**ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG**

**KHOA THỐNG KÊ – TIN HỌC**

**🙡🕮🙣**

**A blue and white logo

Description automatically generated**

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Giảng viên hướng dẫn : Cô Cao Thị Nhâm

Lớp : 48K21.1

Nhóm : 48K211.12

Thành viên : Nguyễn Hữu Toàn

Đinh Thị Diễm Quỳnh

Nguyễn Ngọc Hoàng Ngân

Trương Thị Minh Hiền

Nguyễn Thị Thiên Ân

Đà Nẵng, 2024

# và 2

Yêu cầu:

* Hiệu chỉnh lại các bảng sao cho ít nhất phải ở dạng chuẩn 2NF
* Thiết kế chi tiết các bảng

## A diagram of a network Description automatically generated with medium confidenceClass Diagram

## A diagram of a computer Description automatically generatedSơ đồ quan hệ (Bổ sung)

## Thiết kế chi tiết các bảng

### Bảng nhân viên (Thêm cột TrangThai vào bảng)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: NHANVIEN** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaNV | Char(6) | Not null | PK | Khóa chính của bảng |
| MaCV | Char(6) | Not null |  | Khóa ngoại |
| TenNV | Varchar(20) | Not null |  | Họ và tên |
| NgaySinh | Date | Not null |  |  |
| SDT | Int | Unique |  | Số điện thoại |
| Email | Varchar(20) | Unique |  |  |
| GioiTinh | Bit | Not null |  | * 0: Nữ * 1: Nam |
| DiaChi | Varchar(50) | Not null |  |  |
| TrangThai | Bit | Not null |  | * 0: Đã nghỉ việc * 1: Chưa nghỉ việc |

### Bảng chức vụ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: CHUCVU** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaCV | Char(6) | Not null | PK | Khóa chính của bảng |
| TenCV | Varchar(10) | Not null |  |  |

### Bảng vi phạm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: VIPHAM** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaVP | Char(6) | Not null | PK | Khóa chính của bảng |
| Loi | Varchar(15) | Not null |  | Lỗi vi phạm |
| MaNV | Char(6) | Not null |  | Khóa ngoại |

### Bảng hợp đồng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: HOPDONG** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaHD | Char(5) | Not null | PK | Khóa chính của bảng |
| MaNV | Char(6) | Not null |  | Khóa ngoại |
| LoaiHD | Varchar(20) | Not null |  |  |
| NgayBatDau | Date | Not null |  |  |
| NgayKetThuc | Date |  |  |  |

### Bảng lương

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: LUONG** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaLuong | Char(6) | Not null | PK | Khóa chính của bảng |
| MaNV | Char(6) | Not null |  | Khóa ngoại |
| LuongCB | Float | Not null |  |  |

### Bảng tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: TAIKHOAN** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| TenTK | Varchar(20) | Not null | PK | Khóa chính của bảng |
| MK | Varchar(20) | Not null |  |  |
| MaNV | Char(6) | Not null |  | Khóa ngoại |

### Bảng chấm công

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: CHAMCONG** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaCC | Char(12) | Not null | PK | Khóa chính của bảng |
| TenTK | Varchar(20) | Not null |  | Khóa ngoại |
| NgayCC | Date | Not null |  | Ngày chấm công |
| BatDau | time | Not null |  | Giờ vào ca |
| KetThuc | time | Not null |  | Giờ kết ca |

### Bảng lương chi tiết (Bảng mới bổ sung)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng: BangLuong** | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| MaBL | Char(6) | Not null | PK | Khóa chính của bảng |
| MaNV | char(6) | Not null |  | Khóa ngoại |
| Thang | int | Not null |  | Tháng tính lương |
| Nam | int | Not null |  | Năm tính lương |
| SoGioCong | int | Not null |  | Tổng số giờ làm trong tháng |
| TongLuong | float | Not null |  | Tổng tiền lương của tháng |

Yêu cầu: Xây dựng cơ sở dữ liệu cho bản thiết kế ở bước trên.

CREATE TABLE NHANVIEN (

MaNV CHAR(6) primary key not null,

TenNV VARCHAR(20) not null,

MaCV CHAR(6) not null,

NgaySinh DATE not null,

SDT CHAR(10) not null unique,

Email VARCHAR(30) not null unique,

GioiTinh BIT not null check (GioiTinh in (0,1)),

DiaChi VARCHAR(50) not null,

FOREIGN KEY (MaCV) REFERENCES CHUCVU(MaCV) )

-- Bảng hợp đồng

CREATE TABLE HOPDONG (

MaHD CHAR(5) primary key not null,

MaNV CHAR(6) not null,

LoaiHD VARCHAR(20) not null,

NgayBatDau DATE not null,

NgayKetThuc DATE,

FOREIGN KEY (MaNV) REFERENCES NHANVIEN(MaNV) )

-- Bảng lương

CREATE TABLE LUONG (

MaLuong CHAR(6) primary key not null,

MaNV CHAR(6) not null,

LuongCB FLOAT not null,

FOREIGN KEY (MaNV) REFERENCES NHANVIEN(MaNV) )

-- Bảng tài khoản

CREATE TABLE TAIKHOAN (

TenTK VARCHAR(20) primary key not null,

MaNV CHAR(6) not null,

MK VARCHAR(20) not null,

FOREIGN KEY (MaNV) REFERENCES NHANVIEN(MaNV) )

-- Bảng chấm công

CREATE TABLE CHAMCONG (

MaCC CHAR(12) primary key not null,

TenTK VARCHAR(20) not null,

NgayCC DATE not null,

BatDau TIME not null,

KetThuc TIME not null,

FOREIGN KEY (TenTK) REFERENCES TAIKHOAN(TenTK) )

-- Bảng vi phạm

CREATE TABLE VIPHAM (

MaVP CHAR(6) primary key not null,

MaNV CHAR(6) not null,

Loi VARCHAR(15) not null,

FOREIGN KEY (MaNV) REFERENCES NHANVIEN(MaNV) )

-- Bảng lương ( Bổ sung)

CREATE TABLE BangLuong (

MaBL CHAR(6) primary key not null,

MaNV CHAR(6) not null,

Thang INT not null,

Nam INT not null,

SoGioCong INT not null,

TongLuong FLOAT not null,

FOREIGN KEY (MaNV) REFERENCES NHANVIEN(MaNV) )

--Thêm cột TrangThai vào bảng NHANVIEN

ALTER TABLE NHANVIEN

ADD TrangThai bit -- 0 -> đã nghỉ việc, 1 -> chưa nghỉ việc

Yêu cầu: Xây dựng:

* Các module tạo dữ liệu dump cho các bảng trong cơ sở dữ liệu. Mỗi bảng ít nhất 1000 dòng dữ liệu;
* Ít nhất 10 module trong cơ sở dữ liệu để phục vụ các thao tác xử lý dữ liệu (kiểm tra sự hợp lệ của dữ liệu, xử lý các thao tác nghiệp vụ phức hợp,...)

