

ĐẠI HỌC QUỐC GIA TPHCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI THỰC HÀNH SỐ 3

Đề tài: Mã hóa dữ liệu sử dụng các thuật toán mã hóa công khai

Môn học: Bảo mật cơ sở dữ liệu

Sinh viên thực hiện:

Doãn Hoàng Sơn - 22127365

Võ Hữu Tuấn - 22127439

Giáo viên hướng dẫn:

Cô Nguyễn Thị Hường

Thầy Nguyễn Đình Thúc

Thầy Lê Trọng Anh Tú



Mục lục

1	Tổng quan	2
1.1	Nội dung thực hành	2
1.2	Thông tin nhóm	2
1.3	Phân công công việc	2
2	Cấu trúc chương trình	3
3	Kết quả thực hiện	4
3.1	Câu a: Viết script tạo Database	4
3.2	Câu b: Viết script tạo mới các Table	5
3.3	Câu c: Viết các Stored procedure	7
3.3.1	i. Stored dùng để thêm mới dữ liệu (Insert) vào table NHANVIEN	7
3.3.2	Stored dùng để truy vấn dữ liệu nhân viên (NHANVIEN)	7
3.4	Câu d: Viết các stored procedure và chương trình Python	9
3.4.1	Viết Stored procedure cần thiết	9
3.4.2	Chương trình Python	10
3.5	Câu e: Sử dụng công cụ SQL Profiler	18
3.5.1	Tạo phiên giám sát	18
3.5.2	Giám sát thao tác nhập điểm sinh viên	19
3.5.3	Nhận xét	20
TÀI LIỆU THAM KHẢO		22

1 Tổng quan

1.1 Nội dung thực hành

- Tạo và quản lý khóa
- Tạo bảng và mã hóa dữ liệu sử dụng mã hóa công khai (RSA)
- Tạo stored procedure để truy vấn dữ liệu đã mã hóa

1.2 Thông tin nhóm

Nhóm: 03		
1	Doãn Hoàng Sơn	22127365
2	Võ Hữu Tuấn	22127439

Thông tin nhóm

1.3 Phân công công việc

	Hoàng Sơn	Hữu Tuấn
Câu a	x	
Câu b	x	
Phần i câu c	x	
Phần ii câu c		x
Câu d	x	x
Câu e		x
Báo cáo	x	x

Bảng phân công công việc

2 Cấu trúc chương trình

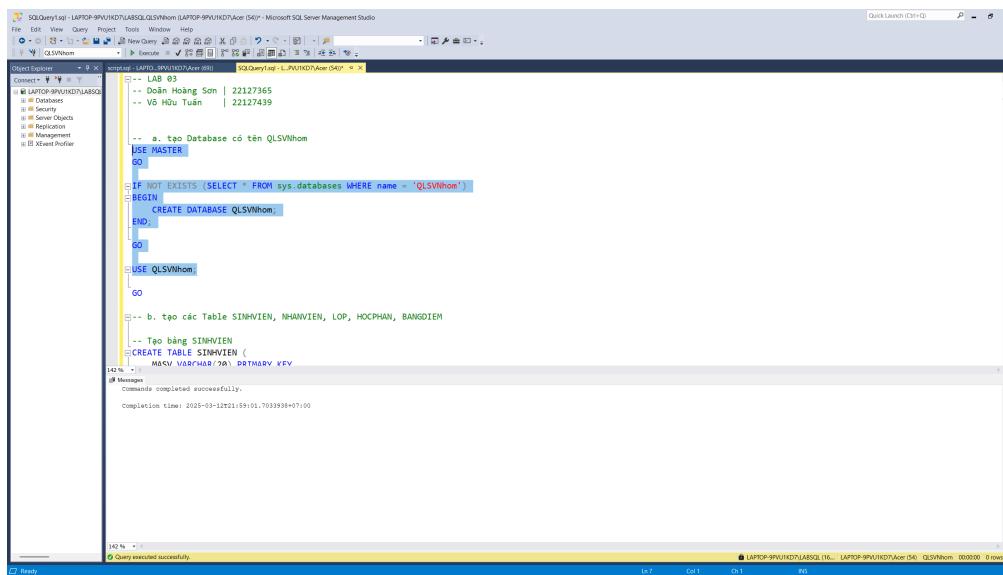
Chương trình bao gồm các thành phần chính sau:

- Database (SQL): bao gồm việc tạo và quản lý dữ liệu cùng các hàm Stored procedure hỗ trợ việc truy vấn dữ liệu
- Giao diện chương trình (Python): Hiển thị giao diện đơn giản và dễ thao tác, sử dụng các thư viện hỗ trợ như:
 - **pyodbc**: Thư viện kết nối SQL Server
 - **ttkbootstrap**: Thư viện hỗ trợ giao diện
 - **tkinter**: Thư viện thông báo

Xem source code tại [đây](#)

3 Kết quả thực hiện

3.1 Câu a: Viết script tạo Database



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'QLSVNhom' is selected under 'LAB 03'. The 'Script' button is highlighted. In the main query window, two scripts are shown:

```
-- a. tao Database có tên QLSVNhom
USE MASTER
GO
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'QLSVNhom')
BEGIN
CREATE DATABASE QLSVNhom;
END
GO

-- b. tao các Table SINHVIEN, NHANVIEN, LOP, HOCPHAN, BANGDIEM
-- Tạo bảng SINHVIEN
CREATE TABLE SINHVIEN (
    ...  

    MAX VARBINARY / A) DTNSV NVF  

142
```

The execution status at the bottom indicates: 'Commands completed successfully.' and 'Completion time: 2020-03-11T21:59:01.7033938+07:00'.

Tạo Database

3.2 Câu b: Viết script tạo mới các Table

```

-- b. tạo các Table SINHVIEN, NHANVIEN, LOP, HOCPHAN, BANGDIEM
-- Tạo bảng SINHVIEN
CREATE TABLE SINHVIEN (
    MASV VARCHAR(20) PRIMARY KEY,
    HOTEN NVARCHAR(100) NOT NULL,
    NGAYSINH DATETIME,
    DIACHI NVARCHAR(200),
    MALOP VARCHAR(20),
    TENMON NVARCHAR(100) UNIQUE NOT NULL,
    MATKHAU VARBINARY(MAX) NOT NULL
);
GO

-- Tạo bảng NHANVIEN
CREATE TABLE NHANVIEN (
    MANV VARCHAR(20) PRIMARY KEY,
    HOTEN NVARCHAR(100) NOT NULL,
    EMAIL VARCHAR(20),
    LUONG VARBINARY(MAX),
    TENMON NVARCHAR(100) UNIQUE NOT NULL,
    MATKHAU VARBINARY(MAX) NOT NULL,
    PUBKEY VARCHAR(20) NOT NULL
);
GO

```

Messages

Commands completed successfully.

Completion time: 2025-03-13T01:29:23.4322298+07:00

Tạo bảng SINHVIEN và NHANVIEN

```

-- Tạo bảng LOP
CREATE TABLE LOP (
    MALOP VARCHAR(20) PRIMARY KEY,
    TENLOP NVARCHAR(100) UNIQUE NOT NULL,
    MATKHAU VARBINARY(MAX) NOT NULL,
    PUBKEY VARCHAR(20) NOT NULL
);
GO

-- Tạo bảng HOCPHAN
CREATE TABLE HOCPHAN (
    MAPH VARCHAR(20) PRIMARY KEY,
    TENDIEM NVARCHAR(100) NOT NULL,
    SOTC INT
);
GO

```

Messages

Commands completed successfully.

