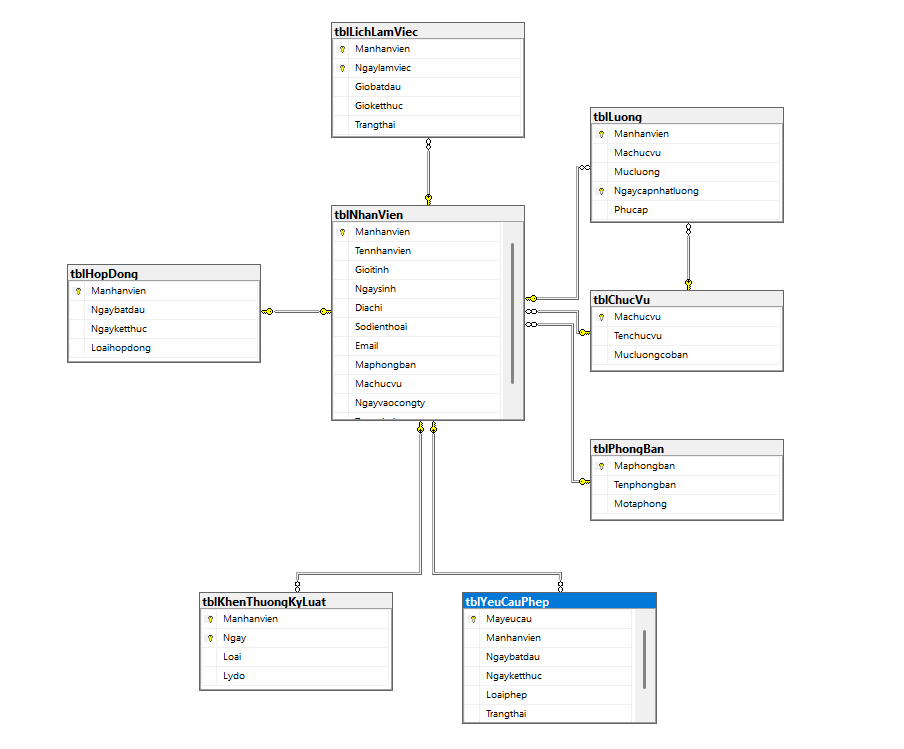
# Chương 3 : Tạo cơ sở dữ liệu

## 3.1. Tạo cơ sở Database

### 3.1.1 DataBase Diagram



### 3.1.1 Câu lệnh SQL

-- Xóa cơ sở dữ liệu nếu đã tồn tại

DROP DATABASE IF EXISTS Quan\_Ly\_Nhan\_Su;

-- Tạo cơ sở dữ liệu mới

CREATE DATABASE Quan\_Ly\_Nhan\_Su;

USE Quan\_Ly\_Nhan\_Su;

-- Xóa bảng nếu đã tồn tại

IF OBJECT\_ID('tblChucVu', 'U') IS NOT NULL

DROP TABLE tblChucVu;

go;

IF OBJECT\_ID('tblPhongBan', 'U') IS NOT NULL

DROP TABLE tblPhongBan;

go;

IF OBJECT\_ID('tblNhanVien', 'U') IS NOT NULL

DROP TABLE tblNhanVien;

go;

IF OBJECT\_ID('tblLuong', 'U') IS NOT NULL

DROP TABLE tblLuong;

go;

IF OBJECT\_ID('tblLichLamViec', 'U') IS NOT NULL

DROP TABLE tblLichLamViec;

go;

IF OBJECT\_ID('tblYeuCauPhep', 'U') IS NOT NULL

DROP TABLE tblYeuCauPhep;

go;

IF OBJECT\_ID('tblKhenThuongKyLuat', 'U') IS NOT NULL

DROP TABLE tblKhenThuongKyLuat;

go;

IF OBJECT\_ID('tblHopDong', 'U') IS NOT NULL

DROP TABLE tblHopDong;

go;