## Tạo dữ liệu cho các bảng

--- Module 1000 dữ liệu bảng CHUCVU

create PROCEDURE p\_InsertCHUCVU

AS

BEGIN

DECLARE @i INT = 1

DECLARE @MaCV CHAR(6)

DECLARE @TenCV VARCHAR(50)

WHILE @i <= 1000

BEGIN

SET @MaCV = RIGHT('000000' + CAST(@i AS VARCHAR(6)), 6)

SET @TenCV = 'Chuc vu ' + CAST(@i AS VARCHAR(50))

INSERT INTO CHUCVU (MaCV, TenCV)

VALUES (@MaCV, @TenCV)

SET @i = @i + 1

END

END

--- Module 1000 dữ liệu bảng NHANVIEN

create PROCEDURE p\_InsertNHANVIEN

AS

BEGIN

DECLARE @i INT = 1,

@MaCV CHAR(6),

@Duong varchar(50),

@TrangThai BIT

DECLARE @steetlist TABLE (Duong varchar(50))

INSERT INTO @steetlist

VALUES ('Nguyen Sinh Sac'), ('Nguyen Huu Tho'), ('Tran Cao Van'),

('Nguyen Van Linh'), ('Ngu Hanh Son'), ('Le Loi'), ('Le Tu Nhat Thong')

WHILE @i < 1020

BEGIN

SELECT TOP 1 @MaCV = MaCV

FROM CHUCVU

ORDER BY NEWID()

SELECT TOP 1 @Duong = Duong

FROM @steetlist

ORDER BY NEWID()

SET @TrangThai = CASE WHEN @i % 2 = 0 THEN 1 ELSE 0 END;

INSERT INTO NHANVIEN (MaNV, TenNV, MaCV, NgaySinh, SDT, Email, GioiTinh, DiaChi, TrangThai)

VALUES (

RIGHT('000000' + CAST(@i AS VARCHAR(6)), 6),

'Nhan Vien ' + CAST(@i AS VARCHAR(20)),

@MaCV,

CAST(DATEADD(DAY, FLOOR(RAND() \* DATEDIFF(DAY, '2000-01-01', GETDATE())), '2000-01-01') AS DATE),

'09' + RIGHT('01029839280' + CAST(@i AS VARCHAR), 8),

'nhanvien' + CAST(@i AS VARCHAR(20)) + '@gmail.com',

CASE WHEN @i % 2 = 0 THEN 1 ELSE 0 END,

'So ' + CAST(@i AS VARCHAR(20)) + ', ' + @Duong + ', TP Da Nang',

@TrangThai -- Thêm trạng thái vào INSERT

)

SET @i = @i + 1

END

END

--- Module 1000 dữ liệu bảng VIPHAM

create PROCEDURE p\_InsertVIPHAM

AS

BEGIN

DECLARE @i INT = 1,

@MaNV CHAR(6),

@MaVP CHAR(6),

@Loi VARCHAR(15)

DECLARE @LoiList TABLE (Loi VARCHAR(15));

INSERT INTO @LoiList (Loi) VALUES ('Di tre'), ('Vang'), ('Khac'), ('Khong mang giay'), ('Ve som')

WHILE @i < 1020

BEGIN

SELECT TOP 1 @MaNV = MaNV

FROM NHANVIEN

ORDER BY NEWID()

SET @MaVP = RIGHT('000000' + CAST(@i AS VARCHAR(6)), 6)

SELECT TOP 1 @Loi = Loi

FROM @LoiList

ORDER BY NEWID()

INSERT INTO VIPHAM (MaVP, MaNV, Loi)

VALUES (@MaVP, @MaNV, @Loi)

SET @i = @i + 1

END

END

--- Module 1000 dữ liệu bảng HOPDONG

create PROCEDURE p\_InsertHOPDONG

AS

BEGIN

DECLARE @i INT = 1

DECLARE @MaNV CHAR(6)

DECLARE @MaHD CHAR(5)

DECLARE @LoaiHD VARCHAR(20)

DECLARE @NgayBatDau DATE

DECLARE @NgayKetThuc DATE

DECLARE @LoaiHDList TABLE (LoaiHD VARCHAR(20))

INSERT INTO @LoaiHDList (LoaiHD) VALUES ('Full-time'), ('Part-time'), ('Thu Viec')

WHILE @i < 1020

BEGIN

SELECT TOP 1 @MaNV = MaNV

FROM NHANVIEN

ORDER BY NEWID()

SET @MaHD = RIGHT('00000' + CAST(@i AS VARCHAR(5)), 5);

SELECT TOP 1 @LoaiHD = LoaiHD

FROM @LoaiHDList

ORDER BY NEWID()

SET @NgayBatDau = DATEADD(DAY, CAST(RAND() \* 1461 AS INT) - 1460, GETDATE())

IF RAND() < 0.8

SET @NgayKetThuc = DATEADD(DAY, CAST(RAND() \* 30 AS INT), @NgayBatDau)

ELSE

SET @NgayKetThuc = NULL

INSERT INTO HOPDONG (MaHD, MaNV, LoaiHD, NgayBatDau, NgayKetThuc)

VALUES (@MaHD, @MaNV, @LoaiHD, @NgayBatDau, @NgayKetThuc);

SET @i = @i + 1

END

END

--- Module 1000 dữ liệu bảng LUONG

create PROCEDURE p\_InsertLUONG

AS

BEGIN

DECLARE @i INT = 1

DECLARE @MaNV CHAR(6)

DECLARE @MaLuong CHAR(6)

DECLARE @LuongCB FLOAT

DECLARE @LoaiHD VARCHAR(20)

WHILE @i < 1020

BEGIN

SELECT TOP 1 @MaNV = MaNV

FROM NHANVIEN

ORDER BY NEWID()

SET @MaLuong = RIGHT('000000' + CAST(@i AS VARCHAR(6)), 6)

SELECT TOP 1 @LoaiHD = LoaiHD

FROM HOPDONG

WHERE MaNV = @MaNV

IF @LoaiHD = 'Full-time'

SET @LuongCB = 25000

ELSE IF @LoaiHD = 'Part-time'

SET @LuongCB = 20000

ELSE

SET @LuongCB = 0

INSERT INTO LUONG (MaLuong, MaNV, LuongCB)

VALUES (@MaLuong, @MaNV, @LuongCB)

SET @i = @i + 1;

END

END

--- Module 1000 dữ liệu bảng TAIKHOAN

create PROCEDURE p\_InsertTAIKHOAN

AS

BEGIN

DECLARE @i INT = 1

DECLARE @TenTK VARCHAR(20)

DECLARE @MaNV CHAR(6)

DECLARE @MK VARCHAR(20)

WHILE @i < 1020

BEGIN

SET @TenTK = 'TK' + RIGHT('000000' + CAST(@i AS VARCHAR(6)), 6)

SELECT TOP 1 @MaNV = MaNV

FROM NHANVIEN

ORDER BY NEWID()

SET @MK = LEFT(CONVERT(VARCHAR(36), NEWID()), 10) + CAST(@i AS VARCHAR(6))

INSERT INTO TAIKHOAN (TenTK, MaNV, MK)

VALUES (@TenTK, @MaNV, @MK)

SET @i = @i + 1

END

END

--- Module 1000 dữ liệu bảng CHAMCONG

create PROCEDURE p\_InsertCHAMCONG

AS

BEGIN

DECLARE @i INT = 0

DECLARE @MaCC CHAR(12)

DECLARE @TenTK VARCHAR(20)

DECLARE @NgayCC DATE

DECLARE @BatDau TIME

DECLARE @KetThuc TIME

WHILE @i < 1600

BEGIN

SET @MaCC = 'CC' + RIGHT('000000' + CAST(@i AS VARCHAR(6)), 6);

SELECT TOP 1 @TenTK = TenTK

FROM TAIKHOAN

ORDER BY NEWID()

SET @NgayCC = DATEADD(DAY, CAST(RAND() \* 1461 AS INT) - 1460, GETDATE())

SET @BatDau = DATEADD(MINUTE, (RAND() \* (960 - 540) + 540), CAST(@NgayCC AS DATETIME))

SET @KetThuc = DATEADD(HOUR, (4 + (RAND() \* 1)), @BatDau)

INSERT INTO CHAMCONG (MaCC, TenTK, NgayCC, BatDau, KetThuc)

VALUES (@MaCC, @TenTK, @NgayCC, @BatDau, @KetThuc)

SET @i = @i + 1;

END

END

-----------------

--1.

exec p\_InsertCHUCVU

--2.

exec p\_InsertNHANVIEN

--3.

exec p\_InsertHOPDONG

--4.

exec p\_InsertLUONG

--5.

exec p\_InsertVIPHAM

--6.

exec p\_InsertTAIKHOAN

--7.