Completion time: 2025-03-13T01:29:23.4322298+07:00

Tạo bảng LOP và HOCPHAN

BÁO CÁO BÀI THỰC HÀNH SỐ 3

Trường Đại học Khoa học Tự nhiên
Bảo mật cơ sở dữ liệu

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a connection to 'LAPTOP-8PVUHD7\Acer (54)' is selected. In the center pane, a query window titled 'SQLQuery1.sql - L:\NVNHD7\Acer (54)\...' displays the following SQL code:

```
CREATE TABLE HOCHPHAN (
    MAHP VARCHAR(20) PRIMARY KEY,
    TENHP NVARCHAR(100) NOT NULL,
    SOTC INT
);

GO

-- Tạo bảng BANGDIEM
CREATE TABLE BANGDIEM (
    MASV VARCHAR(20),
    MAHP VARCHAR(20),
    DIEMTHI VARBINARY(MASV),
    PRIMARY KEY (MASV, MAHP),
    FOREIGN KEY (MASV) REFERENCES SINHVIEN(MASV),
    FOREIGN KEY (MAHP) REFERENCES HOCHPHAN(MAHP)
);

GO

--- c. Viết các Stored procedure
--- 1. SP_INS_PUBLIC_NHANVIEN
--- Stored dùng để thêm mới dữ liệu (Insert) vào table NHANVIEN
CREATE PROCEDURE sp_INS_PUBLIC_NHANVIEN
AS
    BEGIN
        INSERT INTO NHANVIEN (TEN, DIACHI_TP, NAM sinh)
        VALUES ('Nguyễn Văn A', 'Hà Nội', 1990);
    END
    GO

--- 2. SP_UPD_PUBLIC_NHANVIEN
--- Stored dùng để cập nhật dữ liệu (Update) vào table NHANVIEN
CREATE PROCEDURE sp_UPD_PUBLIC_NHANVIEN
AS
    BEGIN
        UPDATE NHANVIEN SET TEN = 'Nguyễn Văn B', DIACHI_TP = 'Hồ Chí Minh' WHERE ID = 1;
    END
    GO
```

The status bar at the bottom indicates 'Messages' and 'Commands completed successfully.' with a completion time of '2025-03-13T01:20:23.4322299+07:00'.

Tạo bảng BANGDIEM

3.3 Câu c: Viết các Stored procedure

3.3.1 i. Stored dùng để thêm mới dữ liệu (Insert) vào table NHANVIEN

- Mã hóa mật khẩu bằng SHA1
- Mã hóa lương bằng RSA với:
 - Public key là mã nhân viên
 - Private key là mật khẩu

```

CREATE PROCEDURE SP_INS_PUBLIC_NHANVIEN
    @Ten NVARCHAR(50),
    @MaNV NVARCHAR(50),
    @Luong MONEY,
    @Mk NVARCHAR(50)
AS
BEGIN
    -- Kiểm tra nếu nhân viên đã tồn tại
    IF EXISTS (SELECT * FROM NHANVIEN WHERE Ten = @Ten)
        BEGIN
            PRINT 'Lỗi: Nhân viên này đã tồn tại';
            RETURN;
        END
    ELSE
        BEGIN
            -- Tạo khóa công khai RSA
            EXEC sp_generatekey @key_id = 'KEY1', @key_size = 2048;
            SELECT @key_id AS 'Khóa công khai RSA';
            PRINT 'Khóa RSA đã tạo thành công';

            -- Giải mã khóa RSA
            SELECT @key_id AS 'Khóa công khai RSA';
            PRINT 'Khóa RSA đã giải mã thành công';

            -- Lưu khóa RSA vào file
            EXEC sp_exportkey @key_id = 'KEY1', @file_name = 'RSAKey.pvk';
            PRINT 'Khóa RSA đã lưu thành công';

            -- Tạo khóa 私匙 (Private Key)
            EXEC sp_generatekey @key_id = 'KEY2', @key_size = 2048;
            SELECT @key_id AS 'Khóa 私匙';
            PRINT 'Khóa 私匙 đã tạo thành công';

            -- Giải mã khóa 私匙
            SELECT @key_id AS 'Khóa 私匙';
            PRINT 'Khóa 私匙 đã giải mã thành công';

            -- Lưu khóa 私匙 vào file
            EXEC sp_exportkey @key_id = 'KEY2', @file_name = 'RSAKey.pvk';
            PRINT 'Khóa 私匙 đã lưu thành công';

            -- Tạo khóa LUONG bằng RSA
            EXEC sp_encryptdata @data = @Luong, @key_id = 'KEY1', @output_name = 'LUONG_RSA';
            PRINT 'Khóa LUONG đã mã hóa thành công';

            -- Tạo khóa MK bằng RSA
            EXEC sp_encryptdata @data = @Mk, @key_id = 'KEY1', @output_name = 'MK_RSA';
            PRINT 'Khóa MK đã mã hóa thành công';

            -- Tạo khóa LUONG và MK bằng RSA
            EXEC sp_encryptdata @data = @Luong, @key_id = 'KEY1', @output_name = 'LUONG_RSA';
            EXEC sp_encryptdata @data = @Mk, @key_id = 'KEY1', @output_name = 'MK_RSA';

            -- Insert dữ liệu vào bảng
            INSERT INTO NHANVIEN (Ten, MaNV, LUONG, LUONG_RSA, MK, MK_RSA)
            VALUES (@Ten, @MaNV, @Luong, @LUONG_RSA, @Mk, @MK_RSA);
        END
    END
END
  
```

Tạo Stored procedure SP_INS_PUBLIC_NHANVIEN

3.3.2 Stored dùng để truy vấn dữ liệu nhân viên (NHANVIEN)

- Giải mã hóa LUONG bằng khóa RSA đã tạo tương ứng cùng Private key là mật khẩu

```

CREATE PROCEDURE SP_SEL_PUBLIC_NHANVIEN
AS
BEGIN
    SET NOCOUNT ON;
    -- Khởi tạo biến
    DECLARE @NAME NVARCHAR(100)
    DECLARE @POSITION NVARCHAR(100)
    DECLARE @GENDER NVARCHAR(100)
    DECLARE @BIRTHDAY NVARCHAR(100)
    DECLARE @ENCRYPTED_PASSWORD NVARCHAR(100)
    DECLARE @SALT NVARCHAR(100)
    -- Khởi tạo biến trả về
    DECLARE @RESULT NVARCHAR(100);
    -- Lấy thông tin nhân viên
    SELECT @NAME = NVARCHAR,
    @POSITION = POSITION,
    @GENDER = GENDER,
    @BIRTHDAY = BIRTHDAY,
    @ENCRYPTED_PASSWORD = ENCRYPTED_PASSWORD,
    @SALT = SALT
    WHERE TENNV = @TENNV;
    -- Kiểm tra nhân viên đã được tạo chìa khóa
    IF NOT EXISTS (SELECT * FROM sys.asymmetric_keys WHERE name = @PKKEY_NAME)
    BEGIN
        PRINT 'Loi: khach vien khong ton tai!';
        RETURN;
    END;
    -- Kiểm tra khóa RSA
    ELSE
    BEGIN
        PRINT 'Loi: khach RSA khong ton tai!';
        RETURN;
    END;
    -- Giải mã bằng
    SET @RESULT = CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE(@ENCRYPTED_PASSWORD, @ENCRYPTED_PASSWORD, @PK));
    -- Kiểm tra nếu giải mã thất bại
    IF @RESULT IS NULL
    BEGIN
        PRINT 'Loi: Giai ma LUONG that bai!';
        RETURN;
    END;
    -- In kết quả
    PRINT 'Kết quả: ' + @RESULT;
    -- Kết thúc
    SELECT
    @NAME AS NV,
    @POSITION AS POSITION,
    @GENDER AS GENDER,
    @BIRTHDAY AS BIRTHDAY;
END;

```

SS 5s - 1 row(s) affected
@Message
Command completed successfully
Completion time: 2024-01-10T11:21:34.1100000+07:00
SS 5s - 1 row(s) affected
Query executed successfully

