

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

BỘ MÔN MẠNG MÁY TÍNH VÀ VIỄN THÔNG HƯỚNG AN TOÀN THÔNG TIN

---

## Báo cáo Lab 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:*

Lưu Thành Đạt (22127063)

Mai Xuân Thường (22127409)

*Giáo viên hướng dẫn:*

Ths. Nguyễn Thị Hường

Ngày 27 tháng 3 năm 2025



## Mục lục

<b>1 Phân công</b>	<b>1</b>
<b>2 Yêu cầu</b>	<b>1</b>
<b>3 Viết các stored procedure</b>	<b>2</b>
<b>4 Viết các stored procedure và chương trình</b>	<b>3</b>
<b>5 Sử dụng công cụ SQL Profile để theo dõi thao tác trong màn hình nhập điểm sinh viên</b>	<b>10</b>

# 1 Phân công

MSSV	Họ tên	Công việc	Mức độ hoàn thành
22127063	Lưu Thành Đạt	Câu c) và script câu d) Kiểm tra và chỉnh sửa script, code Viết báo cáo	100%
22127409	Mai Xuân Thường	Câu e) và code d) Viết báo cáo Demo code	100%

Bảng 1: Bảng phân công công việc

# 2 Yêu cầu

1. Sử dụng lại CSDL đã được cung cấp trong Lab 03.

2. Viết các Stored Procedure sau:

(a) Stored Procedure dùng để thêm mới dữ liệu (**Insert**) vào bảng SINHVIEN, trong đó:

- Thuộc tính MATKHAU được mã hóa (**HASH**) sử dụng thuật toán **SHA1**.
- Thuộc tính LUONG sẽ được mã hóa từ tham số LUONGCB sử dụng thuật toán **RSA 2048**, với khóa bí mật là tham số MK được truyền vào.
- Thuộc tính PUBKEY sẽ lưu trữ tên khóa công khai được tạo ra ứng với nhân viên này. Giá trị này sẽ lưu thông tin khóa công khai được tạo từ client.

(b) Stored Procedure dùng để truy vấn dữ liệu nhân viên (**NHANVIEN**).

3. Viết các Stored Procedure và chương trình để thực hiện các yêu cầu sau:

- Xây dựng **màn hình quản lý đăng nhập** như trong bài lab dành cho cá nhân và xử lý đăng nhập với tài khoản là nhân viên (MANV, MATKHAU).
- Xây dựng **màn hình quản lý nhân viên**.
- Xây dựng **màn hình quản lý lớp học**.
- Xây dựng **màn hình sinh viên của từng lớp**, lưu ý chỉ được phép thay đổi thông tin của những sinh viên thuộc lớp mà nhân viên đó quản lý.

- Xây dựng **màn hình nhập bảng điểm** của từng sinh viên, trong đó:
    - Cột điểm thi sẽ được mã hóa bằng chính **Public Key** của nhân viên (đã đăng nhập).

4. **Lưu ý:** Tất cả các chức năng mã hóa và giải mã đều được thực hiện ở phía client, nghĩa là:

- Mã hóa dữ liệu từ client trước khi lưu xuống CSDL.
  - Giải mã dữ liệu ở client sau khi truy vấn dữ liệu từ CSDL.

5. Sử dụng công cụ SQL Profiler để theo dõi thao tác trong màn hình nhập điểm sinh viên và đưa ra nhận xét.

### 3 Viết các stored procedure

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'ADMIN.QLSVNhom (sa (60))'. The main window displays a script for creating a stored procedure named SP\_INS\_PUBLIC\_ENCRYPT\_NHANVIEN. The script includes logic to check if MANV and TENDN already exist in the NHANVIEN table, and then inserts new data into the table. The execution results pane at the bottom shows a single row of data inserted into the NHANVIEN table.

```
-- Stored dùng để thêm mới dữ liệu (Insert) vào table NHANVIEN
CREATE OR ALTER PROCEDURE SP_INS_PUBLIC_ENCRYPT_NHANVIEN
    @MANV NVARCHAR(10),
    @HOTEN NVARCHAR(100),
    @EMAIL NVARCHAR(100),
    @LUONG NVARCHAR(MAX), -- Lương đã được mã hóa RSA từ client
    @TENDN NVARCHAR(50),
    @MK NVARCHAR(100), -- Mật khẩu đã được mã hóa SHA1 từ client
    @PUB NVARCHAR(MAX) -- Khóa công khai từ client
AS
BEGIN
    -- Kiểm tra trùng lặp MANV
    IF EXISTS (SELECT 1 FROM NHANVIEN WHERE MANV = @MANV)
    BEGIN
        RAISERROR('Mã nhân viên đã tồn tại.', 16, 1);
        RETURN;
    END

    -- Kiểm tra trùng lặp TENDN
    IF EXISTS (SELECT 1 FROM NHANVIEN WHERE TENDN = @TENDN)
    BEGIN
        RAISERROR('Tên đăng nhập đã tồn tại.', 16, 1);
        RETURN;
    END

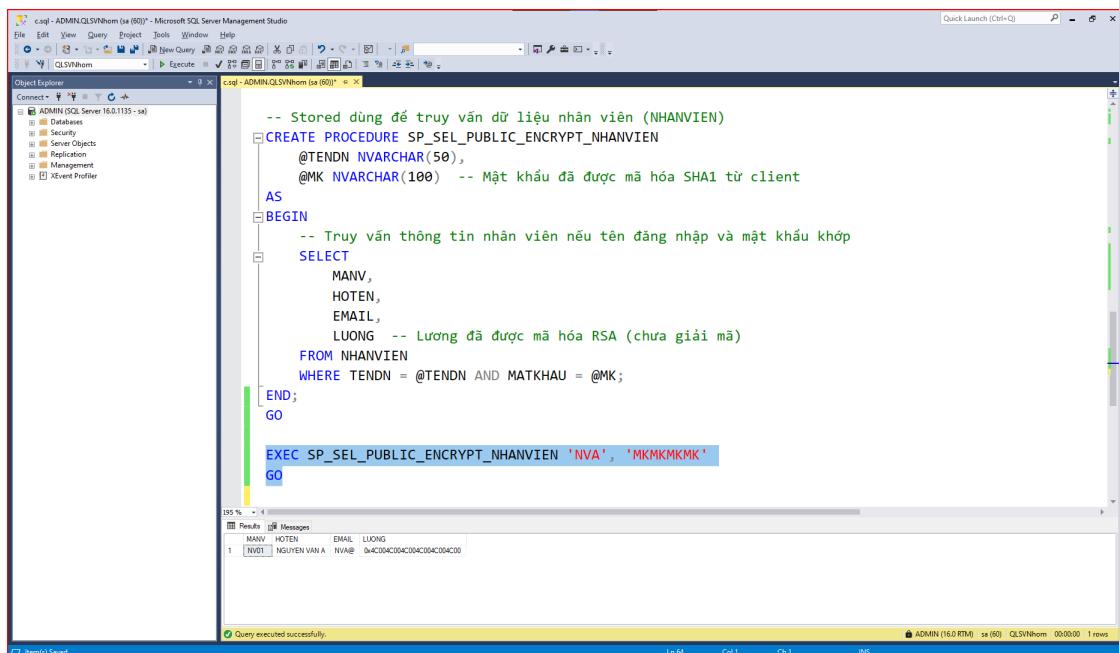
    -- Thêm dữ liệu vào bảng NHANVIEN
    INSERT INTO NHANVIEN (MANV, HOTEN, EMAIL, LUONG, TENDN, MATKHAU, PUBKEY)
    VALUES (@MANV, @HOTEN, @EMAIL, CONVERT(VARBINARY(MAX), @LUONG), @TENDN, CONVERT(VARBINARY(MAX), @MK), @PUB);

    PRINT N'Dữ liệu đã được thêm thành công.';
END;
```