exec p\_InsertCHAMCONG

## Xây dựng module trong cơ sở dữ liệu

**Module 1: Thủ tục tính tổng giờ làm**

|  |  |
| --- | --- |
| Input | @MaNV, @Thang, @Nam |
| Output | @TongSoGioLam |
| Process | 1. Lấy@TuNgay, @DenNgay từ input. 2. Thiết lập giá trị:  * @TuNgay là ngày đầu tiên của tháng và năm đã cho. * @DenNgay là ngày cuối cùng của tháng và năm đã cho.  1. Tính @TongSoGioLam: tổng(lấy phút BatDau -> KetThuc) / 60 để ra ‘giờ’, tại điều kiện: ChamCong.NgayCC nằm giữa @TuNgay and @DenNgay  * @TongSoGioCong không được phép null |

CREATE PROCEDURE p\_TinhTongSoGioLam @MaNV CHAR(6),

@Thang INT,

@Nam INT,

@TongSoGioLam FLOAT OUTPUT

AS

BEGIN

DECLARE @TuNgay DATE, @DenNgay DATE

SET @TuNgay = CAST(CONCAT(@Nam, '-', @Thang, '-01') AS DATE)

SET @DenNgay = EOMONTH(@TuNgay)

SELECT

@TongSoGioLam = SUM(DATEDIFF(MINUTE, BatDau, KetThuc)) / 60

FROM CHAMCONG cc JOIN TAIKHOAN tk ON cc.TenTK = tk.TenTK

JOIN NHANVIEN nv ON tk.MaNV = nv.MaNV

WHERE nv.MaNV = @MaNV AND (NgayCC BETWEEN @TuNgay AND @DenNgay)

SET @TongSoGioLam = ISNULL(@TongSoGioLam, 0)

END

**Module 2: Thủ tục lưu trữ để tính tổng lương cho mỗi nhân viên**

|  |  |
| --- | --- |
| Input | @Thang, @Nam |
| Output | N/A |
| Process | 1. Lấy @Thang và @Nam từ input 2. Tạo một bảng tạm @tableNV để lưu mã nhân viên. 3. Chọn nhân viên hợp lệ, điều kiện: mã nhân viên có TrangThai là 1 (còn làm việc) đưa vào bảng tạm @tableNV. 4. Tính @lentableNV = COUNT (nhân viên trong bảng tạm) và cho biến @dem = 1 5. Sử dụng WHILE để duyệt từng nhân viên trong bảng tạm.  * Với mỗi lần lặp:   + MaBL = lấy Max(MaBL hiện tại) + 1   + Lấy @MaNV từ bảng tạm.   + Gọi p\_TinhTongSoGioLam với @MaNV, @Thang, @Nam, @TongSoGioLam OUTPUT   + Lấy @LuongCB từ bảng LUONG, điều kiện: MaNV = @MaNV   + Tính @TongLuong = @LuongCD \* @TongSoGioLam   + Tăng @dem thêm 1. |

alter PROCEDURE p\_TinhLuongThang @Thang INT,

@Nam INT

AS

BEGIN

DECLARE @MaNV CHAR(6),

@TongSoGioLam FLOAT,

@LuongCB FLOAT,

@TongLuong FLOAT

DECLARE @tableNV TABLE (MaNV CHAR(6))

INSERT INTO @tableNV (MaNV)

SELECT MaNV FROM NHANVIEN WHERE TrangThai = 1

DECLARE @lentableNV INT = (SELECT COUNT(\*) FROM @tableNV)

DECLARE @dem INT = 1

WHILE @dem <= @lentableNV

BEGIN

SELECT TOP 1 @MaNV = MaNV

FROM @tableNV

WHERE MaNV NOT IN (SELECT TOP (@dem - 1) MaNV FROM @tableNV)

EXEC p\_TinhTongSoGioLam @MaNV, @Thang, @Nam, @TongSoGioLam OUTPUT

SELECT @LuongCB = LuongCB FROM LUONG WHERE MaNV = @MaNV

SET @TongSoGioLam = ISNULL(@TongSoGioLam, 0)

SET @LuongCB = ISNULL(@LuongCB, 0)

SET @TongLuong = @TongSoGioLam \* @LuongCB

INSERT INTO BangLuong (MaBL, MaNV, Thang, Nam, SoGioCong, TongLuong)

VALUES (

RIGHT('000000' + CAST(ISNULL((SELECT MAX(CAST(MaBL AS INT)) FROM BangLuong), 0) + 1 AS VARCHAR), 6),

@MaNV,

@Thang,

@Nam,

@TongSoGioLam,

@TongLuong

)

SET @dem = @dem + 1

END

END

**Module 3: Trigger kiểm tra tính hợp lệ của ngày hợp đồng**

|  |  |
| --- | --- |
| Loại | After |
| Sự kiện | Insert, Update |
| Bảng | HOPDONG |
| Process | 1. Lấy NgayBatDau từ bảng inserted 2. Kiểm tra ngày bắt đầu, nếu: NgayKetThuc < NgayBatDau or NgayKetThuc = NgayBatDau and NgayKetThuc is not null  * Đúng: in ‘Ngày kết thúc không được nhỏ hơn hoặc trùng với ngày bắt đầu’ và rollback * Sai: tiếp tục. |

CREATE TRIGGER t\_CheckHopDong

ON HOPDONG

AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @NgayBatDau DATE, @NgayKetThuc DATE

SELECT @NgayBatDau = NgayBatDau FROM INSERTED

SELECT @NgayKetThuc = NgayKetThuc FROM INSERTED

IF (@NgayKetThuc IS NOT NULL AND @NgayKetThuc <= @NgayBatDau)

BEGIN

PRINT (N'Ngày kết thúc không được nhỏ hơn hoặc trùng với ngày bắt đầu!')

ROLLBACK

END

END

--Test

UPDATE HOPDONG

SET NgayKetThuc = '2023-12-31'

WHERE MaHD = 'HD001'

select \* from HOPDONG

**Module 4: Trigger kiểm tra tính hợp lệ của số điện thoại**

|  |  |
| --- | --- |
| Loại | After |
| Sự kiện | Insert, Update |
| Bảng | NHANVIEN |
| Process | 1. Lấy SDT từ bảng inserted 2. Nếu độ dài của SDT, nếu: SDT <> 10  * Đúng: in 'Số điện thoại phải có đúng 10 chữ số!!' và rollback * Sai: tiếp tục |

CREATE TRIGGER t\_CheckSDT

ON NHANVIEN

AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @sdt char(10)

SELECT @sdt = SDT FROM inserted

IF LEN(@sdt) <> 10

BEGIN

PRINT(N'Số điện thoại phải có đúng 10 chữ số!!')

ROLLBACK

END

END

**Module 5: Thủ tục kiểm tra số lần vi phạm của mỗi nhân viên**

|  |  |
| --- | --- |
| Input | N/A |
| Output | N/A |
| Process | 1. Kết nối các bảng: sử dụng JOIN để kết nối ba bảng NHANVIEN, CHUCVU, VIPHAM 2. Thực hiện câu lệnh SELECT để lấy thông tin cần thiết:  * Lấy MaNV, TenNV, TenCV. * Đếm số lần vi phạm COUNT(MaVP) cho từng nhân viên.  1. Dữ liệu được nhóm theo MaNV, TenNV, TenCV. 2. Sắp xếp theo số lần vi phạm giảm dần. |

CREATE PROCEDURE p\_CheckSoLanViPham

AS

BEGIN

SELECT NHANVIEN.MaNV,

NHANVIEN.TenNV,

CHUCVU.TenCV,

COUNT(VIPHAM.MaVP) AS SoLanViPham

FROM NHANVIEN JOIN CHUCVU ON NHANVIEN.MaCV = CHUCVU.MaCV

JOIN VIPHAM ON NHANVIEN.MaNV = VIPHAM.MaNV

GROUP BY NHANVIEN.MaNV, NHANVIEN.TenNV, CHUCVU.TenCV

ORDER BY SoLanViPham DESC

END

**Module 6: Thủ tục kiểm tra thông tin nhân viên đã tồn tại trong hệ thông hay chưa nếu biết họ tên và số điện thoại**

|  |  |
| --- | --- |
| Input | @TenNV, @SDT |
| Output | N/A |
| Process | 1. Lấy @TenNV và @SDT từ input. 2. Kiểm tra xem có bản ghi nào trong bảng NHANVIEN, điều kiện: TenNV = @TenNV và SDT = @SDT.  * Đúng: in N'Nhân viên đã tồn tại trong hệ thống!!' * Sai: in N'Nhân viên không tồn tại trong hệ thống!!' |

CREATE PROCEDURE p\_CheckNhanVien @TenNV VARCHAR(20),

@SDT CHAR(10)

AS

BEGIN

IF EXISTS (SELECT 1 FROM NHANVIEN WHERE TenNV = @TenNV AND SDT = @SDT)

BEGIN

PRINT N'Nhân viên đã tồn tại trong hệ thống!!'