Tạo Stored procedure SP_SEL_PUBLIC_NHANVIEN

3.4 Câu d: Viết các stored procedure và chương trình Python

3.4.1 Viết Stored procedure cần thiết

- SP_INS_PUBLIC_BANGDIEM: dùng để thêm mới hoặc chỉnh sửa dữ liệu (Insert & Update) vào table BANGDIEM

```

CREATE PROCEDURE SP_INS_PUBLIC_BANGDIEM
    (@MANV VARCHAR(20),
     @MASV VARCHAR(20),
     @MAHP VARCHAR(20),
     @DIEMTHI FLOAT)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @MALOP VARCHAR(20)
    DECLARE @PUBKEY_NAME VARCHAR(20)
    DECLARE @ENCRYPTED_DIEM VARBINARY(MAX)

    -- Lấy lớp học của sinh viên
    SELECT @MALOP = MALOP FROM SINHVIEN WHERE MASV = @MASV;

    -- Kiểm tra nếu sinh viên không tồn tại
    IF @MALOP IS NULL
    BEGIN
        PRINT 'Loi: Sinh vien khong ton tai!';
        RETURN;
    END

    -- Lấy Public Key của nhân viên
    SELECT @PUBKEY_NAME = PUBKEY FROM NHANVIEN WHERE MANV = @MANV;

    -- Kiểm tra nếu không có Public Key
    IF @PUBKEY_NAME IS NULL
    BEGIN
        PRINT 'Loi: NHANVIEN khong co PUBLIC KEY!';
        RETURN;
    END

    -- Mã hóa điểm bằng RSA
    SET @ENCRYPTED_DIEM = EncryptByAsymKey(AsymKey_ID(@PUBKEY_NAME), CAST(@DIEMTHI AS NVARCHAR(100)));

    -- Kiểm tra nếu mã hóa thất bại
    IF @ENCRYPTED_DIEM IS NULL
    BEGIN
        PRINT 'Loi: Ma hoa diem khong thanh cong!';
        RETURN;
    END

    -- Thêm hoặc cập nhật điểm trong bảng BANGDIEM
    IF EXISTS (SELECT 1 FROM BANGDIEM WHERE MASV = @MASV AND MAHP = @MAHP)
    BEGIN
        UPDATE BANGDIEM
        SET DIEMTHI = @ENCRYPTED_DIEM
        WHERE MASV = @MASV AND MAHP = @MAHP;
        PRINT 'Cap nhat diem thanh cong!';
    END
    ELSE
    BEGIN
        INSERT INTO BANGDIEM (MASV, MAHP, DIEMTHI)
        VALUES (@MASV, @MAHP, @ENCRYPTED_DIEM);
        PRINT 'Them diem thanh cong!';
    END
END;
GO

```

TạoStored procedure SP_INS_PUBLIC_BANGDIEM

- SP_SEL_PUBLIC_BANGDIEM: dùng để truy vấn dữ liệu trong table BANGDIEM

```

CREATE PROCEDURE SP_SEL_PUBLIC_BANGDIEM
    (@MANV VARCHAR(20), -- Nhân viên đăng nhập
     @PK NVARCHAR(100) -- Mật khẩu của nhân viên để mở khóa giải mã
    )
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @PUBKEY_NAME VARCHAR(20)

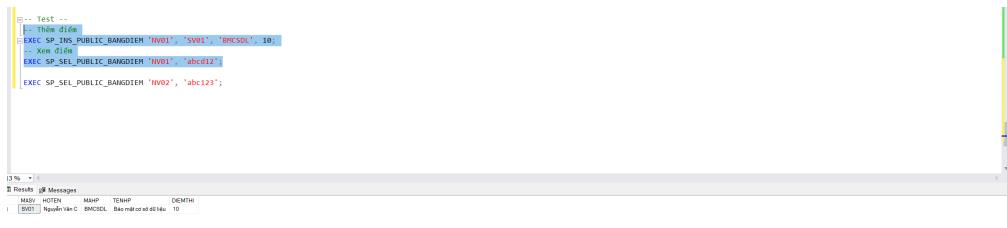
    -- Lấy Public Key của nhân viên
    SELECT @PUBKEY_NAME = PUBKEY FROM NHANVIEN WHERE MANV = @MANV;

    -- Kiểm tra nếu không có Public Key
    IF @PUBKEY_NAME IS NULL
    BEGIN
        PRINT 'Loi: khong tim thay khieu RSA cua nhan vien!';
        RETURN;
    END

    -- Truy vấn điểm thi và giải mã
    PRINT 'Danh gia ma diem...';
    SELECT 
        B.MASV, S.NOTHEN, B.MAHP, H.TENHP,
        CONVERT(NVARCHAR(MAX), DecryptByAsymKey(AsymKey_ID(@PUBKEY_NAME), B.DIEMTHI, @PK)) AS DIEMTHI
    FROM BANGDIEM B
    JOIN SINHVIEN S ON B.MASV = S.MASV
    JOIN NHANVIEN H ON B.MAHP = H.MAHP
    WHERE B.DIEMTHI IS NOT NULL;
END;
GO

```

TạoStored procedure SP_SEL_PUBLIC_BANGDIEM



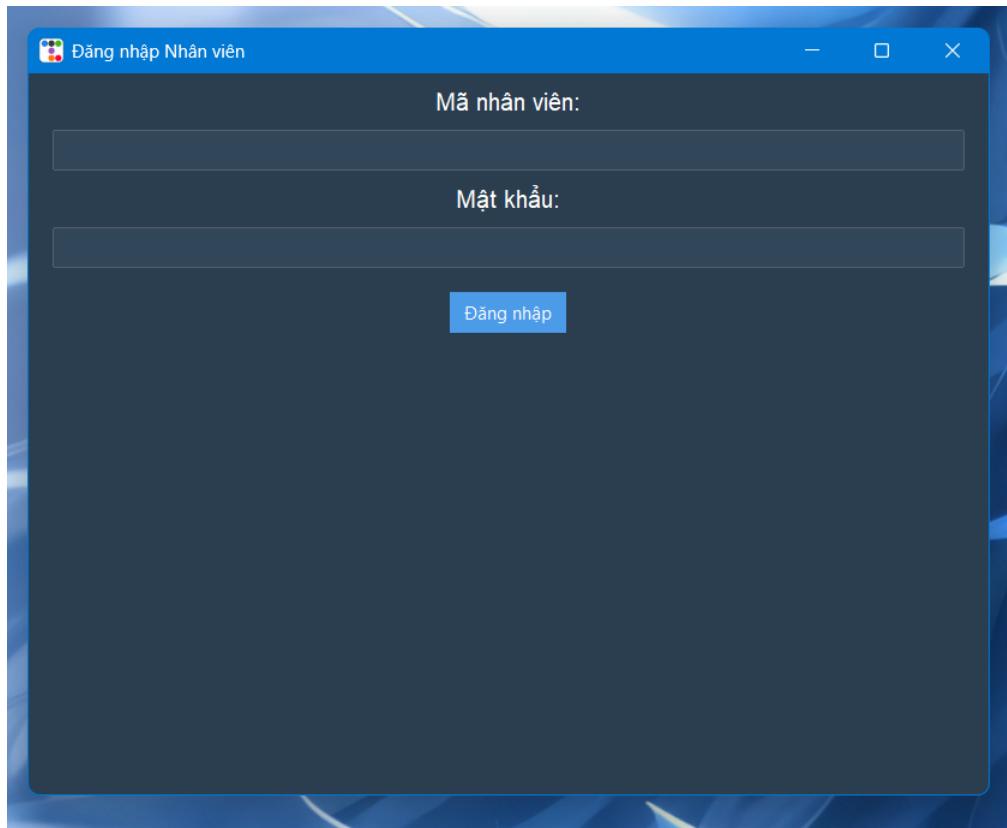
```
-- Test --
EXEC SP_INS_PUBLIC_BANGDIEM 'W001', 'SW01', 'IPRESD1', 10;
-- Xem điểm
EXEC SP_SEL_PUBLIC_BANGDIEM 'W001', 'abc123';
EXEC SP_SEL_PUBLIC_BANGDIEM 'W002', 'abc123';

Results Messages Task List Object Explorer
W001 W001 IPRESD1 10
W001 abc123
W002 abc123
```

