

TRƯỜNG ĐẠI HỌC KINH TẾ  
**KHOA THỐNG KÊ – TIN HỌC**



QUẢN TRỊ CƠ SỞ DỮ LIỆU  
**BÁO CÁO THIẾT KẾ HỆ THỐNG**

*Giảng viên*

: *Cao Thị Nhâm*

*Sinh viên thực hiện*

: *Nguyễn Thị Bích Ngọc*

*Hồ Sỹ Đức Thành*

*Trần Phương Đan*

*Phạm Trần Diệu Khanh*

*Trần Thị Bích Tuyền*

*Lớp*

: *48K14.2*

*Nhóm thực hiện*

: *Nhóm 4*

- Đà Nẵng , tháng 05 năm 2024 -

## MỤC LỤC

CHƯƠNG 1. R1+R2.....	6
1.1.    Thiết kế logic cơ sở dữ liệu.....	6
1.1.1.    Chuẩn hóa dạng 2NF .....	6
1.1.2.    Sơ đồ cơ sở dữ liệu .....	6
1.2.    Mô tả chi tiết các bảng.....	6
1.2.1.    Bảng TAIKHOAN .....	6
1.2.2.    Bảng NHANVIEN .....	7
1.2.3.    Bảng CALAM.....	7
1.2.4.    Bảng LICHLAM.....	8
1.2.5.    Bảng BANGLUONG.....	8
1.2.6.    BẢNG CHAMCONG.....	9
CHƯƠNG 2. R3.....	9
2.1. Xây dựng cơ sở dữ liệu .....	9
2.1.1.Tạo bảng NHANVIEN .....	9
2.1.2.Tạo bảng TAIKHOAN .....	9
2.1.3. Tạo bảng CALAM .....	10
2.1.4.Tạo bảng LICHLAM .....	10
2.1.5.Tạo bảng BANGLUONG .....	10
2.1.6.Tạo bảng CHAMCONG .....	10
2.2. DIAGRAM .....	11
CHƯƠNG 3. R4.....	11
3.1 . Nhũng thay đổi trong database .....	11
3.1.1. Bảng NHANVIEN .....	11
3.1.2 Bảng TAIKHOAN .....	11
3.1.3. Bảng CALAM.....	11
3.1.4.Bảng CHAMCONG.....	12
3.1.5.Bảng LICHLAM .....	12
3.1.6.Bảng BANGLUONG.....	12
3.1.7. DIAGRAM mới .....	13
3.2.DUMP 1000 DỮ LIỆU BẢNG.....	13
3.2.1.Bảng NHANVIEN .....	13
3.2.2. Bảng TAIKHOAN .....	14
3.3.3.Bảng CALAM.....	15

3.3.4.Bảng LICH LAM .....	16
3.3.5.Bảng BANGLUONG .....	17
3.3.6 Bảng CHAM CONG .....	18
3.3. 10 MODULE XỬ LÝ DỮ LIỆU .....	19
3.3.1. MODULE 1 .....	19
3.3.2. MODULE 2 .....	22
3.3.3. MODULE 3 .....	24
3.3.4. MODULE 4 .....	25
3.3.5. MODULE 5 .....	26
3.3.6. MODULE 6 .....	28
3.3.7. MODULE 7 .....	29
3.3.8. MODULE 8 .....	31
3.3.9. MODULE 9 .....	33
3.3.10. MODULE 10 .....	34
CHƯƠNG 4. R5.....	35
4.1.Tổng quan .....	35
4.2.Tạo tài khoản cho người quản lý .....	36
4.3. Tạo tài khoản cho nhân viên .....	39
4.4. Mã hóa .....	43
CHƯƠNG 5. R6.....	46
5.1.Tổng quan .....	46
5.1.1.Backup dữ liệu trên cloud .....	47
5.1.2.Backup bổ sung trên server local .....	47
5.2.Phục hồi dữ liệu trên Server Local .....	54
CHƯƠNG 6. R7.....	58
6.1.Tổng quan .....	58
6.2.Amazon Web Services (AWS) .....	59
6.2.1.Thông tin cơ bản về AWS.....	59
6.2.2.Lý do lựa chọn AWS .....	60
6.3.Triển khai .....	60
6.3.1.Truy cập tài khoản AWS.....	60
6.3.2.Tạo database.....	61
6.3.3.Cấu hình bảo mật và quyền truy cập.....	65
6.3.4. Kết nối từ SQL Server Management Studio (SSMS) .....	66
CHƯƠNG 7. R8.....	68

7.1.Tổng quan .....	68
7.2.Triển khai .....	68

## BẢNG PHÂN CÔNG

CÔNG VIỆC	NGƯỜI THỰC HIỆN
R1,2. Hiệu chỉnh lại các bảng sao cho ít nhất phải ở dạng chuẩn 2NF • Thiết kế chi tiết các bảng	Cả nhóm
R3. Xây dựng cơ sở dữ liệu cho bản thiết kế ở bước trên.	Cả nhóm
R4. 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...)	Cả nhóm
R5. Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu	Đan
R6. Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu	Ngọc
R7. 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)	Khanh
R8. Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection	Thành, Tuyền

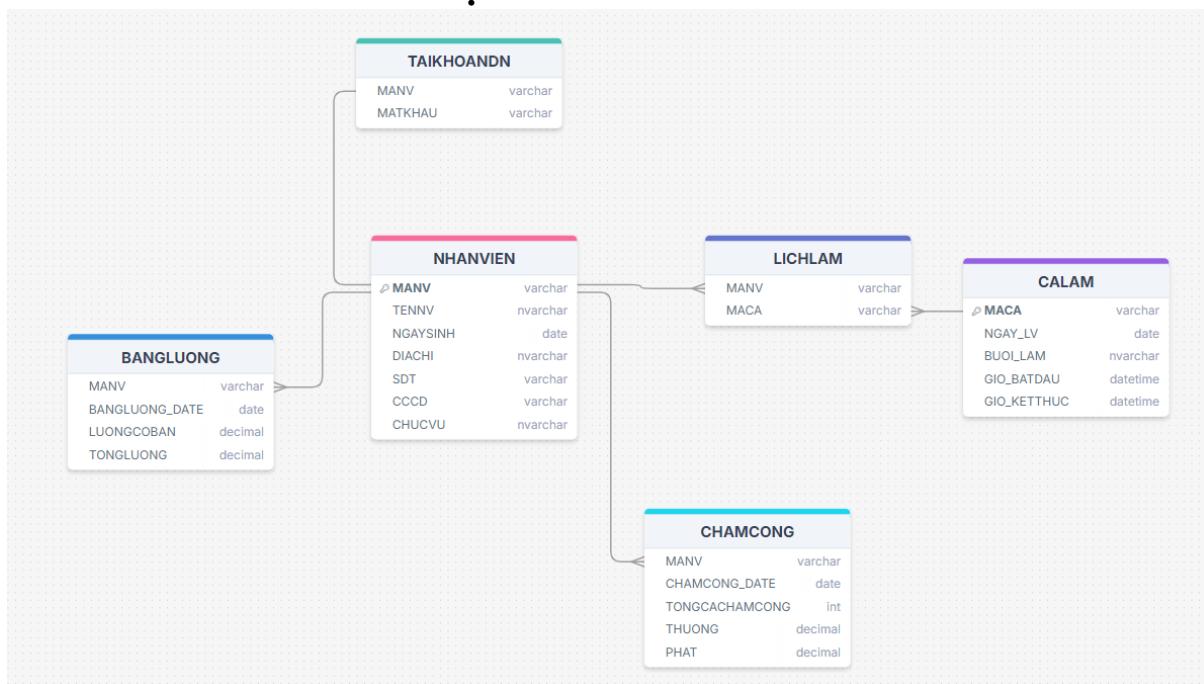
# CHƯƠNG 1. R1+R2

## 1.1.Thiết kế logic cơ sở dữ liệu.

### 1.1.1. Chuẩn hóa dạng 2NF

- TAIKHOAN (MANV, MATKHAU)
- NHANVIEN (MANV, TENNV, NGAYSINH, DIACHI, SDT, CCCD, CHUCVU)
- CALAM (MACA, NGAY\_LV, BUOI\_LAM, GIO\_BATDAU, GIO\_KETTHUC)
- LICHLAM (MANV, MACA)
- BANGLUONG (MANV, BANGLUONG\_DATE, LUONGCOBAN, TONGLUONG)
- CHAMCONG (MANV, CHAMCONG\_DATE, TONGCACHAMCONG, THUONG, PHAT)

### 1.1.2. Sơ đồ cơ sở dữ liệu



## 1.2.Mô tả chi tiết các bảng.

### 1.2.1. Bảng TAIKHOAN

BẢNG : TAIKHOANDN				
Tên	Kiểu dữ liệu	Ràng buộc	Khóa	Mô tả

MANV	varchar(255)	NOT NULL	FK	Khóa ngoại của bảng
MATKHAU	varchar(255)	NOT NULL		Mật khẩu của tài khoản

### 1.2.2. Bảng NHANVIEN

BẢNG : NHANVIEN				
Tên	Kiểu dữ liệu	Ràng buộc	Khóa	Mô tả
MANV	varchar(255)	NOT NULL	PK	Khóa chính của bảng
TENNVL	nvarchar(255)	NOT NULL		Tên nhân viên
NGAYSINH	date	NOT NULL		Ngày sinh của nhân viên
DIACHI	nvarchar(255)	NULL		Địa chỉ của nhân viên
SDT	varchar(255)	NOT NULL		Số điện thoại của nhân viên
CCCD	varchar(255)	NOT NULL		Căn cước công dân của nhân viên
CHUCVU	nvarchar(255)	NOT NULL		Chức vụ của nhân viên

### 1.2.3. Bảng CALAM

BẢNG : CALAM				
Tên	Kiểu dữ liệu	Ràng buộc	Khóa	Mô tả
MACA	varchar(255)	NOT NULL	PK	Khóa chính của bảng

NGAY_LV	Date	NOT NULL		Ngày làm việc của ca làm
BUOI_LAM	nvarchar(255)	NOT NULL		Buổi làm của ca làm ( Sáng/Chiều/Tối)
GIO_BATDAU	Datetime	NOT NULL		Giờ bắt đầu ca làm
GIO_KETTHUC	Datetime	NOT NULL		Giờ kết thúc ca làm

#### 1.2.4. Bảng LICHLAM

BẢNG : LICHLAM				
Tên	Kiểu dữ liệu	Ràng buộc	Khóa	Mô tả
MANV	varchar(255)	NOT NULL	FK	Khóa ngoại của bảng
MACA	varchar(255)	NOT NULL	FK	Khóa ngoại của bảng

#### 1.2.5. Bảng BANGLUONG

BẢNG : BANGLUONG				
Tên	Kiểu dữ liệu	Ràng buộc	Khóa	Mô tả
MANV	varchar(255)	NOT NULL	FK	Khóa ngoại của bảng
BANGLUONG_DATE	datetime(255)	NOT NULL		Thời gian bảng lương
TONGLUONG	Decimal(18,2)	NOT NULL		Tổng lương
LUONGCOBAN	Decimal(18,2)	NOT NULL		Tổng ca chấm công

## 1.2.6. BẢNG CHAMCONG

BẢNG : CHAMCONG				
Tên	Kiểu dữ liệu	Ràng buộc	Khóa	Mô tả
MANV	varchar(255)	NOT NULL	FK	Khóa ngoại của bảng
CHAMCONG_DATE	datetime(255)	NOT NULL		Thời gian chấm công
TONGCACHAMCONG	Int	NOT NULL		Tổng lương
THUONG	Decimal(18,2)	NOT NULL		Tiền thưởng
PHAT	Decimal(18,2)	NOT NULL		Tiền phạt

## CHƯƠNG 2. R3

### 2.1. Xây dựng cơ sở dữ liệu

```
--CREATE DATABASE QL_NHANSU;
USE QL_NHANSU;
```

#### 2.1.1.Tạo bảng NHANVIEN

```
-- Tạo bảng NHANVIEN
CREATE TABLE NHANVIEN (
    MANV VARCHAR(10) PRIMARY KEY NOT NULL,
    TENNV NVARCHAR(255) NOT NULL,
    NGAYSINH DATE NOT NULL,
    DIACHI NVARCHAR(255) NULL ,
    SDT VARCHAR(15) UNIQUE,
    CCCD VARCHAR(50) UNIQUE,
    CHUCVU NVARCHAR(50) NOT NULL,
    CONSTRAINT CK_SDT CHECK (SDT LIKE '0%' AND LEN(SDT) BETWEEN 10 AND 12)
);
```

#### 2.1.2.Tạo bảng TAIKHOAN

```
-- Tạo bảng TAIKHOAN
CREATE TABLE TAIKHOAN (
    MANV VARCHAR(10) NOT NULL PRIMARY KEY,
    MATKHAU VARCHAR(225) NOT NULL,
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV)
);
```

### 2.1.3. Tạo bảng CALAM

```
-- Tạo bảng CALAM
CREATE TABLE CALAM (
    MACA VARCHAR(10) NOT NULL PRIMARY KEY ,
    NGAY_LV DATE NOT NULL,
    BUOI_LAM NVARCHAR(10) NOT NULL,
    GIO_BATDAU TIME NOT NULL,
    GIO_KETTHUC TIME NOT NULL,
    CONSTRAINT CK_GIO CHECK (GIO_BATDAU < GIO_KETTHUC)
);
```

### 2.1.4.Tạo bảng LICHLAM

```
-- Tạo bảng LICHLAM
CREATE TABLE LICHLAM (
    MANV VARCHAR(10) NOT NULL,
    MACA VARCHAR(10) NOT NULL,
    PRIMARY KEY (MANV, MACA),
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
    FOREIGN KEY (MACA) REFERENCES CALAM(MACA),
);

```

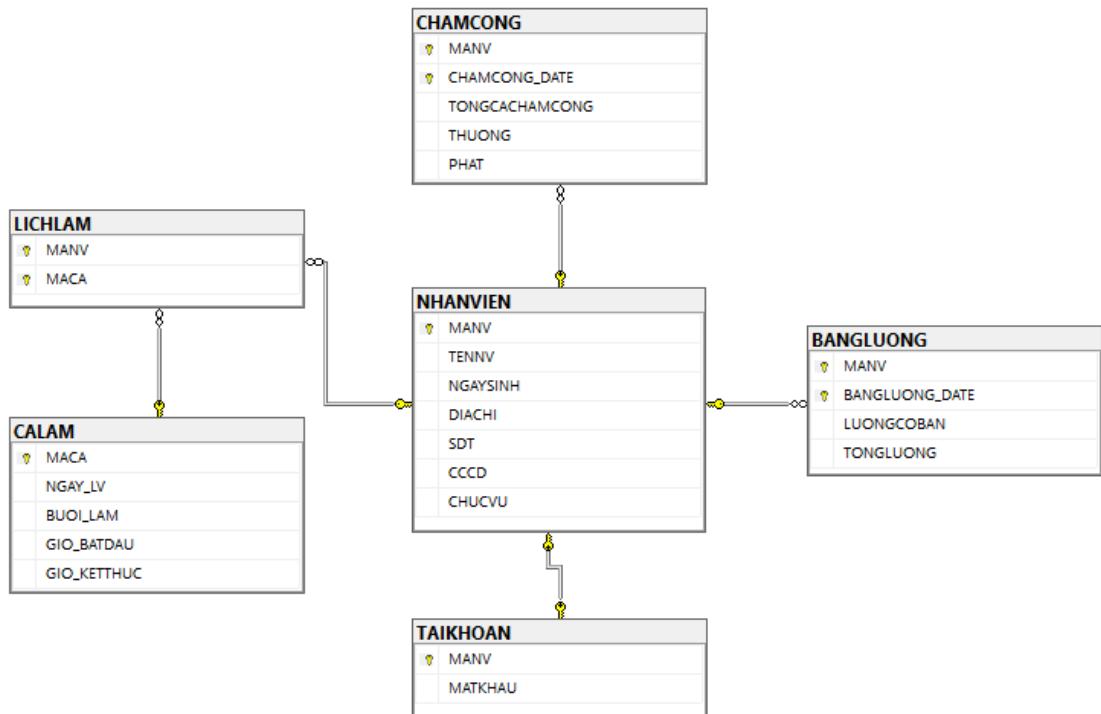
### 2.1.5.Tạo bảng BANGLUONG

```
-- Tạo bảng BANGLUONG
CREATE TABLE BANGLUONG (
    MANV VARCHAR (10) NOT NULL,
    BANGLUONG_DATE DATE NOT NULL,
    LUONGCOBAN DECIMAL(18, 2) NOT NULL,
    TONGLUONG DECIMAL(18, 2) NOT NULL,
    PRIMARY KEY (MANV, BANGLUONG_DATE),
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
    CONSTRAINT CK_LUONG CHECK (LUONGCOBAN >= 0 AND TONGLUONG >= 0)
);
```

### 2.1.6.Tạo bảng CHAMCONG

```
-- Tạo bảng CHAMCONG
CREATE TABLE CHAMCONG (
    MANV VARCHAR (10) NOT NULL,
    CHAMCONG_DATE DATE NOT NULL,
    TONGCACHAMCONG INT NOT NULL,
    THUONG DECIMAL(18, 2) NULL,
    PHAT DECIMAL(18, 2) NULL,
    PRIMARY KEY (MANV, CHAMCONG_DATE),
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
    CONSTRAINT CK_CHAMCONG CHECK (TONGCACHAMCONG >= 0 AND THUONG >= 0 AND PHAT <= 0)
);
```

## 2.2. DIAGRAM



## CHƯƠNG 3. R4

### 3.1 . Những thay đổi trong database

#### 3.1.1. Bảng NHANVIEN

- Thay đổi các cột trong bảng NHANVIEN thành NULL

```
-- Thay đổi các cột trong bảng NHANVIEN thành NULL
ALTER TABLE NHANVIEN ALTER COLUMN TENNV NVARCHAR(255) NULL
ALTER TABLE NHANVIEN ALTER COLUMN NGAYSINH DATE NULL;
ALTER TABLE NHANVIEN ALTER COLUMN DIACHI NVARCHAR(255) NULL
ALTER TABLE NHANVIEN ALTER COLUMN SDT VARCHAR(15) NULL;
ALTER TABLE NHANVIEN ALTER COLUMN CCCD VARCHAR(50) NULL;
ALTER TABLE NHANVIEN ALTER COLUMN CHUCVU NVARCHAR(50) NULL
```

#### 3.1.2 Bảng TAIKHOAN

- Thay đổi các cột trong bảng TAIKHOAN thành NULL

```
-- Thay đổi các cột trong bảng TAIKHOAN thành NULL
ALTER TABLE TAIKHOAN ALTER COLUMN MATKHAU VARCHAR(225) NULL
```

#### 3.1.3. Bảng CALAM

- Thay đổi các cột trong bảng CALAM thành NULL

```
-- Thay đổi các cột trong bảng CALAM thành NULL  
ALTER TABLE CALAM ALTER COLUMN BUOI_LAM NVARCHAR(10) NULL
```

- Thêm cột SOGIOLAMVIEC vào bảng CALAM

```
--Thêm cột SOGIOLAMVIEC vào bảng CALAM  
ALTER TABLE CALAM  
ADD SOGIOLAMVIEC INT NULL
```

### 3.1.4.Bảng CHAMCONG

- Thay đổi các cột trong bảng CHAMCONG thành NULL

```
-- Thay đổi các cột trong bảng CHAMCONG thành NULL  
ALTER TABLE CHAMCONG ALTER COLUMN TONGSOGIOLAM INT NULL
```

- Đổi column TONGCACHAMCONG thành TONGSOGIOLAM

```
--Đổi column TONGCACHAMCONG thành TONGSOGIOLAM  
ALTER TABLE CHAMCONG  
DROP CONSTRAINT CK_CHAMCONG;  
  
ALTER TABLE CHAMCONG  
DROP COLUMN TONGCACHAMCONG;  
  
ALTER TABLE CHAMCONG  
ADD TONGSOGIOLAM INT NULL;  
  
ALTER TABLE CHAMCONG  
ADD CONSTRAINT CK_TONGSOGIOLAM CHECK (TONGSOGIOLAM >= 0 AND THUONG >= 0 AND PHAT <= 0)
```

### 3.1.5.Bảng LICHLAM

- Thêm cột TRANGTHAI vào bảng LICHLAM

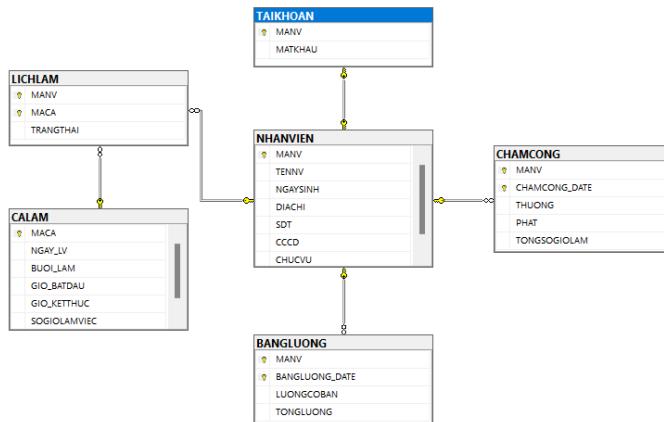
```
--Thêm cột TRANGTHAI vào bảng LICHLAM  
ALTER TABLE LICHLAM  
ADD TRANGTHAI INT NULL  
CONSTRAINT CK_TRANGTHAI CHECK (TRANGTHAI IN (0, 1, 2))
```

### 3.1.6.Bảng BANGLUONG

- Thay đổi các cột trong BANGLUONG thành NULL

```
-- Thay đổi các cột trong bảng BANGLUONG thành NULL  
ALTER TABLE BANGLUONG ALTER COLUMN TONGLUONG DECIMAL(18,2) NULL;  
ALTER TABLE BANGLUONG ALTER COLUMN LUONGCOBAN DECIMAL(18,2) NULL;
```

### 3.1.7. DIAGRAM mới



## 3.2.DUMP 1000 DỮ LIỆU BẢNG

### 3.2.1.Bảng NHANVIEN

```

USE QL_NHANSU;
GO
-- Thủ tục thêm dữ liệu vào bảng NHANVIEN
CREATE OR ALTER PROCEDURE PR_NHANVIEN
AS
BEGIN
    DECLARE @i INT = 1
    DECLARE @Ho NVARCHAR(50), @Dem NVARCHAR(50), @Ten NVARCHAR(50)
    DECLARE @DiaChi NVARCHAR(255)
    DECLARE @CCCD VARCHAR(50)

    -- Danh sách họ, đệm và tên để tạo tên tiếng Việt
    DECLARE @HoList NVARCHAR(50) = N'Nguyễn, Trần, Lê, Phạm, Hoàng, Phan, Vũ, Vũ, Đăng, Bùi, Bùi, Hồ, Ngô, Dương, Đinh'
    DECLARE @DemList NVARCHAR(50) = N'Văn, Hữu, Quốc, Thị, Minh, Thành, Bảo, Ngọc, Linh, Diệu, Phương, Hoàng, Kim'
    DECLARE @TenList NVARCHAR(50) = N'An, Bình, Chi, Dũng, Đạt, Hà, Khanh, Linh, Minh, Nam, Phong, Quân, Trang, Tuân, Văn, Ý, Khanh, Đan, Ngọc, Thành, Tuyên, Ngân, Loan, Giang, Danh'

    -- Danh sách địa chỉ tại Đà Nẵng
    DECLARE @QuanHuyenList NVARCHAR(255) = N'Liên Chiểu, Hải Châu, Ngũ Hành Sơn, Thanh Khê, Sơn Trà, Cẩm Lệ, Hòa Vang'

    DECLARE @DuongList NVARCHAR(255) = N'Nguyễn Văn Linh, Điện Biên Phủ, Lê Duẩn, Hoàng Diệu, Hùng Vương, Trần Phú, Phan Châu Trinh, Ngõ Quyền, Ông Ích Khiêm, Nguyễn Văn Thoại, Hòa Thành, Phan Tứ, Bạch Đằng, Trần Bạch Đằng'

    WHILE @i <= 1200
    BEGIN
        -- Chọn họ, đệm và tên ngẫu nhiên từ các danh sách
        SET @Ho = (SELECT value FROM STRING_SPLIT(@HoList, ',') ORDER BY NEWID() OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY)
        SET @Dem = (SELECT value FROM STRING_SPLIT(@DemList, ',') ORDER BY NEWID() OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY)
        SET @Ten = (SELECT value FROM STRING_SPLIT(@TenList, ',') ORDER BY NEWID() OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY)

        -- Chọn địa chỉ ngẫu nhiên tại Đà Nẵng
        SET @DiaChi = N'Số ' + CAST(1 + ABS(CHECKSUM(NEWID()) % 300) AS NVARCHAR(10)) + N', Đường ' +
            (SELECT value FROM STRING_SPLIT(@DuongList, ',') ORDER BY NEWID() OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY) +
            N', ' + (SELECT value FROM STRING_SPLIT(@QuanHuyenList, ',') ORDER BY NEWID() OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY) +
            N', Đà Nẵng'

        -- Tạo CCCD ngẫu nhiên (12 chữ số)
        SET @CCCD = CAST(100000000000 + ABS(CHECKSUM(NEWID()) % 900000000000) AS VARCHAR(50))

        INSERT INTO NHANVIEN (MANV, TENVN, NGAYSINH, DIACHI, SDT, CCCD, CHUCVU)
        VALUES (
            'NV' + RIGHT('0000' + CAST(@i AS VARCHAR(4)), 4), -- MANV
            @Ho + ' ' + @Dem + ' ' + @Ten, -- Tên tiếng Việt ngẫu nhiên
            DATEADD(DAY, -ABS(CHECKSUM(NEWID()) % 9000) - 5840, GETDATE()), -- Ngày Sinh trước 2004
            @DiaChi, -- Địa chỉ ngẫu nhiên tại Đà Nẵng
            '0' + CAST(10000000 + ABS(CHECKSUM(NEWID()) % 90000000) AS VARCHAR(15)), -- SDT ngẫu nhiên
            @CCCD, -- CCCD ngẫu nhiên
            CASE WHEN ABS(CHECKSUM(NEWID())) % 2 = 0 THEN N'Phá che' ELSE N'Phuc vu' END -- Chức vụ ngẫu nhiên
        )

        SET @i = @i + 1
    END
    -- Thực thi thủ tục
    EXEC PR_NHANVIEN
    -- Kiểm tra kết quả
    SELECT * FROM NHANVIEN