END

ELSE

BEGIN

PRINT N'Nhân viên không tồn tại trong hệ thống!!'

END

END

**Module 7: Trigger xóa thông tin nhân viên**

|  |  |
| --- | --- |
| Loại | Instead Of |
| Sự kiện | Delete |
| Bảng | NHANVIEN |
| Process | 1. Kiểm tra TrangThai trong bảng deleted, nếu: TrangThai = 0  * Đúng: in 'Nhân viên không tồn tại!!' * Sai: update bảng NHANVIEN, set TrangThai = 0, tại điều kiện: MaNV phải ở trong bảng deleted và in 'Đã xóa thành công!!' |

CREATE TRIGGER t\_DeleteNhanVien

ON NHANVIEN

INSTEAD OF DELETE

AS

BEGIN

IF (SELECT TrangThai FROM deleted) = 0

PRINT'Nhân viên không tồn tại!!'

ELSE

BEGIN

UPDATE NHANVIEN

SET TrangThai = 0

WHERE MaNV IN (SELECT MaNV FROM deleted)

PRINT'Đã xóa thành công!!'

END

END

**Module 8: Trigger kiểm tra tính hợp lệ của email**

|  |  |
| --- | --- |
| Loại | After |
| Sự kiện | Insert, Update |
| Bảng | NHANVIEN |
| Process | 1. Lấy Email từ bảng inserted 2. Kiểm tra định dạng của Email, nếu: @Email NOT LIKE '%\_@\_\_%.\_\_%'  * Đúng: in 'Email không hợp lệ!!' và rollback. * Sai: tiếp tục. |

alter TRIGGER t\_CheckEmail

ON NHANVIEN

AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @Email VARCHAR(30)

SELECT @Email = Email FROM INSERTED;

IF @Email NOT LIKE '%\_@\_\_%.\_\_%'

BEGIN

PRINT'Email không hợp lệ!!'

ROLLBACK

END

END

--Test

INSERT INTO NHANVIEN (MaNV, TenNV, MaCV, NgaySinh, SDT, Email, GioiTinh, DiaChi)

VALUES ('NV004', 'Nguyen Van B', '000002', '1990-01-01', '1234567892', 'emailkhonghople12gmail.com', 1, 'Hà Nội')

**Module 9: Thủ tục kiểm tra tài khoản và mật khẩu khi đăng nhập và trả về tên chức vụ để phân quyền truy cập**

|  |  |
| --- | --- |
| Input | @TK, @MK |
| Output | @TenCV |
| Process | 1. Lấy @TK và @MK từ input . 2. Kiểm tra nếu @TK và @MK, điều kiện: @TK IS NULL OR @MK IS NULL OR @TK = '' OR @MK = ''.  * Đúng: in 'Tài khoản hoặc mật khẩu không được để trống' và return. * Sai: tiếp tục.  1. Lấy tên chức vụ gán cho @TenCV từ bảng CHUCVU qua bảng TAIKHOAN và NHANVIEN, điều kiện TenTK = @TK và MK = @MK. 2. Kiểm tra @TenCV, nếu: @TenCV IS NULL  * Đúng: in 'Tài khoản, mật khẩu không hợp lệ hoặc không tìm thấy chức vụ'. * Sai: in  'Chức vụ của nhân viên: ' + @TenCV  1. Nếu không tìm thấy, thông báo ‘Tài khoản, mật khẩu không hợp lệ hoặc không tìm thấy chức vụ’. |

ALTER PROCEDURE p\_CheckChucVu

@TK NVARCHAR(20),

@MK NVARCHAR(20),

@TenCV VARCHAR(10) OUTPUT

AS

BEGIN

IF @TK IS NULL OR @MK IS NULL OR @TK = '' OR @MK = ''

BEGIN

SELECT 'Tài khoản hoặc mật khẩu không được để trống' AS ThongBao

RETURN

END

SELECT @TenCV = cv.TenCV

FROM TAIKHOAN tk

JOIN NHANVIEN nv ON tk.MaNV = nv.MaNV

JOIN CHUCVU cv ON nv.MaCV = cv.MaCV

WHERE tk.TenTK = @TK AND tk.MK = @MK

IF @TenCV IS NULL

BEGIN

SELECT 'Tài khoản, mật khẩu không hợp lệ hoặc không tìm thấy chức vụ' AS ThongBao

END

ELSE

BEGIN

SELECT 'Chức vụ của nhân viên: ' + @TenCV AS ThongBao

END

END

--Test

DECLARE @TenCV VARCHAR(10)

EXEC p\_CheckChucVu @TK = 'TK000002', @MK = '9A61078B-B2', @TenCV = @TenCV OUTPUT

select \* from TAIKHOAN

**Module 10: Thủ tục xem bảng lương của nhân viên**

|  |  |
| --- | --- |
| Input | @MaNV, @Thang, @Nam |
| Output | N/A |
| Process | 1. Lấy MaBL, Thang, Nam, SoGioCong, TongLuong từ BangLuong, tại điều kiện: MaNV = @MaNV và Thang = @Thang và Nam = @Nam 2. Kiểm tra có bao nhiêu hàng được trả về, nếu: @@ROWCOUNT = 0  * Đúng: in 'Không tìm thấy!!'. * Sai: tiếp tục. |

alter PROCEDURE p\_XemBangLuong @MaNV CHAR(6),

@Thang INT,

@Nam INT

AS

BEGIN

SELECT MaBL, Thang, Nam, SoGioCong, TongLuong

FROM BangLuong

WHERE MaNV = @MaNV AND Thang = @Thang AND Nam = @Nam

IF @@ROWCOUNT = 0

BEGIN

SELECT 'Không tìm thấy!!' AS THONGBAO

END

END

--Test

EXEC p\_XemBangLuong @MaNV = '000006', @Thang = 10, @Nam = 2024

Yêu cầu: Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu

## QUẢN LÝ NGƯỜI DÙNG

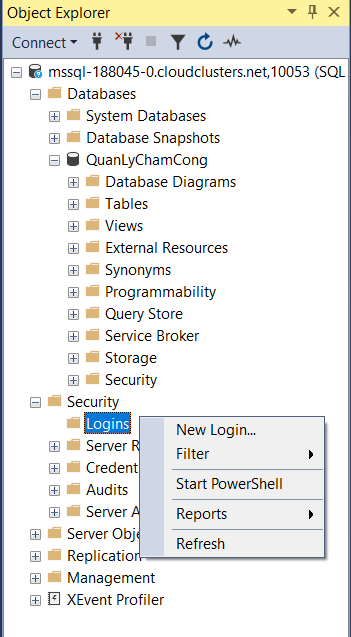
**Tạo tài khoản**

Tạo 2 tài khoản, gồm:

* Tài khoản dành cho nhân viên (NhanVien)
* Tài khoản dành cho quản lý (QuanLy)

Các bước tạo tài khoản:

**Bước 1: Vào thự mục Sercurity -> Logins -> nhấp chuột phải chọn New Login**

****

**Bước 2: Thiết lập thông tin cơ bản của login**

* Tên đăng nhập (Login Name): NhanVien/QuanLy
* Phương thức xác thực: SQL Server authentication (Các thông tin đăng nhập được tạo trong SQL Server không dựa trên tài khoản người dùng Windows, phải cung cấp thông tin xác thực (tên đăng nhập và mật khẩu) mỗi khi kết nối)
* Chọn database cho phép truy cập (Default database): QuanLyChamCong