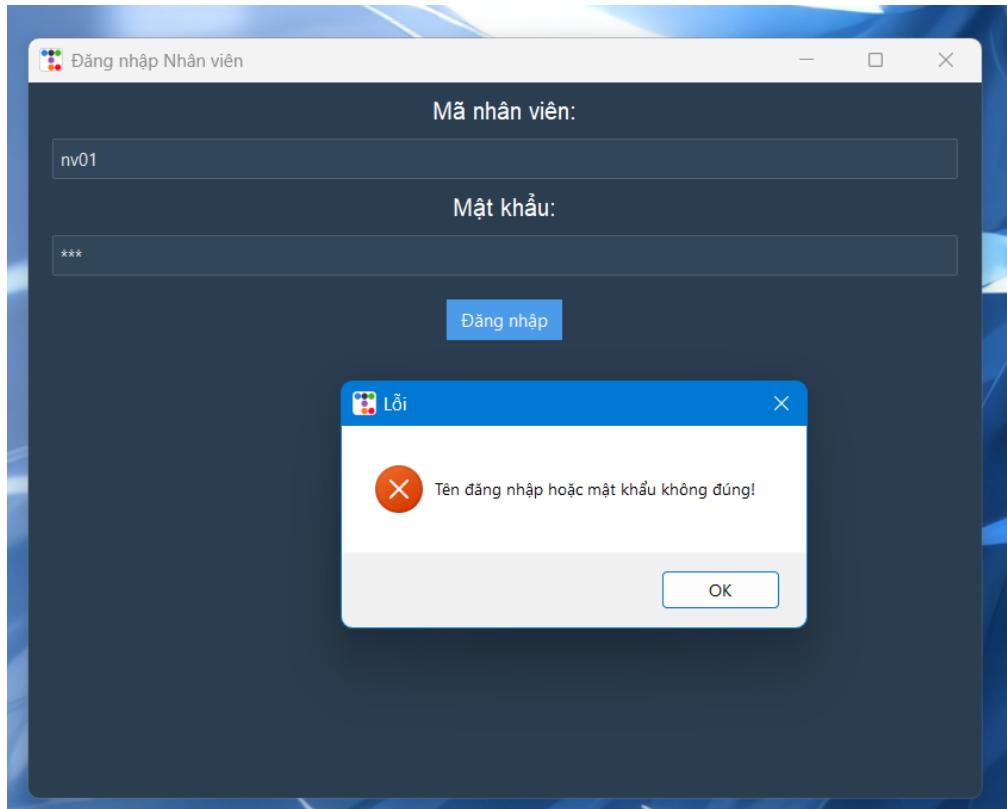
Kết quả thêm và xem điểm

3.4.2 Chương trình Python

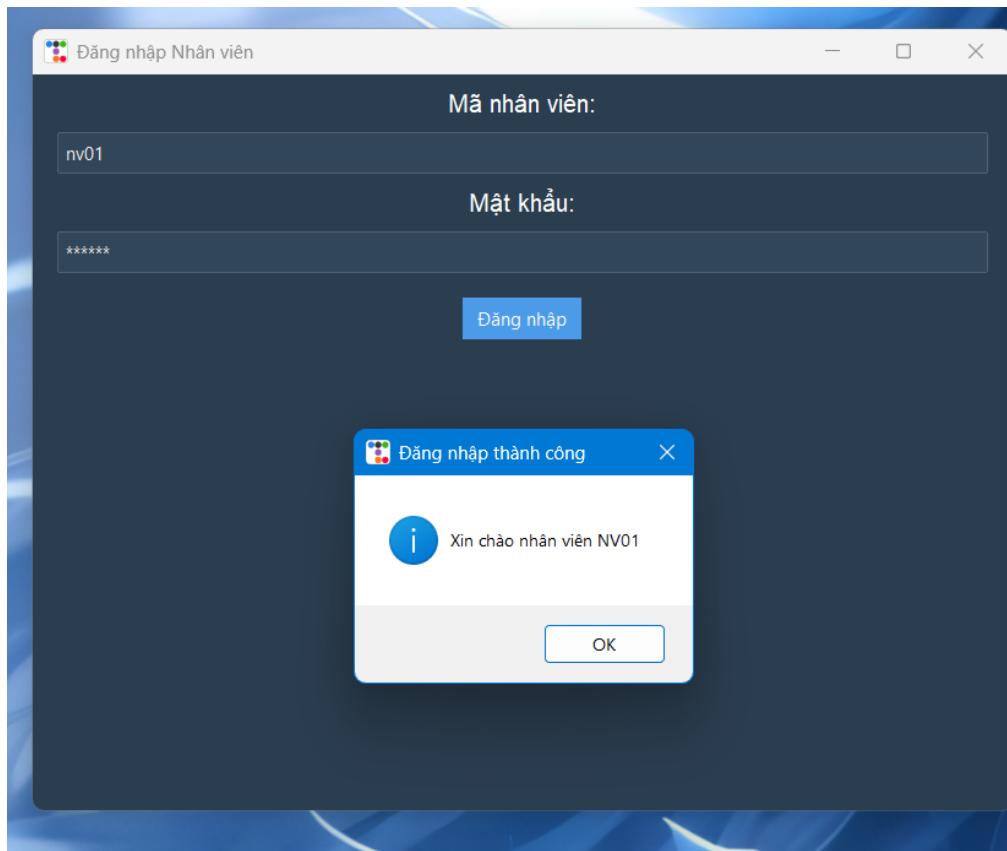
- **Giao diện đăng nhập:** Nhân viên đăng nhập bằng (Mã nhân viên, mật khẩu) đã tạo