	MANV	HOTEN	EMAIL	LUONG
1	NV01	NGUYEN VAN A	HVAB	0x4C00AC004C004C004C004C00

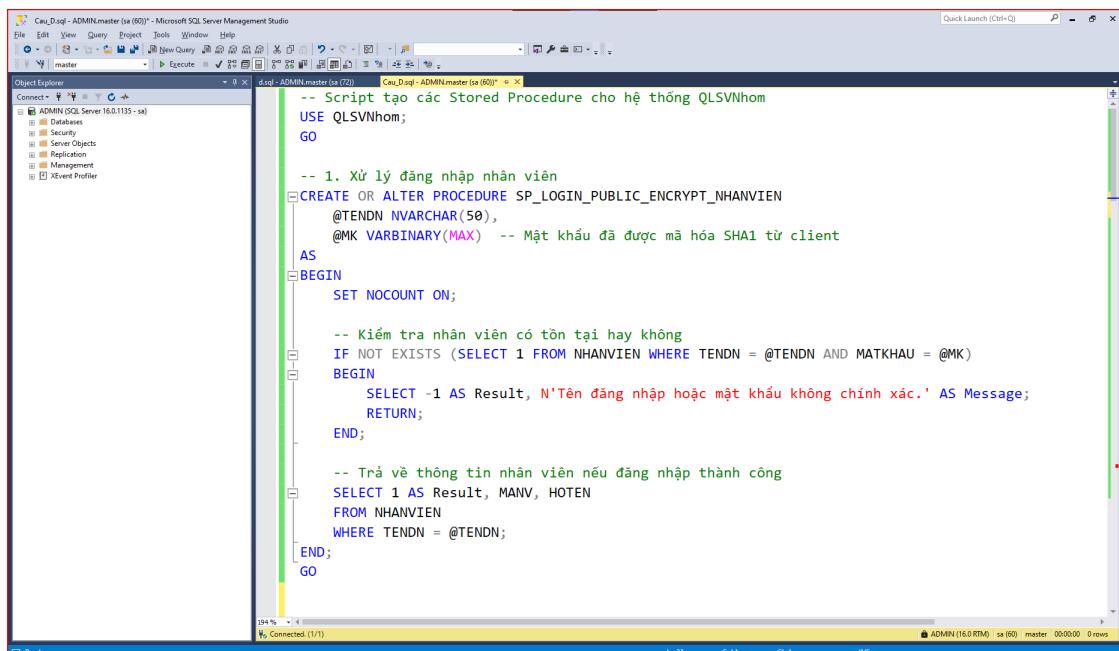
Query executed successfully.

Hình 1: Stored procedure thêm mới nhân viên



Hình 2: Stored procedure truy vấn thông tin nhân viên và kết quả khi thực hiện

#### 4 Viết các stored procedure và chương trình



Hình 3: Stored procedure xử lí đăng nhập

## Báo cáo Lab 3

```
-- thêm lớp mới
CREATE OR ALTER PROCEDURE SP_INS_LOP
    @MALOP VARCHAR(20),
    @TENLOP NVARCHAR(100),
    @MANV VARCHAR(20) -- Nhân viên quản lý lớp
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra trùng mã lớp
    IF EXISTS (SELECT 1 FROM LOP WHERE MALOP = @MALOP)
    BEGIN
        SELECT -2 AS Result; -- Trả về lỗi
        RETURN;
    END

    -- Thêm lớp mới
    INSERT INTO LOP (MALOP, TENLOP, MANV)
    VALUES (@MALOP, @TENLOP, @MANV);

    SELECT 1 AS Result; -- Trả về thành công
END;
GO
```

Hình 4: Stored procedure thêm lớp học

```
-- update lớp học
CREATE OR ALTER PROCEDURE SP_UPDATE_LOP
    @MALOP VARCHAR(20),
    @TENLOP NVARCHAR(100),
    @MANV VARCHAR(20) -- Nhân viên thực hiện cập nhật
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra lớp tồn tại và nhân viên có quyền cập nhật không
    IF NOT EXISTS (SELECT 1 FROM LOP WHERE MALOP = @MALOP AND MANV = @MANV)
    BEGIN
        SELECT -1 AS Result; -- Lớp không tồn tại hoặc không có quyền cập nhật
        RETURN;
    END

    -- Cập nhật thông tin lớp
    UPDATE LOP
    SET TENLOP = @TENLOP
    WHERE MALOP = @MALOP AND MANV = @MANV;

    SELECT 1 AS Result; -- Cập nhật thành công
END;
GO
```