**A screenshot of a computer

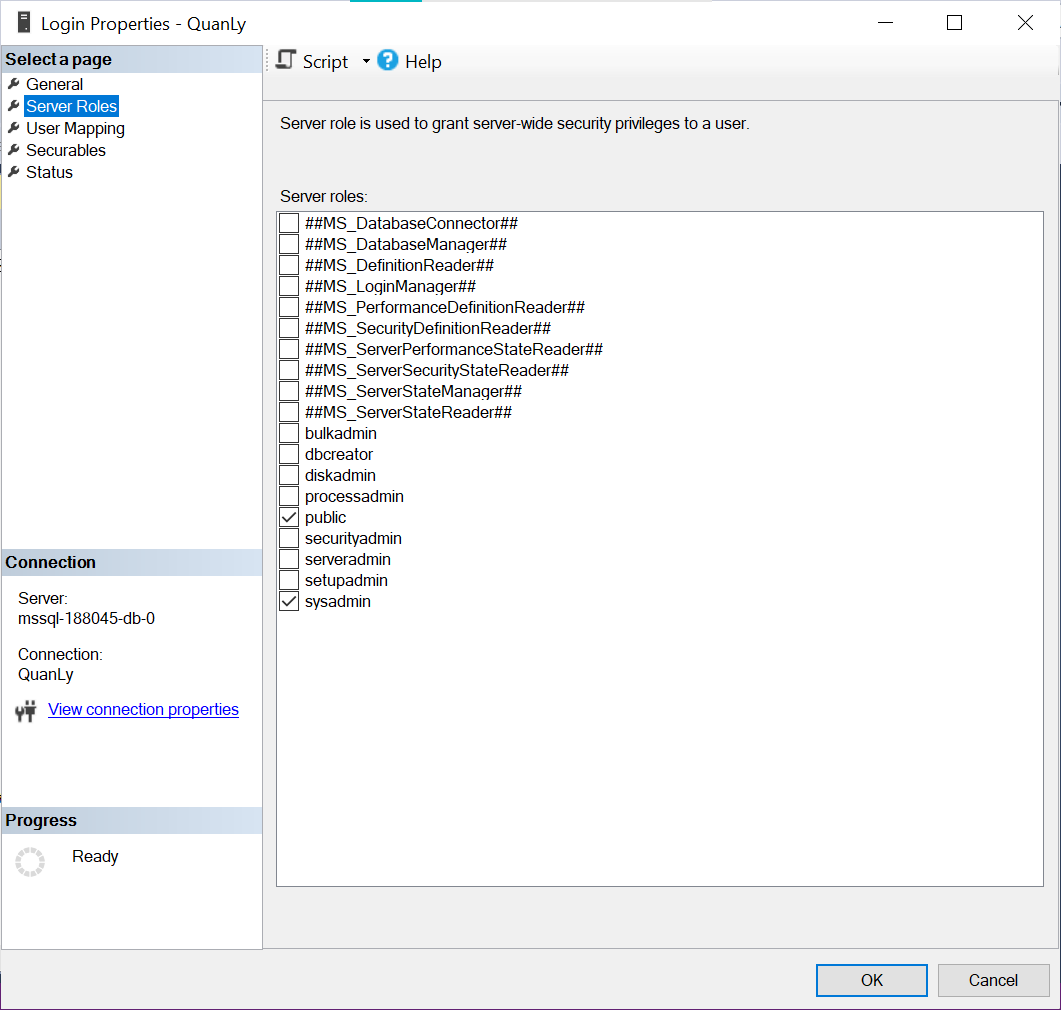
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Bước 3: Thiết lập quyền trên server (server roles)**

* Đối với tài khoản nhân viên (NhanVien) chỉ chọn public (Không thể làm bất kì điều gì tác động tới cơ sở dữ liệu. Chỉ có thể truy cập tới các Object được public bên trong cơ sở dữ liệu)
* Đối với tài khoản quản lý (QuanLy) chọn thêm sysadmin (Có thể làm bất kỳ điều gì trong SQL Server)



A screenshot of a computer

Description automatically generated

**Bước 4:   Lựa chọn database cho phép người dùng kết nối tại tab User mapping**

* Chọn cơ sở dữ liệu: QuanLyChamCong
* Nếu là tài khoản nhân viên thì phần “Database role membership for: QuanLyChamCong” thì chỉ chọn public.
* Nếu là tài khoản quản lý thì phần “Database role membership for: QuanLyChamCong” thì chọn public và db\_owner (có tất cả quyền trong cơ sở dữ liệu, bao gồm đọc (SELECT), ghi (INSERT), xóa (DELETE), sửa (UPDATE), thay đổi cấu trúc (ALTER), tạo mới các đối tượng, v.v.).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of words

Description automatically generated

**Phân quyền người dùng**

**2.1. Nhân Viên**

**Bước 1: Tại database QuanLyChamCong -> Security -> User -> NhanVien -> nhấp chuột phải chọn Properties**

A screenshot of a computer

Description automatically generated

**Bước 2: Tại thẻ Securables -> Chọn Search**

A screenshot of a computer

Description automatically generated

**Bước 3: Cửa sổ Add Object xuất hiện**

* Chọn All objects belonging to the schema: Schema name: dbo

A screenshot of a computer

Description automatically generated

**Bước 4: Phân quyền cho user NhanVien**

* Bảng BANGLUONG: có quyền xem dữ liệu (select)

A screenshot of a computer

Description automatically generated

* Bảng CHAMCONG: có quyền xem dữ liệu (select), thêm dữ liệu (insert)

A screenshot of a computer

Description automatically generated

* Bảng HOPDONG: có quyền xem dữ liệu (select)

A screenshot of a computer

Description automatically generated

* Bảng NHANVIEN: có quyền xem dữ liệu (select)

A screenshot of a computer

Description automatically generated

* Bảng VIPHAM: có quyền xem dữ liệu (select), thêm dữ liệu (insert)

A screenshot of a computer

Description automatically generated

* Bảng TAIKHOAN: có quyền xem dữ liệu (select)

A screenshot of a computer

Description automatically generated

## BẢO MẬT VÀ MÃ HÓA

/\*

Dữ liệu cần mã hóa:

Bảng NHANVIEN: SDT, Email

Bảng TAIKHOAN: MK

\*/

create symmetric key MaHoaDL

with algorithm = AES\_256

encryption by password = 'Mahoa\_dulieu@123'

**-- Mã hóa bảng NHANVIEN**

alter table NHANVIEN

add SDT\_MaHoa varbinary(max),

    Email\_MaHoa varbinary(max)

create proc MaHoaNV

as

begin

    set nocount on

    open symmetric key MaHoaDL

    decryption by password = 'Mahoa\_dulieu@123'

    update NHANVIEN

    set SDT\_MaHoa = encryptbykey(key\_guid('MaHoaDL'), convert(varbinary(max), SDT)),

        Email\_MaHoa = encryptbykey(key\_guid('MaHoaDL'), convert(varbinary(max), Email))