Giao diện đăng nhập



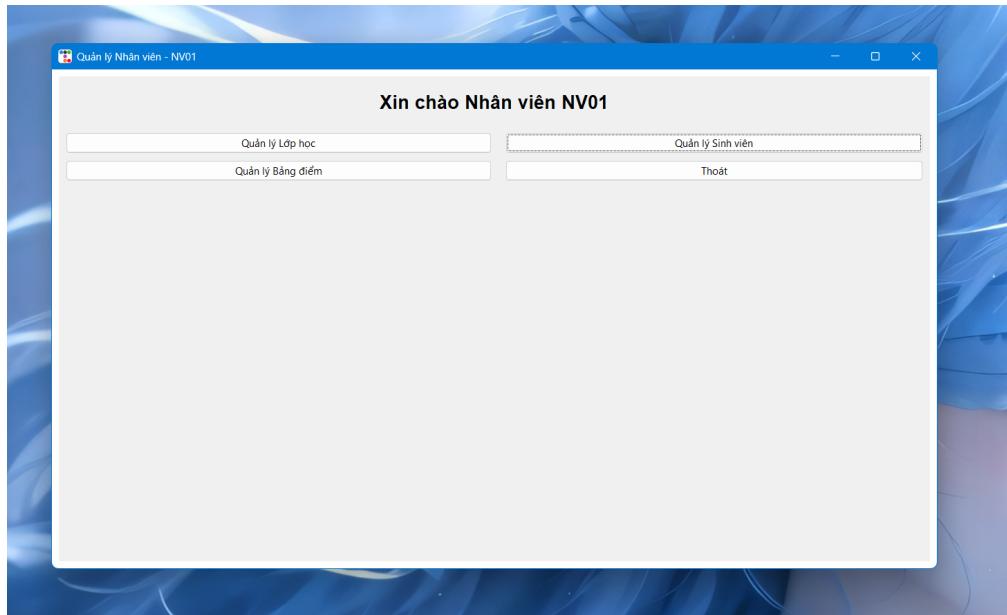
Thông báo khi đăng nhập thất bại



Thông báo khi đăng nhập thành công

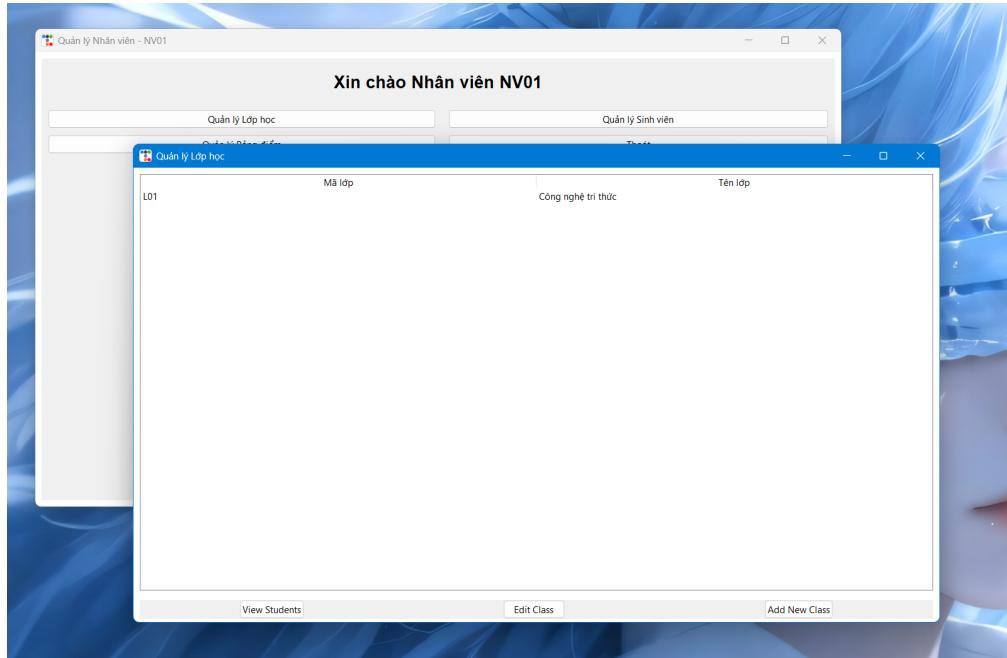
- **Giao diện chính:** hiển thị các tác vụ của nhân viên như:

1. Quản lý lớp học (Xem hoặc chỉnh sửa thông tin lớp học)
2. Quản lý sinh viên (xem hoặc chỉnh sửa thông tin sinh viên)
3. Quản lý bảng điểm (nhập điểm của sinh viên)
4. Thoát: Tắt chương trình



Giao diện chính

- **Giao diện quản lý lớp học:** Hiển thị danh sách các lớp do Nhân viên (đã đăng nhập quản lý) và các button thực hiện các chức năng

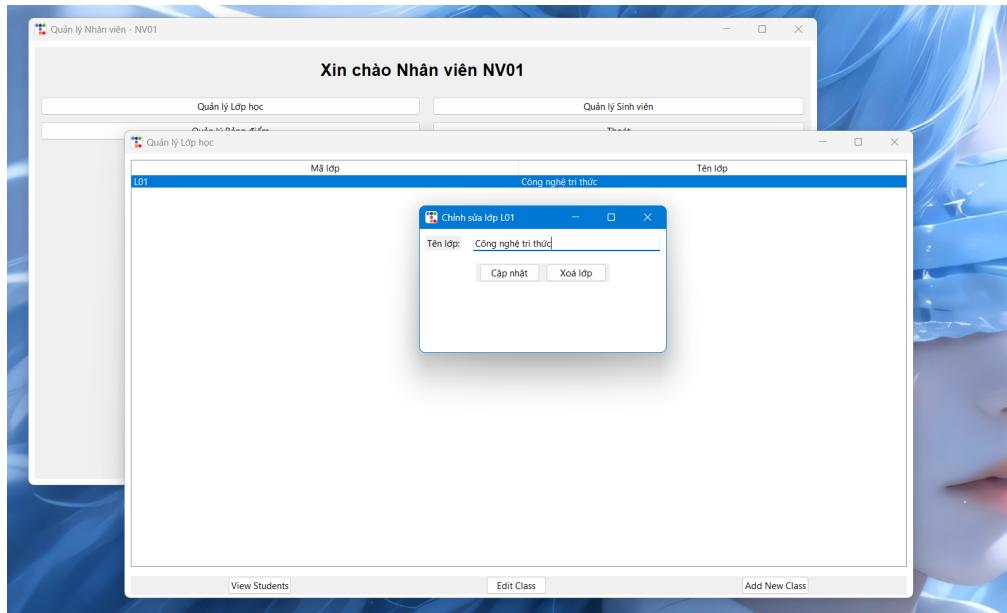


Giao diện quản lý lớp học

Giao diện bao gồm 3 nút chức năng:

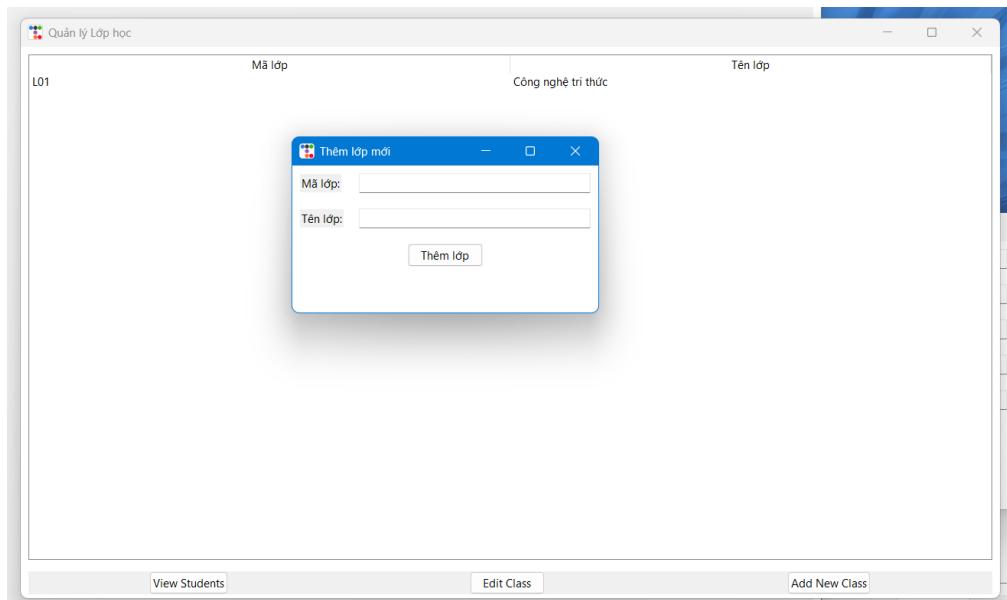
1. "View Students": Cho phép xem danh sách lớp được chọn