-- Tạo bảng Chức Vụ

CREATE TABLE tblChucVu (

Machucvu INT PRIMARY KEY,

Tenchucvu VARCHAR(100) NOT NULL,

Mucluongcoban DECIMAL(10, 2) NOT NULL

);

-- Tạo bảng Phòng Ban

CREATE TABLE tblPhongBan (

Maphongban INT PRIMARY KEY,

Tenphongban VARCHAR(100) NOT NULL,

Motaphong TEXT NOT NULL

);

-- Tạo bảng Nhân Viên

CREATE TABLE tblNhanVien (

Manhanvien INT PRIMARY KEY,

Tennhanvien VARCHAR(100) NOT NULL,

Gioitinh VARCHAR(10),

Ngaysinh DATE,

Diachi VARCHAR(255),

Sodienthoai VARCHAR(15),

Email VARCHAR(100) UNIQUE,

Maphongban INT,

Machucvu INT NOT NULL,

Ngayvaocongty DATE,

Trangthai VARCHAR(50),

FOREIGN KEY (Maphongban) REFERENCES tblPhongBan(Maphongban) ON DELETE CASCADE,

FOREIGN KEY (Machucvu) REFERENCES tblChucVu(Machucvu) ON DELETE CASCADE

);

-- Tạo bảng Lương

CREATE TABLE tblLuong (

Manhanvien INT,

Machucvu INT NOT NULL,

Mucluong DECIMAL(10, 2) NOT NULL,

Ngaycapnhatluong DATE NOT NULL,

Phucap DECIMAL(10, 2),

PRIMARY KEY (Manhanvien, Ngaycapnhatluong),

FOREIGN KEY (Manhanvien) REFERENCES tblNhanVien(Manhanvien) ON DELETE CASCADE,

FOREIGN KEY (Machucvu) REFERENCES tblChucVu(Machucvu) ON DELETE NO ACTION

);

-- Tạo bảng Lịch Làm Việc

CREATE TABLE tblLichLamViec (

Manhanvien INT,

Ngaylamviec DATE,

Giobatdau TIME,

Gioketthuc TIME,

Trangthai VARCHAR(50),

PRIMARY KEY (Manhanvien, Ngaylamviec),

FOREIGN KEY (Manhanvien) REFERENCES tblNhanVien(Manhanvien) ON DELETE CASCADE

);

-- Tạo bảng Yêu Cầu Nghỉ Phép

CREATE TABLE tblYeuCauPhep (

Mayeucau INT PRIMARY KEY,

Manhanvien INT,

Ngaybatdau DATE,

Ngayketthuc DATE,

Loaiphep VARCHAR(50),

Trangthai VARCHAR(50),

FOREIGN KEY (Manhanvien) REFERENCES tblNhanVien(Manhanvien) ON DELETE CASCADE

);

-- Tạo bảng Khen Thưởng & Kỷ Luật

CREATE TABLE tblKhenThuongKyLuat (

Manhanvien INT,

Ngay DATE,

Loai VARCHAR(50),

Lydo TEXT,

PRIMARY KEY (Manhanvien, Ngay),

FOREIGN KEY (Manhanvien) REFERENCES tblNhanVien(Manhanvien) ON DELETE CASCADE

);

-- Tạo bảng Hợp Đồng Lao Động

CREATE TABLE tblHopDong (

Manhanvien INT PRIMARY KEY,

Ngaybatdau DATE,

Ngayketthuc DATE,

Loaihopdong VARCHAR(50),

FOREIGN KEY (Manhanvien) REFERENCES tblNhanVien(Manhanvien) ON DELETE CASCADE

);

## 3.2 Chèn dữ liệu vào SQL

-- Chèn dữ liệu vào bảng tblChucVu

INSERT INTO tblChucVu (Machucvu, Tenchucvu, Mucluongcoban) VALUES

(1, 'Giám Đốc', 25000000),

(2, 'Trưởng Phòng Kinh Doanh', 18000000),

(3, 'Nhân Viên Nhân Sự', 9000000),

(4, 'Kế Toán Trưởng', 15000000),

(5, 'Marketing Manager', 13000000),

(6, 'Nhân Viên Kế Toán', 10000000),

(7, 'Nhân Viên IT', 11000000),

(8, 'Trưởng Phòng IT', 19000000),

(9, 'Nhân Viên Marketing', 9500000),

(10, 'Nhân Viên Kinh Doanh', 10000000);