Hình 5: Stored procedure cập nhật lớp học

## Báo cáo Lab 3

```
-- xoá lớp học
CREATE OR ALTER PROCEDURE SP_DELETE_LOP
    @MALOP VARCHAR(20),
    @MANV VARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra nếu lớp không tồn tại hoặc không thuộc quyền quản lý
    IF NOT EXISTS (SELECT 1 FROM LOP WHERE MALOP = @MALOP AND MANV = @MANV)
    BEGIN
        SELECT -1 AS Result; -- Trả về giá trị có thể đọc từ Python
        RETURN;
    END

    -- Xóa lớp
    DELETE FROM LOP WHERE MALOP = @MALOP;
    SELECT 1 AS Result; -- Trả về giá trị thành công
END;
GO
```

Hình 6: Stored procedure xoá lớp học

```
-- xem danh sách lớp
CREATE OR ALTER PROCEDURE SP_SEL_LOP
AS
BEGIN
    SET NOCOUNT ON;

    -- Lấy danh sách lớp
    SELECT MALOP, TENLOP, MANV FROM LOP;
END;
GO

-- Lấy danh sách sinh viên trong lớp do nhân viên quản lý
CREATE OR ALTER PROCEDURE SP_SEL_SINHVIEN_LOP
    @MANV VARCHAR(20) -- Mã nhân viên đăng nhập
AS
BEGIN
    SET NOCOUNT ON;

    -- Chỉ lấy sinh viên thuộc lớp do nhân viên quản lý
    SELECT SV.MASV, SV.HOTEN, SV.NGAYSINH, SV.DIACHI, L.TENLOP
    FROM SINHVIEN SV
    JOIN LOP L ON SV.MALOP = L.MALOP
    WHERE L.MANV = @MANV;
END;
GO
```

Hình 7: Stored procedure lấy danh sách lớp học và sinh viên

## Báo cáo Lab 3

```

dsql - ADMIN.master (sa (8)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
master | Execute | New Query | Object Explorer | Task List | Results | Grid | Text | Script | Design | Properties | Home | Back | Forward | Refresh | Stop | Close | Quick Launch (Ctrl+Q) | P | x
Object Explorer
Connect ▾ master
ADMIN (SQL Server 16.0.1135 - sa)
Database Security
Server Objects
Replication Management XEvent Profiler
dsql - ADMIN.master (sa (8)) * >
-- Thêm sinh viên vào lớp
CREATE OR ALTER PROCEDURE SP_INS_SINHVIEN
    @MASV NVARCHAR(20),
    @HOTEN NVARCHAR(100),
    @NGAYSINH DATETIME,
    @DIACHI NVARCHAR(200),
    @MALOP NVARCHAR(20),
    @MANV VARCHAR(20) -- Kiểm tra quyền thêm vào lớp
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @HASHED_PASSWORD VARBINARY(20);
    SET @HASHED_PASSWORD = HASHBYTES('SHA2', 'default');

    DECLARE @TENDN NVARCHAR(100);
    SET @TENDN = @MANV;

    -- Kiểm tra quyền quản lý lớp
    IF NOT EXISTS (SELECT 1 FROM LOP WHERE MALOP = @MALOP AND MANV = @MANV)
    BEGIN
        SELECT -1 AS Result; -- Không có quyền hoặc lớp không tồn tại
        RETURN;
    END

    -- Kiểm tra xem sinh viên đã tồn tại chưa
    IF EXISTS (SELECT 1 FROM SINHVIEN WHERE MASV = @MASV)
    BEGIN
        SELECT -2 AS Result; -- Sinh viên đã tồn tại
        RETURN;
    END

    -- Thêm sinh viên mới
    INSERT INTO SINHVIEN (MASV, HOTEN, NGAYSINH, DIACHI, MALOP, TENDN, MATKHAU)
    VALUES (@MASV, @HOTEN, @NGAYSINH, @DIACHI, @MALOP, @TENDN, @HASHED_PASSWORD);

    SELECT 1 AS Result; -- Thành công
END;
GO

```

122 % 1/1 Connected. (1/1) ADMIN (16.0 RTM) | sa (8) | master | 00:00:00 | 0 rows

Hình 8: Stored procedure thêm sinh viên

```

dsql - ADMIN.master (sa (8)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
master | Execute | New Query | Object Explorer | Task List | Results | Grid | Text | Script | Design | Properties | Home | Back | Forward | Refresh | Stop | Close | Quick Launch (Ctrl+Q) | P | x
Object Explorer
Connect ▾ master
ADMIN (SQL Server 16.0.1135 - sa)
Database Security
Server Objects
Replication Management XEvent Profiler
dsql - ADMIN.master (sa (8)) * >
-- Cập nhật thông tin sinh viên
CREATE OR ALTER PROCEDURE SP_UPDATE_SINHVIEN
    @MASV NVARCHAR(20),
    @HOTEN NVARCHAR(100),
    @NGAYSINH DATETIME,
    @DIACHI NVARCHAR(200),
    @MANV VARCHAR(20) -- Kiểm tra quyền
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra quyền
    IF NOT EXISTS (
        SELECT 1 FROM SINHVIEN SV
        JOIN LOP L ON SV.MALOP = L.MALOP
        WHERE SV.MASV = @MASV AND L.MANV = @MANV
    )
    BEGIN
        SELECT -1 AS Result;
        RETURN;
    END

    -- Cập nhật thông tin
    UPDATE SINHVIEN
    SET HOTEN = @HOTEN, NGAYSINH = @NGAYSINH, DIACHI = @DIACHI
    WHERE MASV = @MASV;

    SELECT 1 AS Result;
END;
GO

```

163 % 1/1 Connected. (1/1) ADMIN (16.0 RTM) | sa (8) | master | 00:00:00 | 0 rows