```

	MANV	TENNVT	NGAYSINH	DIACHI	SDT	CCCD	CHUCVU
1	NV0001	Đặng Ngọc Chi	1987-11-28	Số 142, Đường Lê Duẩn, Thanh Khê, Đà Nẵng	0664212927	100262547470	Phuc vu
2	NV0002	Võ Bảo Nam	2003-04-15	Số 149, Đường Ông Ích Khiêm, Ngũ Hành Sơn, Đà Nẵng	0420787638	100006564000	Pha che
3	NV0003	Phạm Bảo Dung	2003-09-17	Số 94, Đường Phan Châu Trinh, Ngũ Hành Sơn, Đà Nẵ...	0959257431	100792059342	Phuc vu
4	NV0004	H Phu Linh	2002-06-09	Số 53, Đường Ngô Quyền, Hải Châu, Đà Nẵng	0333312481	101081059066	Pha che
5	NV0005	Phan Diệu Chi	1988-08-16	Số 24, Đường Hoàng Diệu, Hòa Vang, Đà Nẵng	0573820890	100628962106	Phuc vu
6	NV0006	Lê Quốc An	2003-08-04	Số 263, Đường Phan Tú, Ngũ Hành Sơn, Đà Nẵng	0167324983	100070875981	Pha che
7	NV0007	H Hữu	2004-09-25	Số 5, Đường Trần Phú, Hòa Vang, Đà Nẵng	0141959194	101819328937	Phuc vu
8	NV0008	Trần Phú	1991-02-11	Số 187, Đường Trần Bá Đẳng, Hòa Vang, Đà Nẵng	0677948977	102057979800	Pha che
9	NV0009	Vũ Văn Phong	1986-06-19	Số 73, Đường Nguyễn Văn Linh, Cẩm Lệ, Đà Nẵng	0660212164	100008435758	Phuc vu
10	NV0010	Phan Hữu	2007-04-11	Số 112, Đường Bạch Đằng, Liên Chiểu, Đà Nẵng	0639861344	101954684974	Phuc vu
11	NV0011	Lê Bảo Đạt	2006-07-07	Số 199, Đường Hoài Thanh, Cẩm Lệ, Đà Nẵng	0330432103	101087809625	Phuc vu
12	NV0012	Nguyễn Văn Linh	1993-08-03	Số 72, Đường Lê Duẩn, Sơn Trà, Đà Nẵng	0843231274	101811666302	Phuc vu

### 3.2.2. Bảng TAIKHOAN

```
-- Thủ tục thêm dữ liệu vào bảng TAIKHOAN
CREATE OR ALTER PROCEDURE PR_TAIKHOAN
AS
BEGIN
    DECLARE @i INT = 1
    WHILE @i <= 1200
    BEGIN
        INSERT INTO TAIKHOAN (MANV, MATKHAU)
        VALUES (
            'NV' + RIGHT('000' + CAST(@i AS VARCHAR(4)), 4), -- MANV
            'MK' + RIGHT('0000' + CAST(ABS(CHECKSUM(NEWID()) % 10000) AS VARCHAR(4)), 4))
        SET @i = @i + 1
    END
    -- Thực thi thủ tục
    exec PR_TAIKHOAN
    -- Kiểm tra kết quả
    select * from TAIKHOAN

```

	MANV	MATKHAU
1	NV0001	MK0516
2	NV0002	MK7873
3	NV0003	MK4730
4	NV0004	MK4634
5	NV0005	MK7535
6	NV0006	MK1661
7	NV0007	MK9171
8	NV0008	MK3999
9	NV0009	MK0037
10	NV0010	MK7965
11	NV0011	MK0572
12	NV0012	MK4078
13	NV0013	MK0501
14	NV0014	MK4473
15	NV0015	MK1708
16	NV0016	MK7963
17	NV0017	MK8960
18	NV0018	MK9645
19	NV0019	MK2442
20	NV0020	MK0310
21	NV0021	MK7729
22	NV0022	MK0350
23	NV0023	MK5002
24	NV0024	MK6336
25	NV0025	MK3464

### 3.3.3.Bảng CALAM

```
-- Thủ tục thêm dữ liệu vào bảng CALAM
CREATE OR ALTER PROCEDURE PR_CALAM
AS
BEGIN
    DECLARE @i INT = 1;
    DECLARE @MACA VARCHAR(10);
    DECLARE @NGAY_LV DATE;
    DECLARE @BUOI_LAM NVARCHAR(10);
    DECLARE @GIO_BATDAU TIME;
    DECLARE @GIO_KETTHUC TIME;
    DECLARE @SOGIOLAMVIEC INT;

    WHILE @i <= 1000
    BEGIN
        -- Tạo mã ca ngẫu nhiên
        SET @MACA = 'CA' + RIGHT('000' + CAST(@i AS VARCHAR(4)), 4);

        -- Tạo ngày làm việc ngẫu nhiên trong khoảng từ 01-01-2023 đến 31-12-2023
        SET @NGAY_LV = DATEADD(DAY, ABS(CHECKSUM(NEWID())) % 365, '2023-01-01');

        -- Chọn buổi làm ngẫu nhiên
        SET @BUOI_LAM = CASE
            WHEN @i % 3 = 0 THEN N'Sáng'
            WHEN @i % 3 = 1 THEN N'Chiều'
            ELSE N'Tối'
        END;

        -- Giờ bắt đầu và giờ kết thúc ngẫu nhiên dựa trên buổi làm việc
        IF @BUOI_LAM = N'Sáng'
        BEGIN
            SET @GIO_BATDAU = '06:00';
            SET @GIO_KETTHUC = '11:00';
        END
        ELSE IF @BUOI_LAM = N'Chiều'
        BEGIN
            SET @GIO_BATDAU = '12:00';
            SET @GIO_KETTHUC = '17:00';
        END
        ELSE -- Buổi tối
        BEGIN
            SET @GIO_BATDAU = '18:00';
            SET @GIO_KETTHUC = '23:00';
        END

        -- Chèn dữ liệu vào bảng CALAM
        INSERT INTO CALAM (MACA, NGAY_LV, BUOI_LAM, GIO_BATDAU, GIO_KETTHUC, SOGIOLAMVIEC)
        VALUES (@MACA, @NGAY_LV, @BUOI_LAM, @GIO_BATDAU, @GIO_KETTHUC, @SOGIOLAMVIEC);

        -- Tăng biến đếm
        SET @i = @i + 1;
    END;
END;
```

Exec PR\_CALAM  
-- Kiểm tra kết quả  
SELECT \* FROM CALAM

	MACA	NGAY_LV	BUOI_LAM	GIO_BATDAU	GIO_KETTHUC	SOGIOLAMVIEC
1	CA0001	2023-07-11	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
2	CA0002	2023-04-28	Tối	18:00:00.0000000	23:00:00.0000000	NULL
3	CA0003	2023-08-25	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
4	CA0004	2023-01-17	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
5	CA0005	2023-11-23	Tối	18:00:00.0000000	23:00:00.0000000	NULL
6	CA0006	2023-12-29	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
7	CA0007	2023-07-13	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
8	CA0008	2023-06-26	Tối	18:00:00.0000000	23:00:00.0000000	NULL
9	CA0009	2023-06-03	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
10	CA0010	2023-12-14	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
11	CA0011	2023-09-07	Tối	18:00:00.0000000	23:00:00.0000000	NULL
12	CA0012	2023-10-19	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
13	CA0013	2023-07-25	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
14	CA0014	2023-11-03	Tối	18:00:00.0000000	23:00:00.0000000	NULL
15	CA0015	2023-11-09	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
16	CA0016	2023-12-25	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
17	CA0017	2023-07-13	Tối	18:00:00.0000000	23:00:00.0000000	NULL
18	CA0018	2023-02-09	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
19	CA0019	2023-03-18	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
20	CA0020	2023-10-01	Tối	18:00:00.0000000	23:00:00.0000000	NULL
21	CA0021	2023-08-19	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
22	CA0022	2023-04-18	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
23	CA0023	2023-02-15	Tối	18:00:00.0000000	23:00:00.0000000	NULL
24	CA0024	2023-03-20	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
25	CA0025	2023-11-19	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
26	CA0026	2023-04-20	Tối	18:00:00.0000000	23:00:00.0000000	NULL
27	CA0027	2023-05-15	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
28	CA0028	2023-04-15	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
29	CA0029	2023-08-26	Tối	18:00:00.0000000	23:00:00.0000000	NULL
30	CA0030	2023-03-22	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
31	CA0031	2023-12-28	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
32	CA0032	2023-06-11	Tối	18:00:00.0000000	23:00:00.0000000	NULL
33	CA0033	2023-05-07	Sáng	06:00:00.0000000	11:00:00.0000000	NULL
34	CA0034	2023-12-31	Chiều	12:00:00.0000000	17:00:00.0000000	NULL
35	CA0035	2023-07-20	Tối	18:00:00.0000000	23:00:00.0000000	NULL

### 3.3.4.Bảng LICHLAM

```
-- Thủ tục thêm dữ liệu vào bảng LICHLAM
CREATE OR ALTER PROCEDURE PR_LICHLAM
AS
BEGIN
    DECLARE @i INT = 1;
    DECLARE @TRANGTHAI INT;

    WHILE @i <= 1200
    BEGIN
        -- Xác định trạng thái ngẫu nhiên: 0 - Không di làm, 1 - Đã làm, 2 - Làm trễ
        SET @TRANGTHAI = ABS(CHECKSUM(NEWID())) % 3;

        -- Chèn dữ liệu vào bảng LICHLAM
        INSERT INTO LICHLAM (MANV, MACA, TRANGTHAI)
        VALUES (
            'NV' + RIGHT('000' + CAST(@i AS VARCHAR(4)), 4), -- MANV
            'CA' + RIGHT('000' + CAST(@i AS VARCHAR(4)), 4), -- MACA
            @TRANGTHAI -- TRANGTHAI: 0, 1, hoặc 2
        );

        -- Tăng biến đếm
        SET @i = @i + 1;
    END
END;

-- Thực thi thủ tục
EXEC PR_LICHLAM;

-- Kiểm tra kết quả
SELECT * FROM LICHLAM;
```

	MANV	MACA	TRANGTHAI
1	NV0001	CA0001	1
2	NV0002	CA0002	2
3	NV0003	CA0003	1
4	NV0004	CA0004	2
5	NV0005	CA0005	1
6	NV0006	CA0006	1
7	NV0007	CA0007	2
8	NV0008	CA0008	2
9	NV0009	CA0009	0
10	NV0010	CA0010	2
11	NV0011	CA0011	2
12	NV0012	CA0012	2
13	NV0013	CA0013	0
14	NV0014	CA0014	1
15	NV0015	CA0015	2
16	NV0016	CA0016	1
17	NV0017	CA0017	1
18	NV0018	CA0018	0
19	NV0019	CA0019	0
--	--	--	--

### 3.3.5.Bảng BANGLUONG

```
-- Thủ tục thêm dữ liệu vào bảng BANGLUONG
CREATE OR ALTER PROCEDURE PR_BANGLUONG
AS
BEGIN
    DECLARE @Counter INT = 1;

    WHILE @Counter <= 1200
    BEGIN
        -- Tạo giá trị cho các cột
        DECLARE @MANV VARCHAR(10) = CONCAT('NV', RIGHT('000' + CAST(@Counter AS VARCHAR(4)), 4))

        -- Thiết lập BANGLUONG_DATE cho từng tháng, bắt đầu từ tháng hiện tại và lùi dần về trước
        DECLARE @BANGLUONG_DATE DATE = DATEADD(MONTH, -(@Counter % 12), GETDATE())

        DECLARE @LUONGCOBAN DECIMAL(18, 2) = ROUND(1000000 + (RAND() * 4000000), 0)
        DECLARE @TONGLUONG DECIMAL(18, 2) = @LUONGCOBAN + ROUND((RAND() * 2000000), 0)

        -- Kiểm tra tồn tại của nhân viên trước khi chèn dữ liệu
        IF EXISTS (SELECT 1 FROM NHANVIEN WHERE MANV = @MANV)
        BEGIN
            -- Chèn dữ liệu vào bảng BANGLUONG
            INSERT INTO BANGLUONG (MANV, BANGLUONG_DATE, LUONGCOBAN, TONGLUONG)
            VALUES (@MANV, @BANGLUONG_DATE, @LUONGCOBAN, @TONGLUONG)
        END
        -- Tăng biến đếm
        SET @Counter = @Counter + 1
    END
    -- Thực thi thủ tục
    EXEC PR_BANGLUONG;
    -- Kiểm tra dữ liệu trong bảng BANGLUONG
    SELECT * FROM BANGLUONG;

```

	MANV	BANGLUONG_DATE	LUONGCOBAN	TONGLUONG
1	NV0001	2024-09-21	1909548.00	2830369.00
2	NV0002	2024-08-21	3144135.00	4739638.00
3	NV0003	2024-07-21	2387469.00	2818340.00
4	NV0004	2024-06-21	3831696.00	5658432.00
5	NV0005	2024-05-21	2004659.00	3432755.00
6	NV0006	2024-04-21	2199742.00	3391335.00
7	NV0007	2024-03-21	1631723.00	2796447.00
8	NV0008	2024-02-21	1843721.00	2721717.00
9	NV0009	2024-01-21	3940924.00	5078181.00
10	NV0010	2023-12-21	1140489.00	1358611.00
11	NV0011	2023-11-21	3981807.00	5456792.00
12	NV0012	2024-10-21	1186339.00	2664465.00
13	NV0013	2024-09-21	3421992.00	5011397.00
14	NV0014	2024-08-21	4798556.00	5761624.00
15	NV0015	2024-07-21	4141569.00	6023195.00
16	NV0016	2024-06-21	4205255.00	5258386.00
17	NV0017	2024-05-21	2656932.00	4151304.00
18	NV0018	2024-04-21	4128439.00	5618027.00
19	NV0019	2024-03-21	1145311.00	2836008.00
20	NV0020	2024-02-21	1736540.00	2339435.00
21	NV0021	2024-01-21	2643988.00	3670521.00
22	NV0022	2023-12-21	4645660.00	6371870.00
23	NV0023	2023-11-21	4078137.00	4986548.00
24	NV0024	2024-10-21	1579778.00	1822997.00
25	NV0025	2024-09-21	1886544.00	2863994.00
26	NV0026	2024-08-21	4317693.00	6024536.00
27	NV0027	2024-07-21	3242618.00	4576825.00
28	NV0028	2024-06-21	2127031.00	2201925.00
29	NV0029	2024-05-21	3380549.00	4439063.00
30	NV0030	2024-04-21	3103672.00	3364176.00
31	NV0031	2024-03-21	1771058.00	3352462.00
32	NV0032	2024-02-21	1694175.00	2861581.00
33	NV0033	2024-01-21	1142846.00	2969140.00

### 3.3.6 Bảng CHAMCONG

```
-- Thủ tục thêm dữ liệu vào bảng CHAMCONG
CREATE OR ALTER PROCEDURE PR_CHAMCONG
AS
BEGIN
    DECLARE @i INT = 1;
    DECLARE @MANV VARCHAR(10);
    DECLARE @CHAMCONG_DATE DATE;
    DECLARE @TONGSOGIOLAM INT;
    DECLARE @THUONG DECIMAL(18, 2);
    DECLARE @PHAT DECIMAL(18, 2);

    WHILE @i <= 1200
    BEGIN
        -- Tạo giá trị cho các cột
        SET @MANV = 'NV' + RIGHT('000' + CAST(@i AS VARCHAR(4)), 4);
        SET @CHAMCONG_DATE = DATEADD(DAY, @i % 30, '2023-01-01');
        SET @TONGSOGIOLAM = ROUND(RAND() * 24, 0); -- Bã sửa đổi
        SET @THUONG = ROUND(RAND() * 1000000, 2);
        SET @PHAT = -ROUND(RAND() * 500000, 2);

        -- Chèn dữ liệu vào bảng CHAMCONG
        IF EXISTS (SELECT 1 FROM NHANVIEN WHERE MANV = @MANV)
        BEGIN
            INSERT INTO CHAMCONG (MANV, CHAMCONG_DATE, TONGSOGIOLAM, THUONG, PHAT)
            VALUES (@MANV, @CHAMCONG_DATE, @TONGSOGIOLAM, @THUONG, @PHAT);
        END
        SET @i = @i + 1;
    END
END;
```

47 %

Results Messages

	MANV	CHAMCONG_DATE	THUONG	PHAT	TONGSOGIOLAM
1	NV0001	2023-01-02	106444.63	-217831.45	4
2	NV0002	2023-01-03	81627.55	-399268.03	9
3	NV0003	2023-01-04	481963.92	-44290.66	20
4	NV0004	2023-01-05	228737.39	-304009.21	3
5	NV0005	2023-01-06	783985.77	-274451.51	5
6	NV0006	2023-01-07	377241.53	-215854.98	24
7	NV0007	2023-01-08	558731.49	-377192.66	20
8	NV0008	2023-01-09	513721.86	-51558.25	6
9	NV0009	2023-01-10	987161.67	-264668.78	14
10	NV0010	2023-01-11	362156.48	-184642.88	20
11	NV0011	2023-01-12	378663.88	-82381.81	1
12	NV0012	2023-01-13	405503.92	-7793.20	5
13	NV0013	2023-01-14	444221.50	-148735.49	18
14	NV0014	2023-01-15	296719.88	-91704.66	10
15	NV0015	2023-01-16	743542.82	-292109.30	8
16	NV0016	2023-01-17	925874.78	-242218.94	8
17	NV0017	2023-01-18	31164.23	-333702.49	17
18	NV0018	2023-01-19	154272.64	-274263.48	22
19	NV0019	2023-01-20	924067.85	-100412.82	11
20	NV0020	2023-01-21	476685.09	-359963.02	20
21	NV0021	2023-01-22	556164.17	-234104.45	10
22	NV0022	2023-01-23	65333.52	-35966.09	1
23	NV0023	2023-01-24	144525.53	-250457.58	11
24	NV0024	2023-01-25	926062.84	-2928.14	8
25	NV0025	2023-01-26	134396.62	-106257.85	9
26	NV0026	2023-01-27	176886.80	-308294.45	13
27	NV0027	2023-01-28	597904.99	-274136.30	3
28	NV0028	2023-01-29	135578.44	-401387.18	14
29	NV0029	2023-01-30	46242.82	-331359.60	5
30	NV0030	2023-01-01	760252.32	-197865.13	1
31	NV0031	2023-01-02	348454.61	-325941.10	10

### 3.3. 10 MODULE XỬ LÝ DỮ LIỆU

#### 3.3.1. MODULE 1

<b>Tên module</b>	Trigger tự động cộng số công và tính số tiền phạt theo tháng dựa trên trạng thái đi làm của nhân viên .
<b>Mô tả</b>	<ul style="list-style-type: none"> <li>Khi nhân viên đăng ký ca làm và có đi làm vào ngày hôm đó (trạng thái :1) thì được ghi nhận số giờ làm bằng cách cộng số giờ làm việc vào tổng số giờ làm việc trong tháng</li> <li>Nếu nhân viên không đi làm (trạng thái 0) tự động thêm tiền phạt 100k trong tháng và không được ghi nhận số giờ làm.</li> <li>Nếu nhân viên đi làm trễ (trạng thái 2) và tổng số lần đi làm trễ trên 3 lần thì tự động thêm tiền phạt 100k trong tháng và vẫn được ghi nhận số giờ làm.</li> </ul> <p>Note: trạng thái 1 là có đi làm và đi làm đúng giờ , trạng thái 0 là không đi làm , trạng thái 2 là trễ giờ làm</p>
<b>Phân tích</b>	<p>Bảng:LICHLAM</p> <p>Loại: After</p> <p>Sự kiện :Update</p> <p>Process:</p> <ol style="list-style-type: none"> <li>Lấy @TRANGTHAI , @MANV , @MACA từ inserted</li> <li>Lấy @Ngay_LV,@SOGIOLAMVIEC từ bảng CALAM theo @MACA</li> <li>Lấy @TONGCACHAMCONG từ bảng CHAMCONG theo @MANV và theo tháng và năm của @NGAY_LV</li> <li></li> <li>Nếu @TRANGTHAI=1 (nhân viên có đi làm và đi đúng giờ)</li> </ol>

-->Cập nhật bảng CHAMCONG: TONGCACHAMCONG =  
TONGCACHAMCONG + @SOGIOLAMVIEC

Điều kiện: MANV=@MANV AND YEAR(CHAMCONG\_DATE) = @YEAR  
AND MONTH(CHAMCONG\_DATE) = @MONTH

4.2. Nếu @TRANGTHAI=0 (nhân viên không đi làm)

->Cập nhật bảng CHAMCONG: PHAT=PHAT-100000

Điều kiện: MANV=@MANV AND YEAR(CHAMCONG\_DATE) = @YEAR  
AND MONTH(CHAMCONG\_DATE) = @MONTH

4.3. Nếu @TRANGTHAI=2 (nhân viên đi làm trễ)

→Cập nhật bảng CHAMCONG :TONGCACHAMCONG =  
TONGCACHAMCONG + @SOGIOLAMVIEC

Điều kiện: MANV=@MANV AND YEAR(CHAMCONG\_DATE) = @YEAR  
AND MONTH(CHAMCONG\_DATE) = @MONTH

- Đếm số lần nhân viên đó đi trễ trong tháng

- Nếu nhân viên đó đi trễ >=3 lần trong tháng đó

-->Cập nhật bảng CHAMCONG :PHAT=PHAT-100000

Điều kiện: MANV=@MANV AND YEAR(CHAMCONG\_DATE) = @YEAR  
AND MONTH(CHAMCONG\_DATE) = @MONTH

## Code

```
CREATE OR ALTER TRIGGER trg_TrangThai
ON LICHIAM
AFTER UPDATE
AS
BEGIN
    DECLARE @TONGSOGIOLAM INT,
            @TRANGTHAI INT,
            @MANV VARCHAR(10),
            @MACA_CALAM VARCHAR(10),
            @MACA VARCHAR(10),
            @SOGIOLAMVIEC INT,
            @COUNT INT,
            @NGAY_LV DATE,
            @YEAR INT,
            @MONTH INT

    SELECT @TRANGTHAI = TRANGTHAI, @MANV = MANV, @MACA = MACA
    FROM INSERTED

    SELECT @NGAY_LV = NGAY_LV, @SOGIOLAMVIEC = SOGIOLAM
    FROM CALAM
    WHERE MACA=@MACA

    SET @YEAR = YEAR(@NGAY_LV)
    SET @MONTH = MONTH(@NGAY_LV)

    SELECT @TONGSOGIOLAM = TONGSOGIOLAM
    FROM CHAMCONG
    WHERE MANV = @MANV AND YEAR(CHAMCONG_DATE) = @YEAR AND MONTH(CHAMCONG_DATE) = @MONTH

    IF @TRANGTHAI = 1
    BEGIN
        UPDATE CHAMCONG
        SET TONGSOGIOLAM = ISNULL(@TONGSOGIOLAM, 0) + ISNULL(@SOGIOLAMVIEC, 0)
        WHERE MANV = @MANV AND YEAR(CHAMCONG_DATE) = @YEAR AND MONTH(CHAMCONG_DATE) = @MONTH
    END
    ELSE IF @TRANGTHAI = 0
    BEGIN
        UPDATE CHAMCONG
        SET PHAT = ISNULL(PHAT, 0) - 100000
        WHERE MANV = @MANV AND YEAR(CHAMCONG_DATE) = @YEAR AND MONTH(CHAMCONG_DATE) = @MONTH;
    END
    ELSE IF @TRANGTHAI = 2
    BEGIN
        SELECT @COUNT = COUNT(L.MACA)
        FROM CALAM C JOIN LICHIAM L ON C.MACA=L.MACA
        WHERE YEAR(NGAY_LV) = @YEAR AND MONTH(NGAY_LV) = @MONTH AND TRANGTHAI=2 AND MANV=@MANV

        UPDATE CHAMCONG
        SET PHAT = ISNULL(PHAT, 0) - 100000
        WHERE MANV = @MANV AND YEAR(CHAMCONG_DATE) = @YEAR AND MONTH(CHAMCONG_DATE) = @MONTH

        IF @COUNT >= 3
        BEGIN

            UPDATE CHAMCONG
            SET TONGSOGIOLAM = ISNULL(@TONGSOGIOLAM, 0) + ISNULL(@SOGIOLAMVIEC, 0)
            WHERE MANV = @MANV AND YEAR(CHAMCONG_DATE) = @YEAR AND MONTH(CHAMCONG_DATE) = @MONTH
        END
    END
END
END
```

Test	<pre> /*TEST TRƯỜNG HỢP 3 */ -----TẠO CA LÀM VIỆC MỚI VỚI CÙNG THÁNG VÀ NĂM  INSERT INTO CALAM (MACA, NGAY_LV, BUOI_LAM, GIO_BATDAU, GIO_KETTHUC) VALUES ('CA2400', '2024-11-20', 'Sáng', '08:00', '12:00'),        ('CA2401', '2024-11-21', 'Sáng', '08:00', '12:00'),        ('CA2402', '2024-11-22', 'Sáng', '08:00', '12:00');  -----NHÂN VIÊN BẮNG KÝ CA  INSERT INTO LICHLAM (MANV, MACA, TRANGTHAI) VALUES ('NV0002', 'CA2400', NULL), ('NV0002', 'CA2401', NULL), ('NV0002', 'CA2402', NULL); -----NHÂN VIÊN UPDATE TRẠNG THÁI ĐI LÀM TRỄ UPDATE LICHLAM SET TRANGTHAI = 2 WHERE MANV = 'NV0002' AND MACA IN ('CA2400', 'CA2401', 'CA2402')  /*TEST: TRƯỜNG HỢP 2 */  INSERT INTO LICHLAM (MANV, MACA, TRANGTHAI) VALUES ('NV0004', 'CA0014', NULL);  UPDATE LICHLAM SET TRANGTHAI = 0 WHERE MANV = 'NV0004' AND MACA = 'CA0014'  /*TEST TRƯỜNG HỢP 1 */  INSERT INTO LICHLAM (MANV, MACA, TRANGTHAI) VALUES ('NV0004', 'CA0013', NULL); </pre>
------	---

### 3.3.2 MODULE 2

Tên module	Trigger tự động thêm số giờ làm cho ca làm . Nếu ca làm việc diễn ra vào ngày lễ (30/4-1/5 và 2/9) , tết dương lịch (1/1) thì số giờ làm việc nhân 3. Nếu không thì số giờ làm việc để bình thường.
Mô tả	<ul style="list-style-type: none"> <li>Khi quản lý thêm ca làm cho nhân viên.Tự động cập nhật số giờ làm việc = giờ kết thúc - giờ tan làm</li> <li>Nếu ca làm có ngày làm việc vào ngày lễ (cụ thể : 30/4-1/4 , 2/9) và ngày tết dương lịch (1/1) thì số giờ làm việc sẽ tăng gấp 3 lần so với số giờ bình thường.</li> <li>Mục đích của việc này là để đến cuối tháng khi tính lương , nhân viên nào làm việc vào ngày lễ-tết sẽ được nhân 3 số lương bình thường.</li> </ul>
Phân tích	<p>Bảng : CALAM</p> <p>Loại: After</p> <p>Sự kiện: Insert</p>