-- Chèn dữ liệu vào bảng tblPhongBan

INSERT INTO tblPhongBan (Maphongban, Tenphongban, Motaphong) VALUES

(1, 'Phòng Kinh Doanh', 'Quản lý hoạt động kinh doanh'),

(2, 'Phòng Nhân Sự', 'Quản lý nhân viên và phúc lợi'),

(3, 'Phòng IT', 'Quản lý công nghệ thông tin'),

(4, 'Phòng Marketing', 'Quảng bá và tiếp thị sản phẩm'),

(5, 'Phòng Tài Chính', 'Quản lý tài chính công ty');

-- Chèn dữ liệu vào bảng tblNhanVien

INSERT INTO tblNhanVien (Manhanvien, Tennhanvien, Gioitinh, Ngaysinh, Diachi, Sodienthoai, Email, Maphongban, Machucvu, Ngayvaocongty, Trangthai) VALUES

(3, 'Pham Thanh Son', 'Nam', '1992-09-20', 'Đà Nẵng', '0123456791', 'sonpt@example.com', 3, 4, '2021-03-22', 'Đang làm'),

(4, 'Le Minh Tam', 'Nữ', '1988-11-02', 'Cần Thơ', '0123456792', 'tamlm@example.com', 4, 1, '2019-08-30', 'Đang làm'),

(5, 'Hoang Thi Bich', 'Nữ', '1993-01-10', 'Hải Phòng', '0123456793', 'bichht@example.com', 5, 5, '2020-11-18', 'Nghỉ việc'),

(1, 'Nguyen Thi Lan', 'Nữ', '1990-05-01', 'Hà Nội', '0123456789', 'lannt@example.com', 1, 2, '2020-01-01', 'Đang làm'),

(2, 'Tran Minh Tu', 'Nam', '1985-07-14', 'Hồ Chí Minh', '0123456790', 'tutm@example.com', 2, 3, '2018-04-15', 'Đang làm');

-- Chèn dữ liệu vào bảng tblLuong

INSERT INTO tblLuong (Manhanvien, Machucvu, Mucluong, Ngaycapnhatluong, Phucap) VALUES

(3,7, 10000000, '2021-06-01', 500000),

(4,4, 20000000, '2021-06-01', 3000000),

(5,5, 12000000, '2021-06-01', 2500000),

(1, 2, 18000000, '2023-01-01', 2000000),

(2, 3, 9000000, '2023-01-01', 1000000);

-- Insert dữ liệu vào tblLichLamViec

INSERT INTO tblLichLamViec (Manhanvien, Ngaylamviec, Giobatdau, Gioketthuc, Trangthai) VALUES

(1, '2021-06-01', '08:00', '17:00', 'Có mặt'),

(2, '2021-06-01', '08:00', '17:00', 'Có mặt'),

(3, '2021-06-01', '08:30', '17:30', 'Có mặt'),

(4, '2021-06-01', '09:00', '18:00', 'Nghỉ phép'),

(5, '2021-06-01', '08:00', '17:00', 'Vắng mặt');

-- Insert dữ liệu vào tblYeuCauPhep

INSERT INTO tblYeuCauPhep (MayeuCau, Manhanvien, Ngaybatdau, Ngayketthuc, Loaiphep, Trangthai) VALUES

(1, 1, '2021-07-01', '2021-07-05', 'Nghỉ phép', 'Được phê duyệt'),

(2, 2, '2021-07-10', '2021-07-12', 'Nghỉ ốm', 'Chờ phê duyệt'),

(3, 3, '2021-08-15', '2021-08-20', 'Nghỉ phép', 'Được phê duyệt'),

(4, 4, '2021-09-01', '2021-09-03', 'Nghỉ lễ', 'Được phê duyệt'),

(5, 5, '2021-09-05', '2021-09-07', 'Nghỉ phép', 'Chờ phê duyệt');