2. Edit Class: Chính sửa thông tin của lớp học



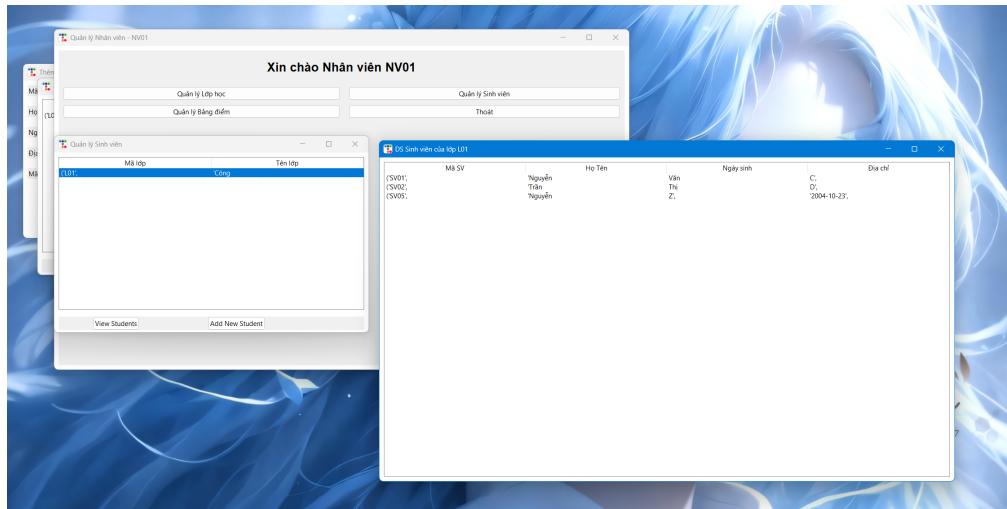
Giao diện chỉnh sửa thông tin lớp học

3. "Add new class": Thêm một lớp học mới (mặc định mã nhân viên đang đăng nhập là nhân viên quản lý)



Giao diện thêm lớp học mới

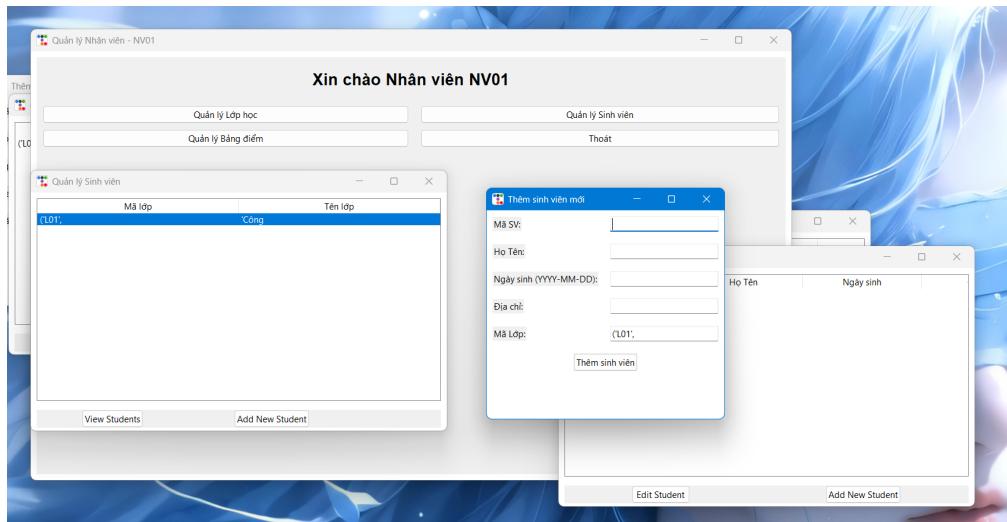
- **Giao diện quản lý sinh viên:** Hiển thị danh sách các sinh viên do Nhân viên quản lý và các button thực hiện các chức năng



Giao diện quản lý sinh viên

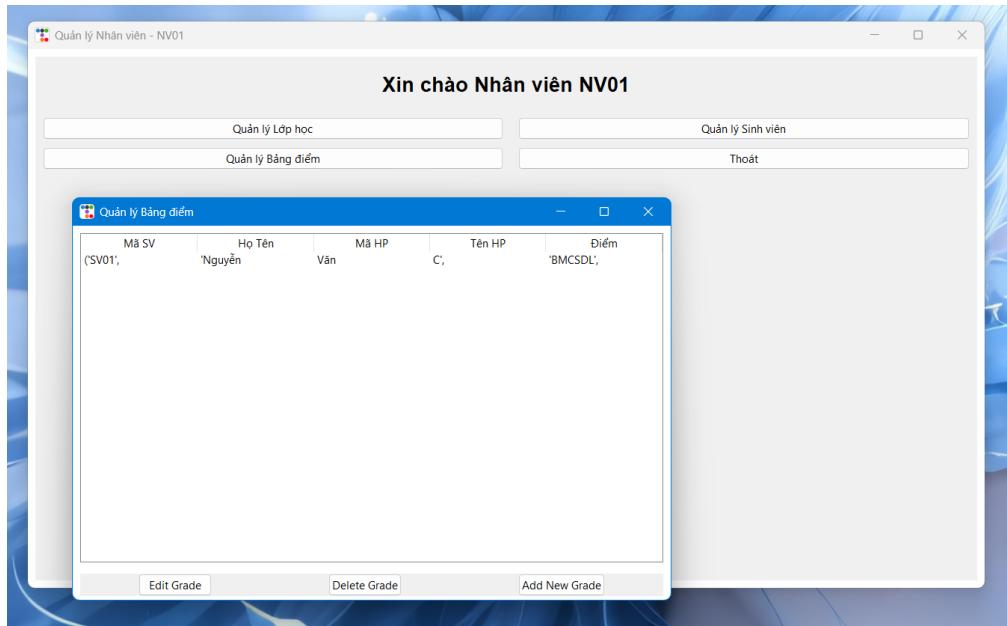
Giao diện bao gồm 2 nút chức năng:

- "View Students": Cho phép xem danh sách sinh viên của lớp được chọn
- "Add New Student": Thêm sinh viên mới



Giao diện thêm sinh viên mới

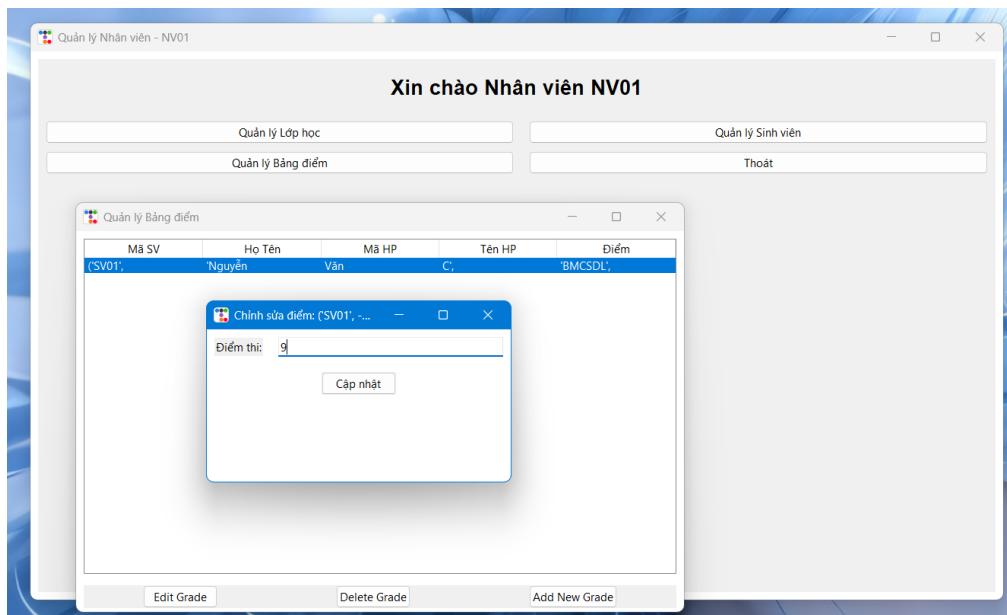
- Giao diện quản lý điểm: Hiển thị danh sách điểm của sinh viên do Nhân viên đã nhập và các button thực hiện các chức năng