	<p>Process:</p> <p>1.Lấy @MACA , @GIO_BATDAU, @GIO_KETTHUC, @NGAY_LV từ bảng inserted</p> <p>2.Tìm số giờ làm việc -&gt; @SOGIOLAMVIEC = DATEDIFF(HOUR, @GIO_BATDAU, @GIO_KETTHUC)</p> <p>3.Nếu ca làm có ngày làm việc vào ngày 30/4 , 1/5 ,2/9 và 1/1 --&gt; Cập nhật bảng CALAM: SOGIOLAMVIEC = @SOGIOLAMVIEC * 3 Điều kiện : MACA=@MACA</p> <p>4.Nếu không: --&gt; Cập nhật bảng CALAM: SOGIOLAMVIEC = @SOGIOLAMVIEC Điều kiện : MACA=@MACA</p>
Code	<pre> CREATE TRIGGER trg_TangLuong ON CALAM AFTER INSERT AS BEGIN     DECLARE @MACA VARCHAR(10), @SOGIOLAMVIEC INT, @NGAY_LV DATE,             @GIO_BATDAU TIME, @GIO_KETTHUC TIME;      SELECT @MACA = MACA, @GIO_BATDAU = GIO_BATDAU,            @GIO_KETTHUC = GIO_KETTHUC, @NGAY_LV = NGAY_LV     FROM inserted;      SET @SOGIOLAMVIEC = DATEDIFF(HOUR, @GIO_BATDAU, @GIO_KETTHUC);      IF (MONTH(@NGAY_LV) = 4 AND DAY(@NGAY_LV) = 30) OR        (MONTH(@NGAY_LV) = 5 AND DAY(@NGAY_LV) = 1) OR        (MONTH(@NGAY_LV) = 9 AND DAY(@NGAY_LV) = 2) OR        (MONTH(@NGAY_LV) = 1 AND DAY(@NGAY_LV) = 1)     BEGIN         UPDATE CALAM         SET SOGIOLAMVIEC = @SOGIOLAMVIEC * 3         WHERE MACA = @MACA;     END     ELSE     BEGIN         UPDATE CALAM         SET SOGIOLAMVIEC = @SOGIOLAMVIEC         WHERE MACA = @MACA;     END END </pre>
Test	<pre> select * from calam ---TEST trường hợp ca làm vào ngày lễ - tết insert into calam(MACA,NGAY_LV,BUOI_LAM,GIO_BATDAU,GIO_KETTHUC)     values ('CA1216','2024-1-1','N'Sáng','08:00:00','09:00:00') ---TEST trường hợp ca làm vào ngày bình thường insert into calam(MACA,NGAY_LV,BUOI_LAM,GIO_BATDAU,GIO_KETTHUC)     values ('CA1217','2024-2-1','N'Sáng','08:00:00','09:00:00') </pre>

### 3.3.3 MODULE 3

<b>Tên module</b>	Khi thông tin nhân viên mới được thêm vào , hệ thống tự động cập nhật thông tin nhân viên đó vào cơ sở dữ liệu.
<b>Mô tả</b>	Hệ thống tự động thêm thông tin nhân viên vào cơ sở dữ liệu khi các thông tin của nhân viên được thỏa mãn .Mục đích của việc này là để đảm bảo dữ liệu nhập vào của nhân viên.
<b>Phân tích</b>	<p>Bảng: NHANVIEN</p> <p>Loại: After</p> <p>Sự kiện: Insert</p> <p>Process:</p> <ol style="list-style-type: none"> <li>1.Nhập Thông tin: Nhân viên điền thông tin vào web.</li> <li>2.Xác thực Dữ liệu: Hệ thống kiểm tra tính hợp lệ của thông tin (định dạng số điện thoại, trường bắt buộc).</li> <li>3.Nếu có lỗi, hệ thống thông báo cho người dùng “Thông tin nhập không hợp lệ, mời nhập lại” để nhân viên chỉnh sửa.</li> <li>4.Nếu số điện thoại &lt;10 hoặc &gt;12, hiển thị thông báo lỗi và rollback</li> <li>5.Nếu địa chỉ, chức vụ, tên nhân viên mà null, hiển thị thông báo lỗi và rollback.</li> <li>6.Nếu CCCD và SDT bị trùng, hiển thị thông báo lỗi và null.</li> <li>7.Nếu thông tin hợp lệ, hệ thống sẽ cập nhật tự động vào cơ sở dữ liệu.</li> </ol>

Code	<pre> CREATE OR ALTER TRIGGER tr_capnhatd1 ON NHANVIEN AFTER INSERT AS     DECLARE @MANV VARCHAR(10);     DECLARE @NGAYSINH DATE;     DECLARE @SDT VARCHAR(15);     DECLARE @CCCD VARCHAR(50);     DECLARE @CHUCVU NVARCHAR(50);     DECLARE @TENNIV NVARCHAR(255);     DECLARE @DIACHI NVARCHAR(255);      BEGIN         -- Lấy thông tin từ bản ghi mới chèn         SELECT @MANV = inserted.MANV,                @NGAYSINH = inserted.NGAYSINH,                @SDT = inserted.SDT,                @CCCD = inserted.CCCD,                @CHUCVU = inserted.CHUCVU,                @TENNIV = inserted.TENNIV,                @DIACHI = inserted.DIACHI         FROM inserted;         -- Kiểm tra các điều kiện cần thiết         IF LEN(@SDT)&lt;10 OR LEN(@SDT)&gt;12         BEGIN             RAISERROR('Thông tin nhập không hợp lệ, mời nhập lại', 16, 1);             ROLLBACK; -- Hoặc xử lý khác tùy ý         END         IF @DIACHI IS NULL         BEGIN             RAISERROR('Thông tin nhập không hợp lệ, mời nhập lại', 16, 1);             ROLLBACK; -- Hoặc xử lý khác tùy ý         END         IF @CHUCVU IS NULL         BEGIN             RAISERROR('Thông tin nhập không hợp lệ, mời nhập lại', 16, 1);             ROLLBACK; -- Hoặc xử lý khác tùy ý         END         IF @TENNIV IS NULL         BEGIN             RAISERROR('Thông tin nhập không hợp lệ, mời nhập lại', 16, 1);             ROLLBACK; -- Hoặc xử lý khác tùy ý         END         IF EXISTS (SELECT 1 FROM NHANVIEN WHERE CCCD = @CCCD)         BEGIN             RAISERROR('Thông tin nhập không hợp lệ, mời nhập lại', 16, 1);             ROLLBACK; -- Hoặc xử lý khác tùy ý         END         IF EXISTS (SELECT 1 FROM NHANVIEN WHERE SDT = @SDT)         BEGIN             RAISERROR('Thông tin nhập không hợp lệ, mời nhập lại', 16, 1);             ROLLBACK; -- Hoặc xử lý khác tùy ý         END         -- Các hành động khác nếu cần         -- Ví dụ: ghi nhật ký, cập nhật bảng khác, v.v.         UPDATE NHANVIEN         SET NGAYSINH = @NGAYSINH,             SDT = @SDT,             CCCD = @CCCD,             CHUCVU = @CHUCVU,             TENNIV = @TENNIV,             DIACHI = @DIACHI         WHERE MANV = @MANV;     END; </pre>
Test	<pre> INSERT INTO NHANVIEN (MANV, NGAYSINH, SDT, CCCD, CHUCVU, TENNIV, DIACHI) VALUES ('NV001', '1990-01-01', '0123456789', 'CCCD001', 'Phá chế', 'Nguyễn Văn A', 'Hà Nội'); </pre>

### 3.3.4 MODULE 4

Tên module	Trigger tự động random mật khẩu tài khoản cho nhân viên khi quản lý thêm một mã nhân viên mới trong tài khoản.
Mô tả	Nếu quản lý thêm một mã nhân viên mới ở bảng NHANVIEN thì hệ thống sẽ tạo tự động random mật khẩu ban đầu cho nhân viên đó. Mục đích là để tạo tài khoản cho nhân viên.
Phân	Bảng: TAIKHOAN

tích	<p>Sự kiện : After</p> <p>Loại : Insert</p> <p>Process :</p> <ol style="list-style-type: none"> <li>1.Lấy @Manv từ bảng insterted</li> <li>2.Dùng biến RAND()*10000 để lấy giá trị random mật khẩu từ 0 đến 9999</li> <li>3.Chuyển giá trị random đó sang số nguyên với 4 chữ số</li> <li>4.Mật khẩu được thêm với điều kiện : @MATKHAU='MK'+ 4 chữ số random phía sau</li> </ol>
Code	<pre> CREATE OR ALTER TRIGGER trg_RandomPassword ON TAIKHOAN AFTER INSERT AS BEGIN     DECLARE @MANV VARCHAR(10),             @MATKHAU VARCHAR(225)      SELECT @MANV = MANV     FROM INSERTED      SET @MATKHAU = 'MK' + RIGHT('0000' + CAST(CAST(RAND() * 10000 AS INT) AS VARCHAR(4)), 4)      UPDATE TAIKHOAN     SET MATKHAU = @MATKHAU     WHERE MANV = @MANV END </pre>
Test	<pre> INSERT INTO NHANVIEN (MANV, TENNV, NGAYSINH, DIACHI, SDT, CCCD, CHUCVU) VALUES ('NV2005', 'Nguyen Van A', '1990-01-01', '123 Street', '0123475679', 'CCCD123755', 'Phuc vu') </pre>

### 3.3.5 MODULE 5

Tên module	Thủ tục đưa ra thông báo khi nhân viên đăng ký ca làm . Nếu số lượng người đăng kí 1 ca đã đủ , đưa ra thông báo " Ca đã đủ số lượng đăng ký ". Nếu không , đưa ra thông báo đăng kí thành công. Quy định ( 1 Pha chế , 2 Phục vụ)
Mô tả	Thủ tục này tự động đăng ký ca làm việc cho nhân viên tại quán cà phê, đảm bảo: 1 nhân viên pha chế. 2 nhân viên phục vụ.
Phân tích	Input: @MaNhanVien, @MaCa, @ChucVu:

Output: Thông báo thành công hoặc thất bại: Thủ tục sẽ in ra thông báo:"Nhân viên đã đăng ký ca này, không thể đăng ký lại!" nếu nhân viên đã đăng ký ca đó.

"Ca này đã đủ Pha Ché, không thể đăng ký thêm!" nếu số nhân viên Pha Ché trong ca đã đủ.

"Ca này đã đủ Phục Vụ, không thể đăng ký thêm!" nếu số nhân viên Phục Vụ trong ca đã đủ.

"Đăng ký ca làm thành công!" nếu đăng ký thành công.

Process:

1. Kiểm tra nhân viên đã đăng ký ca làm chưa
2. Kiểm tra số lượng nhân viên phục vụ trong ca COUNT(\*) để đếm số lượng nhân viên có chức vụ "Phục Vụ" đã đăng ký trong ca cụ thể.
- 3.Kiểm tra số lượng nhân viên pha chế trong ca COUNT(\*) để đếm số lượng nhân viên có chức vụ "Pha Ché" đã đăng ký trong ca.
4. Kiểm tra điều kiện trước khi đăng ký
5. Đăng ký nhân viên vào ca

Code	<pre> CREATE OR ALTER PROCEDURE DangKyCaLam     @MaNhanVien VARCHAR(10),     @MaCa VARCHAR(10),     @ChucVu NVARCHAR(50) AS BEGIN     DECLARE @SoPhucVu INT, @SoPhaChe INT;      -- Kiểm tra xem nhân viên đã đăng ký ca này chưa     IF EXISTS (SELECT 1 FROM LICHCLAM WHERE MANV = @MaNhanVien AND MACA = @MaCa)     BEGIN         PRINT 'Nhân viên đã đăng ký ca này, không thể đăng ký lại!';         RETURN;     END      -- Kiểm tra số lượng nhân viên phục vụ đã đăng ký trong ca     SELECT @SoPhucVu = COUNT(*)     FROM LICHCLAM LL     JOIN NHANVIEN NV ON LL.MANV = NV.MANV     WHERE LL.MACA = @MaCa AND NV.CHUCVU = N'Phục Vu';      -- Kiểm tra số lượng nhân viên pha chế đã đăng ký trong ca     SELECT @SoPhaChe = COUNT(*)     FROM LICHCLAM LL     JOIN NHANVIEN NV ON LL.MANV = NV.MANV     WHERE LL.MACA = @MaCa AND NV.CHUCVU = N'Pha Che';      -- Kiểm tra điều kiện trước khi đăng ký     IF @ChucVu = N'Pha Chế' AND @SoPhaChe &gt;= 1     BEGIN         PRINT 'Ca này đã đủ Pha Chế, không thể đăng ký thêm!';     END     ELSE IF @ChucVu = N'Phục Vu' AND @SoPhucVu &gt;= 2     BEGIN         PRINT 'Ca này đã đủ Phục Vu, không thể đăng ký thêm!';     END     ELSE     BEGIN         -- Thêm nhân viên vào lịch làm việc         INSERT INTO LICHCLAM (MANV, MACA)         VALUES (@MaNhanVien, @MaCa);         PRINT 'Đăng ký ca làm thành công!';     END END; </pre>
Test	<pre> -- Thử đăng ký cho nhân viên NV01 vào ca CA002 (Phục Vu) EXEC DangKyCaLam 'NV0002', 'CA0002', N'Phuc Vu'; EXEC DangKyCaLam 'NV0002', 'CA0002', N'Phuc Vu'; EXEC DangKyCaLam 'NV0003', 'CA0002', N'Phuc Vu'; EXEC DangKyCaLam 'NV0004', 'CA0002', N'Phuc Vu';  -- Kiểm tra dữ liệu SELECT * FROM NHANVIEN; SELECT * FROM LICHCLAM; </pre>

### 3.3.6 MODULE 6

Tên module	Giới hạn số ca làm việc của nhân viên
Mô tả	<p>Giới hạn số ca làm việc của nhân viên không vượt quá 15 ca/ 1 tuần .</p> <p>Nếu nhân viên đã đạt đến số lượng ca tối đa (15 ca trong một tuần), thủ tục sẽ thông báo để ngăn họ đăng ký thêm.Ngược lại, nếu số ca làm còn nằm trong giới hạn, nhân viên sẽ được phép tiếp tục đăng ký ca làm.</p>
Phân	Input: @MaNhanVien, @NgayHienTai:

tích	<p>Output:</p> <p>Nếu số ca làm trong tuần đã đạt 15: "Bạn đã đăng ký đủ số ca trong tuần. Vui lòng không đăng ký thêm đến khi qua tuần tiếp theo!"</p> <p>Nếu số ca làm còn dưới 15: "Bạn có thể tiếp tục đăng ký ca làm việc."</p> <p>Process:</p> <ol style="list-style-type: none"> <li>1. Tính số ca làm việc trong 7 ngày gần nhất của nhân viên từ bảng LICHLAM</li> <li>2. Kiểm tra số ca làm việc có vượt quá giới hạn <ul style="list-style-type: none"> <li>+ Nếu số ca lớn hơn hoặc bằng 15: In ra thông báo "Bạn đã đăng ký đủ số ca trong tuần.Vui lòng không đăng ký thêm đến khi qua tuần tiếp theo!".</li> <li>+ Nếu số ca ít hơn 15: In ra thông báo "Bạn có thể tiếp tục đăng ký ca làm việc.".</li> </ul> </li> </ol>
Code	<pre> CREATE OR ALTER PROCEDURE KiemTraCaLamTrongTuan     @MaNhanVien VARCHAR(10),     @NgayHienTai DATE AS BEGIN     DECLARE @SoCaLamTrongTuan INT;      -- Tính số ca làm của nhân viên trong 7 ngày gần nhất     SELECT @SoCaLamTrongTuan = COUNT(*)     FROM LICHLAM LL     JOIN CALAM CL ON LL.MACA = CL.MACA     WHERE LL.MANV = @MaNhanVien         AND CL.NGAY_LV BETWEEN DATEADD(DAY, -7, @NgayHienTai) AND @NgayHienTai;      -- Kiểm tra nếu số ca làm vượt quá 15 ca     IF @SoCaLamTrongTuan &gt;= 15     BEGIN         PRINT N'Bạn đã đăng ký đủ số ca trong tuần. Vui lòng không đăng ký thêm đến khi qua tuần tiếp theo!';     END     ELSE     BEGIN         PRINT N'Bạn có thể tiếp tục đăng ký ca làm việc.';     END END; </pre>
Test	<pre> --test -- Giả sử hôm nay là ngày 2024-10-20 và kiểm tra cho nhân viên NV01 EXEC KiemTraCaLamTrongTuan 'NV0002', '2024-10-20'; SELECT * FROM CALAM; SELECT * FROM LICHLAM; SELECT * FROM NHANVIEN; </pre>

### 3.3.7 MODULE 7

Tên module	Trigger tự động tính toán và cập nhật tổng lương (TONGLUONG) cho nhân viên
------------	--

Mô tả	<p>Trigger sẽ tự động tính tổng lương cho nhân viên dựa trên dữ liệu từ bảng CHAMCONG và cập nhật vào bảng BANGLUONG.</p> <p>Dữ liệu đầu vào gồm mã nhân viên, tổng số giờ làm việc, chức vụ, thưởng, và phạt.</p> <p>Dựa vào chức vụ để biết lương cơ bản của nhân viên.</p> <p>Tổng lương sẽ được tính dựa vào công thức sau: TONGLUONG = LUONGCOBAN * TONGGIOLAM + THUONG/PHAT</p> <p>Trigger sẽ thực hiện vào các sự kiện INSERT và UPDATE trên bảng CHAMCONG</p>
Phân tích	<p>Bảng: CHAMCONG</p> <p>Loại: after</p> <p>Sự kiện: insert, update</p> <p>Process:</p> <ol style="list-style-type: none"> <li>1. Từ bảng CHAMCONG, Trigger lấy các thông tin về MANV, CHAMCONG_DATE, TONGSOGIOLAM, THUONG, PHAT</li> <li>2. Tính tổng số giờ làm trong tháng của nhân viên</li> <li>3. Cập nhật TONGSOGIO vào bảng CHAMCONG</li> <li>4. Xác định CHUCVU của nhân viên từ bảng NHANVIEN. Dựa vào CHUCVU, trigger sẽ tự động cập nhập lương cơ bản cho nhân viên:           <ol style="list-style-type: none"> <li>4.1 Nếu là phục vụ: lương cơ bản là 20,000.</li> <li>4.2 Nếu là quản lý: lương cơ bản là 25,000.</li> </ol> </li> <li>5. Trigger sẽ tự động tính TONGLUONG bằng công thức sau:  <math display="block">\text{TONGLUONG} = \text{LUONGCOBAN} * \text{TONGGIOLAM} + \text{THUONG} - \text{ABS(PHAT)}</math> </li> <li>6. Cập nhật TONGLUONG trong bảng BANGLUONG.</li> </ol>

## Code

```

CREATE OR ALTER TRIGGER TRG_TINHTONGLUONG
ON CHAMCONG
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @MANV VARCHAR(10),
            @CHAMCONG_DATE DATE,
            @THUONG DECIMAL(18, 2),
            @PHAT DECIMAL(18, 2),
            @CHUCVU NVARCHAR(50),
            @LUONGCOBAN DECIMAL(18, 2),
            @TONGLUONG DECIMAL(18, 2),
            @SOGIOLAM INT,
            @YEAR INT,
            @MONTH INT;

    SELECT TOP 1
        @MANV = i.MANV,
        @CHAMCONG_DATE = i.CHAMCONG_DATE,
        @THUONG = ISNULL(i.THUONG, 0),
        @PHAT = ISNULL(i.PHAT, 0)
    FROM inserted i;

    SET @YEAR = YEAR(@CHAMCONG_DATE);
    SET @MONTH = MONTH(@CHAMCONG_DATE);

    SELECT @SOGIOLAM = SUM(c.SOGIOLAMVIEC)
    FROM CALAM c
    JOIN CHAMCONG cc ON cc.MANV = @MANV
    WHERE cc.MANV = @MANV
    AND YEAR(cc.CHAMCONG_DATE) = @YEAR
    AND MONTH(cc.CHAMCONG_DATE) = @MONTH;

    UPDATE CHAMCONG
    SET TONGSGOTOLAM = @SOGIOLAM
    WHERE MANV = @MANV
    AND YEAR(CHAMCONG_DATE) = @YEAR
    AND MONTH(CHAMCONG_DATE) = @MONTH;

    SELECT @CHUCVU = n.CHUCVU
    FROM NHANVIEN n
    WHERE n.MANV = @MANV;

    SET @LUONGCOBAN = CASE
        WHEN @CHUCVU = N'Phuc vu' THEN 20000
        WHEN @CHUCVU = N'Pha Che' THEN 25000
        ELSE 0
    END;

    -- Tính tổng lương
    SET @TONGLUONG = (@LUONGCOBAN * @SOGIOLAM) + @THUONG - ABS(@PHAT);

    -- Cập nhật hoặc thêm mới thông tin vào bảng BANGLUONG
    IF EXISTS (SELECT 1 FROM BANGLUONG WHERE MANV = @MANV)
    BEGIN
        UPDATE BANGLUONG
        SET TONGLUONG = @TONGLUONG, LUONGCOBAN = @LUONGCOBAN, BANGLUONG_DATE = GETDATE()
        WHERE MANV = @MANV;
    END
    ELSE
    BEGIN
        INSERT INTO BANGLUONG (MANV, TONGLUONG, LUONGCOBAN, BANGLUONG_DATE)
        VALUES (@MANV, @TONGLUONG, @LUONGCOBAN, GETDATE());
    END;
END;

```

## Test

```

INSERT INTO CHAMCONG (MANV, CHAMCONG_DATE, THUONG, PHAT)
VALUES ('NV0001', '2023-10-20', 50000, 0);

INSERT INTO CHAMCONG (MANV, CHAMCONG_DATE, THUONG, PHAT)
VALUES ('NV0002', '2023-10-20', 50000, 0);

SELECT * FROM BANGLUONG WHERE MANV = 'NV0001';

```

### 3.3.8 MODULE 8

Tên module	Tính phần thưởng và cập nhập tổng lương cho người làm việc hiệu quả nhất
Mô tả	Tính toán và cập nhật phần thưởng cho nhân viên có tổng số giờ làm việc cao nhất trong tháng. Tổng phần thưởng là 500,000 VNĐ, sẽ được chia đều cho số lượng nhân viên làm việc hiệu quả nhất, sau đó cập nhật vào cột TONGLUONG trong bảng BANGLUONG.
Phân	Input: Không có đầu vào

tích	<p>Output: cập nhật bảng lương</p> <p>Process:</p> <ol style="list-style-type: none"> <li>1. Thiết lập tổng tiền thưởng</li> <li>2. Tìm số giờ làm việc nhiều nhất trong tháng</li> <li>3. Đếm số người có số giờ làm việc cao nhất</li> <li>4. Chia phần thưởng và cập nhật tổng lương</li> <li>5. Cập nhật tổng lương của nhân viên trong bảng BANGLUONG</li> </ol>
Code	<pre> CREATE OR ALTER PROCEDURE SP_THUONG_HIEUQUA AS BEGIN     DECLARE @MAX_GIO INT,             @SO_NGUOI INT,             @TONG_TIEN_THUONG DECIMAL(18, 2),             @THUONG_MOI_NGUOI DECIMAL(18, 2);      SET @TONG_TIEN_THUONG = 500000;      -- Tìm số giờ làm việc nhiều nhất và số nhân viên có số giờ đó trong tháng hiện tại     SELECT @MAX_GIO = MAX(TONGSOGIOLAM),            @SO_NGUOI = COUNT(DISTINCT MANV)     FROM CHAMCONG     WHERE MONTH(CHAMCONG_DATE) = MONTH(GETDATE())     AND YEAR(CHAMCONG_DATE) = YEAR(GETDATE());      SET @THUONG_MOI_NGUOI = @TONG_TIEN_THUONG / @SO_NGUOI;      UPDATE BANGLUONG     SET TONGLUONG = TONGLUONG + @THUONG_MOI_NGUOI     WHERE MANV IN (         SELECT MANV         FROM CHAMCONG         WHERE TONGSOGIOLAM = @MAX_GIO         AND MONTH(CHAMCONG_DATE) = MONTH(GETDATE())         AND YEAR(CHAMCONG_DATE) = YEAR(GETDATE())     );      SELECT BL.MANV,            BL.TONGLUONG - @THUONG_MOI_NGUOI AS TONGLUONG_TRUOC, -- Tổng lương trước khi cộng thưởng            @THUONG_MOI_NGUOI AS THUONG_DUOC_CHIA, -- Phần thưởng được chia            BL.TONGLUONG AS TONGLUONG_SAU -- Tổng lương sau khi cộng thưởng     FROM BANGLUONG BL     WHERE BL.MANV IN (         SELECT MANV         FROM CHAMCONG         WHERE TONGSOGIOLAM = @MAX_GIO         AND MONTH(CHAMCONG_DATE) = MONTH(GETDATE())         AND YEAR(CHAMCONG_DATE) = YEAR(GETDATE())     ); END; </pre>
Test	<pre> INSERT INTO BANGLUONG (MANV, BANGLUONG_DATE, LUONGCOBAN, TONGLUONG) VALUES     ('NV0051', '2024-10-01', 5000000, 7000000),     ('NV0092', '2024-10-01', 6000000, 8500000),     ('NV0093', '2024-10-01', 5500000, 6000000),     ('NV0074', '2024-10-01', 5500000, 5500000);  EXEC SP_THUONG_HIEUQUA; </pre>

### 3.3.9 MODULE 9

<b>Tên module</b>	Tự động cập nhật tiền thưởng/phạt
<b>Mô tả</b>	Tự động cập nhật tiền thưởng/phạt dựa trên tongsogiolam trong bảng CHAMCONG biết rằng tiền lương là 20.000/giờ làm và nếu làm đủ 100 giờ thì tiền thưởng sẽ = 20% tổng số giờ làm, nếu làm không đủ 100 giờ thì tiền phạt sẽ = 20% tổng số giờ làm
<b>Phân tích</b>	<p>Bảng: CHAMCONG</p> <p>Loại: AFTER INSERT</p> <p>Sự kiện: INSERT</p> <p>Process:</p> <ol style="list-style-type: none"> <li>Trigger này chạy khi có dữ liệu mới được thêm vào bảng CHAMCONG.</li> <li>Kiểm tra mã nhân viên và tổng số giờ làm việc trong bảng tạm inserted</li> <li>Lấy mốc 100 giờ làm làm mốc thưởng/phạt, nếu đủ 100 giờ làm thì lấy TONGSOGIOLAM nhân với LUONGCOBAN nhân với 0.2 (20%) rồi cập nhật dữ liệu vào cột THUONG nếu không đủ 100 giờ làm thì lấy TONGSOGIOLAM nhân với LUONGCOBAN nhân với 0.2 (20%) rồi cập nhật dữ liệu vào cột PHAT</li> </ol>