-- Insert dữ liệu vào tblKhenThuongKyLuat

INSERT INTO tblKhenThuongKyLuat (Manhanvien, Ngay, Loai, Lydo) VALUES

(1, '2021-06-15', 'Khen thưởng', 'Hoàn thành xuất sắc công việc'),

(2, '2021-06-20', 'Kỷ luật', 'Đi muộn 3 lần trong tháng'),

(3, '2021-06-25', 'Khen thưởng', 'Đạt thành tích xuất sắc trong dự án'),

(4, '2021-07-01', 'Khen thưởng', 'Có sáng kiến giúp cải tiến công việc'),

(5, '2021-07-10', 'Kỷ luật', 'Không hoàn thành công việc đúng thời hạn');

-- Insert dữ liệu vào tblHopDong

INSERT INTO tblHopDong (Manhanvien, Ngaybatdau, Ngayketthuc, Loaihopdong) VALUES

(1, '2021-06-01', '2023-06-01', 'Toàn thời gian'),

(2, '2019-08-15', '2022-08-15', 'Bán thời gian'),

(3, '2020-02-01', '2023-02-01', 'Toàn thời gian'),

(4, '2021-05-15', '2023-05-15', 'Thời vụ'),

(5, '2021-11-01', '2022-11-01', 'Bán thời gian');

## 3.3 In bảng dữ liệu

*3.3.1 In ra các bảng có trong SQL:*

|  |
| --- |
| **Câu lệnh:** |
| SELECT TABLE\_NAME  FROM INFORMATION\_SCHEMA.TABLES  WHERE TABLE\_TYPE = 'BASE TABLE'; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblChucVu; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblPhongBan; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblLuong; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblLichLamViec; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblYeuCauPhep; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblKhenThuongKyLuat; |
| **Kết quả:** |
|  |
| **Câu lệnh:** |
| SELECT \* FROM tblHopDong; |
| **Kết quả:** |
|  |

# Chương 4: Xây dựng các view

## 4.1 View danh sách nhân viên kèm theo tên phòng ban

Chức năng này dùng để xem danh sách nhân viên và phong ban nhân viên đó đang làm ở công ty.

|  |
| --- |
| **Câu lệnh View:** |
| -- 1. View danh sách nhân viên kèm theo tên phòng ban  CREATE VIEW vw\_NhanVien\_PhongBan AS  SELECT nv.Manhanvien, nv.Tennhanvien, pb.Tenphongban, nv.Trangthai  FROM tblNhanVien nv  JOIN tblPhongBan pb ON nv.Maphongban = pb.Maphongban;  -- Chức năng: Hiển thị danh sách nhân viên kèm theo tên phòng ban của họ.  -- Câu lệnh hiện ra view  select \* from vw\_NhanVien\_PhongBan |
| **Kết quả:** |
|  |

## 4.2 View danh sách nhân viên kèm theo chức vụ

- View này được tạo ra để xem được danh sách các nhân viên và kèm theo các chức vụ của mỗi người trong nhân viên và trạng thái của nhân viên đó

|  |
| --- |
| **Câu lệnh View:** |
| -- 2. View danh sách nhân viên kèm theo chức vụ  CREATE VIEW vw\_NhanVien\_ChucVu AS  SELECT nv.Manhanvien, nv.Tennhanvien, cv.Tenchucvu, nv.Trangthai  FROM tblNhanVien nv  JOIN tblChucVu cv ON nv.Machucvu = cv.Machucvu;  -- Chức năng: Hiển thị danh sách nhân viên với chức vụ của họ.  SELECT \* FROM vw\_NhanVien\_ChucVu |
| **Kết quả:** |
|  |