Hình 9: Stored procedure cập nhật sinh viên

## Báo cáo Lab 3

```

-- Xóa sinh viên
CREATE OR ALTER PROCEDURE SP_DELETE_SINHVIEN
    @MASV NVARCHAR(20),
    @MANV VARCHAR(20) -- Kiểm tra quyền
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra quyền
    IF NOT EXISTS (
        SELECT 1 FROM SINHVIEN SV
        JOIN LOP L ON SV.MALOP = L.MALOP
        WHERE SV.MASV = @MASV AND L.MANV = @MANV
    )
    BEGIN
        SELECT -1 AS Result;
        RETURN;
    END

    -- Xóa sinh viên
    DELETE FROM SINHVIEN WHERE MASV = @MASV;

    SELECT 1 AS Result;
END;
GO

```

Hình 10: Stored procedure xoá sinh viên

```

-- 5. Nhập điểm sinh viên với điểm thi đã được mã hóa
CREATE OR ALTER PROCEDURE SP_INS_PUBLIC_ENCRYPT_BANGDIEM
    @MASV NVARCHAR(20),
    @MAHP NVARCHAR(20),
    @DIENTHIDB VARBINARY(MAX)
AS
BEGIN
    SET NOCOUNT ON;

    -- Kiểm tra sinh viên có tồn tại không
    IF NOT EXISTS (SELECT 1 FROM SINHVIEN WHERE MASV = @MASV)
    BEGIN
        SELECT -1 AS Result;
        RETURN;
    END

    -- Kiểm tra học phần có tồn tại không
    IF NOT EXISTS (SELECT 1 FROM HOCPHAN WHERE MAHP = @MAHP)
    BEGIN
        SELECT -1 AS Result;
        RETURN;
    END

    -- Kiểm tra sinh viên đã có điểm chưa
    IF EXISTS (SELECT 1 FROM BANGDIEM WHERE MASV = @MASV AND MAHP = @MAHP)
    BEGIN
        SELECT -2 AS Result;
        RETURN;
    END

    -- Nhập điểm
    INSERT INTO BANGDIEM (MASV, MAHP, DIENTHIDB)
    VALUES (@MASV, @MAHP, @DIENTHIDB);

    -- Trả về kết quả thành công
    SELECT 1 AS Result;
END;
GO

```

Hình 11: Stored procedure thêm điểm đã mã hoá

## Báo cáo Lab 3

```
-- 6. Truy vấn điểm thi với điểm thi chưa giải mã
CREATE OR ALTER PROCEDURE SP_SEL_PUBLIC_ENCRYPT_BANGDIEM
    @MASV VARCHAR(20),
    @MAHP VARCHAR(20),
    @MANV VARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;

    -- Trả về điểm chưa giải mã
    IF EXISTS (SELECT 1 FROM BANGDIEM WHERE MASV = @MASV AND MAHP = @MAHP)
    BEGIN
        SELECT MASV, MAHP, DIEMTHI
        FROM BANGDIEM
        WHERE MASV = @MASV AND MAHP = @MAHP;
    END
    ELSE
    BEGIN
        SELECT NULL AS MASV, NULL AS MAHP, NULL AS DIEMTHI;
    END
END;
GO
```

Hình 12: Stored procedure truy xuất điểm chưa giải mã

```
-- 4. Quản lý nhân viên
CREATE OR ALTER PROCEDURE SP_SEL_NHANVIEN
AS
BEGIN
    SET NOCOUNT ON;

    -- Lấy danh sách lớp
    SELECT MANV, HOTEN, EMAIL, LUONG FROM NHANVIEN;
END;
GO
```

Hình 13: Stored procedure quản lí nhân viên

## Báo cáo Lab 3

```

def generate_rsa_keys(username: str, password: str):
    """Generate RSA keys for a specific user if they don't exist, otherwise load from cache."""
    load_keys() # Ensure cache is updated

    if username in KEY_CACHE:
        return KEY_CACHE[username]["private_key"], KEY_CACHE[username]["public_key"]

    # Generate new RSA keys
    private_key = RSA.generate(2048)
    private_pem = private_key.export_key(passphrase=password, pkcs=8)
    public_pem = private_key.publickey().export_key()

    # Encode keys to Base64 for JSON storage
    KEY_CACHE[username] = {
        "private_key": base64.b64encode(private_pem).decode(),
        "public_key": base64.b64encode(public_pem).decode()
    }

    save_keys()
    return private_pem, public_pem

```

Hình 14: Chương trình sinh khoá RSA

```

def save_keys():
    """Save the current cache to the JSON file."""
    with open(KEYS_FILE, "w") as f:
        json.dump(KEY_CACHE, f, indent=4)

```

Hình 15: Chương trình lưu khoá vào file

```

def load_keys():
    """Load all keys from the JSON file into cache."""
    if os.path.exists(KEYS_FILE):
        try:
            with open(KEYS_FILE, "r") as f:
                KEY_CACHE.update(json.load(f))
        except json.JSONDecodeError:
            print("🔴 Error: Corrupted key file, resetting...")
            with open(KEYS_FILE, "w") as f:
                json.dump({}, f) # Reset file
    else:
        with open(KEYS_FILE, "w") as f:
            json.dump({}, f) # Create empty JSON file

```

Hình 16: Chương trình truy xuất khoá

```

def load_private_key(username: str):
    """Load the private key for a specific user."""
    load_keys()

    if username in KEY_CACHE:
        private_pem = base64.b64decode(KEY_CACHE[username]["private_key"])
        return private_pem
    return None

```

Hình 17: Chương trình truy xuất khoá bí mật

```

def get_current_public_key(username: str):
    """Load the public key for a specific user."""
    load_keys()

    if username in KEY_CACHE:
        public_pem = base64.b64decode(KEY_CACHE[username]["public_key"])
        return public_pem
    return None