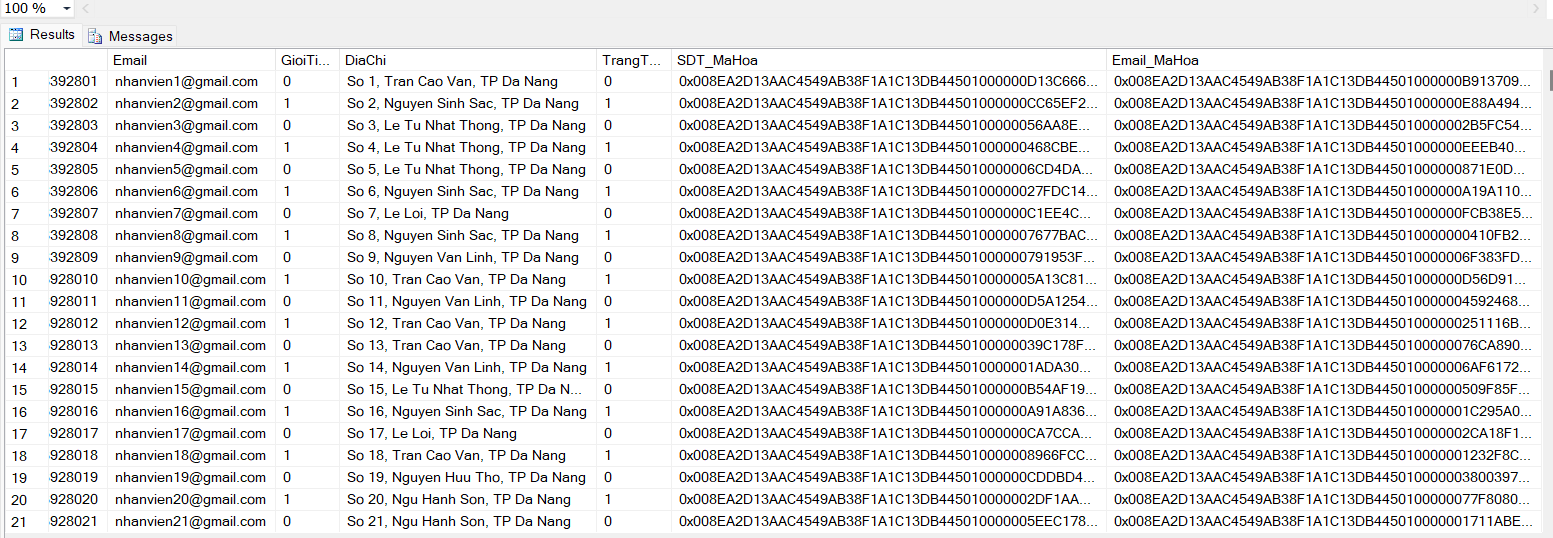
    close symmetric key MaHoaDL

end

--Test

exec MaHoaNV

select \* from NHANVIEN



**--Giải mã**

open symmetric key MaHoaDL

decryption by password = 'Mahoa\_dulieu@123'

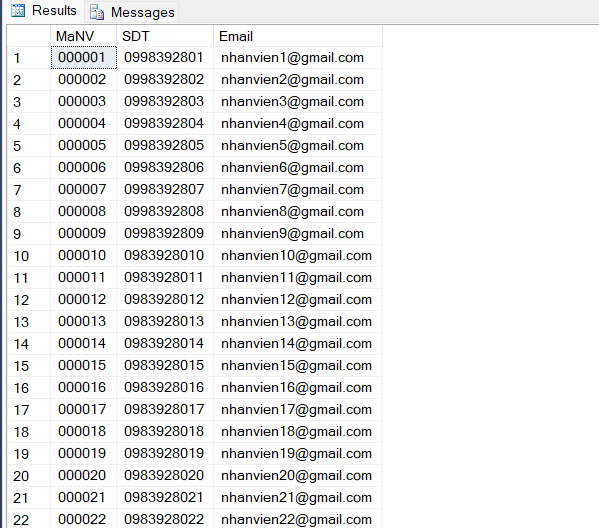
select MaNV,

    convert(char(10), decryptbykey(SDT\_MaHoa)) as SDT,

    convert(varchar(30), decryptbykey(Email\_MaHoa)) as Email

from NHANVIEN

close symmetric key MaHoaDL



**-- Mã hóa bảng TAIKHOAN**

alter table TAIKHOAN

add MK\_MaHoa varbinary(max)

create proc MaHoaTK

as

begin

set nocount on

open symmetric key MaHoaDL

decryption by password = 'Mahoa\_dulieu@123'

update TAIKHOAN

set MK\_MaHoa = encryptbykey(key\_guid('MaHoaDL'), convert(varbinary(max), MK))

close symmetric key MaHoaDL

end

**--Test**

exec MaHoaTK

select \* from TAIKHOAN

A screenshot of a computer

Description automatically generated

**--Giải mã**

open symmetric key MaHoaDL

decryption by password = 'Mahoa\_dulieu@123'

select TenTK, MaNV,

convert(varchar(20), decryptbykey(MK\_MaHoa)) as MK

from TAIKHOAN

close symmetric key MaHoaDL

A white screen with black text

Description automatically generated with medium confidence

Yêu cầu: Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu

## Tạo backup tự động

### Tạo Backup device để lưu trữ bản sao lưu của cơ sở dữ liệu

**Bước 1: Vào thư mục Server Objects -> Nhấp chuột phải vào Backup Devices -> New Backup Device**

**A screenshot of a computer

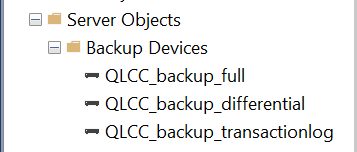
Description automatically generated**

**Bước 2: Đặt tên cho thiết bị và lựa chọn đường dẫn tệp sao lưu**

* QLCC\_backup\_full (Full backup)
* QLCC\_backup\_differential (Differential backup)
* QLCC\_backup\_transactionlog (Transaction Log backup)

A screenshot of a computer

Description automatically generated



### Tạo lịch backup tự động

**Bước 1: Khởi động SQL Server Agent**

A screenshot of a computer

Description automatically generated

**Bước 2: Tạo kế hoạch. Tại thư mục Management -> Nhấp chuột phải vào Maintenance Plants -> Chọn Maintenance Plan Wizard**

A screenshot of a computer

Description automatically generated

**Bước 3: Nhập tên, chọn Separate schedules for each task để thiết lập lịch riêng biệt cho từng task -> Next**

A screenshot of a computer program

Description automatically generated

**Bước 4:**

* Back Up Database (Full): thực hiện backup toàn bộ các đối tượng, bảng hệ thống và dữ liệu trong cơ sở dữ liệu.
* Back Up Database (Differential): chỉ backup dữ liệu thay đổi kể từ lần full backup trước đó.
* Back Up Database (Transaction Log): lưu trữ những thay đổi kể từ transaction log backup trước đó và loại trừ đi những transaction đã được commit vào cơ sở dữ liệu.
* Maintenance Cleanup Task: để tự động xóa các bản backup cũ, các tệp log hoặc các tệp không còn cần thiết nữa để giải phóng dung lượng đĩa

A screenshot of a computer program

Description automatically generated

**Bước 5: Thiết lập lịch tự động backup**

* **Full backup:** 23:59 tối chủ nhật hàng tuần
* Tại tab General, chọn database QuanLyChamCong

A screenshot of a computer

Description automatically generated

* Tại tab Destination, chọn Back up databases across one or more files -> Add -> Chọn Backup device: QLCC\_backup

A screenshot of a computer

Description automatically generated

* Tại mục Scheduled, chọn Change -> thiết lập thời gian full backup tự động lặp lại hàng tuần vào chủ nhật lúc 23:59:00

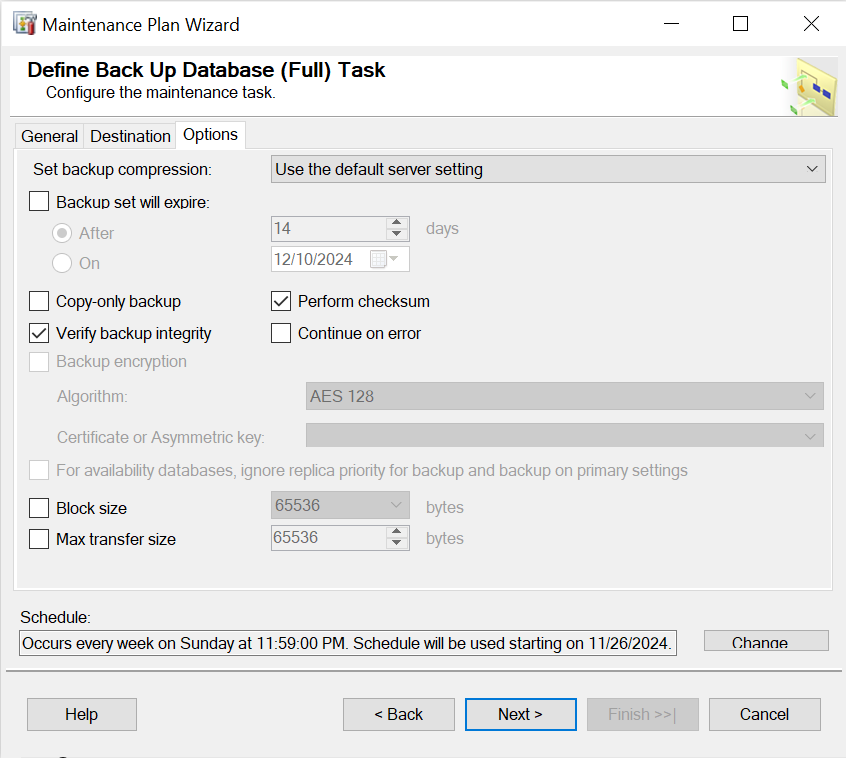
A screenshot of a computer

Description automatically generated

* Tại tab Options. Chọn:

Verify backup integrity: để đảm bảo file backup không bị lỗi và có thể sử dụng để khôi phục.