Code	<pre> CREATE TRIGGER TR_TinhThuongPhat9 ON CHAMCONG AFTER INSERT, UPDATE AS BEGIN     DECLARE @MANV VARCHAR(10);     DECLARE @TONGSOGIOLAM INT;     DECLARE @LUONGCOBAN DECIMAL(18, 2) = 20000; -- Lương cơ bản 20.000/giờ     DECLARE @THUONG DECIMAL(18, 2) = 0;     DECLARE @PHAT DECIMAL(18, 2) = 0;      -- Lấy mã nhân viên và tổng số giờ làm việc từ bản ghi mới thêm vào     SELECT @MANV = MANV, @TONGSOGIOLAM = TONGSOGIOLAM     FROM inserted;      -- Nếu tổng số giờ làm &gt;= 100 thì thưởng 20%     IF @TONGSOGIOLAM &gt;= 100     BEGIN         SET @THUONG = (@TONGSOGIOLAM * @LUONGCOBAN) * 0.2; -- 20% tiền thưởng         SET @PHAT = 0;     END     -- Nếu tổng số giờ làm &lt; 100 thì phạt 20%     ELSE     BEGIN         SET @PHAT = (@TONGSOGIOLAM * @LUONGCOBAN) * 0.2; -- 20% tiền phạt         SET @THUONG = 0;     END      -- Cập nhật cột THUONG và PHAT trong bảng CHAMCONG     UPDATE CHAMCONG     SET THUONG = @THUONG, PHAT = @PHAT     WHERE MANV = @MANV AND CHAMCONG_DATE = (SELECT MAX(CHAMCONG_DATE) FROM CHAMCONG WHERE MANV = @MANV); END; </pre>
Test	<pre> --test: INSERT INTO CHAMCONG (MANV, CHAMCONG_DATE, TONGSOGIOLAM) VALUES ('NV1001', '2024-10-20', 105);  UPDATE CHAMCONG SET TONGSOGIOLAM = 105 WHERE MANV = 'NV0001' AND CHAMCONG_DATE = '2024-10-19';  SELECT * FROM CHAMCONG WHERE MANV = 'NV1001'; </pre>

### 3.3.10 MODULE 10

Tên modul e	Xóa thông tin nhân viên khi ngày làm việc gần nhất của nhân viên đó là 10 năm trở lại đây.
Mô tả	Nếu ngày làm việc của nhân viên đó gần nhất là 10 năm trở lại thì hệ thống sẽ xóa thông tin của nhân viên. Mục đích là để giảm tải dung lượng cho hệ thống.
Phân tích	<p>Input: MANV</p> <p>Output: Thông tin nhân viên làm việc trên 10 năm bị xóa</p> <p>Process:</p> <p>1.Lấy ngày làm việc gần nhất của nhân viên khi nhân viên đăng ký ca làm trong bảng LICHLAM</p>

	2.Nếu ngày làm việc gần nhất là cách đây hơn 10 năm → Tiến hành xóa tất cả thông tin của nhân viên đó trong hệ thống.
Code	<pre> CREATE OR ALTER PROCEDURE SP_DeleteEmployee     @MANV VARCHAR(10)     AS     BEGIN         DECLARE @LastWorkDate DATE;          SELECT @LastWorkDate = MAX(NGAY_LV)         FROM LICHLAM L join CALAM C ON L.MACA=C.MACA         WHERE MANV = @MANV;          IF @LastWorkDate &lt;= DATEADD(YEAR, -10, GETDATE())         BEGIN             DELETE FROM CHAMCONG WHERE MANV = @MANV;             DELETE FROM BANGLUONG WHERE MANV = @MANV;             DELETE FROM LICHLAM WHERE MANV = @MANV;             DELETE FROM TAIKHOAN WHERE MANV = @MANV;             DELETE FROM NHANVIEN WHERE MANV = @MANV;         END     END; </pre>
Test	<pre> INSERT INTO CALAM (MACA, NGAY_LV, BUOI_LAM, GIO_BATDAU, GIO_KETTHUC, SOGIOLAM) VALUES ('CA2006', '2005-10-22', 'Sáng', '08:00:00', '12:00:00', 4);  INSERT INTO LICHLAM (MANV, MACA, TRANGTHAI) VALUES ('NV2005', 'CA2006', '2')  declare @a varchar(10) execute SP_DeleteEmployee 'NV2005' </pre>

## CHƯƠNG 4. R5

### Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu

#### 4.1.Tổng quan

Cơ chế bảo mật cơ sở dữ liệu SQL Server là giải pháp cho phép các quản trị viên cơ sở dữ liệu thiết lập quyền hạn cho người dùng hoặc nhóm người dùng khai thác cơ sở dữ liệu. Bảo mật cơ sở dữ liệu SQL Server giúp phân quyền người dùng trên cơ sở dữ liệu.

Đối với phương thức chứng thực, SQL server cung cấp hai phương thức là Windows Authentication và SQL Server Authentication.

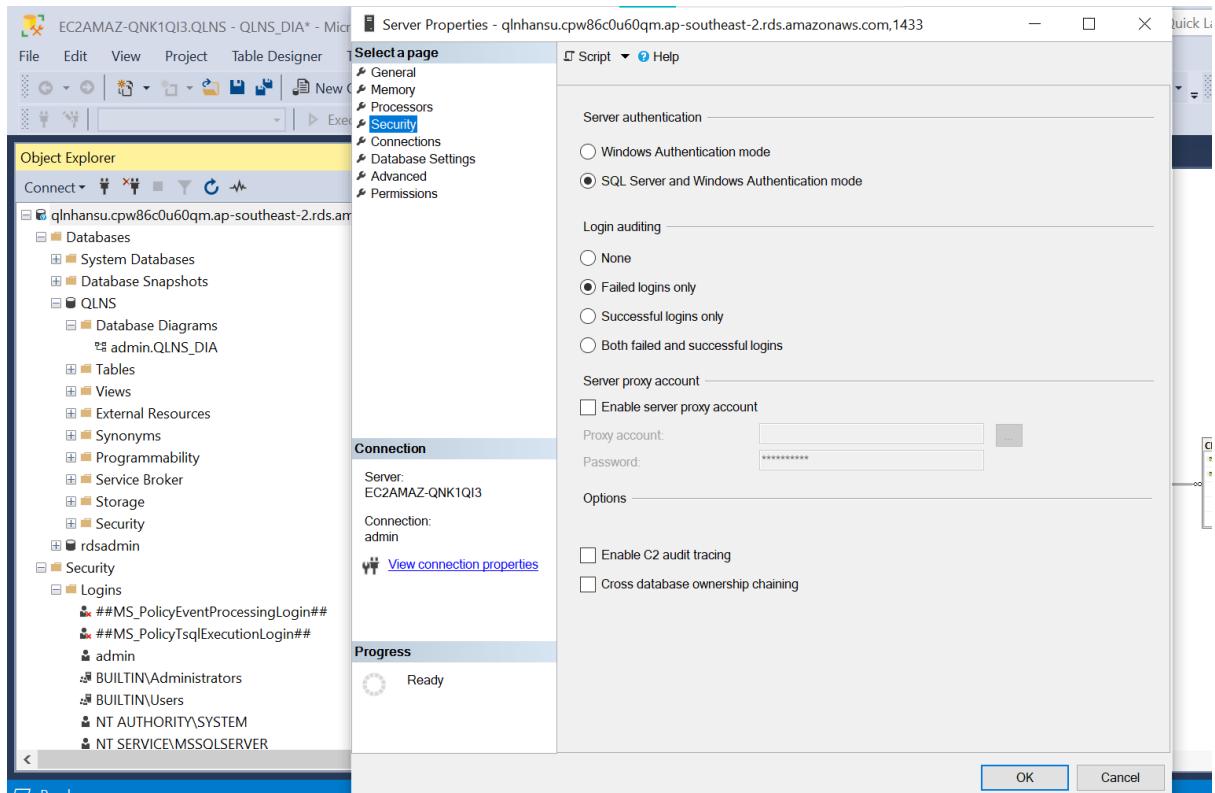
- Windows Authentication: sử dụng user login Windows trên chính máy tính cài đặt SQL Server. Windows User bao gồm cả local user và domain user nếu máy tính cài đặt SQL server có join domain.
- SQL Server Authentication: Là tài khoản được tạo và quản lý bởi chính SQL Server.

Nhóm lựa chọn bảo mật theo SQL Server Authentication.

## 4.2.Tạo tài khoản cho người quản lý

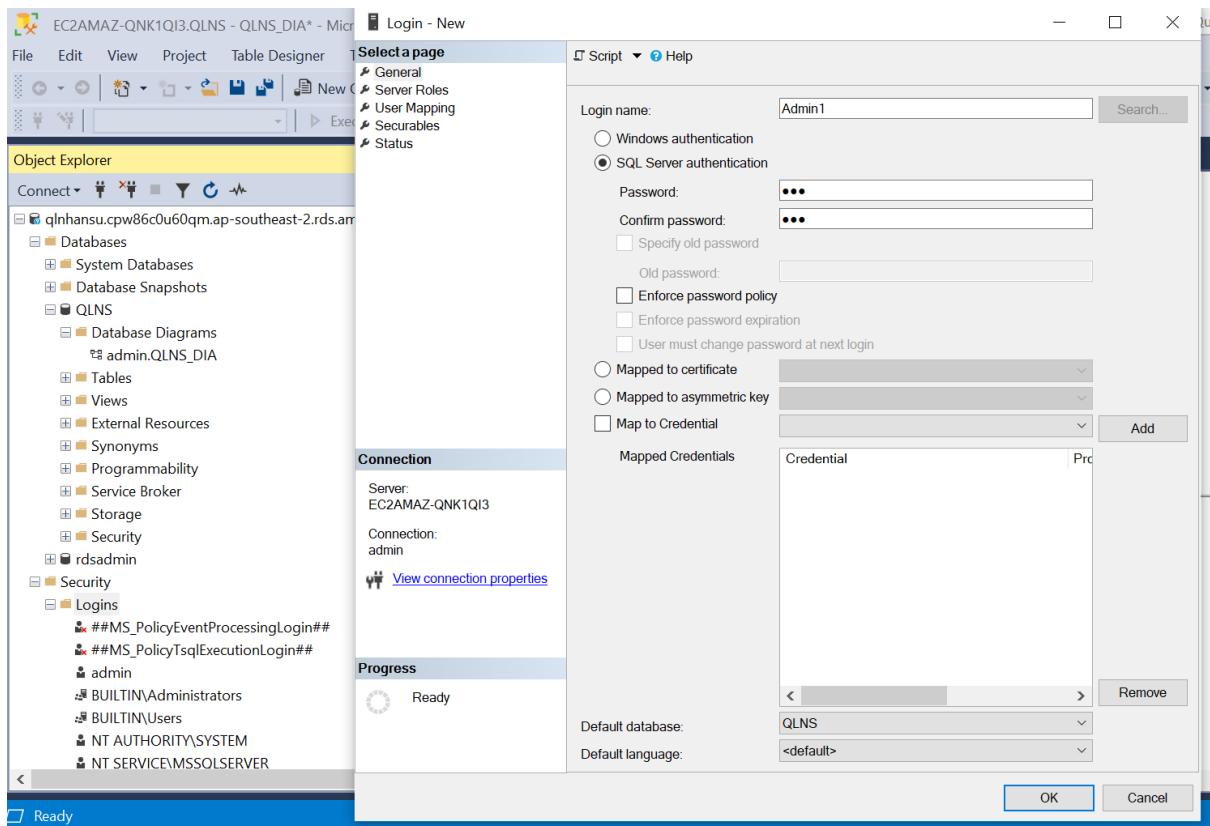
Trong quá trình lập tài khoản, bên cạnh việc cho phép truy cập vào database, người quản trị sẽ cấp các quyền cần thiết để tài khoản thao tác trong database đó.

### Bước 1: Thiết lập chế độ chứng thực Windows và SQL Server Authentication mode

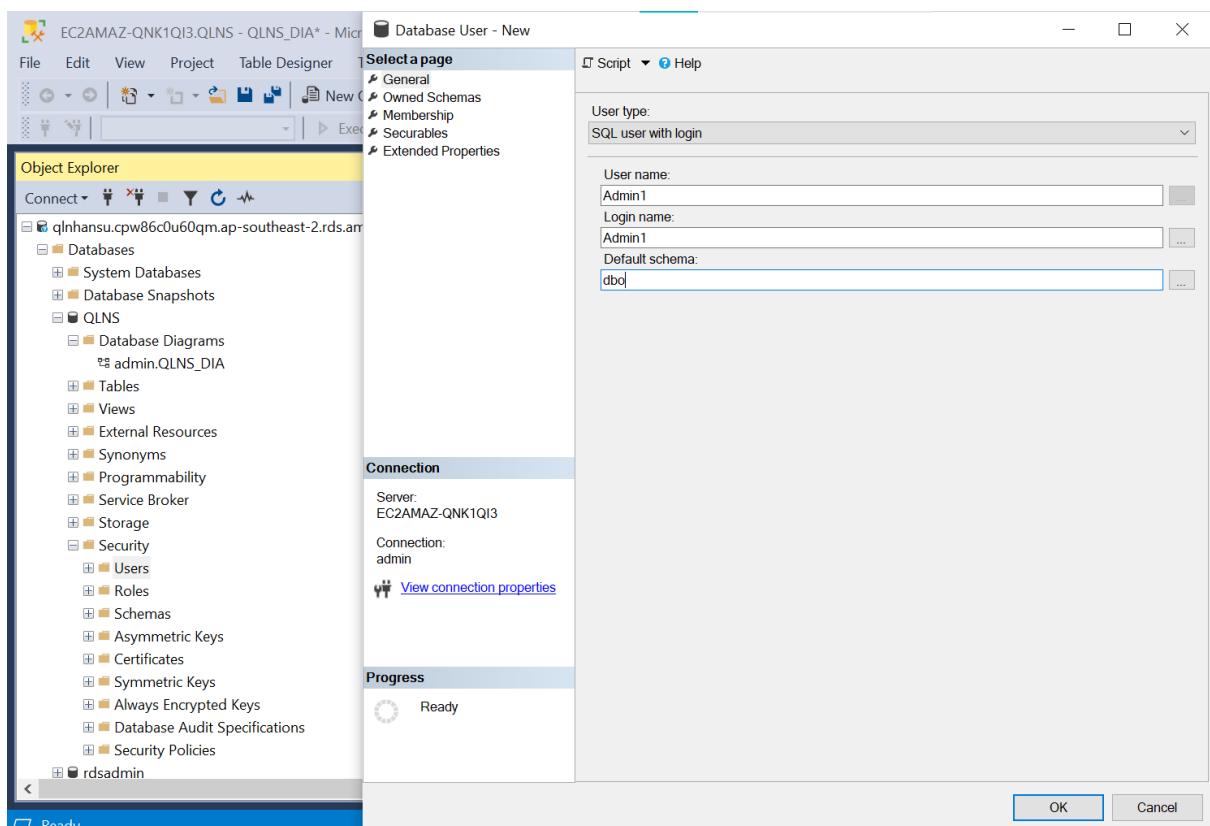


Hình 4.2.Thiết lập chế độ chứng thực

### Bước 2: Tạo tài khoản Login có tên tài khoản là Admin1

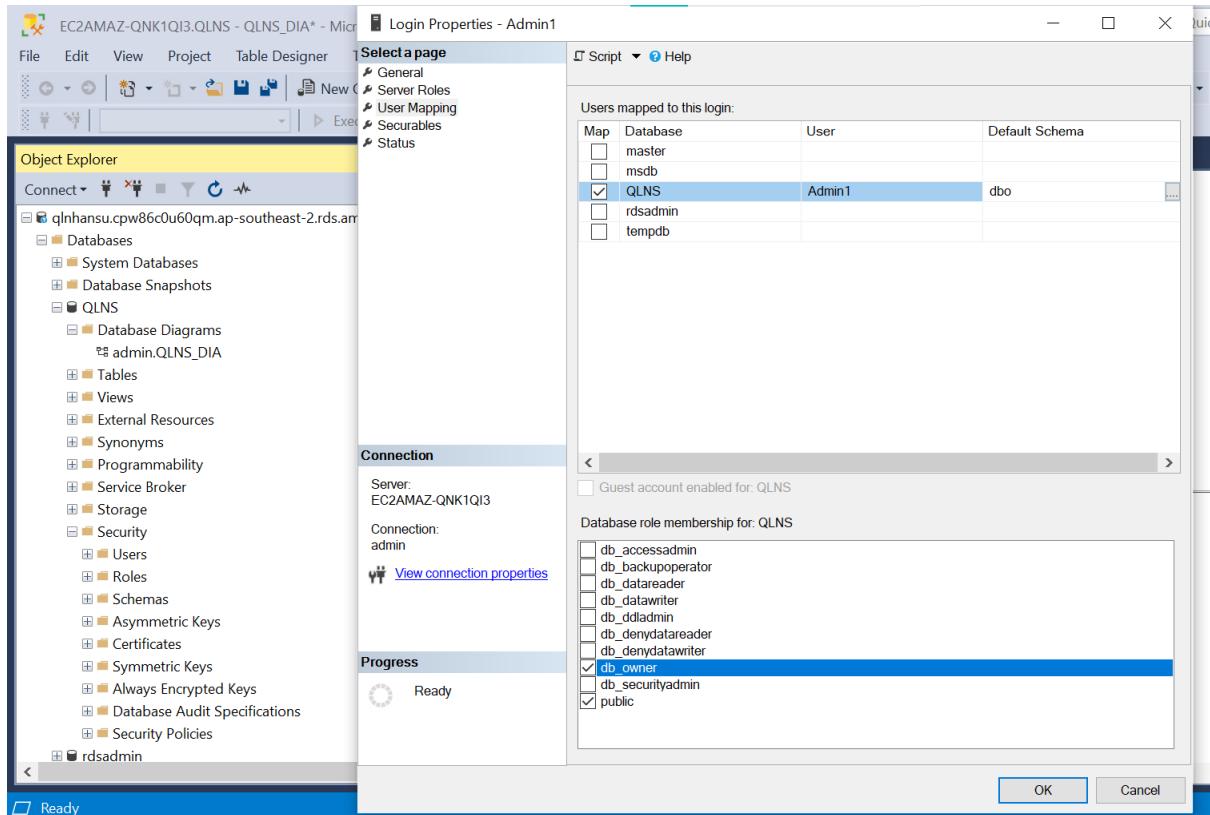


Hình 4.1.2 - Tạo tài khoản Login có tên tài khoản là Admin1



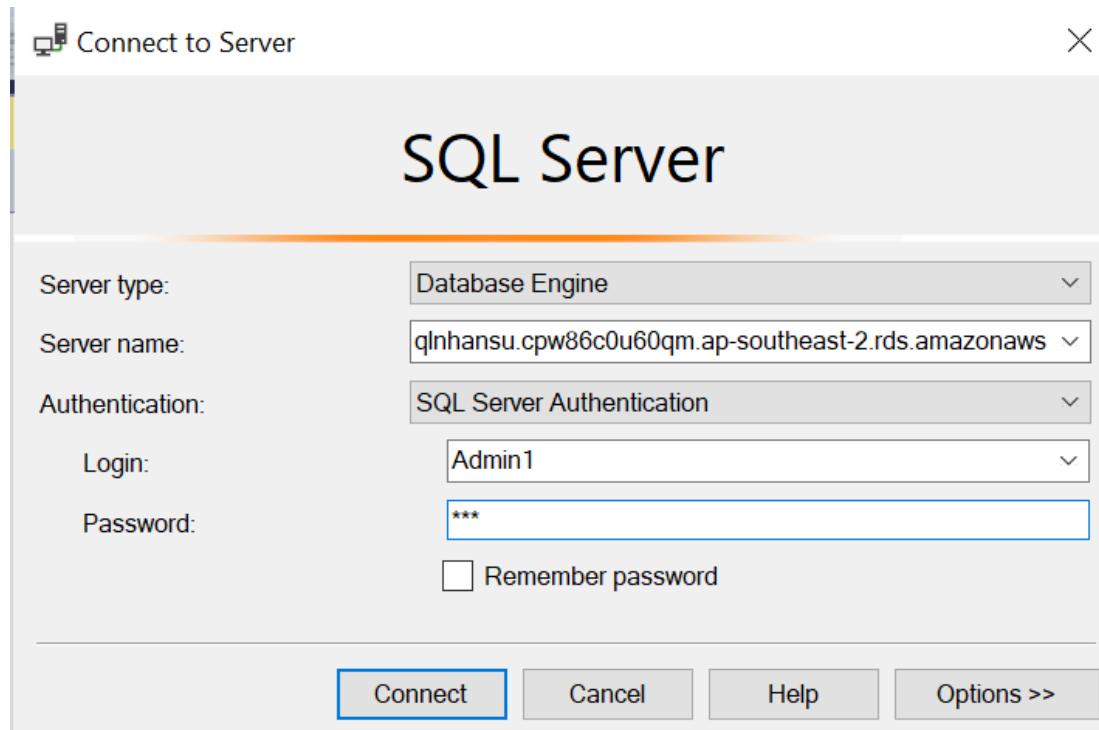
Hình 4.1.3. Tạo tài khoản user cho Admin

Đối với tài khoản Admin: Admin sẽ được cấp toàn quyền sử dụng database ( thiết lập quyền trên Server, thiết lập quyền trên cơ sở dữ liệu và thiết lập trạng thái).

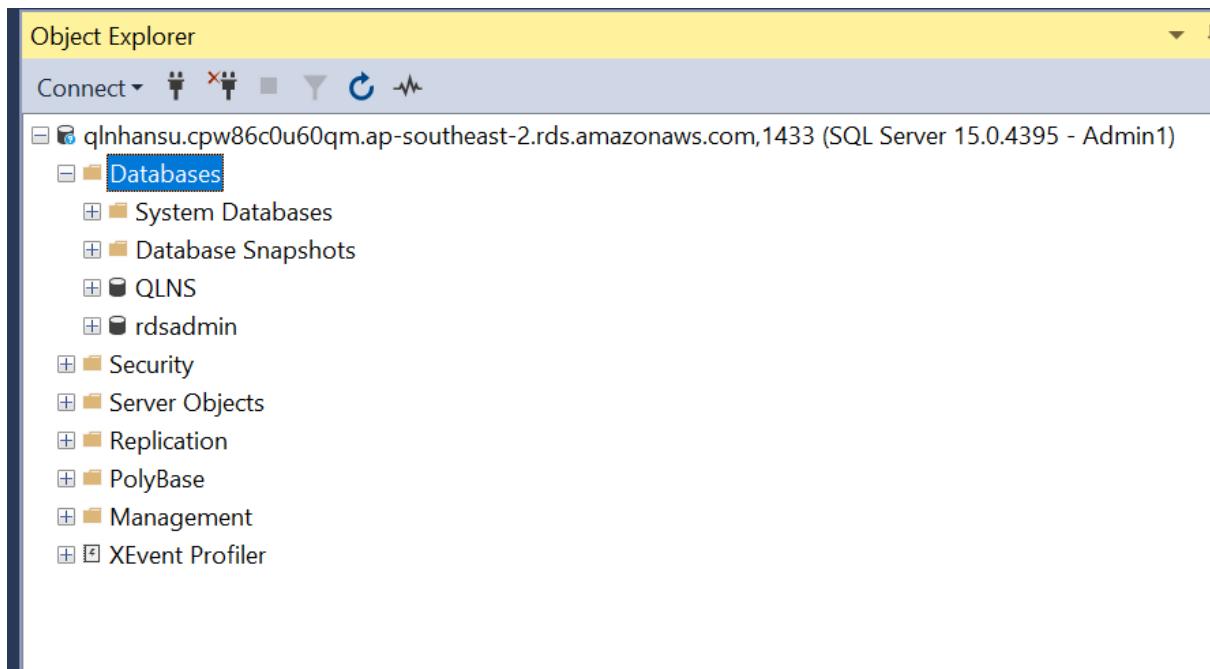


Hình 4.1.4 Phân quyền cho tài khoản Admin1

Sau khi phân quyền, thực hiện thao tác đăng nhập tài khoản được tạo



Hình 4.1.5 Login server bằng tài khoản Admin1



Đăng nhập thành công với tài khoản Admin1

A screenshot of the SQL Server Management Studio interface. The Object Explorer shows the same connection as the previous screenshot. In the center, there is a query editor window titled 'SQLQuery1.sql'. It contains the following SQL code:

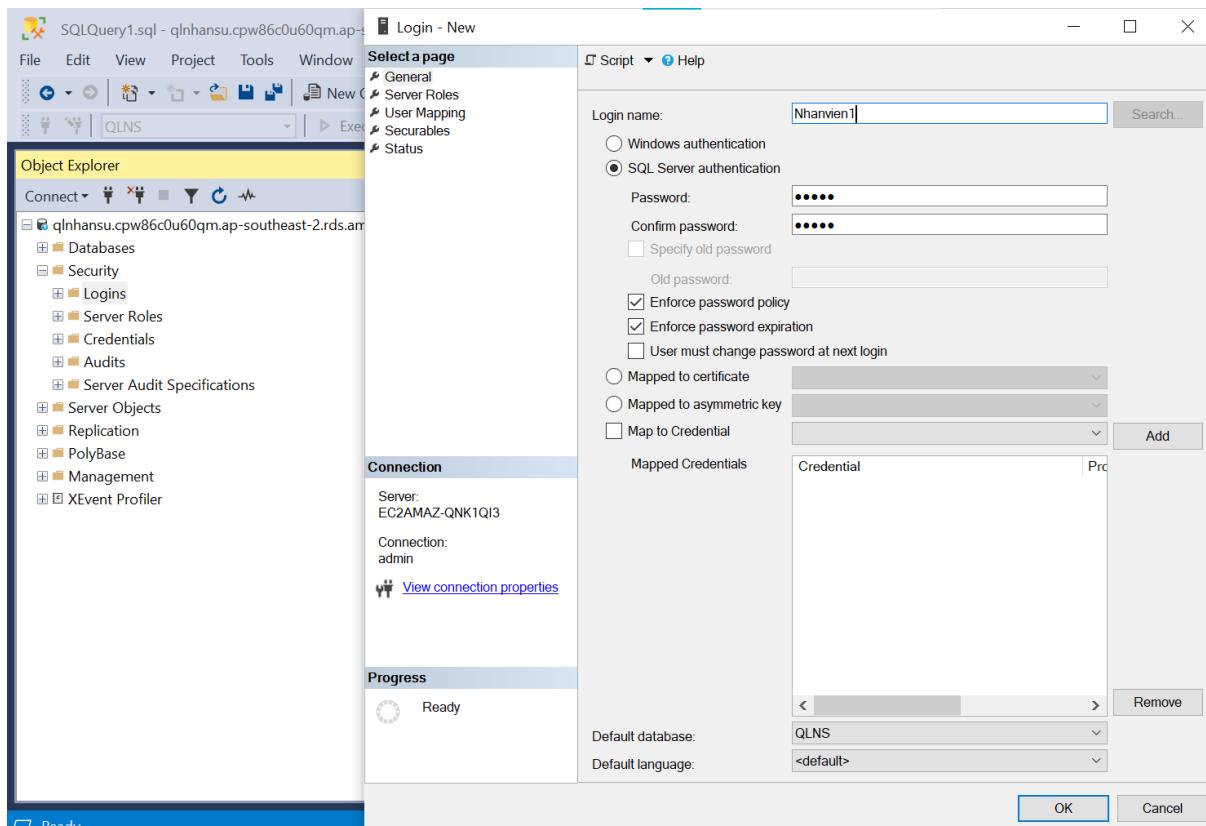
```
1 UPDATE CALAM
2 SET GIO_KETTHUC='13:00:00'
3 WHERE MACA='CA0123';
```

The status bar at the bottom right indicates 'Completion time: 2024-11-21T17:13:33.4205924+07:00'.