## 4.3 View Lương nhân viên mới

Chức năng này dùng để xem và kiểm soát được nhân viên mới vào trong công ty

|  |
| --- |
| **Câu lệnh View:** |
| -- 3. View lương nhân viên mới nhất  CREATE VIEW vw\_LuongMoiNhat AS  SELECT l.Manhanvien, nv.Tennhanvien, l.Mucluong, l.Ngaycapnhatluong  FROM tblLuong l  JOIN tblNhanVien nv ON l.Manhanvien = nv.Manhanvien  WHERE l.Ngaycapnhatluong = (SELECT MAX(Ngaycapnhatluong) FROM tblLuong WHERE Manhanvien = l.Manhanvien);  -- Chức năng: Hiển thị mức lương mới nhất của từng nhân viên.  go;  -- Xem nhân viên mới  SELECT \* FROM vw\_LuongMoiNhat |
| **Kết quả:** |
|  |

## 4.4 View Nhân viên được khen thưởng

- Hiện thị ra danh sách nhân viên được khen thưởng và hiện ra các thuộc tính bao gồm: Mã nhân viên, Tên nhân viên, Loại, Lý do

|  |
| --- |
| **Câu lệnh View:** |
| -- 4. View danh sách nhân viên đã từng được khen thưởng  CREATE VIEW vw\_NhanVien\_KhenThuong AS  SELECT kt.Manhanvien, nv.Tennhanvien, kt.Loai, kt.Lydo  FROM tblKhenThuongKyLuat kt  JOIN tblNhanVien nv ON kt.Manhanvien = nv.Manhanvien  WHERE kt.Loai = 'Khen thưởng';  -- Chức năng: Liệt kê những nhân viên đã từng được khen thưởng.  SELECT \* FROM vw\_NhanVien\_KhenThuong |
| **Kết quả:** |
|  |

## 4.5 View lịch làm việc của nhân viên

- View này dùng để xem lịch làm việc của từng nhân viên

|  |
| --- |
| **Câu lệnh View:** |
| CREATE VIEW vw\_LichLamViec AS  SELECT llv.Manhanvien, nv.Tennhanvien, llv.Ngaylamviec, llv.Giobatdau, llv.Gioketthuc, llv.Trangthai  FROM tblLichLamViec llv  JOIN tblNhanVien nv ON llv.Manhanvien = nv.Manhanvien;  -- in ra kết quả  SELECT \* FROM vw\_LichLamViec |
| **Kết quả:** |
|  |

## 5.6 View Danh sách hợp đồng

- View này cho xem hiện ra hợp đồng hết hạn, từ đó cho ta biết được nhân viên nào đã hết hợp đồng.

|  |
| --- |
| **Câu lệnh View:** |
| CREATE VIEW vw\_HopDongHieuLuc AS  SELECT hd.Manhanvien, nv.Tennhanvien, hd.Ngaybatdau, hd.Ngayketthuc, hd.Loaihopdong  FROM tblHopDong hd  JOIN tblNhanVien nv ON hd.Manhanvien = nv.Manhanvien  WHERE hd.Ngayketthuc >= GETDATE();  Select \* from vw\_HopDongHieuLuc |
| **Kết quả:** |
|  |

## 5.7 View yêu cầu nghỉ phép

- Trong View này sẽ xem được phê duyệt có được nghỉ làm hay không.

|  |
| --- |
| **Câu lệnh View:** |
| CREATE VIEW vw\_HopDongHieuLuc AS  SELECT hd.Manhanvien, nv.Tennhanvien, hd.Ngaybatdau, hd.Ngayketthuc, hd.Loaihopdong  FROM tblHopDong hd  JOIN tblNhanVien nv ON hd.Manhanvien = nv.Manhanvien  WHERE hd.Ngayketthuc >= GETDATE(); |
| **Kết quả:** |
|  |

## 5.8 View xem tổng số viên từng phòng

- Xem tổng số lượng nhân viên của từng phòng một, từ đó cho ta sẽ dõi được số lượng nhân viên trong từng phòng và số lượng nhân viên của công ty

|  |
| --- |
| **Câu lệnh View:** |
| CREATE VIEW vw\_TongNhanVien\_PhongBan AS  SELECT pb.Tenphongban, COUNT(nv.Manhanvien) AS SoLuongNhanVien  FROM tblPhongBan pb  LEFT JOIN tblNhanVien nv ON pb.Maphongban = nv.Maphongban  GROUP BY pb.Tenphongban;  -- Chức năng: Thống kê số lượng nhân viên trong từng phòng ban.  select \* from vw\_TongNhanVien\_PhongBan |
| **Kết quả:** |
|  |