Perform checksum: để kiểm tra dữ liệu trong quá trình backup, giúp phát hiện lỗi.



* **Differential backup:** các ngày thứ 2, 3, 4, 5, 6, 7
* Thực hiện các bước tương tự như Full Backup

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

* Tạo lịch thực hiện Differential Backup vào thứ 2, 3, 4, 5, 6, 7 hàng tuần lúc 23:59:00

A screenshot of a computer

Description automatically generated

* **Transaction log backup:** hàng ngày và lặp lại 5 giờ 1 lần, bắt đầu từ 9:15 đến 23:15
* Thực hiện các bước tương tự như Full Backup

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Tạo lịch Transaction Log backup hàng ngày, cách 5 tiếng thực hiện backup một lần, bắt đầu từ 9:15 đến 23:15

A screenshot of a computer

Description automatically generated

**Bước 6: Tạo lịch xóa các file backup cũ tự động.**

* Chọn Backup files để xóa các file backup
* Chọn đường dẫn đến nới lưu trữ các bản backup tại Folder, nhập phần mở rộng file là “.bak”
* Tại File age, chọn 4 tuần để chỉ xóa các file cũ hơn 4 tuần

A screenshot of a computer

Description automatically generated

* Chọn Change để tạo lịch. Thực hiện xóa vào lúc 1:00 thứ hai hàng tuần

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Tạo Backup thực hiện trên cloud cluster

A screenshot of a computer

Description automatically generated**Bước 1: Lựa chọn thao tác backup**

A screenshot of a computer

Description automatically generated**Bước 2: Thiết lập các tùy chọn backup**

**Bước 3: Thiết lập backup thành công**

A screenshot of a computer

Description automatically generated

Yêu cầu: Giả sử lượng dữ liệu rất lớn, không đủ lưu trữ trong một ổ đĩa. Hãy đưa ra phương án giải quyết TỐT NHẤT cho vấn đề đã đặt ra và thực hiện trên CSDL đã xây dựng. (Nêu lý do lựa chọn giải pháp và thực hiện triển khai giải pháp cho hệ thống)

Triển khai và quản lý cơ sở dữ liệu là một trong những hoạt động phức tạp, tốn thời gian và tốn chi phí nhất trong CNTT. MySQL Cluster là một cơ sở dữ liệu phân tán được thiết kế cho việc mở rộng và tính sẵn sàng cao. Nó cung cấp khả năng truy cập thời gian thực trong bộ nhớ với tính nhất quán giao dịch trên các phân vùng và bộ dữ liệu phân tán. Nó được thiết kế cho các ứng dụng quan trọng.

MySQL Cluster  có các chức năng:

- Khả năng mở rộng (scalability) cao

- Khả năng sẵn sàng uptime (availability) 99,999%

- Bảo toàn dữ liệu với khả năng sao lưu giữa các node

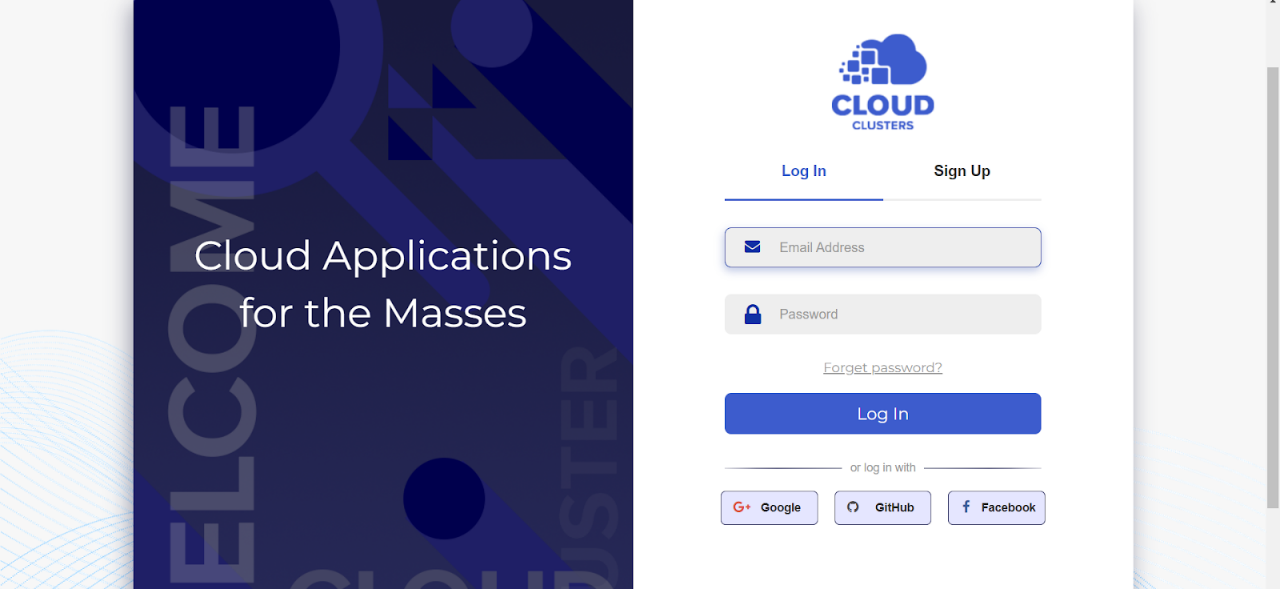
- Dễ dàng quản lý và bảo trì

- Tính linh động và khả năng truy cập mọi lúc mọi nơi

- …

**Để lưu cơ sở dữ liệu trên Cloud Clusters cần tạo tài khoản để thực hiện**

**Bước 1. Đăng nhập trên Cloud Clusters**



**Bước 2: Tiếp theo ta chọn SQL Server, chọn Free Trial, sau đó chọn Confirm**

A screenshot of a computer

Description automatically generated

**Bước 3: Chọn Manage**

A screenshot of a computer

Description automatically generated

**Bước 4. Tạo Database “QuanLyChamCong” và user “QuanLy” có tất cả các quyền**

* DB & User → Create Database
* DB & User → Create User

A screenshot of a chat

Description automatically generated

**Bước 5. Lấy dữ liệu từ trên Cloud Clusters bằng cách connect trong SQL server với server name**

Overview —> Copy mssql-188045-0.cloudclusters.net,10053

A screenshot of a computer

Description automatically generated

**Bước 6. Đăng nhập vào SQL**

* Server name: mssql-188045-0.cloudclusters.net,10053
* Login: QuanLy
* Password: đã tạo ở Cloud Clusters

A screenshot of a computer

Description automatically generated

**Bước 7. Add các user NhanVien và cấp quyền được Read & Write**

* DB & User → Create User

**A screenshot of a chat

Description automatically generated**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generatedBước 8. Đẩy database và add các user thành công**

**A screenshot of a computer

Description automatically generated**

→ Kiểm tra trên Cloud Clusters.