Hình 4.1.6- Quản lý có toàn quyền thao tác như thêm, sửa, xóa, ... với database

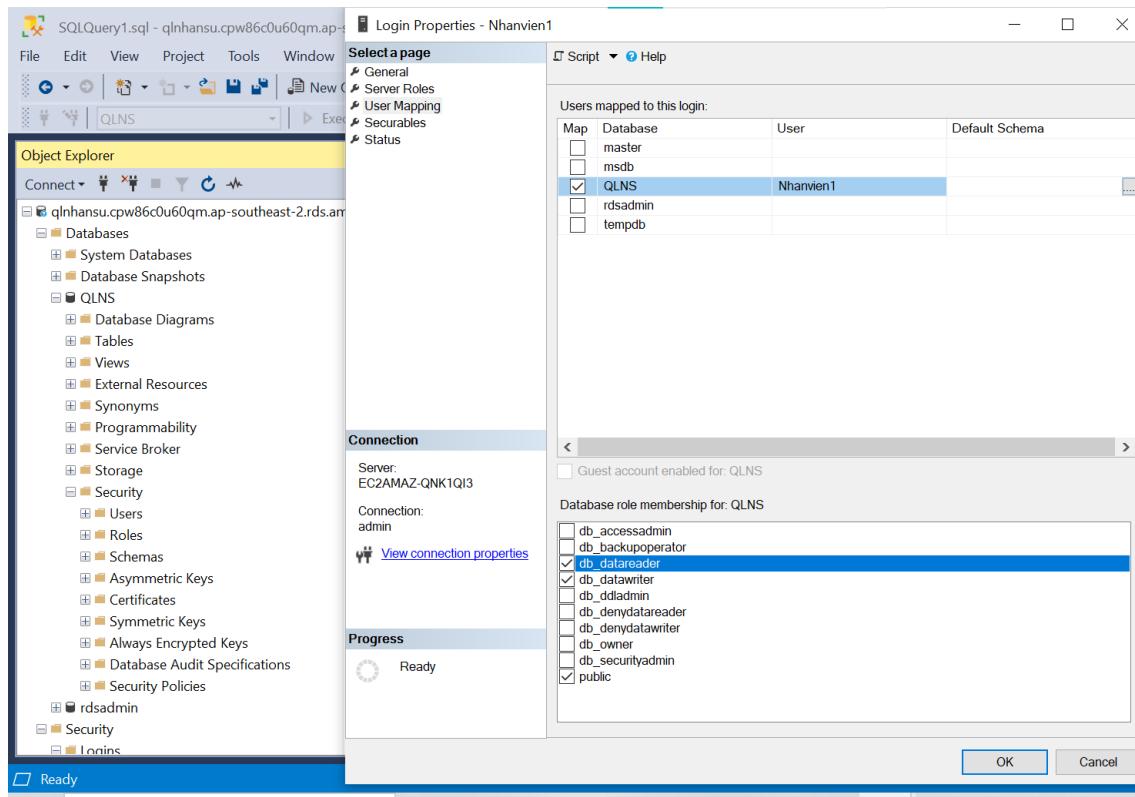
### 4.3. Tạo tài khoản cho nhân viên

Bước 1: Tạo tài khoản cho Nhân Viên có tên Nhanvien1

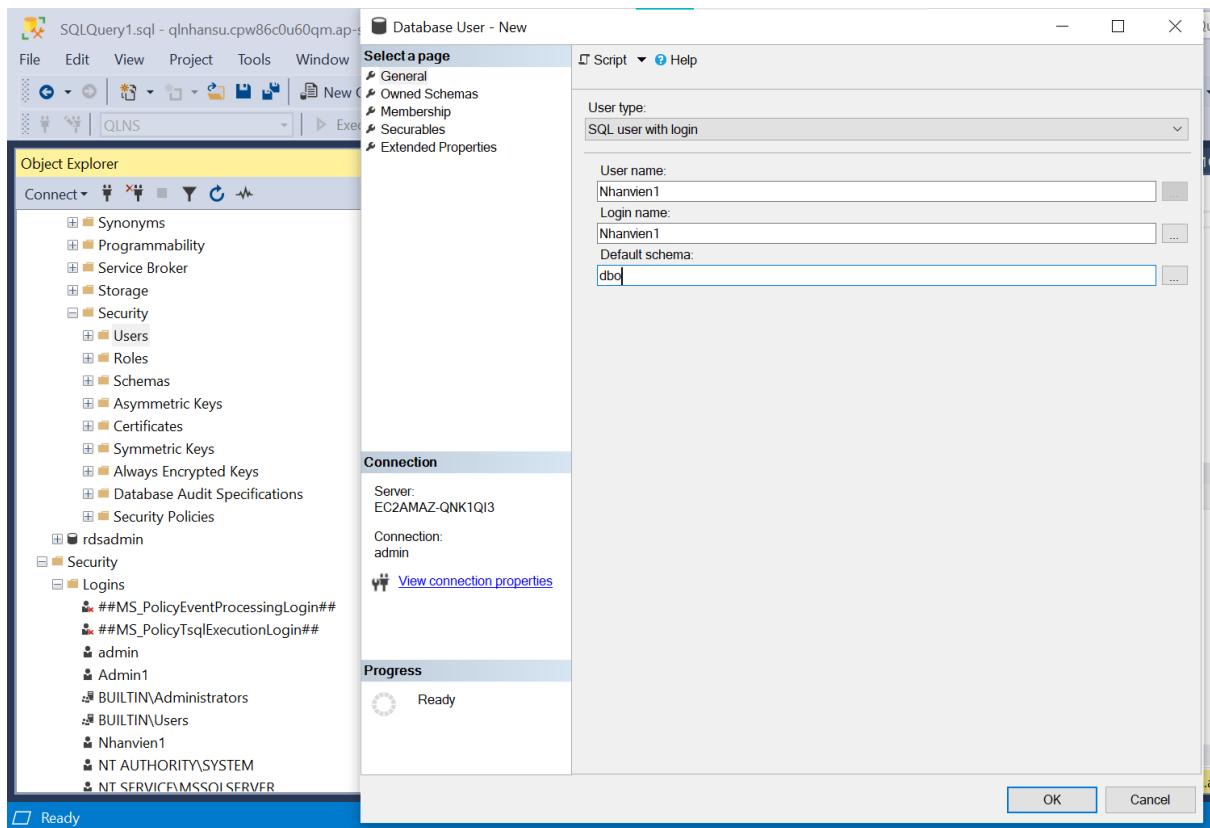


Hình 6.2.1 - Tạo tài khoản cho Nhân Viên có tên Nhanvien1

### Bước 2: Thực hiện phân quyền cho tài khoản Nhanvien1.

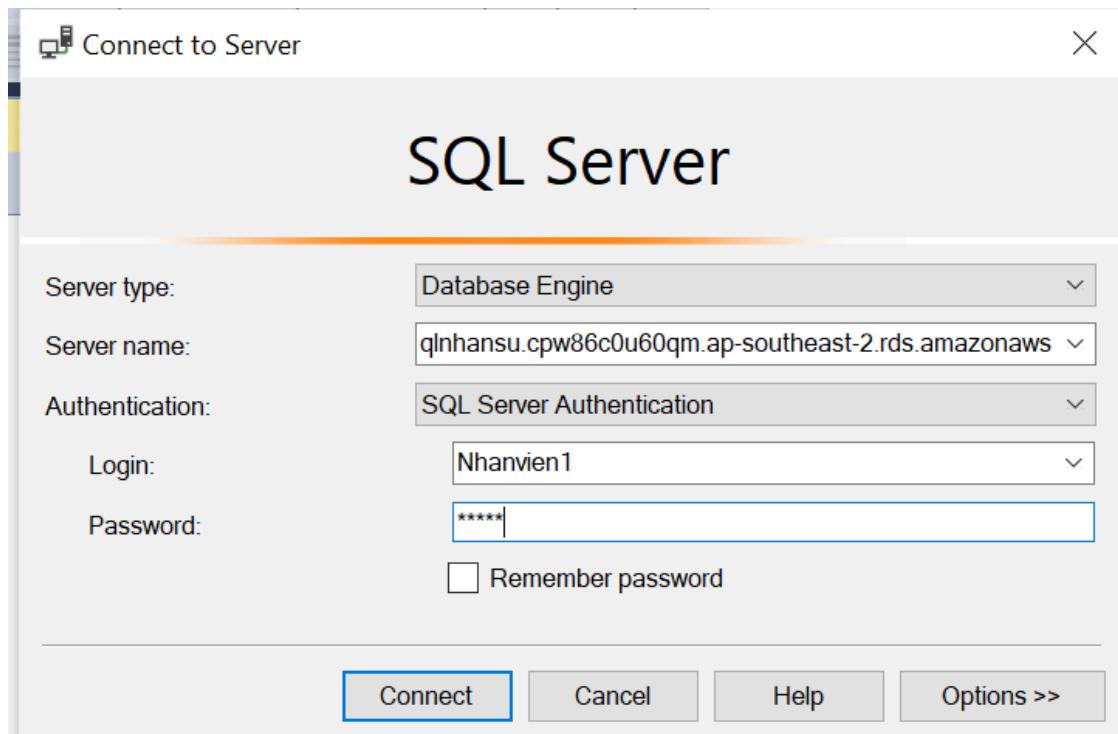


Hình 4.2.2- Thực hiện phân quyền cho tài khoản Nhanvien1.



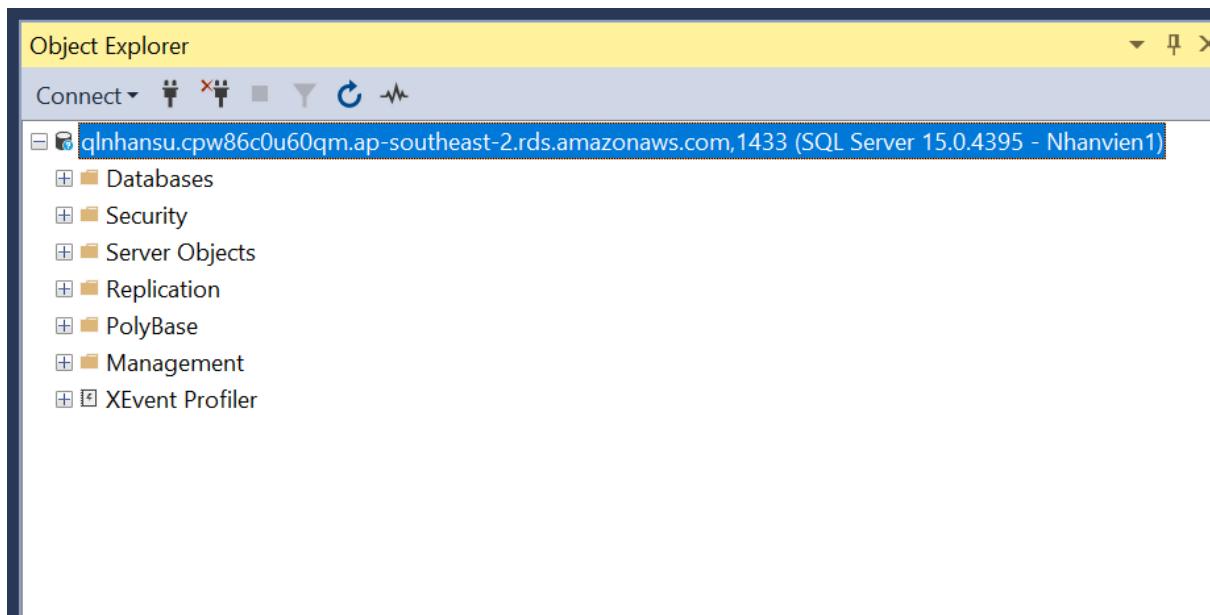
Hình 4.2.3. Tạo tài khoản user cho Nhanvien1.

**Bước 3: Đăng nhập tài khoản Nhanvien vừa được tạo.**



Hình 4.2.4. Đăng nhập Login cho tài khoản Nhanvien1.

Đăng nhập thành công



Đối với tài khoản Nhanvien1 chỉ có quyền xem, thêm dữ liệu vào bảng và không có quyền tạo thêm bảng trong database.

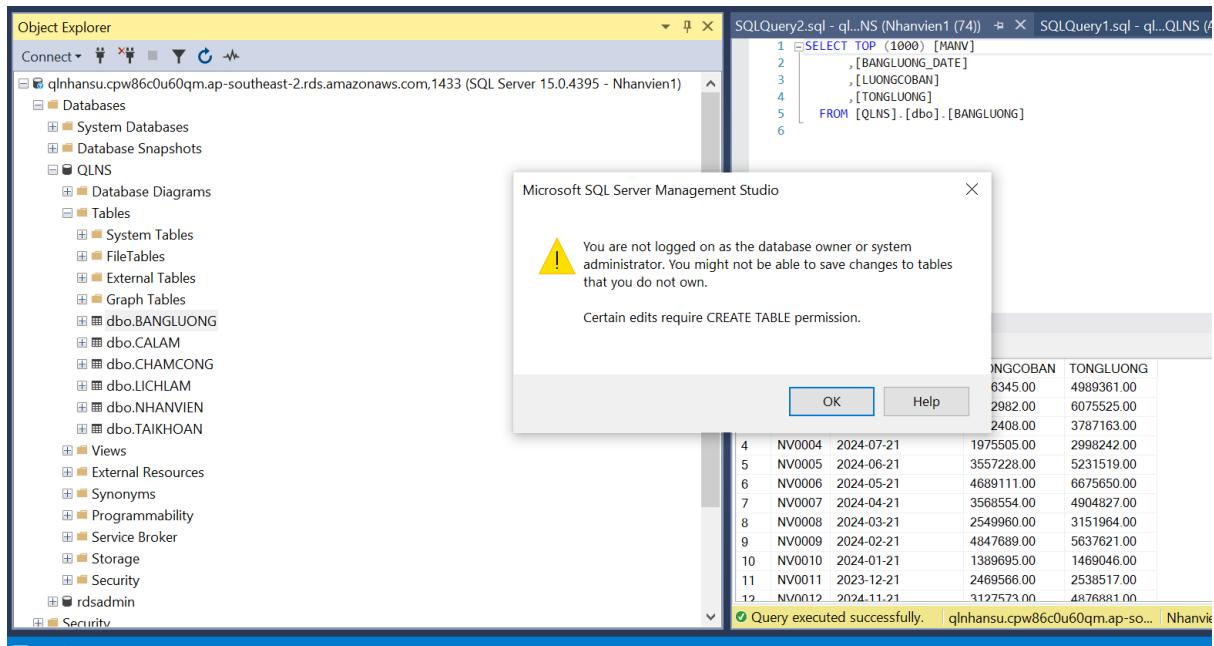
```

1   SELECT TOP (1000) [MANV]
2       ,[BANGLUONG_DATE]
3       ,[LUONGCOBAN]
4       ,[TONGLUONG]
5   FROM [QLNS].[dbo].[BANGLUONG]
6

```

	MANV	BANGLUONG_DATE	LUONGCOBAN	TONGLUONG
1	NV0001	2024-10-21	4436345.00	4989361.00
2	NV0002	2024-09-21	4442982.00	6075525.00
3	NV0003	2024-08-21	2662408.00	3787163.00
4	NV0004	2024-07-21	1975505.00	2998242.00
5	NV0005	2024-06-21	3557228.00	5231519.00
6	NV0006	2024-05-21	4689111.00	6675650.00
7	NV0007	2024-04-21	3568554.00	4904827.00
8	NV0008	2024-03-21	2549960.00	3151964.00
9	NV0009	2024-02-21	4847689.00	5637621.00
10	NV0010	2024-01-21	1389695.00	1469046.00
11	NV0011	2023-12-21	2469566.00	2538517.00
12	NV0012	2024-11-21	3127573.00	4876981.00

Hình 4.2.5 - Quyền của Nhanvien1.



Hình 4.2.6 - Quyền của Nhanvien1.

#### 4.4. Mã hóa

Nhóm sử dụng phương pháp mã hoá 2 chiều bằng mã hoá 2 chiều EncryptByPassPhrase trong SQL server. Thực hiện mã hoá thông tin 2 cột SDT, CCCD trong bảng NHANVIEN và MATKHAU trong bảng TAIKHOAN bằng phương pháp mã hoá mật khẩu.

căn cước công dân và mật khẩu thành viên nhằm bảo mật dữ liệu cho quán, tránh tình trạng có người xâm nhập vào sửa số liệu, đánh cắp thông tin cản trở cho công việc kinh doanh của quán.

**a. Mã hóa thông tin 3 cột SDT và CCCD trong bảng NHANVIEN và MATKHAU trong bảng TAIKHOAN bằng phương pháp mã hoá mật khẩu**

```

1 --tao cot luu du lieu ma hoa
2 ALTER TABLE NHANVIEN ADD
3 [ENCRYPTED SDT] VARBINARY (MAX)
4 GO
5
6 ALTER TABLE NHANVIEN ADD
7 [ENCRYPTED CCCD] VARBINARY (MAX)
8 GO
9
10 ALTER TABLE TAIKHOAN ADD
11 [ENCRYPTED MATKHAU] VARBINARY (MAX)
12 GO
13

```

**b. Cập nhật dữ liệu được mã hóa vào cột vừa tạo**

```
14 -- cap nhap du lieu da duoc ma hoa
15 UPDATE NHANVIEN
16 [SET [ENCRYPTED SDT]= ENCRYPTBYPASSPHRASE('SDT',CONVERT(VARCHAR(100),SDT))
17 GO
18
19 UPDATE NHANVIEN
20 [SET [ENCRYPTED CCCD]= ENCRYPTBYPASSPHRASE('CCCD',CONVERT(VARCHAR(100),CCCD))
21 GO
22
23 UPDATE TAIKHOAN
24 [SET [ENCRYPTED MATKHAU]= ENCRYPTBYPASSPHRASE('MATKHAU',CONVERT(VARCHAR(100),MATKHAU))
25 GO
26
```

6 % ▶

Messages

```
(1201 rows affected)
(1201 rows affected)
(1200 rows affected)
Completion time: 2024-11-19T17:33:23.6672856+07:00
```

**c. Xóa cột chưa được mã hóa**

```
-- xoa cot chua ma hoa
ALTER TABLE NHANVIEN DROP CONSTRAINT CK_SDT;
ALTER TABLE NHANVIEN
DROP CONSTRAINT UQ__NHANVIEN__A955A0AAC2A54662;
ALTER TABLE NHANVIEN DROP COLUMN SDT
ALTER TABLE NHANVIEN DROP COLUMN CCCD
ALTER TABLE TAIKHOAN DROP COLUMN MATKHAU
```

**d. Thực hiện truy vấn**

34 | SELECT \* FROM NHANVIEN

86 % ▾

	CHUCVU	ENCRYPTED SDT	ENCRYPTED CCCD
1	Biên Phủ, Hải Châu, Đà Nẵng	Pha chế	0x0200000004AE2E344658F7A0594BB1532B350EE3B61592FF...
2	ch Khiêm, Cẩm Lê, Đà Nẵng	Pha chế	0x02000000D230944752414A25F70CEF1D01D552DCBCAAE50...
3	ach Đẳng, Sơn Trà, Đà Nẵng	Phuc vu	0x020000006B089182DD33507BB46236F5D4E940F86C561972...
4	Bạch Đằng, Cẩm Lê, Đà Nẵng	Phuc vu	0x020000009E7F06041A4E54265F46B1C9FED4FE899AB2FBB...
5	Vương, Hải Châu, Đà Nẵng	Phuc vu	0x02000000FC0D62F653CA9680CE0FBE6390F262ED09A27B4...
6	Bạch Đằng, Ngũ Hành Sơn, Đà Nẵ...	Phuc vu	0x020000001688A2CB41EAAB81388E680B43258015B4142C9D...
7	Hoài Thanh, Hòa Vang, Đà Nẵng	Pha chế	0x0200000019AB08D3CC7A74E83228BEED9E3249C8958016B3...
8	Vương, Sơn Trà, Đà Nẵng	Phuc vu	0x0200000016307802AF72BB5ABA24F6BB2EA58A516B3E17E...
9	h Khiêm, Sơn Trà, Đà Nẵng	Pha chế	0x020000004E5F3920597D11798675C245C14382D117E4CF29...
10	Diệu, Sơn Trà, Đà Nẵng	Phuc vu	0x02000000D2C280047E84FBE4A2E7DD2B4A8D3497CEA3B2...
11	ch Khiêm, Ngũ Hành Sơn, Đà Nẵng	Pha chế	0x0200000081BEA5607629963400CBA44BD4E3872410E25EA0...
12	Quyền, Hải Châu, Đà Nẵng	Pha chế	0x020000006980B2BF16FAB03C26C9BC0960153959F5ABA83F...
13	ễn Văn Thoại, Liên Chiểu, Đà Nẵng	Phuc vu	0x02000000D4666157F18EC911EE39F557F070D42E080792D9...
14	Biên Phủ, Sơn Trà, Đà Nẵng	Phuc vu	0x0200000018A79E27807CF08EBC4BE51DAD3CD467699F849...
15	ễn Văn Thoại, Ngũ Hành Sơn, Đà ...	Pha chế	0x02000000BF033B2592F79A73C34A37C69A81299FB2B99F61...
16	g Diệu, Cẩm Lê, Đà Nẵng	Phuc vu	0x02000000400959FE9A807BF6E08B92085C6A278070A900BA...

36 | SELECT \* FROM TAIKHOAN

86 % ▾

	MANV	ENCRYPTED MATKHAU
1	NV0001	0x02000000E2A037356A497636CF18735BD2243E48B142655A9...
2	NV0002	0x020000004C79C3829E569F5E6A152423396BB62DDD97C58F...
3	NV0003	0x02000000E3A67A2DB7A0B7260453535C5822E779D18162216...
4	NV0004	0x02000000ECAED063101D0E91E030E06DFD203FFACE7DEC...
5	NV0005	0x020000006310B6A14D8001C5EF042ECEBE246DA8E82D51D...
6	NV0006	0x0200000054F2E1CDB9D038C0EDE27913B2BF95863E4078FB...
7	NV0007	0x020000009BEC68DAF0892D05E2870305F7005D69425E51D2...
8	NV0008	0x02000000D654FE4034530B797C4ECCFC1B13320338AE8122...
9	NV0009	0x020000004D68F297EA14269DA5EF2FFF60B48A39A9D0E1BF...
10	NV0010	0x020000001326640C75F4EE1083156BED1633748BF5F325478...
11	NV0011	0x02000000E4F5478615784450820AC709A512100BFED315295...
12	NV0012	0x0200000026FE54CED212CCD30B82E63426888017A3315DD6...
13	NV0013	0x020000006FBF9C5B3FFDE27C3091E1199AF61EDA60602817...
14	NV0014	0x02000000D7C4276861771D6C44A30C3E5F2C787397ECB4AB...
15	NV0015	0x02000000D9B1AC91D828769F6D808BA25B489527E724C416...
16	NV0016	0x0200000080AC93CDA45CE72031D13806573E0AADAAA52E...
17	NV0017	0x02000000151289DC5A7E24D03D4CE83C8037089475F2F92B...

e. Giải mã dữ liệu thông qua Decryptbypassphrase

```
-- giai ma du lieu
SELECT MANV,
CONVERT(CHAR, CONVERT(VARCHAR(100), DECRYPTBYPASSPHRASE('SDT', [ENCRYPTED SDT]))) AS SDT FROM NHANVIEN
SELECT MANV,
CONVERT(CHAR, CONVERT(VARCHAR(100), DECRYPTBYPASSPHRASE('CCCD', [ENCRYPTED CCCD]))) AS CCCD FROM NHANVIEN
SELECT MANV,
CONVERT(CHAR, CONVERT(VARCHAR(100), DECRYPTBYPASSPHRASE('MATKHAU', [ENCRYPTED MATKHAU]))) AS MATKHAU FROM TAIKHOAN
```

Kết quả sau khi giải mã

	MANV	SDT
1	NV0001	0752525163
2	NV0002	0118058845
3	NV0003	0976039807
4	NV0004	0368798479
5	NV0005	0804251904

	MANV	CCCD
4	NV0004	100868378881
5	NV0005	101324208620

	MANV	MATKHAU
3	NV0003	MK6696
4	NV0004	MK6806
5	NV0005	MK9993
6	NV0006	MK2229
7	NV0007	MK6776
8	NV0008	MK0993
9	NV0009	MK3800
10	NV0010	MK6448

## CHƯƠNG 5. R6

### Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu.

#### 5.1.Tổng quan

Backup dữ liệu là một bước thiết yếu để bảo vệ thông tin quan trọng trước các rủi ro như lỗi hệ thống, tấn công mạng, hoặc sai sót người dùng. Quy trình này đảm bảo khả năng khôi phục nhanh chóng, duy trì hoạt động liên tục, bảo toàn phiên bản lịch sử, và đáp ứng yêu cầu pháp lý. Backup cũng giúp doanh nghiệp đối phó với ransomware mà không cần trả tiền chuộc, bảo vệ danh tiếng và sự ổn định.

Nhóm chúng em lựa chọn phương pháp backup đầy đủ (full backup) để sao lưu toàn bộ dữ liệu, bao gồm các bảng và thông tin quan trọng. Phương pháp này đảm bảo dữ liệu được khôi phục dễ dàng, nhất quán và an toàn, dù tiêu tốn nhiều dung lượng và thời gian. Full backup

mang lại sự tin cậy cao, giúp doanh nghiệp sẵn sàng đối phó với các tình huống như lỗi phần cứng hay sự cố mạng.