## 4.9. View danh sách nhân viên nữ

- View này cho phép bạn thống các nhân viên nữ và các chức vụ của nhân viên đó.

|  |
| --- |
| **Câu lệnh View:** |
| CREATE VIEW vw\_NhanVienNu AS  SELECT Manhanvien, Tennhanvien, Gioitinh, Ngaysinh, Diachi, Sodienthoai, Email  FROM tblNhanVien  WHERE Gioitinh = 'Nữ';  -- Chức năng: Hiển thị danh sách tất cả nhân viên nữ trong công ty.  select \* from vw\_NhanVienNu |
| **Kết quả:** |
|  |

## 4.10 View Lương cao nhất

- View hiện ra số lương cao nhất và thông tin của người đó.

|  |
| --- |
| **Câu lệnh View:** |
| CREATE VIEW vw\_NhanVien\_LuongCaoNhat AS  SELECT nv.Manhanvien, nv.Tennhanvien, cv.Tenchucvu, l.Mucluong  FROM tblNhanVien nv  JOIN tblLuong l ON nv.Manhanvien = l.Manhanvien  JOIN tblChucVu cv ON nv.Machucvu = cv.Machucvu  WHERE l.Mucluong = (SELECT MAX(Mucluong) FROM tblLuong); |
| **Kết quả:** |
|  |

# CHƯƠNG 7 PHÂN QUYỀN VÀ BẢO VỆ CƠ SỞ DỮ LIỆU

## 7.1. Tạo tài khoản

Trong phần này, chúng ta sẽ tìm hiểu cách tạo tài khoản người dùng và phân quyền trong SQL Server. Quản lý tài khoản và quyền truy cập là một trong những yếu tố quan trọng để đảm bảo an ninh dữ liệu. Bên dưới là các câu lệnh về cách tạo tài khoản.

|  |
| --- |
| **Tạo tài khoản quản lý** |
| CREATE LOGIN QuanLy WITH PASSWORD = 'Password123!';  CREATE USER QuanLyUser FOR LOGIN QuanLy; |
| **Tạo tài khoản nhân viên** |
| CREATE LOGIN NhanVien WITH PASSWORD = 'Password123!';  CREATE USER NhanVienUser FOR LOGIN NhanVien; |
| **Tạo tài khoản khách hàng** |
| CREATE LOGIN KhachHang WITH PASSWORD = 'Password123!';  CREATE USER KhachHangUser FOR LOGIN KhachHang; |

## 7.2. Cấp quyền

Bên trên, chúng ta đã tìm hiểu các câu lệnh tạo tài khoản trong SQL. Bây giờ, hãy chuyển sang các lệnh cấp quyền để xác định quyền truy cập và hành động mà các tài khoản có thể thực hiện trên cơ sở dữ liệu.

|  |
| --- |
| **Cấp quyền quản lý** |
| --Cấp quyền quản lý quản lý có toàn quyền trong database  GRANT CONTROL ON DATABASE::Quan\_Ly\_Nhan\_Su TO QuanLyUser; |
| **Cấp quyền tài khoản nhân viên** |
| GRANT SELECT ON tblLuong TO NhanVienUser;  DENY UPDATE, DELETE ON tblLuong TO NhanVienUser; |
| **Cấp quyền tài khoản khách hàng** |
| GRANT SELECT ON tblPhongBan TO KhachHangUser;  GRANT SELECT ON tblChucVu TO KhachHangUser; |

Bên trên là cấp quyền 3 quyền cho hệ thống ngoài ra bên dưới đây là câu lệnh kiểm tra tất cả các quyền của tài khoàn quản lý

|  |
| --- |
| **Câu lệnh** |
| -- Kiểm tra tất cả quyền của tài khoản Quản lý  EXECUTE AS USER = 'QuanLyUser';  SELECT \* FROM fn\_my\_permissions(NULL, 'DATABASE');  REVERT; |
| **Kết quả** |
|  |