```

Hình 18: Chương trình truy xuất khoá công khai

## Báo cáo Lab 3

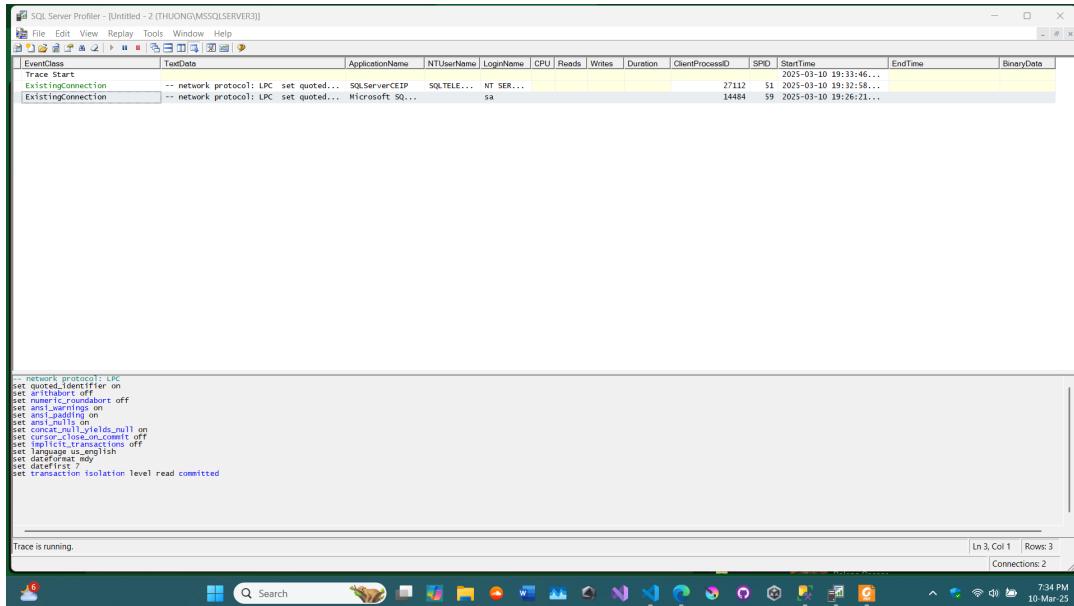
```
#  **Encrypt Data with Public Key**
def rsa_encrypt(data: str, public_key_pem: bytes):
    public_key = RSA.import_key(public_key_pem)
    cipher = PKCS1_OAEP.new(public_key)
    encrypted_data = cipher.encrypt(data.encode())
    return encrypted_data
```

Hình 19: Chương trình mã hoá bằng RSA

```
#  **Decrypt Data with Private Key**
def rsa_decrypt(encrypted_data: bytes, private_key_pem: bytes, password: str):
    private_key = RSA.import_key(private_key_pem, passphrase=password)
    cipher = PKCS1_OAEP.new(private_key)
    decrypted_data = cipher.decrypt(encrypted_data).decode()
    return decrypted_data
```