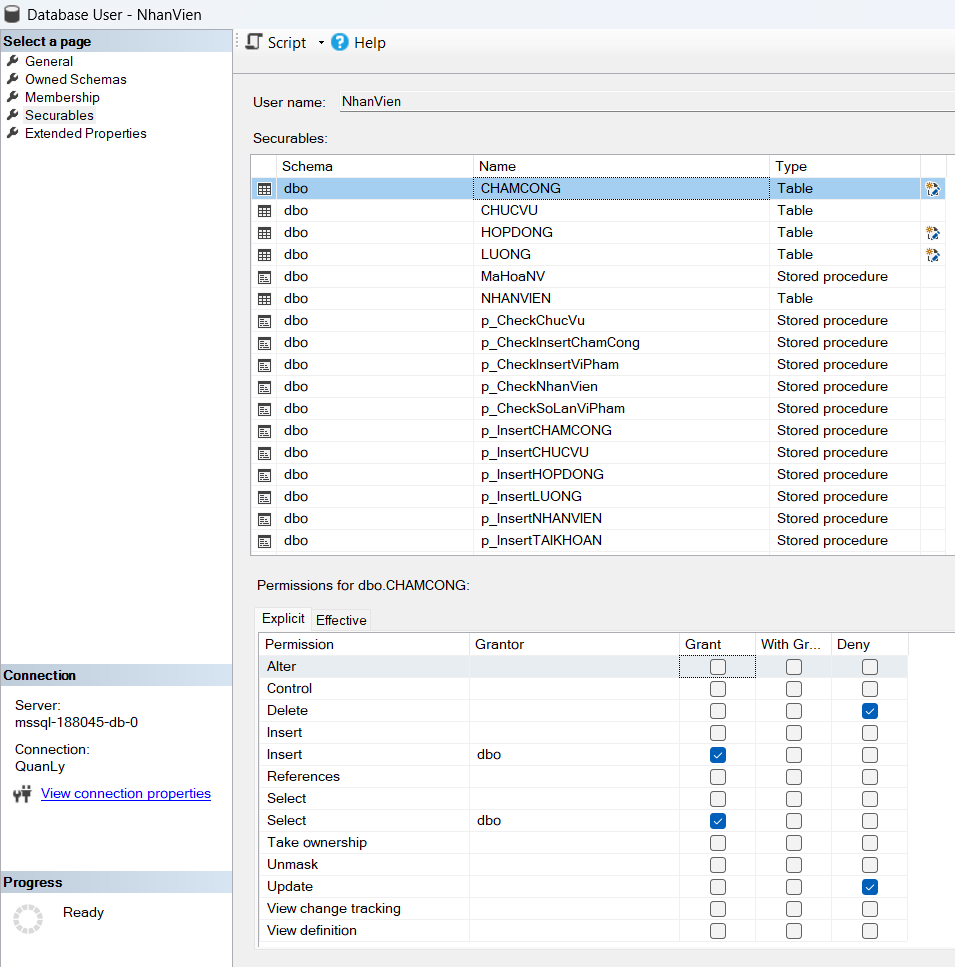
A screenshot of a computer

Description automatically generated

Yêu cầu: Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection

**PHƯƠNG PHÁP HẠN CHẾ SQL INJECTION**

**Bước 1: Phân quyền truy cập:**

* Sau khi phân quyền truy cập ở R5, nhân viên có quyền insert vào bảng ChamCong và ViPham, các bảng còn lại chỉ được select

A screenshot of a computer

Description automatically generated

**Bước 2: Xử lý phần quyền được insert của nhân viên:**

* Tạo thủ tục dùng các câu lệnh tham số, xử lý dữ liệu đầu vào, dùng thông báo chung để báo lỗi.
* **Bảng CHAMCONG:**

-- Module 11: Nhân viên insert vào bảng CHAMCONG

create PROCEDURE InsertChamCong

    @TenTK VARCHAR(20),

    @NgayCC DATE,

    @BatDau TIME,

    @KetThuc TIME

AS

BEGIN

    BEGIN TRY

        IF @TenTK IS NULL OR @TenTK = ''

            OR @NgayCC IS NULL OR @BatDau IS NULL OR @KetThuc IS NULL

        BEGIN

            PRINT 'Vui lòng kiểm tra lại thông tin!!'

            RETURN

        END

        IF NOT EXISTS (SELECT 1 FROM TAIKHOAN WHERE TenTK = @TenTK)

        BEGIN

            PRINT 'Vui lòng kiểm tra lại thông tin!!'

            RETURN

        END

        DECLARE @MaCC CHAR(12),

@MaxMaCC INT

        SELECT @MaxMaCC = MAX(CAST(RIGHT(MaCC, 8) AS INT))

FROM CHAMCONG

        SET @MaCC = 'CC' + RIGHT('00000000' + CAST(ISNULL(@MaxMaCC, 0) + 1 AS VARCHAR), 6)

        INSERT INTO CHAMCONG (MaCC, TenTK, NgayCC, BatDau, KetThuc)

        VALUES (@MaCC, @TenTK, @NgayCC, @BatDau, @KetThuc)

        PRINT 'Dữ liệu chấm công đã được thêm thành công!!'

    END TRY

    BEGIN CATCH

        PRINT 'Vui lòng kiểm tra lại thông tin!!'

    END CATCH

END --Test module 11:

EXEC InsertChamCong @TenTK = 'TK000640', @NgayCC = '', @BatDau = '08:00:00', @KetThuc = '17:00:00'

* **Bảng VIPHAM**

-- Module 12: Nhân viên insert vào bảng VIPHAM

 create PROCEDURE InsertViPham

    @MaNV CHAR(6),

    @Loi VARCHAR(15)

AS

BEGIN

    BEGIN TRY

        IF @MaNV IS NULL OR @Loi IS NULL

           OR @MaNV = '' OR @Loi = ''

        BEGIN

            PRINT 'Vui lòng kiểm tra lại thông tin!!'

            RETURN

        END

        IF NOT EXISTS (SELECT 1 FROM NHANVIEN WHERE MaNV = @MaNV)

        BEGIN

            PRINT 'Nhân viên không tồn tại!!'

            RETURN

        END

        DECLARE @MaVP CHAR(6),

                @MaxMaVP INT

        SELECT @MaxMaVP = MAX(CAST(RIGHT(MaVP, 6) AS INT))

        FROM VIPHAM

        SET @MaVP = RIGHT('000000' + CAST(ISNULL(@MaxMaVP, 0) + 1 AS VARCHAR), 6)

        INSERT INTO VIPHAM (MaVP, MaNV, Loi)

        VALUES (@MaVP, @MaNV, @Loi)

        PRINT 'Dữ liệu vi phạm đã được thêm thành công.'

    END TRY

    BEGIN CATCH

        PRINT 'Đã xảy ra lỗi. Vui lòng thử lại sau.'

    END CATCH

END

-- Test module 12:

EXEC InsertViPham @MaNV = '001008', @Loi = 'Đi muộn'

**Bước 3: Xử lý dữ liệu đầu vào phần đăng nhập**

-- Module 13: Xử lý đăng nhập

create PROCEDURE sp\_DangNhap

@TenTK NVARCHAR(50),

@MatKhau NVARCHAR(255)

AS

BEGIN

IF NOT EXISTS (SELECT 1 FROM TAIKHOAN WHERE TenTK = @TenTK)

BEGIN

PRINT 'Tài khoản không tồn tại hoặc mật khẩu không chính xác!!'

RETURN

END

DECLARE @MatKhauDB NVARCHAR(255);

SELECT @MatKhauDB = MK FROM TAIKHOAN WHERE TenTK = @TenTK

  IF @MatKhauDB IS NOT NULL AND @MatKhau = @MatKhauDB

 BEGIN

PRINT 'Đăng nhập thành công!!'

  END

ELSE

BEGIN

PRINT 'Tài khoản không tồn tại hoặc mật khẩu không chính xác!!'

END

END

-- Test module 13

Exec sp\_DangNhap 'TK000001', 'D400EA25-D1'