Kết quả hiện ra trong kiểm tra quyền của Quản lý không chỉ là 16 quyền hiện ra màn hình mà còn nhiều quyền bên dưới hình ảnh kết quả bên trên chỉ là mang tính tượng trưng. Tiếp đến là các câu lệnh kiểm tra tất cả các quyền của nhân viên

|  |
| --- |
| **Câu lệnh** |
| EXECUTE AS USER = 'NhanVienUser';  SELECT \* FROM fn\_my\_permissions(NULL, 'DATABASE');  REVERT; |
| **Kết quả** |
|  |

Tiếp đến là câu lệnh kiểm tra quyền tài khoản khách hàng

|  |
| --- |
| **Câu lệnh** |
| --Kiểm tra quyền của tài khoản Khách hàng  EXECUTE AS USER = 'KhachHangUser';  SELECT \* FROM fn\_my\_permissions(NULL, 'DATABASE');  REVERT; |
| **Kết quả** |
|  |

Bên trên là những câu lệnh để phân quyền trong SQL, nhưng làm thế nào để kiểm tra xem từng tài khoản SQL có những quyền gì. Tiếp đến phần sau tôi sẽ làm nói về cách bảo mật mã hóa dữ liệu của mình sao cho không bị dò dị thông tin người dùng ra bên ngoài.

## 7.3. Mã hóa

Mã hóa dữ liệu là một phương pháp bảo mật quan trọng nhằm bảo vệ thông tin nhạy cảm khỏi các nguy cơ rò rỉ, đánh cắp hoặc truy cập trái phép. Trong bối cảnh ngày nay, khi các cuộc tấn công mạng ngày càng tinh vi, mã hóa không chỉ là một tùy chọn mà đã trở thành tiêu chuẩn không thể thiếu trong quản lý dữ liệu.

Lợi ích của mã hóa:

* Bảo mật dữ liệu: Dữ liệu được mã hóa sẽ trở nên vô dụng với kẻ tấn công nếu không có khóa giải mã.
* Tuân thủ pháp luật: Nhiều quy định như GDPR hay HIPAA yêu cầu dữ liệu nhạy cảm phải được mã hóa.
* Bảo vệ thông tin nhạy cảm: Ngăn chặn việc truy cập trái phép đối với thông tin cá nhân, tài chính hoặc thông tin kinh doanh quan trọng.

Có hai phương pháp mã hóa dữ liệu:

**ENCRYPTBYPASSPHRASE():** Mã hóa bằng một cụm mật khẩu, không cần tạo khóa, dễ dùng nhưng phụ thuộc vào mật khẩu.

|  |
| --- |
| **Câu lệnh SQL** |
| SELECT ENCRYPTBYPASSPHRASE('MyPassphrase', 'Thông tin nhạy cảm');  SELECT CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE('MyPassphrase', ColumnData)); |

**ENCRYPTBYKEY():** Mã hóa bằng Symmetric Key, cung cấp bảo mật cao hơn nhờ quản lý khóa trong cơ sở dữ liệu.

|  |
| --- |
| **Câu lệnh SQL** |
| CREATE SYMMETRIC KEY MyKey  WITH ALGORITHM = AES\_256  ENCRYPTION BY PASSWORD = 'StrongPassword!';  OPEN SYMMETRIC KEY MyKey  DECRYPTION BY PASSWORD = 'StrongPassword!';  INSERT INTO MyTable (ColumnData)  VALUES (ENCRYPTBYKEY(KEY\_GUID('MyKey'), 'Thông tin nhạy cảm'));  SELECT CAST(DECRYPTBYKEY(ColumnData) AS NVARCHAR(MAX)) AS DecryptedData  FROM MyTable;  CLOSE SYMMETRIC KEY MyKey; |

Phương pháp sử dụng trong bài Tôi chọn ENCRYPTBYKEY() vì bảo mật cao hơn và phù hợp cho hệ thống quy mô lớn, đảm bảo dữ liệu an toàn và dễ quản lý