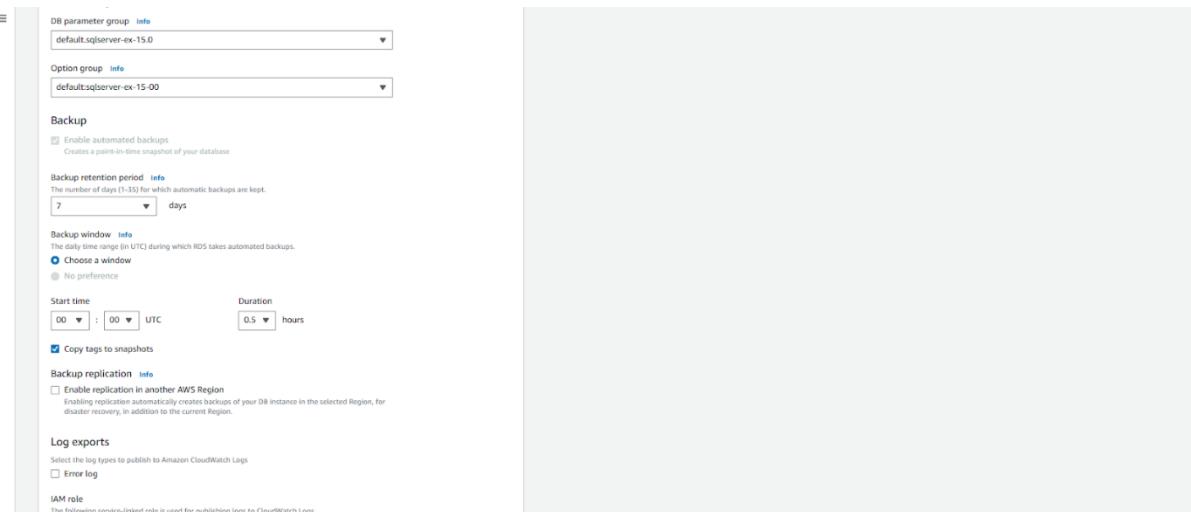
Để tối ưu hóa việc bảo vệ dữ liệu, nhóm đề xuất hai phương án thực hiện backup trên SQL Server như sau:

### 5.1.1. Backup dữ liệu trên cloud

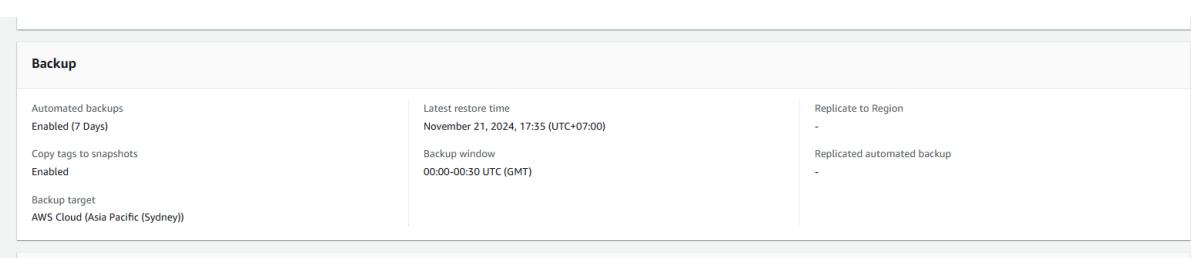
- Đây là phương án chính trong kế hoạch ban đầu.
- Dữ liệu sẽ được sao lưu định kỳ và lưu trữ trên nền tảng cloud để đảm bảo tính ổn định, bảo mật, và dễ dàng phục hồi trong trường hợp xảy ra sự cố.

Các thao tác thực hiện Backup dữ liệu trên cloud như sau:

- Sao lưu tự động: AWS RDS sẽ lưu trữ các bản sao lưu tự động của cơ sở dữ liệu trong 7 ngày.
- Cửa sổ sao lưu (Backup window): Thời gian cửa sổ sao lưu tự động từ 00:00 đến 00:30



The screenshot shows the AWS RDS console configuration for a database. It includes fields for DB parameter group (default\_sqldriver-ex-15.0) and Option group (default\_sqldriver-ex-15-00). Under the 'Backup' section, 'Enable automated backups' is checked. The 'Backup retention period' is set to 7 days. The 'Backup window' is set from 00:00 to 00:30 UTC. The 'Duration' is set to 0.5 hours. The 'Copy tags to snapshots' checkbox is checked. There are also sections for 'Backup replication' (unchecked) and 'Log exports' (unchecked). An 'IAM role' section at the bottom indicates a 'RDS service-linked role'.

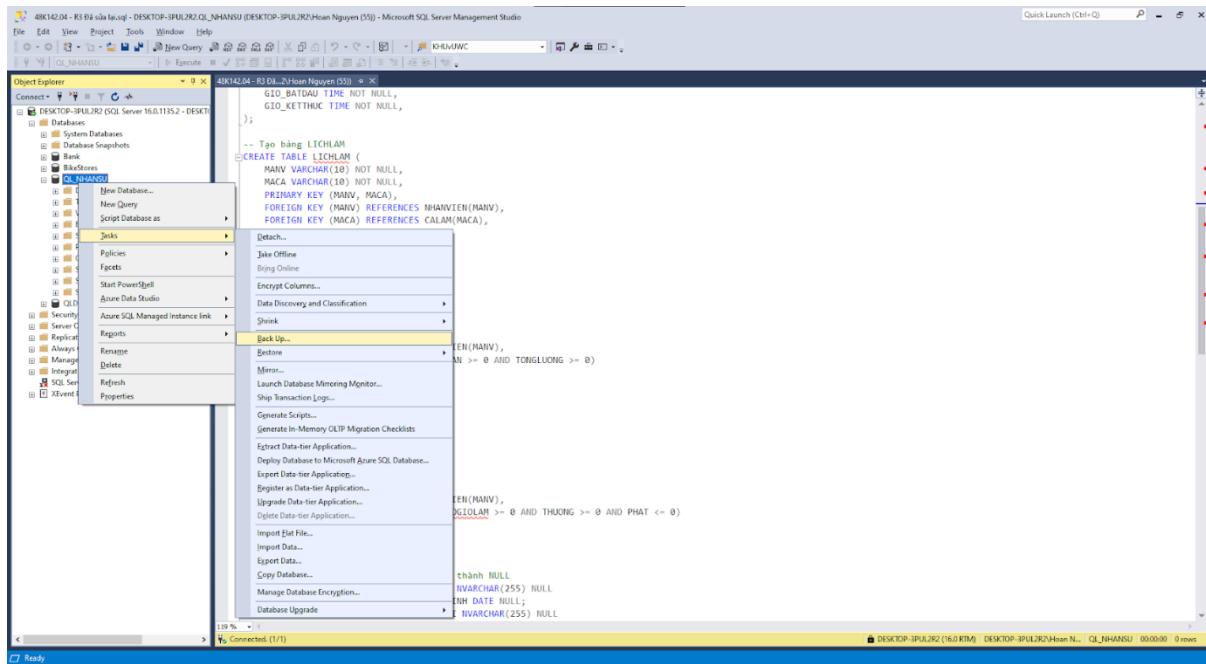
The screenshot shows the status of automated backups. It displays the latest restore time as November 21, 2024, 17:35 (UTC+07:00). The backup window is listed as 00:00-00:30 UTC (GMT). The replication status is shown as 'Replicated automated backup'.

### 5.1.2. Backup bổ sung trên server local

- Nhằm đáp ứng yêu cầu kiểm tra trực tiếp trong quá trình thực hiện, nhóm sẽ thực hiện thêm một bản backup trên server local.
- Bản backup này giúp dễ dàng kiểm tra và xử lý khi cần thiết, đặc biệt trong trường hợp không thể kết nối đến cloud.

Các thao tác thực hiện backup thực hiện trên SQL Server Management Studio như sau:

**Bước 1: Lựa chọn thao tác backup trên cơ sở dữ liệu cần sao lưu**

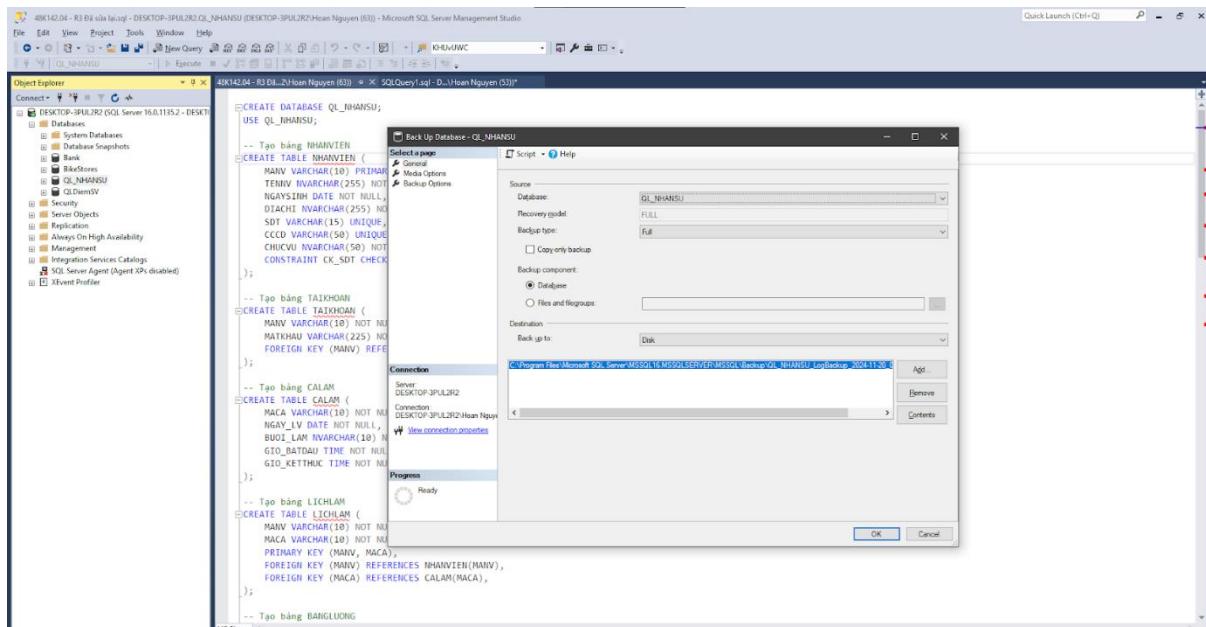


Lựa chọn thao tác backup trên cơ sở dữ liệu cần sao lưu

## Bước 2: Thiết lập các tùy chọn backup

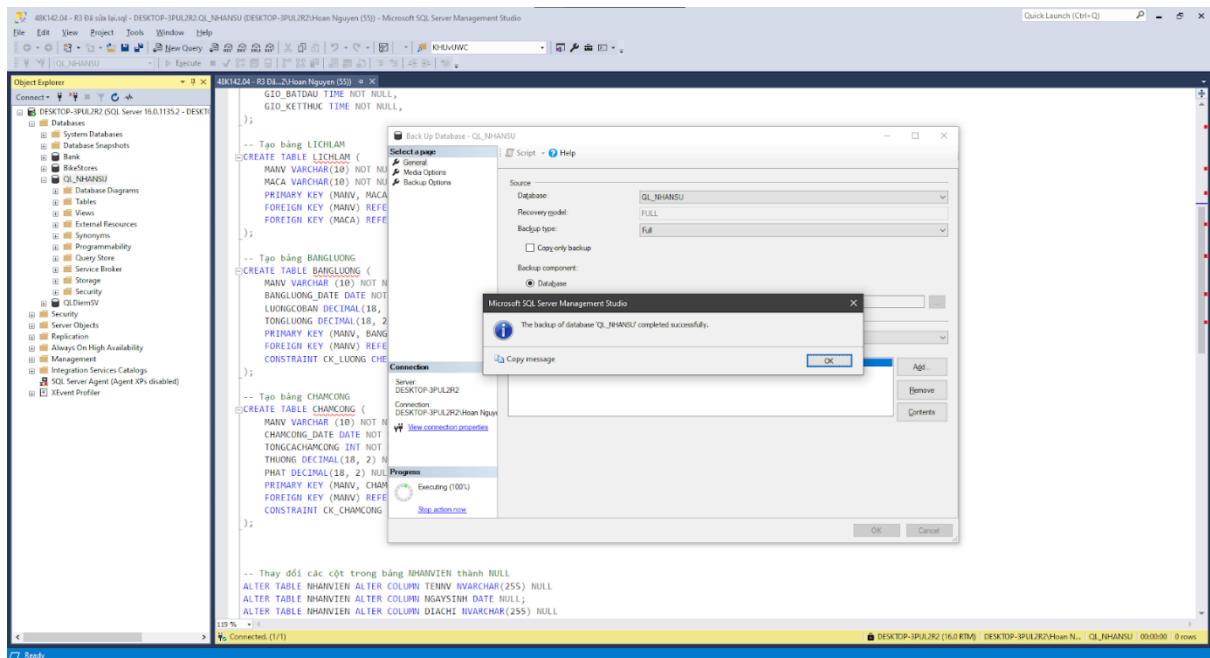
Ở bước này em Chọn cơ sở dữ liệu cần sao lưu là **QL\_NHANSU**, với **loại sao lưu là Full Backup** để đảm bảo toàn bộ dữ liệu được lưu trữ đầy đủ và an toàn. Bản sao lưu sẽ được lưu tại đường dẫn: *C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup*

Điều này đảm bảo dữ liệu luôn có sẵn để phục hồi khi cần thiết, đồng thời vị trí lưu trữ được chọn là thư mục mặc định của SQL Server để dễ dàng quản lý và truy xuất.



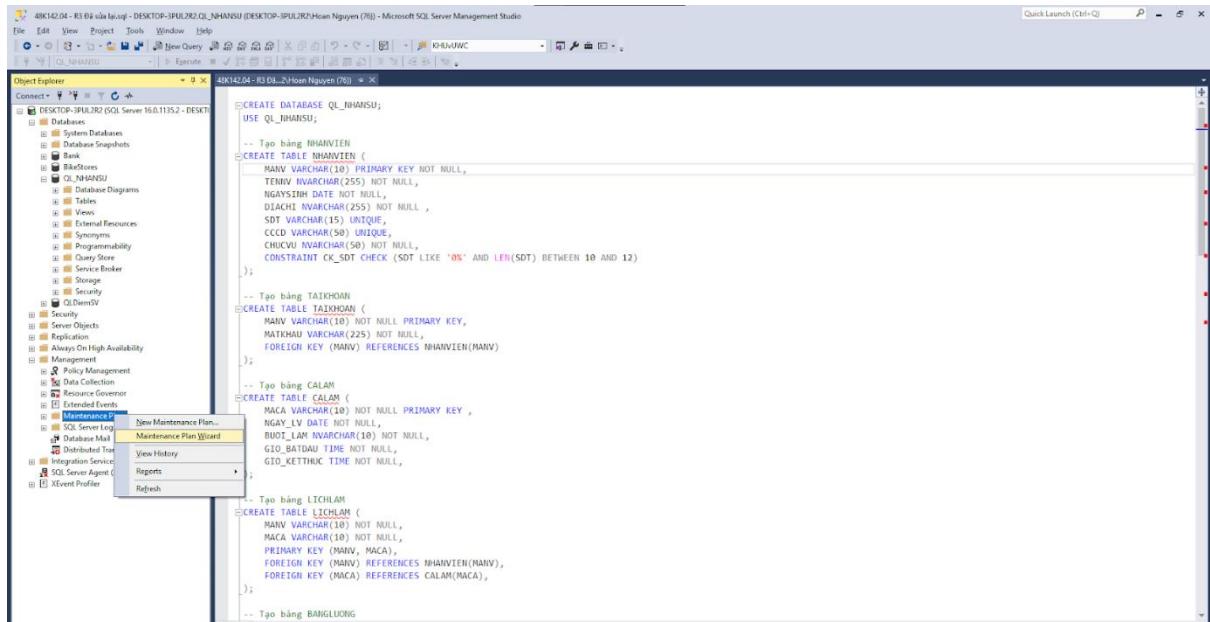
Thiết lập các tùy chọn backup

### Bước 3: Backup thành công

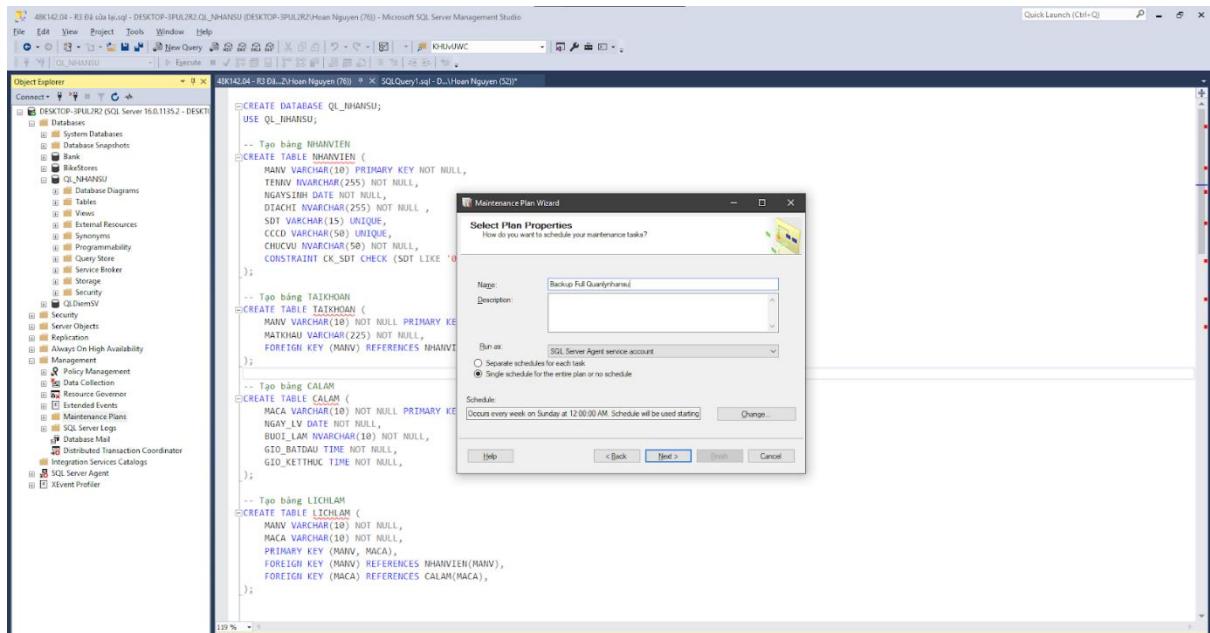


*Backup thành công*

### Bước 4: Thực hiện tự động backup dữ liệu

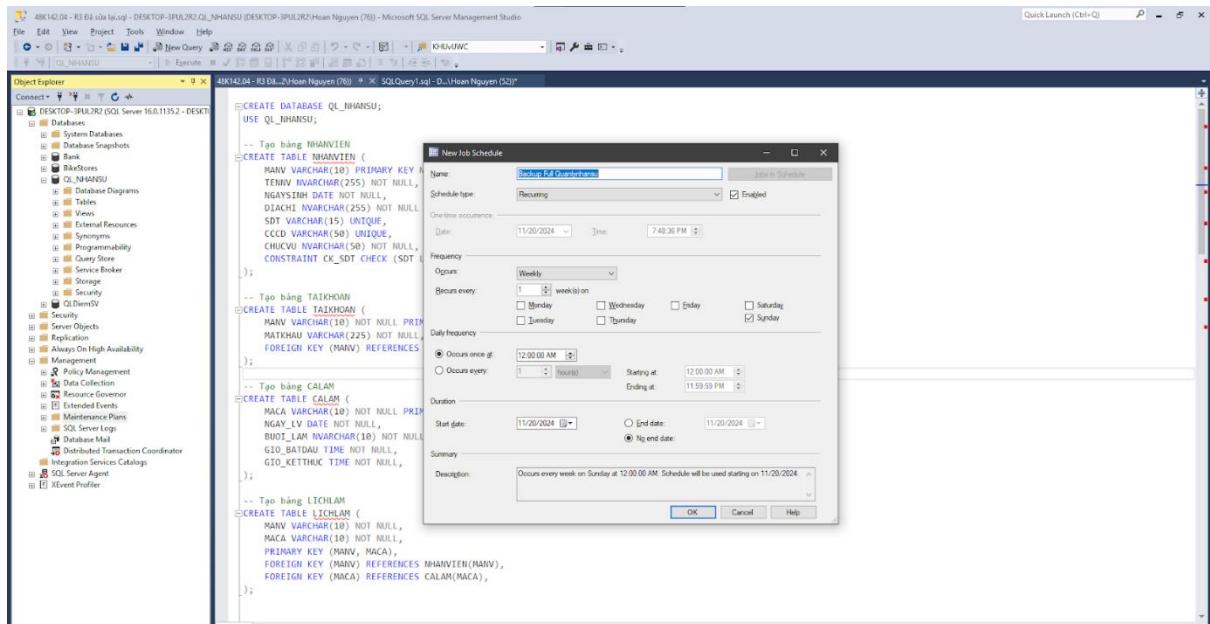


*Chọn mục Maintenance Plans trong phần Management.*



*Chọn tùy chọn Single schedule để thiết lập thời gian backup*

Với phương án lựa chọn **full backup**, em đã quyết định thực hiện sao lưu **hàng tuần vào chủ nhật hàng tuần**, vào lúc **00:00 sáng**, khi lưu lượng truy cập hệ thống ở mức thấp nhất. Thời điểm này giúp đảm bảo việc sao lưu diễn ra suôn sẻ mà không ảnh hưởng đáng kể đến hoạt động của người dùng và hiệu suất tổng thể của hệ thống.



*Chọn lịch chạy tự động*

```

CREATE DATABASE QL_NHANSU;
USE QL_NHANSU;

-- Tạo bảng NHANVIEEN
CREATE TABLE NHANVIEEN (
    MANV VARCHAR(10) PRIMARY KEY NOT NULL,
    NGAYSINH DATE NOT NULL,
    DIACHI NVARCHAR(255) NOT NULL ,
    SDT VARCHAR(15) UNIQUE,
    CCCD VARCHAR(50) UNIQUE,
    CHUCVU NVARCHAR(50) NOT NULL,
    CONSTRAINT CK_SDTCHECK CHECK (SDT LIKE '0')
);

-- Tạo bảng TAIKHOAN (
CREATE TABLE TAIKHOAN (
    MANV VARCHAR(10) NOT NULL PRIMARY KEY,
    MATKHAU VARCHAR(225) NOT NULL,
    FOREIGN KEY (MANV) REFERENCES NHANVIEEN
);

-- Tạo bảng CALAM (
CREATE TABLE CALAM (
    MACA VARCHAR(10) NOT NULL PRIMARY KEY,
    NGAY_LV DATE NOT NULL,
    BUDO_LAM NVARCHAR(10) NOT NULL,
    GIO_BATDAU TIME NOT NULL,
    GIO_KETTHUC TIME NOT NULL,
);

-- Tạo bảng LICHLAM (
CREATE TABLE LICHLAM (
    MANV VARCHAR(10) NOT NULL,
    MACA VARCHAR(10) NOT NULL,
    PRIMARY KEY (MANV, MACA),
    FOREIGN KEY (MANV) REFERENCES NHANVIEEN(MANV),
    FOREIGN KEY (MACA) REFERENCES CALAM(MACA),
);

-- Tạo bảng BANGLUONG

```

*Chọn loại tự động backup*

General   Destination   Options

Backup type:  Full

Database(s):

Backup component:

Backup type:

- All databases
- System databases
- All user databases (excluding master, model, msdb, tempdb)
- These databases

These databases:

- model
- master
- QL\_NHANSU
- tempdb

Schedule: Not scheduled (On Demand)

*Chọn database cần thực hiện tự động backup*

```

CREATE DATABASE QL_NHANSU;
USE QL_NHANSU;

-- Tạo bảng NHANVIEN
CREATE TABLE NHANVIEN (
    MANV VARCHAR(10) PRIMARY KEY NOT NULL,
    NGAYSINH DATE NOT NULL,
    DIACHI NVARCHAR(255) NOT NULL ,
    SDT VARCHAR(15) UNIQUE,
    CCCD VARCHAR(50) UNIQUE,
    CHUCVU NVARCHAR(50) NOT NULL,
    CONSTRAINT CK_SDT CHECK (SDT LIKE '0')
);

-- Tạo bảng TAIKHOAN (
CREATE TABLE TAIKHOAN (
    MANV VARCHAR(10) NOT NULL PRIMARY KEY,
    MATKHAU VARCHAR(225) NOT NULL,
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV)
);

-- Tạo bảng CALAM (
CREATE TABLE CALAM (
    MACA VARCHAR(10) NOT NULL PRIMARY KEY,
    NGAY_LV_DATE NOT NULL,
    BUDI_LAM NVARCHAR(10) NOT NULL,
    GIO_BATDAU TIME NOT NULL,
    GIO_KETTHUC TIME NOT NULL,
);

```

*Tùy chỉnh các tùy chọn sao lưu*

```

CREATE DATABASE QL_NHANSU;
USE QL_NHANSU;

-- Tạo bảng NHANVIEN
CREATE TABLE NHANVIEN (
    MANV VARCHAR(10) PRIMARY KEY NOT NULL,
    NGAYSINH DATE NOT NULL,
    DIACHI NVARCHAR(255) NOT NULL ,
    SDT VARCHAR(15) UNIQUE,
    CCCD VARCHAR(50) UNIQUE,
    CHUCVU NVARCHAR(50) NOT NULL,
    CONSTRAINT CK_SDT CHECK (SDT LIKE '0')
);

-- Tạo bảng TAIKHOAN (
CREATE TABLE TAIKHOAN (
    MANV VARCHAR(10) NOT NULL PRIMARY KEY,
    MATKHAU VARCHAR(225) NOT NULL,
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV)
);

-- Tạo bảng CALAM (
CREATE TABLE CALAM (
    MACA VARCHAR(10) NOT NULL PRIMARY KEY,
    NGAY_LV_DATE NOT NULL,
    BUDI_LAM NVARCHAR(10) NOT NULL,
    GIO_BATDAU TIME NOT NULL,
    GIO_KETTHUC TIME NOT NULL,
);

```

*Chọn thiết bị lưu trữ sao lưu*

```

CREATE DATABASE QL_NHANSU;
USE QL_NHANSU;

-- Tạo bảng NHANVIEN
CREATE TABLE NHANVIEN (
    MANV VARCHAR(10) PRIMARY KEY NOT NULL,
    TENNV NVARCHAR(255) NOT NULL,
    NGAYSINH DATE NOT NULL,
    DIACHI NVARCHAR(255) NOT NULL ,
    SDT VARCHAR(15) UNIQUE ,
    CCCD VARCHAR(50) UNIQUE ,
    CHUCVU NVARCHAR(50) NOT NULL,
    CONSTRAINT CK_SDTCHECK CHECK (SDT LIKE '9[0-9]{9}' )
);

-- Tạo bảng TAIKHOM
CREATE TABLE TAIKHOM (
    MANV VARCHAR(10) NOT NULL PRIMARY KEY,
    MATKHAU VARCHAR(225) NOT NULL,
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV)
);

-- Tạo bảng CALAM
CREATE TABLE CALAM (
    MACA VARCHAR(10) NOT NULL PRIMARY KEY,
    NGAY_LV_DATE NOT NULL,
    BUDI_LAN NVARCHAR(10) NOT NULL,
    GIO_BATDAU TIME NOT NULL,
    GIO_KETTHUC TIME NOT NULL,
);
;

-- Tạo bảng LICHLAN
CREATE TABLE LICHLAN (
    MANV VARCHAR(10) NOT NULL,
    MACA VARCHAR(10) NOT NULL,
    PRIMARY KEY (MANV, MACA),
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
    FOREIGN KEY (MACA) REFERENCES CALAM(MACA),
);

-- Tạo bảng BANGLUONG