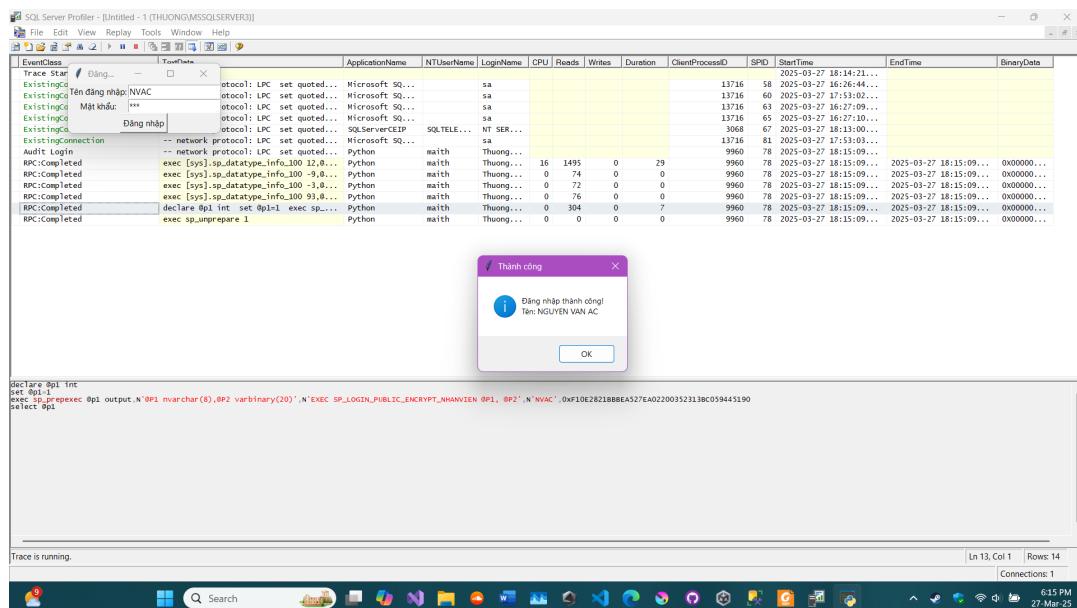
Hình 20: Chương trình giải mã

## 5 Sử dụng công cụ SQL Profiler để theo dõi thao tác trong màn hình nhập điểm sinh viên

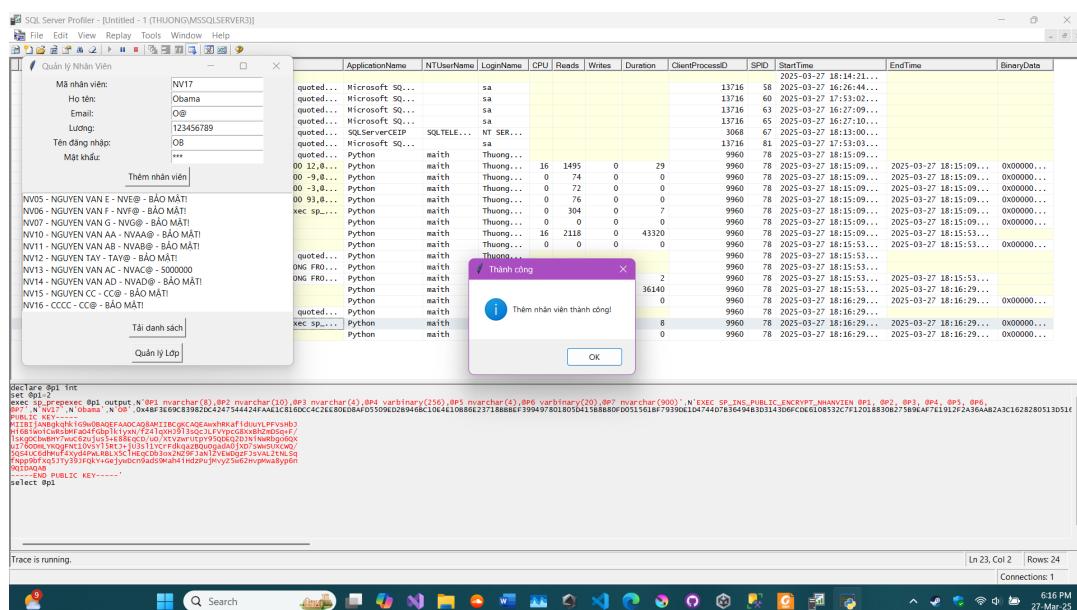


Hình 21: Vào thành công SQL Profiler

## Báo cáo Lab 3

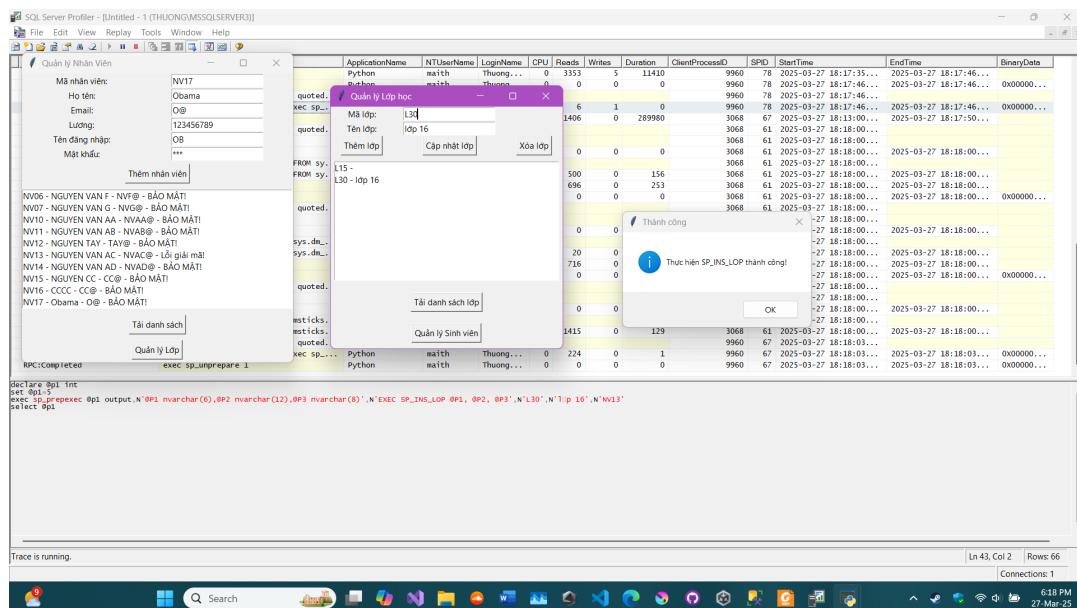