Giao diện quản lý bảng điểm

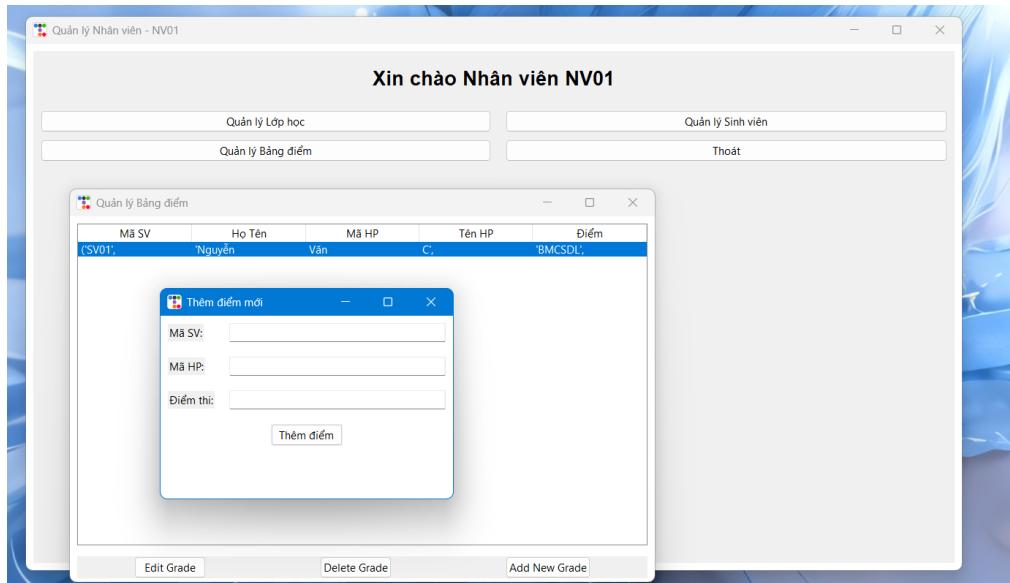
Giao diện bao gồm 3 nút chức năng:

1. "Edit Grade": Cho phép sửa điểm đã tạo



Giao diện chỉnh sửa điểm

2. "Delete Grade": Xóa điểm đã tạo
3. "Add New Grade": Thêm điểm mới



Giao diện thêm điểm mới

3.5 Câu e: Sử dụng công cụ SQL Profiler

3.5.1 Tạo phiên giám sát

1. Mở SQL Server Management Studio (SSMS)

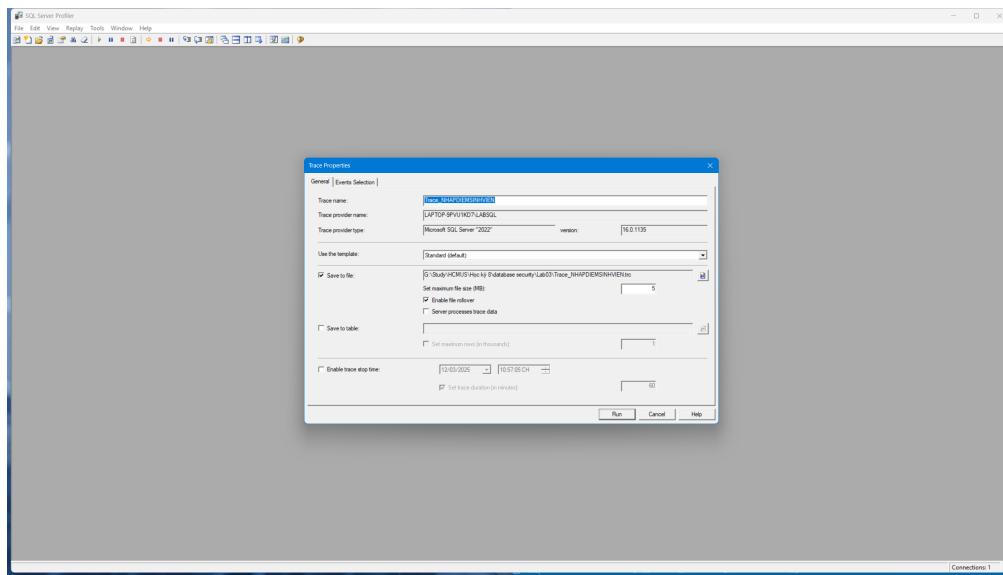
2. Chọn mục **Tools** trên thanh taskbar

3. Chọn **SQL Server Profiler**

4. Đăng nhập vào Server Database muốn giám sát

5. Tạo phiên giám sát mới:

- Trace name: Tên phiên giám sát
- Use the template: Chọn bản Standard (default)
- Chọn Save to file và chọn thư mục lưu trữ



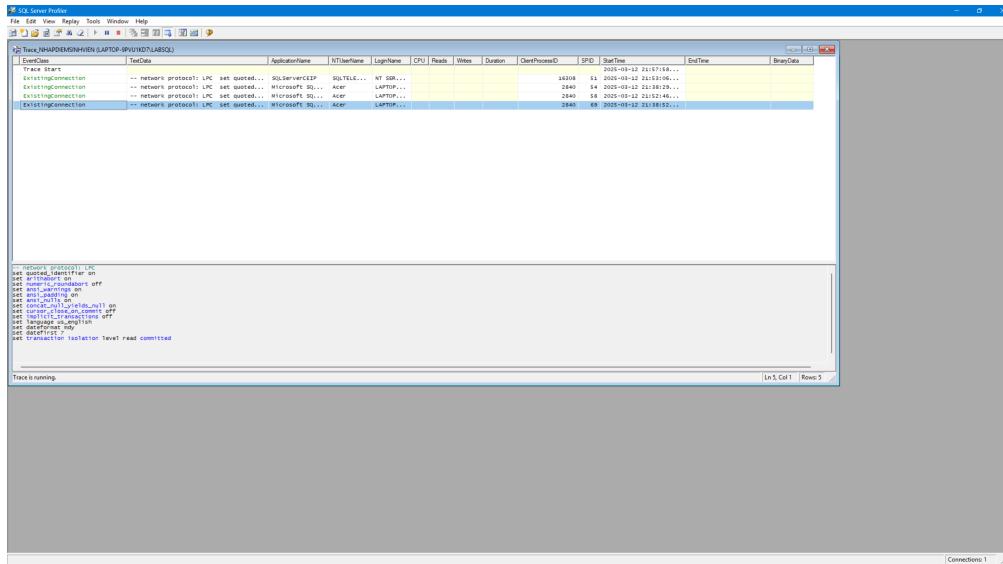
Tạo phiên giám sát mới

6. Kiểm tra mục Events Selection và đánh dấu các mục sau để quan sát:

- **TSQL-SQL:BatchCompleted:** Theo dõi câu lệnh SQL hoàn tất.
- **Stored Procedures-RPC:Completed:** Theo dõi stored procedure.
- **Errors and Warnings-User Error Message:** Ghi nhận lỗi nếu có