Để trước khi mã hóa một bảng (ví dụ mã hóa bảng nhân viên) thì ta cần tạo Symtrickey để sử dụng vào mã hóa.

|  |
| --- |
| **Câu lệnh SQL** |
| CREATE SYMMETRIC KEY MySymmetricKey  WITH ALGORITHM = AES\_256  ENCRYPTION BY PASSWORD = 'StrongPassword!'; |

Bước tiếp là tạo 1 bảng mới để chứa thông tin được mã hóa

|  |
| --- |
| **Câu lệnh SQL** |
| CREATE TABLE Nhanvien\_Encrypted (  Manhanvien VARCHAR(50), -- Lưu ý: Bảng gốc có Manhanvien là INT  Tennhanvien VARCHAR(100),  Email\_MaHoa VARBINARY(MAX),  Sodienthoai\_MaHoa VARBINARY(MAX),  Maphongban INT,  Machucvu INT  ); |

Đến phần mã hóa thông tin Nhân Viên trong cột Email, và Số điẹn thoại của nhân viên tráh bị dò dị thông tin.

|  |
| --- |
| **Câu lệnh SQL** |
| OPEN SYMMETRIC KEY MySymmetricKey  DECRYPTION BY PASSWORD = 'StrongPassword!';  INSERT INTO Nhanvien\_Encrypted (Manhanvien, Tennhanvien, Email\_MaHoa, Sodienthoai\_MaHoa, Maphongban, Machucvu)  SELECT  Manhanvien,  Tennhanvien,  ENCRYPTBYKEY(KEY\_GUID('MySymmetricKey'), CAST(Email AS NVARCHAR(MAX))),  ENCRYPTBYKEY(KEY\_GUID('MySymmetricKey'), CAST(Sodienthoai AS NVARCHAR(MAX))),  Maphongban,  Machucvu  FROM  tblNhanVien;  CLOSE SYMMETRIC KEY MySymmetricKey;  GO  SELECT \* FROM Nhanvien\_Encrypted; |
| **Kết quả** |
|  |

## 7.4. Giải mã hóa

Để giải được mã hóa thì ta cũng cần có khóa Symmetric nhưng vì ví dụ trên khi ta tạo khóa đã tồn tại. Bây giờ ta chỉ cần mở khóa Synmmetric là được. Bên dưới là câu lệnh mở khóa Symmetric và câu lệnh giải mã hoá.

|  |
| --- |
| **Câu lệnh SQL** |
| -- open Symmetric  OPEN SYMMETRIC KEY MySymmetricKey  DECRYPTION BY PASSWORD = 'StrongPassword!';  GO  -- xem giai ma hoa  SELECT  CONVERT(VARCHAR(MAX),DECRYPTBYKEY(Email\_MaHoa)) as Emaol\_De  ,  CONVERT(VARCHAR(MAX),DECRYPTBYKEY(Sodienthoai\_MaHoa)) as Sodienthoai\_De  FROM  Nhanvien\_Encrypted;  GO  -- đóng khóa symmetric  CLOSE SYMMETRIC KEY MySymmetricKey;  GO |
| **Kết quả** |
|  |

Bên trên là một ví dụ về cách mã hóa và giải mã dữ liệu của một bảng trong cơ sở dữ liệu. Nếu bạn cần áp dụng cho các bảng khác, bạn chỉ cần thay đổi các thuộc tính phù hợp của bảng đó. Quy trình vẫn tương tự, cho phép bạn dễ dàng bảo mật dữ liệu trong nhiều bảng khác nhau.

Phần trình bày về mã hóa và giải mã hóa xin dừng tại đây. Tuy nhiên, ngoài phương pháp sử dụng khóa Symmetric (khóa đối xứng) mà tôi đã minh họa, còn có nhiều phương pháp mã hóa khác để tăng cường bảo mật. Bạn có thể chọn cách tạo và quản lý khóa riêng phù hợp với nhu cầu cụ thể của mình.