Hình 22: Đăng nhập

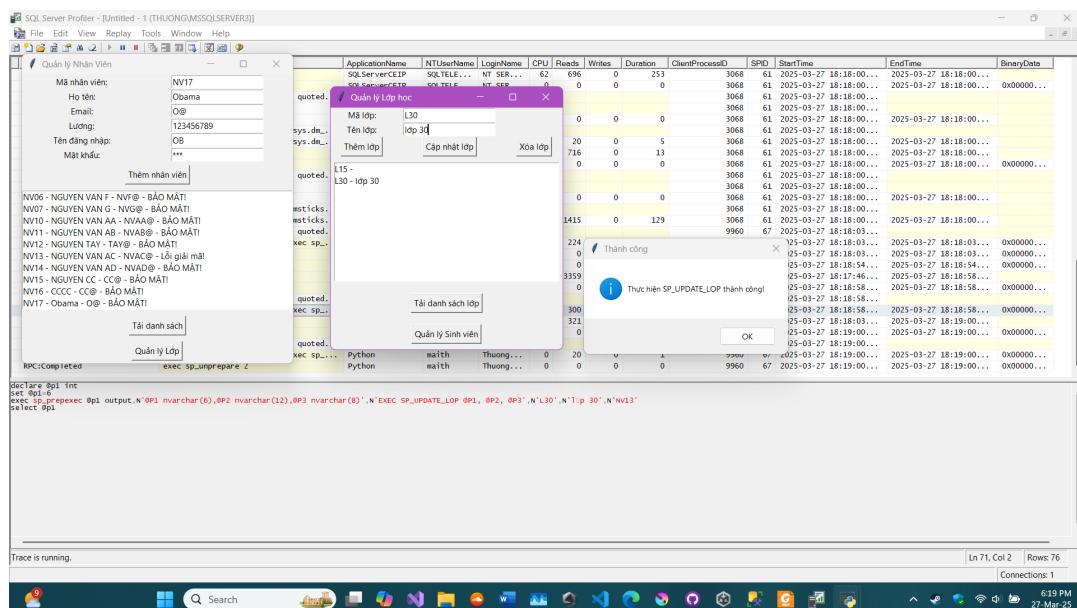


Hình 23: Thêm nhân viên

## Báo cáo Lab 3

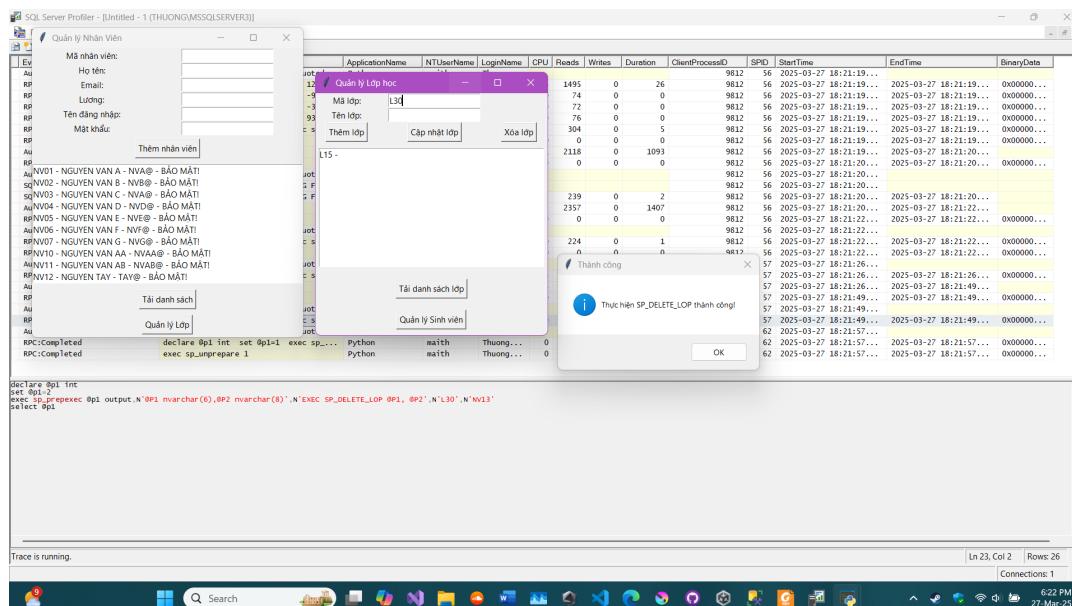


Hình 24: Thêm lớp

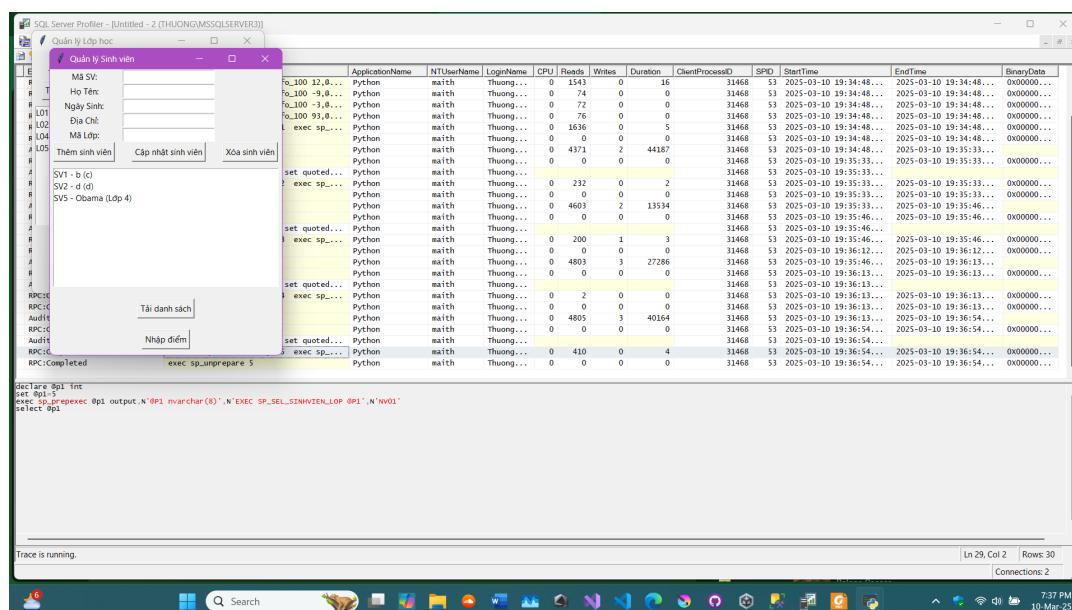


Hình 25: Cập nhật lớp

## Báo cáo Lab 3

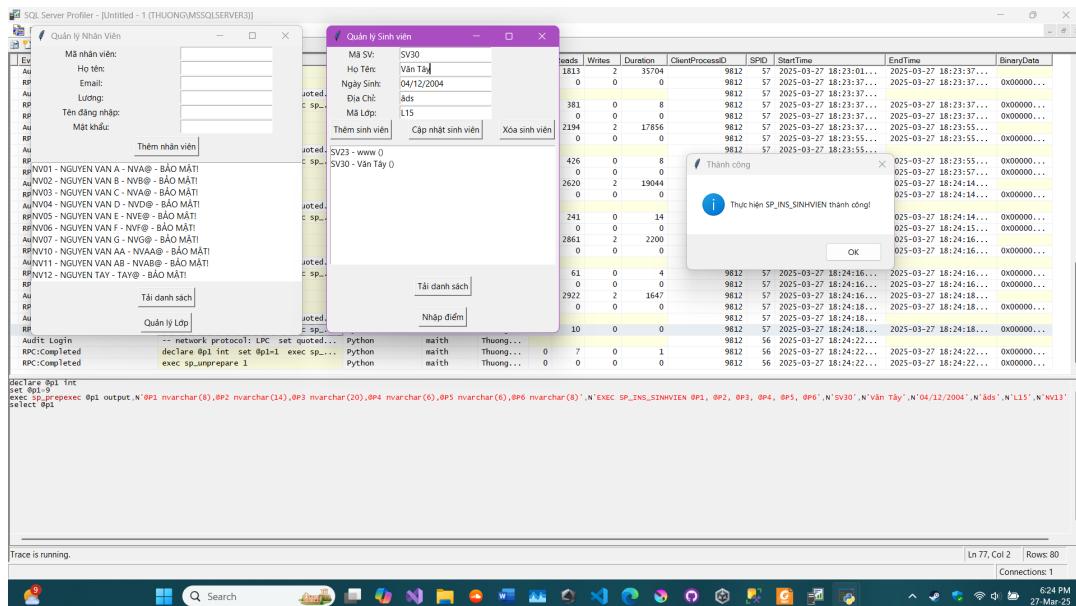


