngaytra DATE NOT NULL,
    Vieccanlam NVARCHAR(255) NOT NULL,
    FOREIGN KEY (Makhachhang) REFERENCES
Quanlykhachhang(Makhachhang),
    FOREIGN KEY (Manhanvien) REFERENCES Nhanvien(Manhanvien)
);
GO

```

- Bảng Hopdong_Nhanvien (Quan hệ nhiều-nhiều giữa Hopdong và Nhanvien)

```

CREATE TABLE Hopdong_Nhanvien (
    Mahopdong VARCHAR(10) NOT NULL,
    Manhanvien VARCHAR(10) NOT NULL,
    PRIMARY KEY (Mahopdong, Manhanvien),
    FOREIGN KEY (Mahopdong) REFERENCES
Quanlyhopdong(Mahopdong),
    FOREIGN KEY (Manhanvien) REFERENCES Nhanvien(Manhanvien)
);
GO

```

- Bảng Quanlymauquantrac (Đổi tên từ Quanlymauthu)

```

CREATE TABLE Quanlymauquantrac (
    Mamau VARCHAR(10) NOT NULL PRIMARY KEY,

```

	30	
--	----	--

--	--	--

```

Mahopdong VARCHAR(10) NOT NULL,
Tenmau NVARCHAR(100) NOT NULL,
Noidung NVARCHAR(MAX) NOT NULL,
Ngaylay DATE NOT NULL,
Ngaytra DATE NOT NULL,
Ketqua NVARCHAR(50) NOT NULL,
Manhanvien VARCHAR(10) NOT NULL,
FOREIGN KEY (Mahopdong) REFERENCES
Quanlyhopdong(Mahopdong),
FOREIGN KEY (Manhanvien) REFERENCES Nhanvien(Manhanvien)
);
GO

```

- Bảng Phieuketqua

```

CREATE TABLE Phieuketqua (
Maphieuketqua INT NOT NULL IDENTITY PRIMARY KEY,
Mamauthu VARCHAR(10) NOT NULL,
Mahopdong VARCHAR(10) NOT NULL,
Ngaylaymau DATE NOT NULL,
Ngaytraketqua DATE NOT NULL,
Trangthaiketquaphantich NVARCHAR(MAX) NOT NULL,
Trangthaixuly NVARCHAR(30),
FOREIGN KEY (Mamauthu) REFERENCES Quanlymauquantrac(Mamau),
FOREIGN KEY (Mahopdong) REFERENCES
Quanlyhopdong(Mahopdong)
);
GO

```

- Bảng chi tiết thông số

	31	
--	----	--

--	--	--

```

CREATE TABLE Chitietthongso (
    ID INT NOT NULL IDENTITY PRIMARY KEY,
    Maphieuketqua INT NOT NULL,
    Thongso NVARCHAR(100) NOT NULL,
    Donvi NVARCHAR(50) NOT NULL,
    Phuongphapphantich NVARCHAR(200) NOT NULL,
    Ketquaphantich NVARCHAR(MAX) NOT NULL,
    Quychuansosanh NVARCHAR(MAX) NOT NULL,
    FOREIGN KEY (Maphieuketqua) REFERENCES
    Phieuketqua(Maphieuketqua)
);
GO

```

-Phần thêm sao lưu dữ liệu

```

CREATE TABLE QuanLyMauQuanTrac_History (
    Mamau VARCHAR(10),
    Mahopdong VARCHAR(10),
    Tenmau NVARCHAR(100),
    Noidung NVARCHAR(MAX),
    Ngaylay DATE,
    Ngaytra DATE,
    Ketqua NVARCHAR(50),
    Manhanvien VARCHAR(10),
    Hanhdong NVARCHAR(10), -- Loại hành động (UPDATE, DELETE)
    Ngaychinhsửa DATETIME DEFAULT GETDATE(), -- Thời gian hành
    động
    Banggoc NVARCHAR(100),
    Khoachinh NVARCHAR(100),
    PRIMARY KEY (Mamau, Ngaychinhsửa)

```

	32	
--	----	--

--	--	--

);

GO

CREATE TABLE QuanLyKhachHang_History (

Makhachhang VARCHAR(10),

Tencongty NVARCHAR(100),

Nguoidaidien NVARCHAR(50),

Diachi NVARCHAR(255),

Mahopdong VARCHAR(10),

Sodienthoai VARCHAR(15),

Email NVARCHAR(100),

Hanhdong NVARCHAR(10), -- Loại hành động (UPDATE, DELETE)

Ngaychinh sua DATETIME DEFAULT GETDATE(), -- Thời gian hành động

Banggoc NVARCHAR(100),

Khoachinh NVARCHAR(100),

PRIMARY KEY (Makhachhang, Ngaychinh sua)

);

GO

CREATE TABLE QuanLyHopDong_History (

Mahopdong VARCHAR(10),

Makhachhang VARCHAR(10),

Manhanvien VARCHAR(10),

Quy FLOAT,

Trangthai NVARCHAR(50),

Ngaylap DATE,

Ngaytra DATE,

	33	
--	----	--

--	--	--

```

Vieccanlam NVARCHAR(255),
Hanhdong NVARCHAR(10), -- Loại hành động (UPDATE, DELETE)
Ngaychinh sua DATETIME DEFAULT GETDATE(), -- Thời gian hành
động
Banggoc NVARCHAR(100),
Khoachinh NVARCHAR(100),
PRIMARY KEY (Mahopdong, Ngaychinh sua)
);
GO

```

```

CREATE TABLE PhieuKetQua_History (
    Maphieuketqua INT,
    Mamauthu VARCHAR(10),
    Mahopdong VARCHAR(10),
    Ngaylaymau DATE,
    Ngaytraketqua DATE,
    Trangthaiketquaphantich NVARCHAR(MAX),
    Trangthaixuly NVARCHAR(30),
    Hanhdong NVARCHAR(10), -- Loại hành động (UPDATE, DELETE)
    Ngaychinh sua DATETIME DEFAULT GETDATE(), -- Thời gian hành
động
    Banggoc NVARCHAR(100),
    Khoachinh NVARCHAR(100),
    PRIMARY KEY (Maphieuketqua, Ngaychinh sua)
);
GO

```

	34	
--	----	--