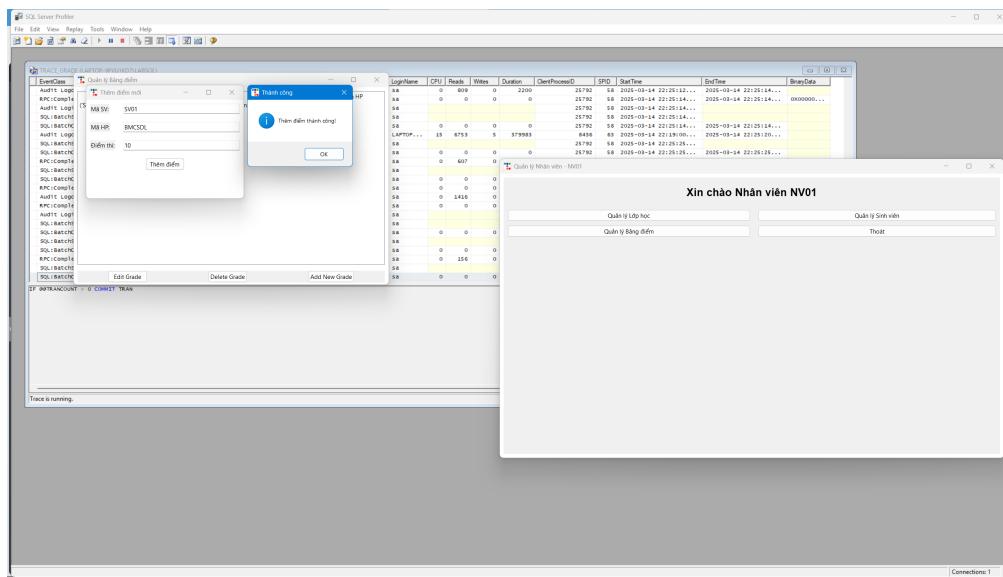
7. Sau đó nhấn **Run** để bắt đầu giám sát

8. Lúc này SQL Profile trả về màn hình theo dõi các thao tác trên SQL trong phiên giám sát.



Màn hình theo dõi của phiên giám sát

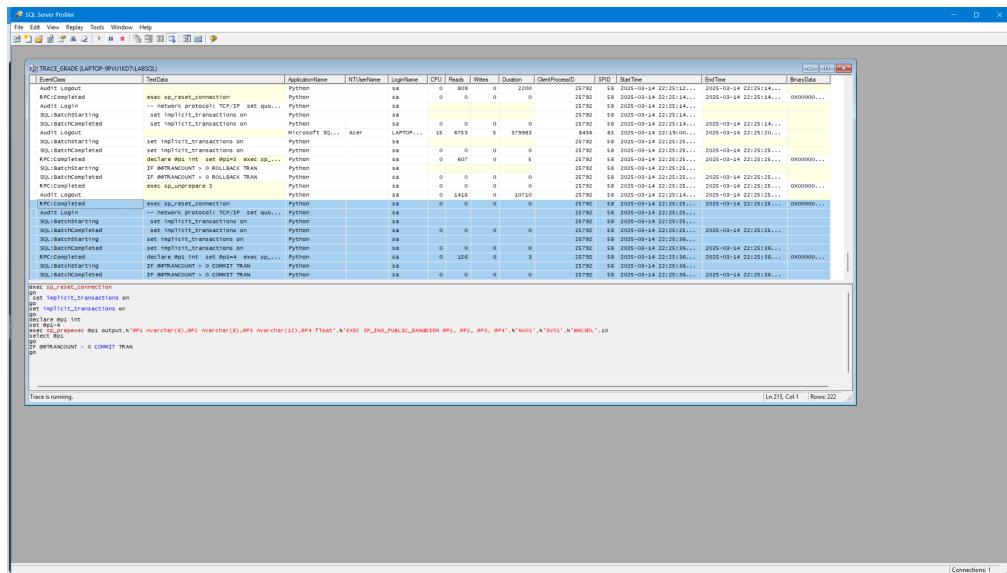
3.5.2 Giám sát thao tác nhập điểm sinh viên



Thêm điểm bằng chương trình

BÁO CÁO BÀI THỰC HÀNH SỐ 3

Trường Đại học Khoa học Tự nhiên
Bảo mật cơ sở dữ liệu



Kết quả giám sát của SQL Profiler

3.5.3 Nhận xét

SQL Profiler đã ghi nhận lại các truy vấn (event) liên quan đến việc thêm dữ liệu vào bảng điểm:

- Kết nối đến Server: sự kiện "Audit Login" cho biết chương trình python đã đăng nhập thông qua giao thức TCP/IP với tài khoản "sa"

Audit Login	-- network protocol: TCP/IP SET QUO...	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
RPC:Completed	declare @ol int set @ol=4 exec sp...	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	IF INTRANODE > 0 ROLLBACK TRAN	Python	sa	25792	\$ 2025-03-14 22:25:25...
RPC:Completed	exec sp_unguard 3	Python	sa	25792	\$ 2025-03-14 22:25:25...
RPC:Completed	exec sp_unguard 3	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	exec sp_loseconnection	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	exec sp_loseconnection	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
RPC:Completed	declare @ol int set @ol=4 exec sp...	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	IF INTRANODE > 0 ROLLBACK TRAN	Python	sa	25792	\$ 2025-03-14 22:25:25...
RPC:Completed	exec sp_unguard 3	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	exec sp_loseconnection	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	exec sp_loseconnection	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
RPC:Completed	exec sp_reset_connection	SQLServerCEIP	SQLTELE... NT SER...	22600	\$ 1 2025-03-14 22:24:06...
RPC:Completed	exec sp_reset_connection	SQLServerCEIP	SQLTELE... NT SER...	22600	\$ 1 2025-03-14 22:24:06...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQL:BatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...

Login server

- Thực thi các truy vấn: các sự kiện "SQL:Batch..." cho thấy việc xử lý các giao dịch (transactions)

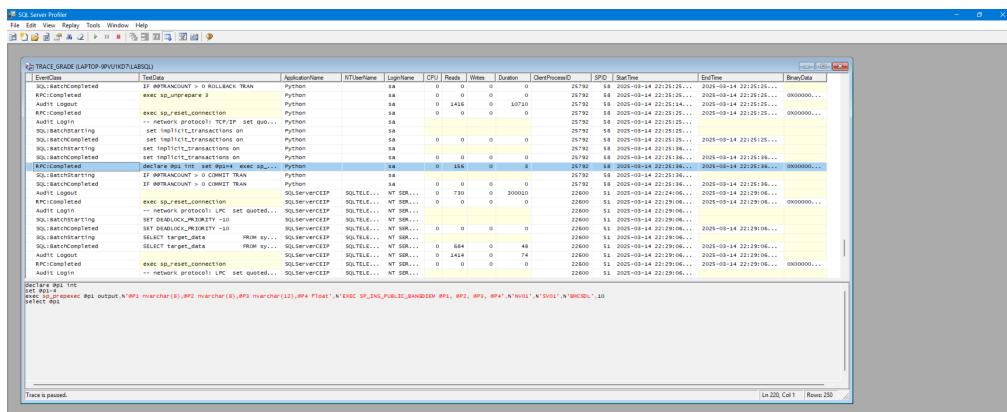
SQlBatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQlBatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQlBatchStarting	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...
SQlBatchCompleted	set implicit_transactions on	Python	sa	25792	\$ 2025-03-14 22:25:25...

Xử lý transactions

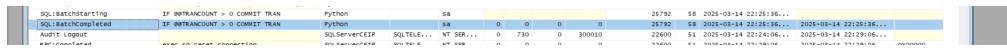
- Xử lý việc nhập điểm: sự kiện "RPC" (Remote Procedure Call) cùng trạng thái "Completed" cho biết việc thực thi stored procedure SP_INS_PUBLIC_BANGDIEM để thêm điểm đã

BÁO CÁO BÀI THỰC HÀNH SỐ 3

Trường Đại học Khoa học Tự nhiên
Bảo mật cơ sở dữ liệu



Kết quả thực thi stored procedure



Các event xử lý việc logout

thành công!

- Và sau đó là các lệnh liên quan để audit và transactions để logout

TÀI LIỆU THAM KHẢO

- [1] CREATE ASYMMETRIC KEY (Transact-SQL) - SQL Server | Microsoft Learn
- [2] OPEN SYMMETRIC KEY (Transact-SQL) - SQL Server | Microsoft Learn
- [3] Fix Msg 8116 “Argument data type date is invalid for argument 1 of substring function” in SQL Server
- [4] sachadee/Dlib: Dlib compiled binary (.whl) for python 3.7, 3.8, 3.9 for windows x64