```

*Chọn finish để hoàn thành backup tự động*

Action	Status	Message
Creating maintenance plan 'Backup_Full_Qu...	Success	
Adding tasks to the maintenance plan	Success	
Adding scheduling options	Success	
Adding reporting options	Success	
Saving maintenance plan 'Backup_Full_Qu...	Success	

*Thiết lập Tự động backup dữ liệu thành công*

```

-- Tạo bảng LICHLAN
CREATE TABLE LICHLAN (
MANV VARCHAR(10) NOT NULL,
MACA VARCHAR(10) NOT NULL,
PRIMARY KEY (MANV, MACA),
FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
FOREIGN KEY (MACA) REFERENCES CALAM(MACA),
);
;

-- Tạo bảng BANGLUONG
CREATE TABLE BANGLUONG (
MANV VARCHAR (10) NOT NULL,
BANGLUONG_DATE DATE NOT NULL,
LUONGCOBAN DECIMAL(18, 2) NOT NULL,
TONGLUONG DECIMAL(18, 2) NOT NULL,
PRIMARY KEY (MANV, BANGLUONG_DATE),
FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
CONSTRAINT CK_LUONG CHECK (LUONGCOBAN >= 0 AND TONGLUONG >= 0)
);

-- Tạo bảng CHANGCONG
CREATE TABLE CHANGCONG (
MANV VARCHAR (10) NOT NULL,
CHANGCONG_DATE DATE NOT NULL,
TONGCACHANGCONG INT NOT NULL,
THUONG DECIMAL(18, 2) NULL,
PHAT DECIMAL(18, 2) NULL,
PRIMARY KEY (MANV, CHANGCONG_DATE),
FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
CONSTRAINT CK_CHANGCONG CHECK (TONGSOGIOLAM >= 0 AND THUONG >= 0 AND PHAT <= 0)
);

-- Thay đổi các cột trong bảng NHANVIEN thành NULL
ALTER TABLE NHANVIEN ALTER COLUMN TEINV NVARCHAR(255) NULL
;

```

Trong mục *SQL Server Agent* chọn *Start Job at Step* để bắt đầu chạy backup tự động đã tạo

```

-- Tạo bảng LICHLAN
CREATE TABLE LICHLAN (
MANV VARCHAR(10) NOT NULL,
MACA VARCHAR(10) NOT NULL,
PRIMARY KEY (MANV, MACA),
FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
FOREIGN KEY (MACA) REFERENCES CALAM(MACA),
);
;

-- Tạo bảng BANGLUONG
CREATE TABLE BANGLUONG (
MANV VARCHAR (10) NOT NULL,
BANGLUONG_DATE DATE NOT NULL,
LUONGCOBAN DECIMAL(18, 2) NOT NULL,
TONGLUONG DECIMAL(18, 2) NOT NULL,
PRIMARY KEY (MANV, BANGLUONG_DATE),
FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
CONSTRAINT CK_LUONG CHECK (LUONGCOBAN >= 0 AND TONGLUONG >= 0)
);

-- Tạo bảng CHANGCONG
CREATE TABLE CHANGCONG (
MANV VARCHAR (10) NOT NULL,
CHANGCONG_DATE DATE NOT NULL,
TONGCACHANGCONG INT NOT NULL,
THUONG DECIMAL(18, 2) NULL,
PHAT DECIMAL(18, 2) NULL,
PRIMARY KEY (MANV, CHANGCONG_DATE),
FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),
CONSTRAINT CK_CHANGCONG CHECK (TONGSOGIOLAM >= 0 AND THUONG >= 0 AND PHAT <= 0)
);

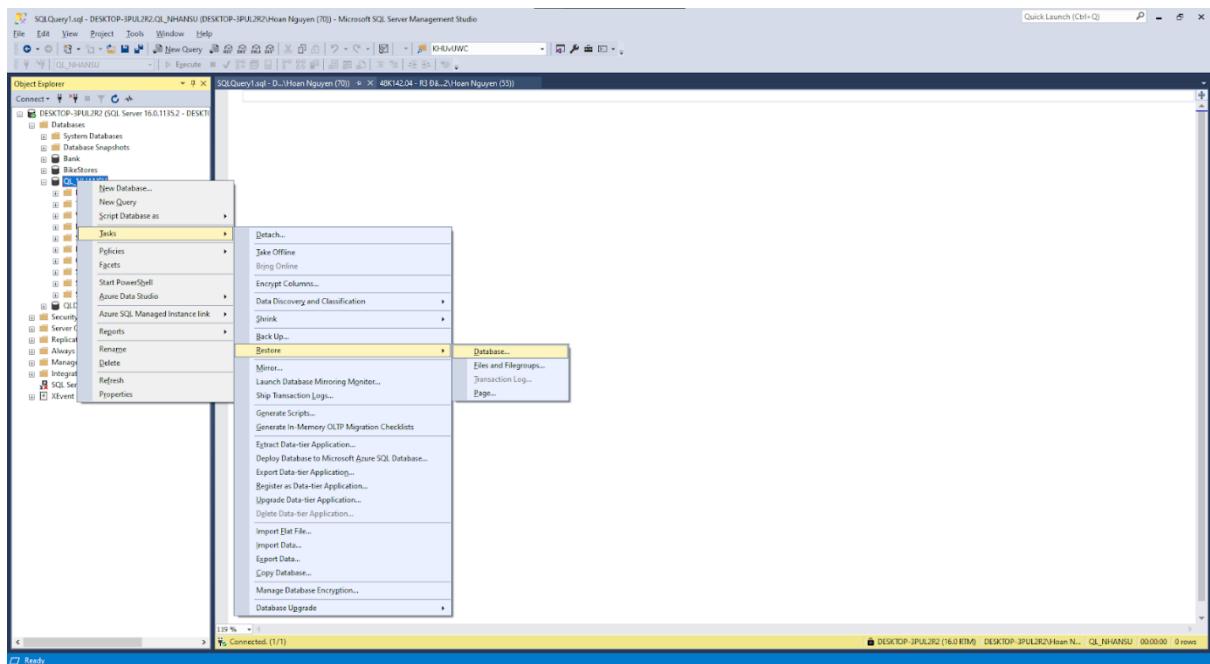
-- Thay đổi các cột trong bảng NHANVIEN thành NULL
ALTER TABLE NHANVIEN ALTER COLUMN TEINV NVARCHAR(255) NULL
;

```

"Backup Full Quanlynhansu.Subplan\_1" được chạy thành công

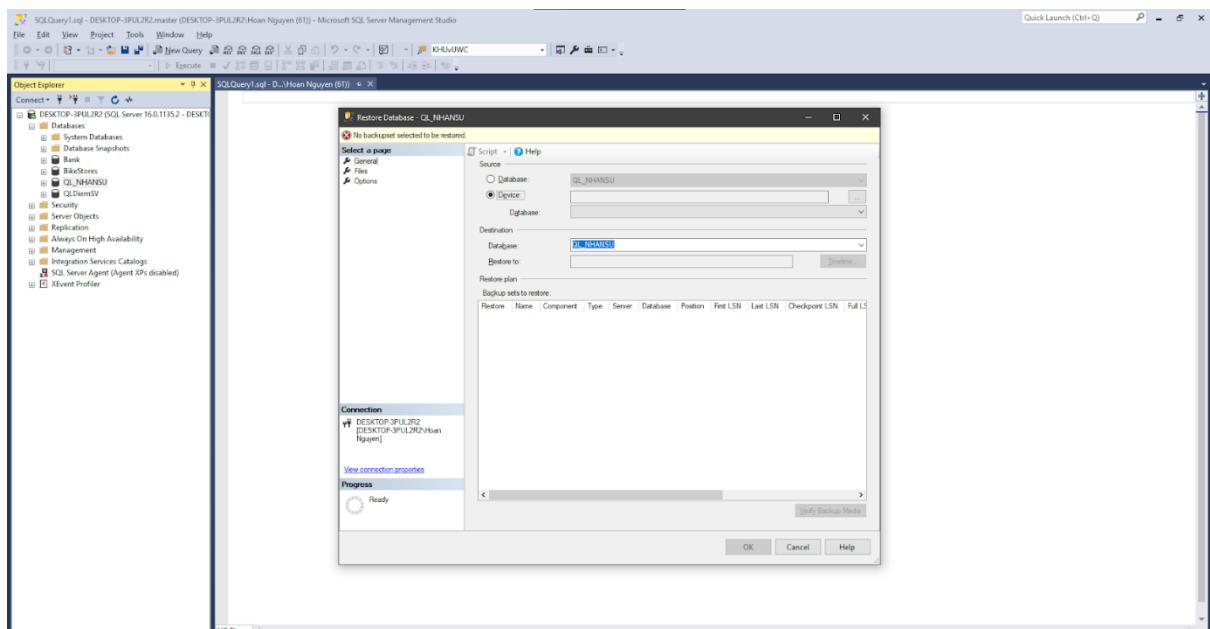
## 5.2.Phục hồi dữ liệu trên Server Local

Bước 1: Lựa chọn loại phục hồi

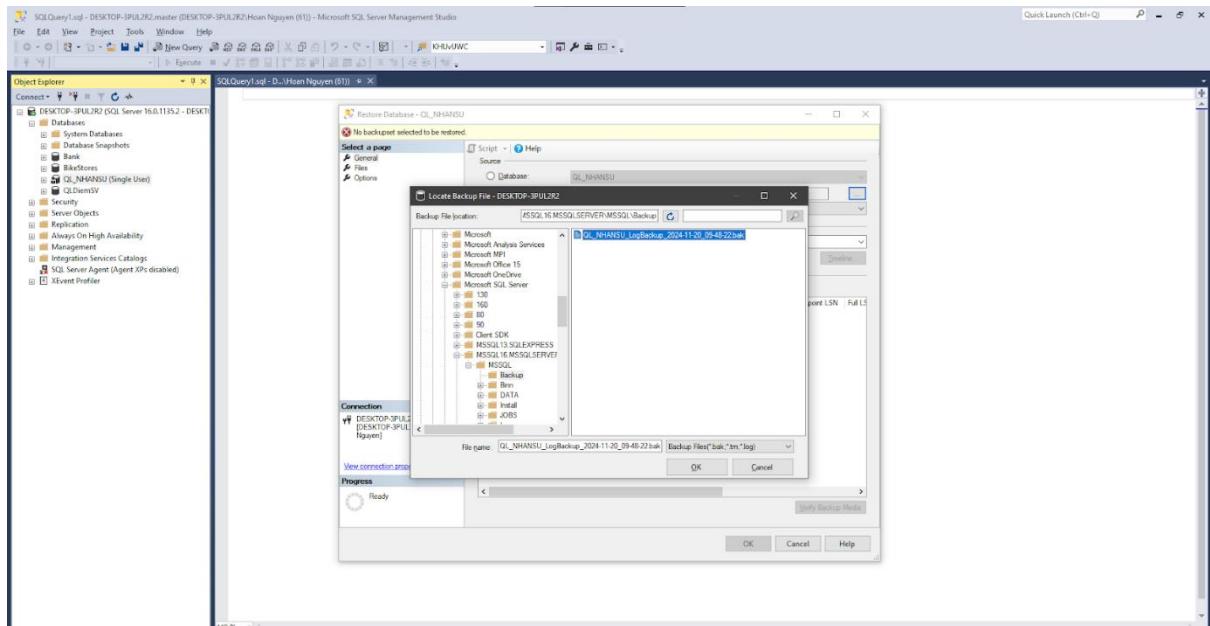


Lựa chọn loại phục hồi

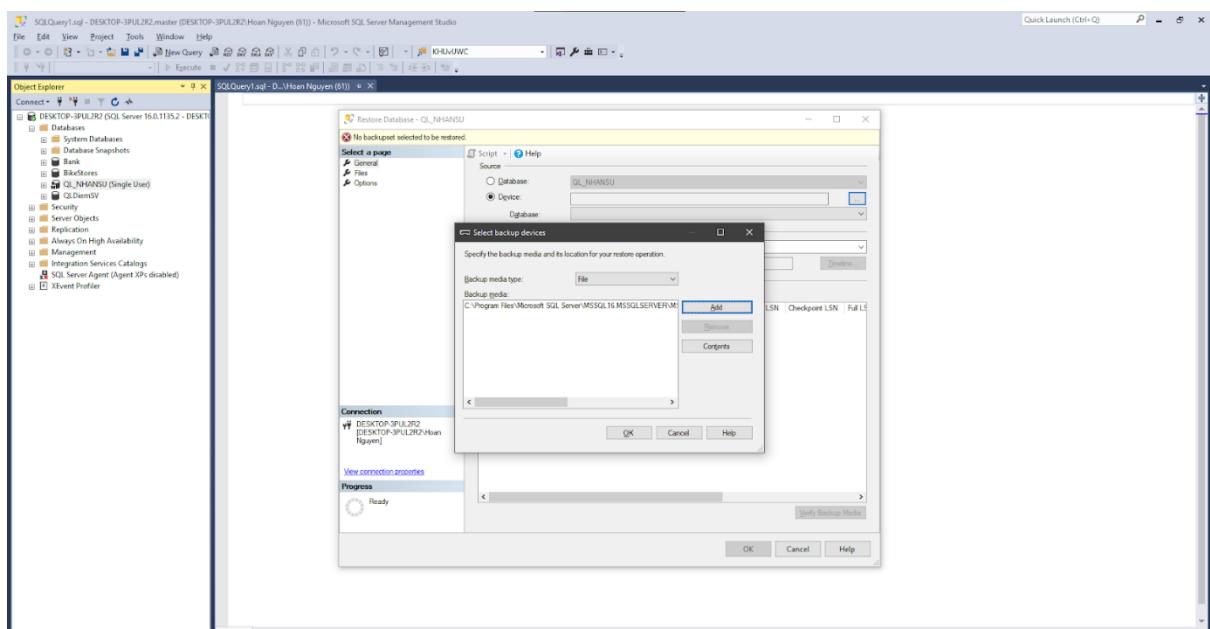
## Bước 2: Chọn thiết bị lưu trữ backup



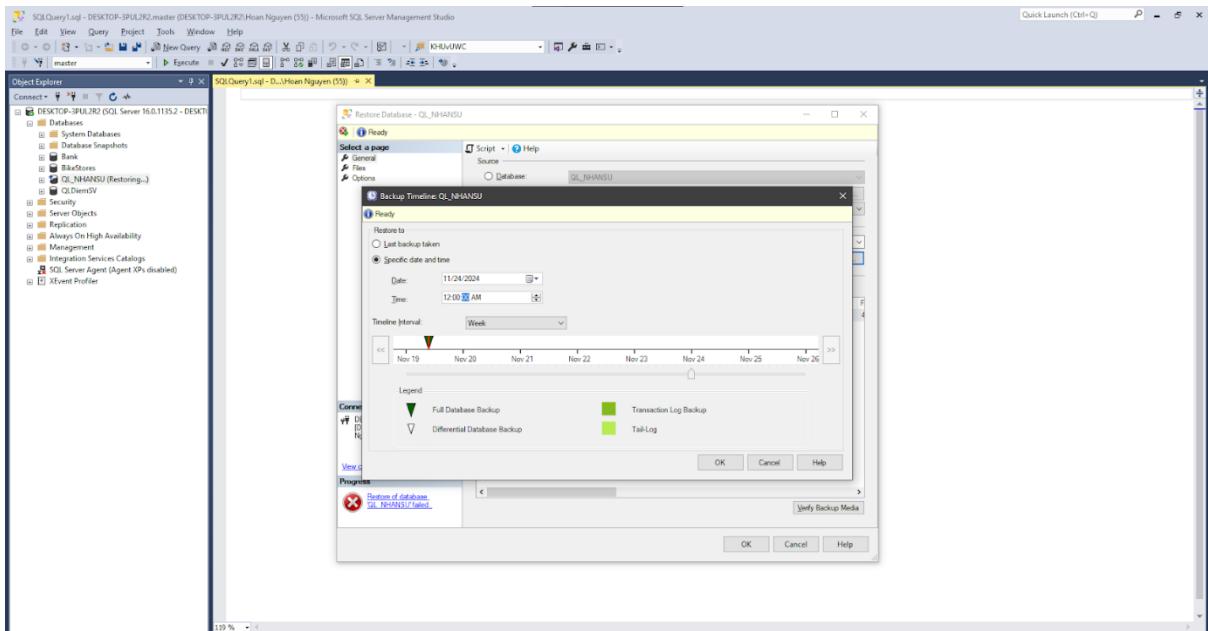
Chọn thiết bị lưu trữ backup



*Chọn file đã backup để phục hồi*

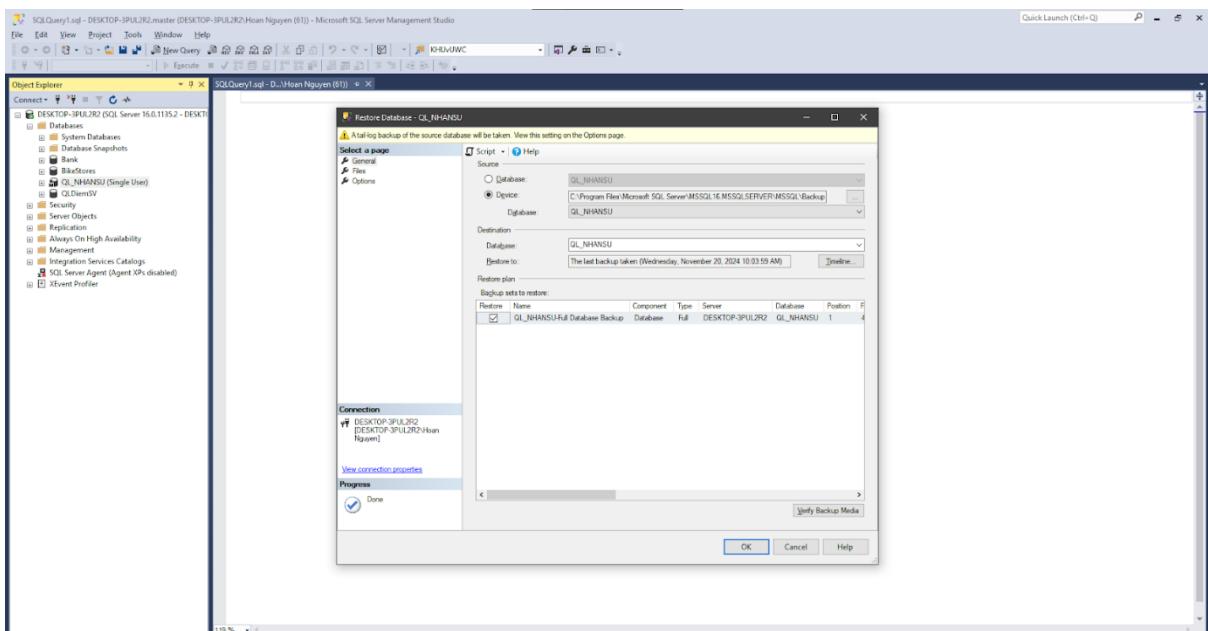


*Chọn Ok để thiết lập các tùy chọn phục hồi*



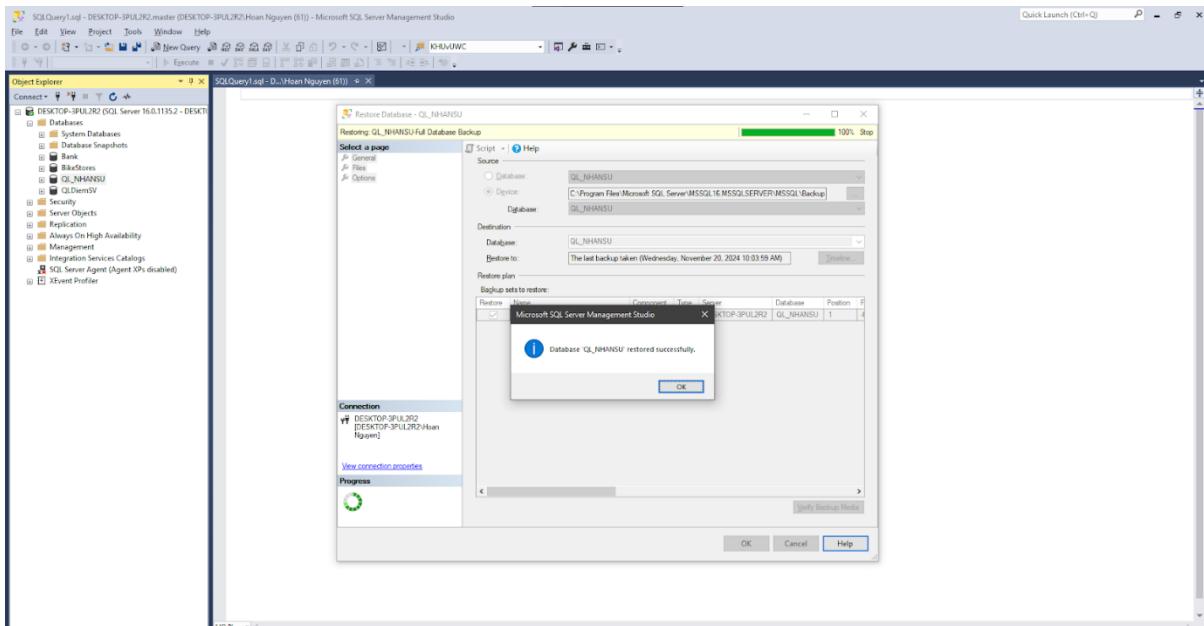
*Lựa chọn thời gian phục hồi*

### Bước 3: Thiết lập tùy chọn phục hồi



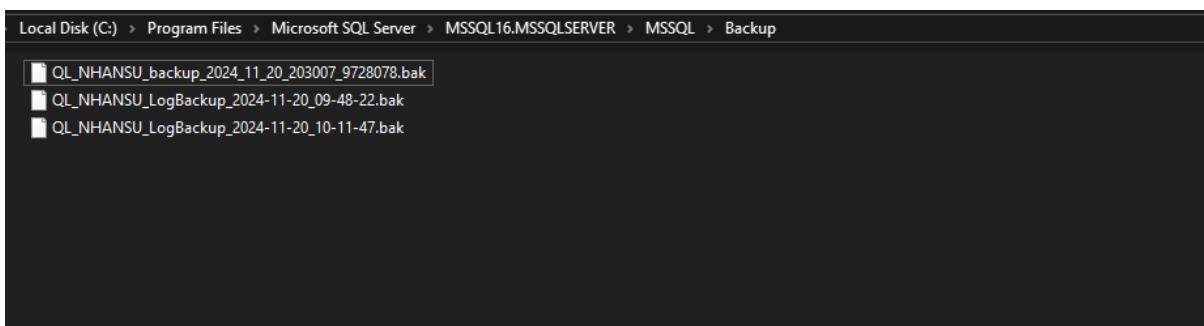
*Thiết lập tùy chọn phục hồi*

### Bước 4: Phục hồi thành công



*Phục hồi thành công*

#### Bước 4: Kiểm tra và quản lý các phiên bản phục hồi trong thư mục đã chọn để phục hồi



*Kiểm tra và Quản lý các phiên bản backup và phục hồi*

## CHƯƠNG 6. R7

**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)**

### 6.1.Tổng quan

Hệ thống quản lý nhân sự hiện tại đang đối mặt với nhiều thách thức lớn, đặc biệt là khối lượng dữ liệu ngày càng tăng liên quan đến thông tin nhân viên, chấm công, bảng lương, đào tạo và quản lý hiệu suất. Sự phát triển không ngừng của doanh nghiệp khiến hệ thống phải xử lý lượng dữ liệu khổng lồ hàng ngày, trong khi hệ thống lưu trữ cũ đã bắt đầu quá tải, dẫn đến tình

trang chậm trễ trong việc truy xuất và xử lý thông tin. Điều này không chỉ ảnh hưởng đến hiệu suất làm việc của bộ phận nhân sự mà còn gây bất tiện cho các hoạt động điều hành.

Bên cạnh đó, khả năng truy cập của hệ thống còn bị giới hạn. Điều này tạo ra khó khăn khi doanh nghiệp cần hỗ trợ làm việc từ xa hoặc tại các chi nhánh (trong tương lai). Sự thiếu linh hoạt trong truy cập không chỉ làm giảm hiệu quả công việc mà còn hạn chế khả năng đáp ứng các nhu cầu thay đổi của doanh nghiệp.

Ngoài ra, vấn đề bảo mật và sao lưu dữ liệu cũng là một mối lo ngại lớn. Hệ thống hiện tại không được trang bị các cơ chế bảo mật tiên tiến như mã hóa dữ liệu hay xác thực đa yếu tố, khiến dữ liệu nhạy cảm dễ bị tấn công trước các rủi ro như tấn công mạng hoặc rò rỉ thông tin. Đặc biệt, khả năng sao lưu và khôi phục dữ liệu khi xảy ra sự cố còn nhiều hạn chế, làm tăng nguy cơ mất mát dữ liệu quan trọng, ảnh hưởng trực tiếp đến hoạt động kinh doanh. Những bất cập này đặt ra yêu cầu cấp thiết cho việc nâng cấp và chuyển đổi hệ thống quản lý nhân sự lên một nền tảng hiện đại, linh hoạt và an toàn hơn.

Những hạn chế hiện tại của hệ thống quản lý nhân sự cho thấy rằng việc triển khai dịch vụ Cloud giúp giải quyết hiệu quả các vấn đề về lưu trữ, bảo mật và truy cập, đồng thời mở ra cơ hội nâng cao hiệu quả quản lý và đáp ứng tốt hơn nhu cầu phát triển của doanh nghiệp.

## 6.2.Amazon Web Services (AWS)

### 6.2.1.Thông tin cơ bản về AWS

Amazon Web Services (AWS) là nền tảng điện toán đám mây hàng đầu thế giới, cung cấp hơn 200 dịch vụ toàn diện bao gồm lưu trữ, máy chủ, cơ sở dữ liệu, trí tuệ nhân tạo (AI), học máy (Machine Learning), và nhiều dịch vụ khác. Được phát triển bởi Amazon, AWS hiện là lựa chọn tin cậy của các doanh nghiệp lớn trên toàn cầu như Netflix, Samsung, và Airbnb. Hạ tầng của AWS trải rộng trên 99 vùng khả dụng (Availability Zones) tại hơn 30 khu vực địa lý (Regions), đảm bảo tốc độ xử lý nhanh chóng và khả năng phục hồi mạnh mẽ.



Hình 6.2.1.Logo AWS

## **6.2.2.Lý do lựa chọn AWS**

- Độ tin cậy và quy mô lớn**

AWS có hệ thống hạ tầng rộng khắp, đảm bảo tính sẵn sàng cao và tốc độ truy cập nhanh, phù hợp cho doanh nghiệp có nhiều chi nhánh hoặc nhân viên làm việc từ xa.

Được chứng nhận bởi các tiêu chuẩn bảo mật quốc tế như ISO 27001, SOC 2, và GDPR, AWS mang lại sự tin cậy trong việc bảo vệ dữ liệu nhân sự.

- Khả năng mở rộng linh hoạt**

AWS hỗ trợ mở rộng tài nguyên nhanh chóng mà không cần đầu tư thêm vào phần cứng, phù hợp với sự tăng trưởng dữ liệu liên tục của hệ thống quản lý nhân sự.

Doanh nghiệp có thể bắt đầu với quy mô nhỏ và tăng dần tài nguyên khi cần thiết, giúp tối ưu hóa chi phí.

- Chi phí hợp lý**

AWS cung cấp mô hình thanh toán "Pay-as-you-go" linh hoạt, chỉ tính phí dựa trên mức sử dụng thực tế. Điều này giúp doanh nghiệp tránh lãng phí và quản lý chi phí hiệu quả hơn.

AWS còn cung cấp nhiều công cụ tối ưu hóa chi phí như AWS Cost Explorer và AWS Budgets, giúp doanh nghiệp dễ dàng theo dõi và kiểm soát ngân sách.

- Dịch vụ phong phú và hỗ trợ tích hợp**

AWS cung cấp đa dạng dịch vụ như Amazon S3 (lưu trữ), Amazon EC2 (máy chủ ảo), và Amazon RDS (quản lý cơ sở dữ liệu), đáp ứng đầy đủ các nhu cầu của hệ thống HRM.

Các dịch vụ của AWS dễ dàng tích hợp với các phần mềm quản lý nhân sự phổ biến hiện nay hoặc các giải pháp nội bộ của doanh nghiệp.

- Bảo mật vượt trội**

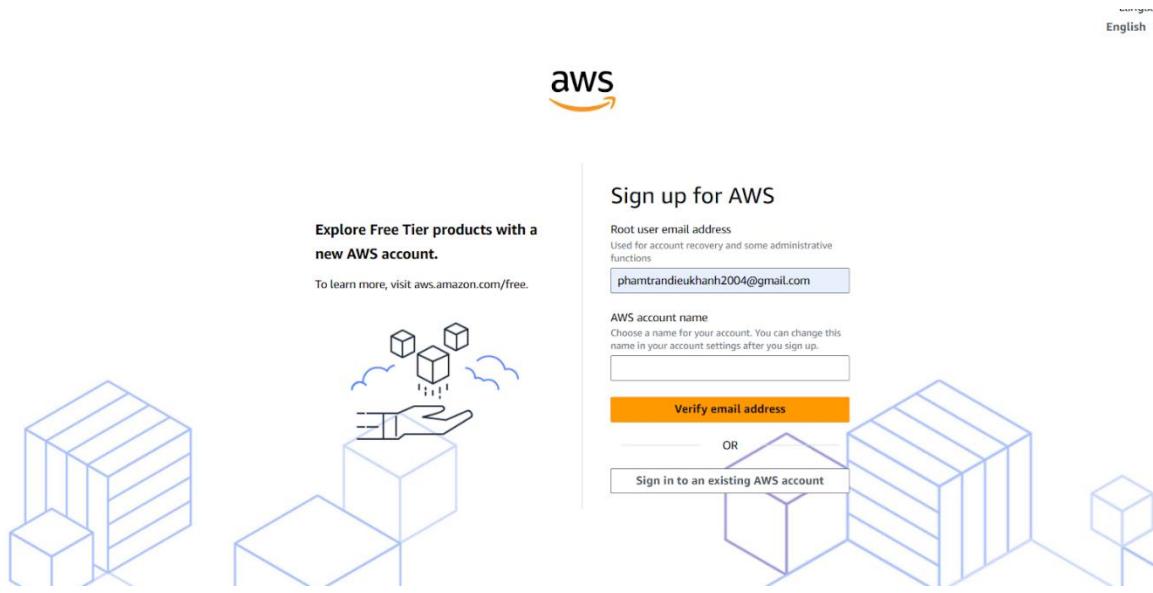
AWS cung cấp các tính năng bảo mật mạnh mẽ như mã hóa dữ liệu, xác thực đa yếu tố (MFA), và hệ thống phát hiện mối đe dọa tự động AWS Guard Duty.

Hệ thống sao lưu tự động và khả năng khôi phục nhanh của AWS đảm bảo dữ liệu nhân sự luôn an toàn, ngay cả khi xảy ra sự cố.

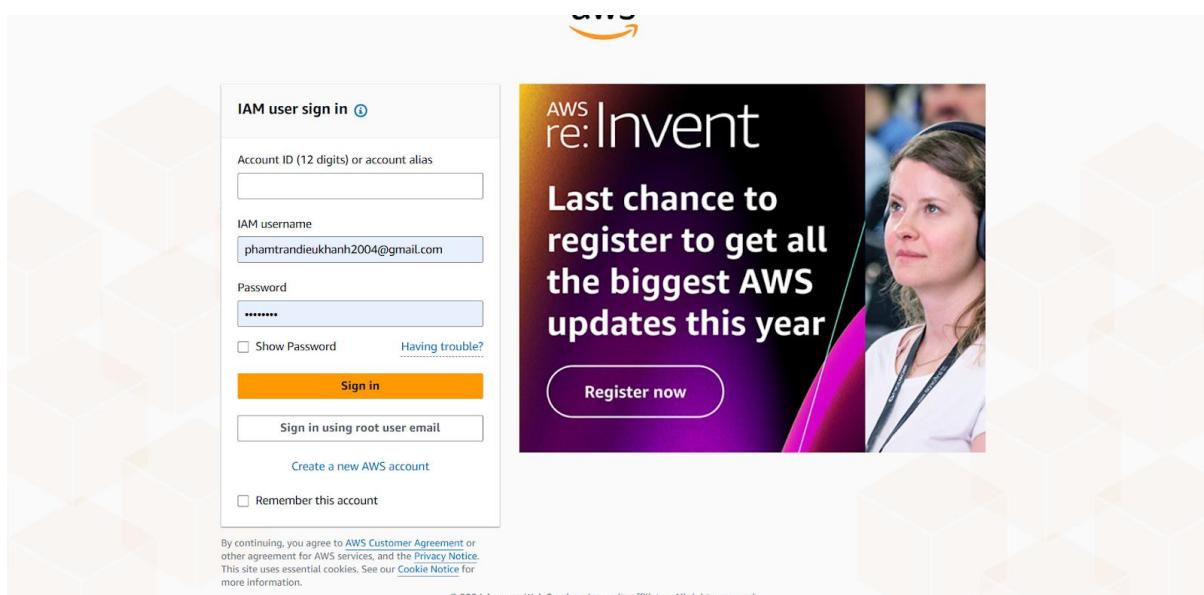
## **6.3.Triển khai**

### **6.3.1.Truy cập tài khoản AWS**

- Nếu chưa có tài khoản tiến hành đăng ký tài khoản AWS



- Nếu đã có tài khoản tiến hành đăng nhập vào AWS.



### 6.3.2.Tạo database

- Truy cập AWS Management Console

Điều hướng đến dịch vụ RDS (Relational Database Service).

The screenshot shows the AWS RDS Dashboard for the Asia Pacific (Sydney) region. On the left, a sidebar lists various RDS management options like Databases, Query Editor, and Performance insights. The main panel displays a summary of resources: DB Instances (0/40), DB Clusters (0/40), and Snapshots (0). It also shows supported platforms (VPC) and a default network (vpc-06d4ccf731f00a5d4). A banner at the top introduces Aurora I/O-Optimized. To the right, a 'Recommended services' section is shown with a note about no recommendations yet, and a 'Recommended for you' section featuring links to 'Amazon RDS Backup and Restore using AWS Backup' and 'Migrate SSRS to RDS for SQL Server'.

- Chọn "Create database"

Trong giao diện RDS, nhấp vào nút Create database.

The screenshot shows the 'Databases' page within the RDS service. It displays a single database entry named 'qlnhansu' with the status 'Available'. The page includes a search bar, a toolbar with 'Group resources', 'Modify', 'Actions', 'Restore from S3', and 'Create database' buttons, and a detailed table view of the database properties.

- Cấu hình cơ sở dữ liệu

Engine type: Chọn Microsoft SQL Server.

Database creation method: Chọn Standard Create để tùy chỉnh cấu hình.

Edition: Lựa chọn phiên bản SQL Server Express.

The screenshot shows the 'Engine options' section of the AWS RDS console. On the left, there's a grid of database engine icons with their names: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server (selected), and IBM Db2. To the right of the grid is a detailed description of Microsoft SQL Server, stating it's a relational database management system developed by Microsoft. Below this, the configuration details for the selected Microsoft SQL Server instance are shown, including the database management type (Amazon RDS), edition (SQL Server Express Edition), license (License Included), and engine version (SQL Server 2019 15.00.4395.2.v1).

- **Cấu hình phiên bản cơ sở dữ liệu (DB instance)**

DB instance class: Chọn db.t3.medium cho môi trường thử nghiệm.

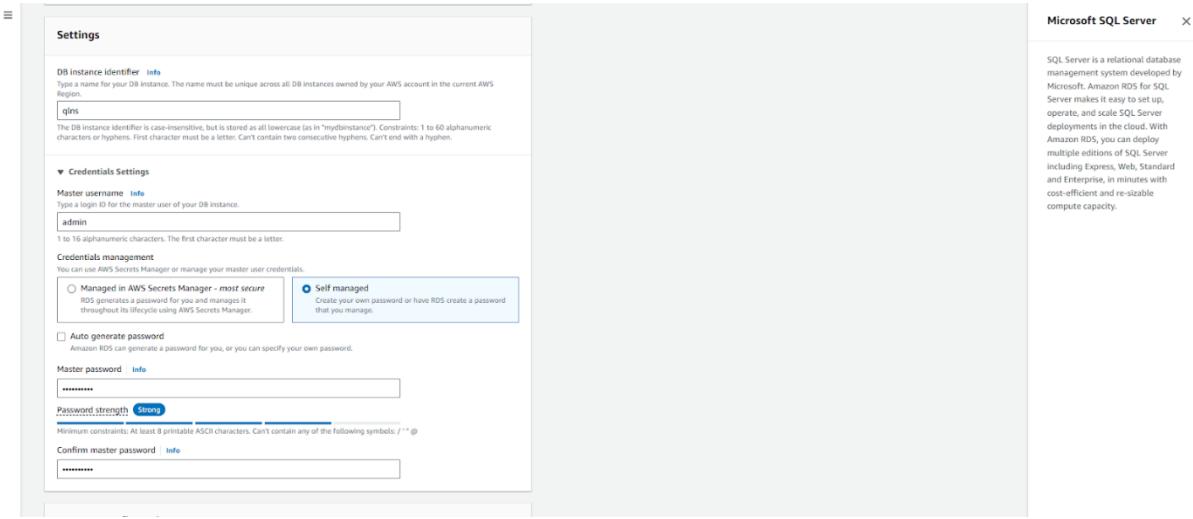
Storage (dung lượng lưu trữ): Chọn 20 GB - là dung lượng tối đa cho môi trường thử nghiệm.

The screenshot shows the 'Storage' configuration section of the AWS RDS DB instance creation wizard. It includes fields for selecting the storage type (General Purpose SSD (gp2)), setting allocated storage (20 GiB), and a note about storage optimization after modification. Other sections visible include 'Burstable classes (includes t classes)' (set to db.t3.micro) and 'Connectivity'.

- **Đặt thông tin đăng nhập**

Master username: Nhập tên tài khoản quản trị .

Master password: Nhập mật khẩu và xác nhận.

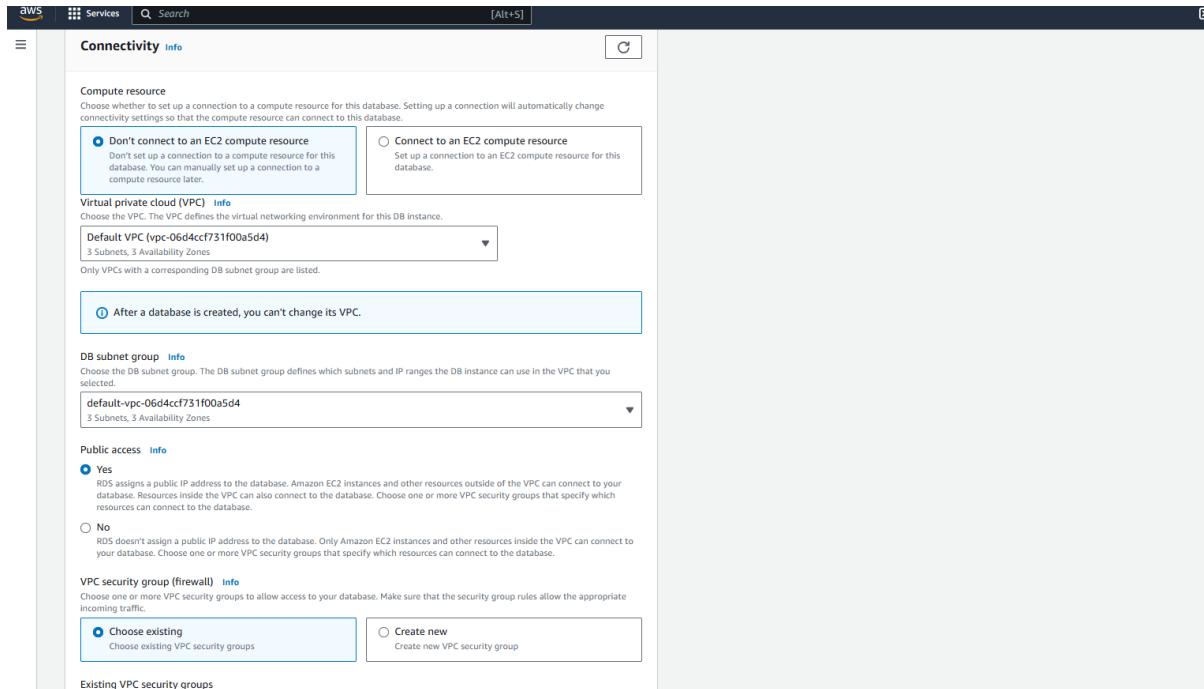


- **Cấu hình mạng**

VPC (Virtual Private Cloud): Chọn VPC mặc định .

Subnet group: Đảm bảo cơ sở dữ liệu có thể truy cập từ các subnet cần thiết.

Public access: Chọn Yes để cơ sở dữ liệu có thể được truy cập qua Internet.



- **Tạo cơ sở dữ liệu**

Nhấp vào nút Create database và chờ AWS tạo phiên bản cơ sở dữ liệu.

### 6.3.3. Cấu hình bảo mật và quyền truy cập

- **Tạo Security Group**

Trong AWS Management Console, điều hướng đến dịch vụ EC2 → Security Groups.

Tìm Security Group được gán cho cơ sở dữ liệu.

Thêm một Inbound Rule mới:

- Type: MS SQL (TCP/1433).
- Source: Đặt My IP và tạo Inbound Rule riêng cho từng IP của các thành viên còn lại.

- **Xác nhận Endpoint của cơ sở dữ liệu**

Truy cập lại RDS, tìm cơ sở dữ liệu đã tạo, và ghi lại Endpoint (địa chỉ DNS) và Port (1433).

The screenshot shows the AWS RDS console with the 'Connectivity & security' tab selected. The page displays detailed information about the endpoint and port, networking settings (VPC, subnet group, and subnets), and security configurations (VPC security groups, certificate authority, and DB instance certificate expiration date).

Endpoint & port	Networking	Security
Endpoint qlnhansu.cpw86c0u60qm.ap-southeast-2.rds.amazonaws.com Port 1433	Availability Zone ap-southeast-2a  VPC vpc-06d4ccf731f00a5d4  Subnet group default-vpc-06d4ccf731f00a5d4  Subnets subnet-0b5316f964218ec01 subnet-0f846d87485fdf8d1 subnet-01e1507306590e788  Network type IPv4	VPC security groups default (sg-01fc10025b00728d7) Active  Publicly accessible Yes  Certificate authority rds-ca-rsa2048-g1  Certificate authority date May 25, 2061, 04:42 (UTC+07:00)  DB instance certificate expiration date November 17, 2027, 17:08 (UTC+07:00)

### 6.3.4. Kết nối từ SQL Server Management Studio (SSMS)

- **Mở SSMS**

Chạy ứng dụng SQL Server Management Studio trên máy tính.

- **Kết nối tới cơ sở dữ liệu AWS RDS**

Server type: Chọn Database Engine.

Server name: Nhập Endpoint từ cơ sở dữ liệu AWS RDS.

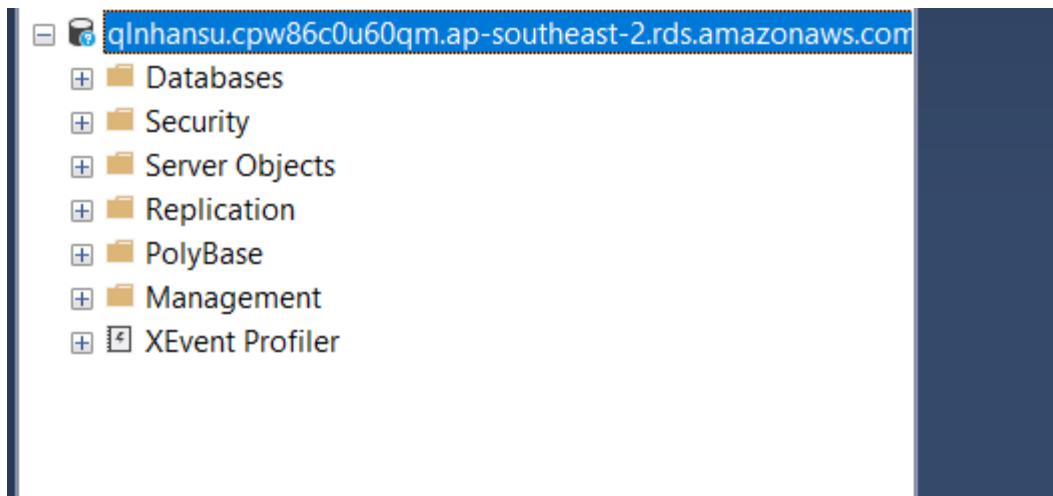
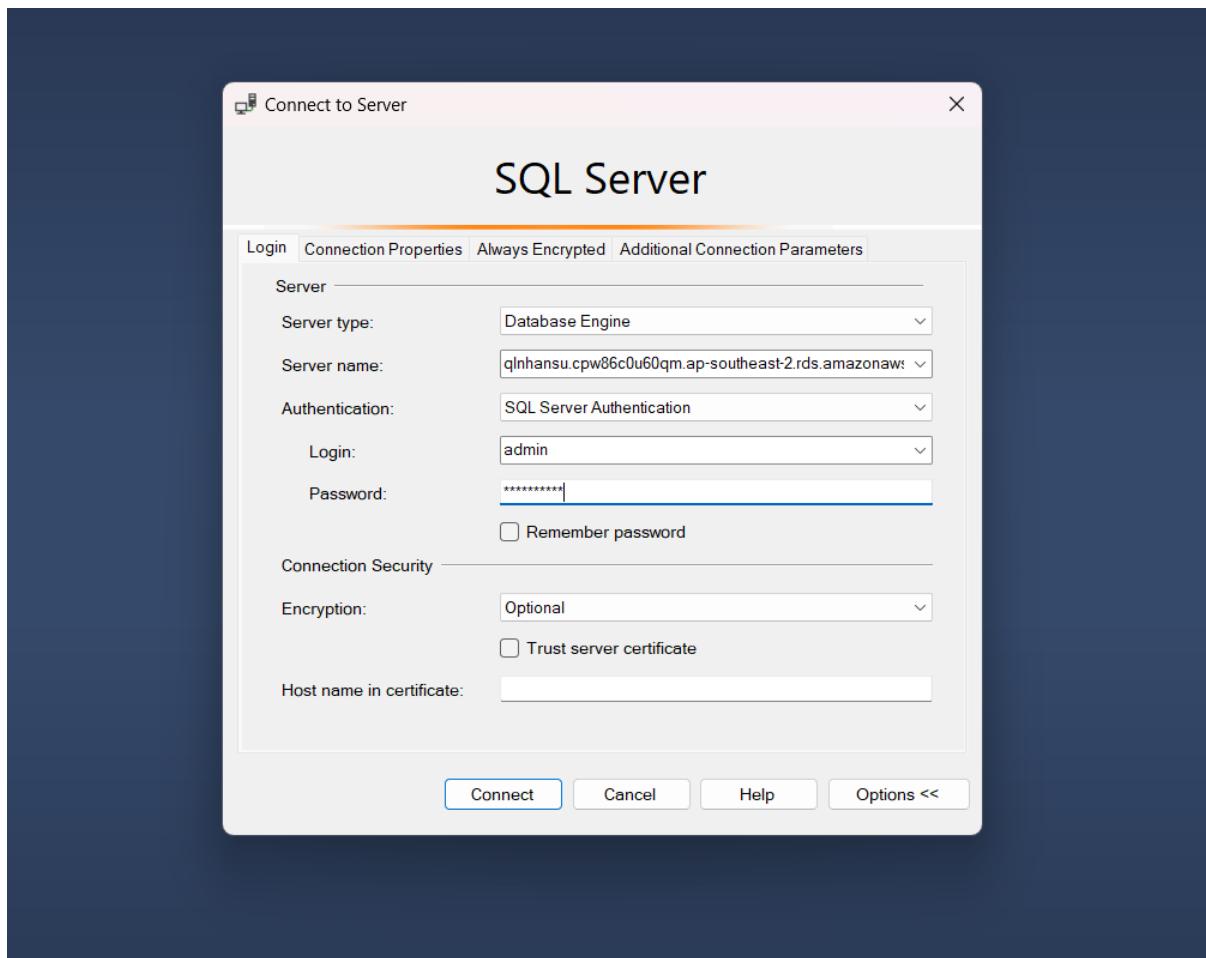
Authentication: Chọn SQL Server Authentication.

Login: Nhập Master username.

Password: Nhập mật khẩu đã cấu hình.

- **Xác nhận kết nối**

Nhập Connect để kết nối với cơ sở dữ liệu trên AWS.



## CHƯƠNG 7. R8

### Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection

#### 7.1.Tổng quan

Tấn công SQL Injection là một kỹ thuật lợi dụng những lỗ hổng về câu truy vấn của các ứng dụng. Được thực hiện bằng cách chèn thêm một đoạn SQL để làm sai lệnh đi câu truy vấn ban đầu, từ đó có thể khai thác dữ liệu từ database. SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác như một người quản trị web, trên cơ sở dữ liệu của ứng dụng.( SQL Injection có thể cho phép những kẻ tấn công thực hiện các thao tác delete, insert, update,..)

- Các dòng lệnh Inject SQL có thể thay đổi câu lệnh SQL và gây nguy hiểm cho khả năng bảo mật của hệ thống. Khi cơ sở dữ liệu có lỗ hổng, kẻ xấu có thể gây ra SQL Injection dựa trên  $1=1$  luôn đúng, hoặc dựa trên “ $=$ ” luôn đúng

- SQL Injection là một trong những mối quan tâm lớn về an ninh thông tin trong hệ thống quản lý cơ sở dữ liệu. Để đối phó với SQL Injection, nhóm chúng em đã chọn giải pháp sử dụng Stored Procedures:

- **Lý do**
  - Thực hiện truy vấn an toàn:
    - Stored Procedures cho phép bạn đặt các thủ tục xử lý truy vấn trực tiếp trong cơ sở dữ liệu
    - Các thủ tục này được định nghĩa trước và chỉ được thực thi thông qua gọi các thủ tục, giảm khả năng thực hiện truy vấn không an toàn trực tiếp từ ứng dụng
  - Ngăn chặn SQL Injection
    - Bằng cách sử dụng các thủ tục lưu trữ, bạn có thể tránh được việc người dùng có thể chèn các đoạn mã SQL độc hại vào câu truy vấn,
    - Các tham số trong Stored Procedures thường được xử lý một cách an toàn, không tạo cơ hội cho SQL Injection.
  - Tăng tính tương thích
    - Sử dụng Stored Procedures tạo ra một lớp trung gian giữa ứng dụng và cơ sở dữ liệu. Điều này làm giảm sự phụ thuộc trực tiếp vào câu truy vấn SQL từ phía ứng dụng
    - Nếu có sự thay đổi trong cấu trúc cơ sở dữ liệu, chỉ cần điều chỉnh Stored Procedures mà không ảnh hưởng đến mã nguồn ứng dụng.
  - Quản lý quyền truy cập:
    - Stored Procedures có thể được xác định với quyền truy cập cụ thể, giúp kiểm soát quyền truy cập vào dữ liệu
    - Người quản trị cơ sở dữ liệu có thể cấp quyền thực thi Stored Procedures mà không cần cấp quyền trực tiếp trên các bảng hoặc cơ sở dữ liệu
  - Hiệu suất:
    - Stored Procedures thường được biên dịch và tối ưu hóa một lần khi được tạo ra, cung cấp hiệu suất tốt hơn so với các truy vấn SQL động.
    - Điều này giúp giảm tải cho cả ứng dụng và cơ sở dữ liệu.

#### 7.2.Triển khai

Sau đây là thủ tục để đảm bảo an toàn trước các tấn công SQL Injection

- Thực hiện thủ tục

```
CREATE OR ALTER PROCEDURE uspp_login
    @username NVARCHAR(100),
    @password NVARCHAR(100),
    @ret INT OUTPUT -- Tham số trả về trạng thái
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra tài khoản
    IF EXISTS (
        SELECT 1
        FROM dbo.TAIKHOAN
        WHERE MANV = @username AND MATKHAU = @password
    )
    BEGIN
        SET @ret = 1; -- Thành công
        -- PRINT N'D Đăng nhập thành công!';
    END
    ELSE
    BEGIN
        SET @ret = 0; -- Thất bại
        -- PRINT N'T Tên người dùng hoặc mật khẩu không hợp lệ!';
    END
END;
```

- Trường hợp khi nhập sai username và password

The screenshot shows a SQL query window with the following code:

```
DECLARE @result INT;
EXEC uspp_login
    @username = N'user_test',
    @password = N'pass_test',
    @ret = @result OUTPUT;
PRINT N'Kết quả: ' + CAST(@result AS NVARCHAR(10));
```

The status bar at the bottom left indicates "77 %". The "Messages" tab in the bottom right shows the output:

Kết quả: 0  
Completion time: 2024-11-25T14:33:22.8149102+07:00

- Trường hợp bị tấn công SQL Injection

```
DECLARE @result INT;

EXEC uspp_login
    @username = N'valid_user',
    @password = N' OR 1=1',
    @ret = @result OUTPUT;

PRINT 'Kết quả: ' + CAST(@result AS NVARCHAR(10));
```

Completion time: 2024-11-25T14:24:54.7026569+07:00

- Trường hợp khi nhập đúng username và password

```
DECLARE @result INT;

EXEC uspp_login
    @username = N'NV0001',
    @password = N'MK8946',
    @ret = @result OUTPUT;

PRINT N'Kết quả: ' + CAST(@result AS NVARCHAR(10));
```

Completion time: 2024-11-25T14:36:53.7750134+07:00