Hình 26: Xoá lớp

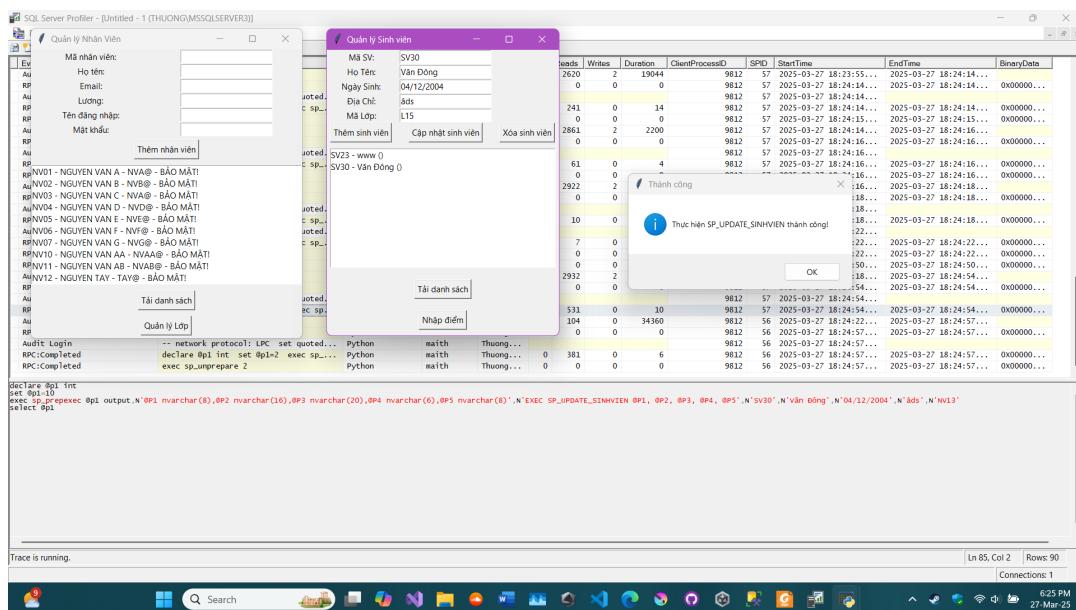


Hình 27: Mở quản lí sinh viên

## Báo cáo Lab 3

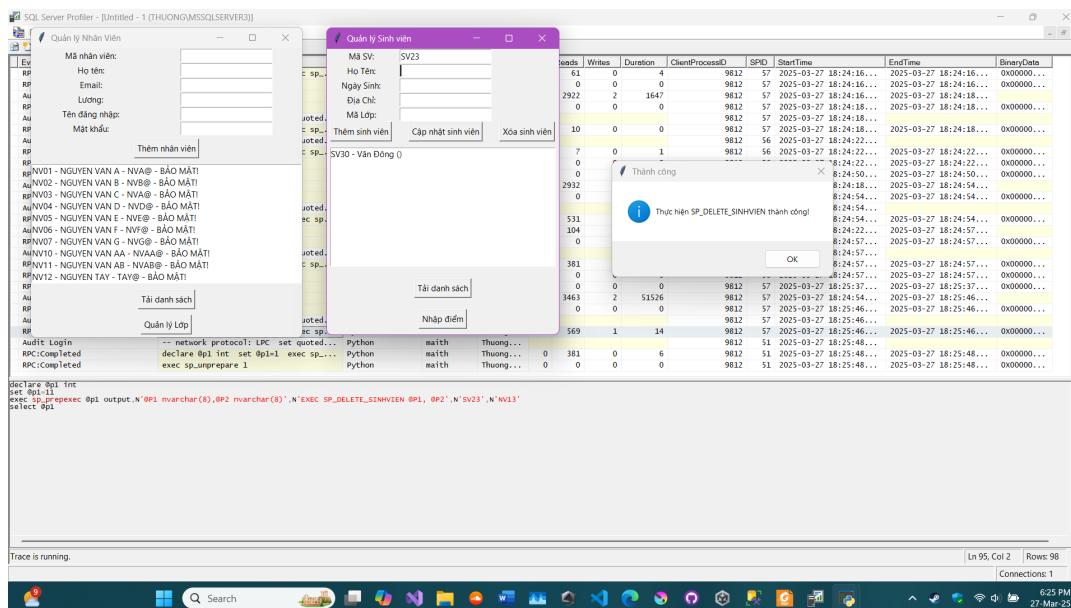


Hình 28: Thêm sinh viên

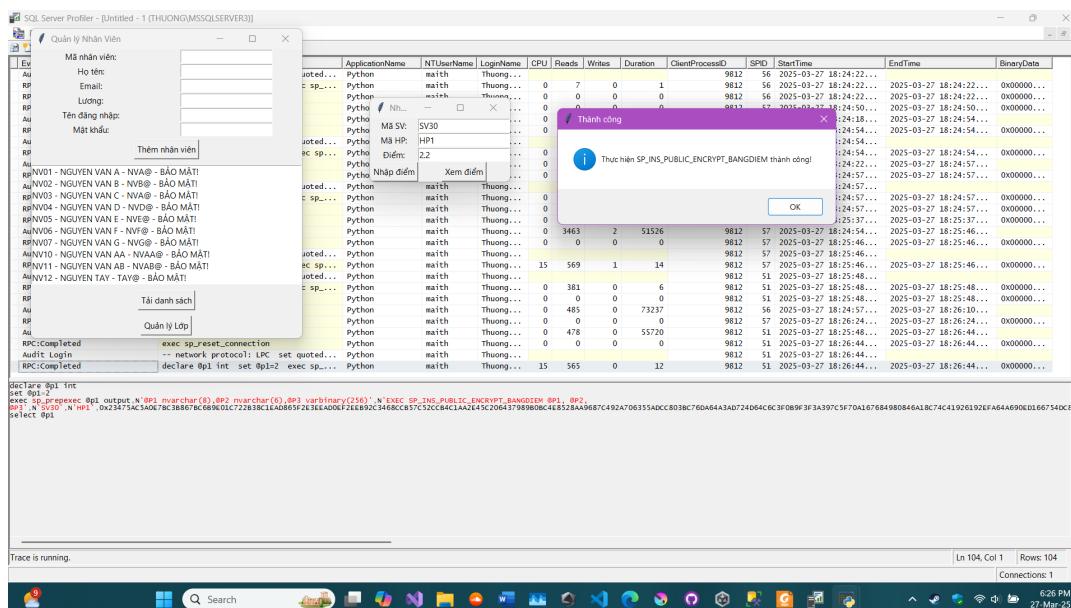


Hình 29: Cập nhật sinh viên

## Báo cáo Lab 3

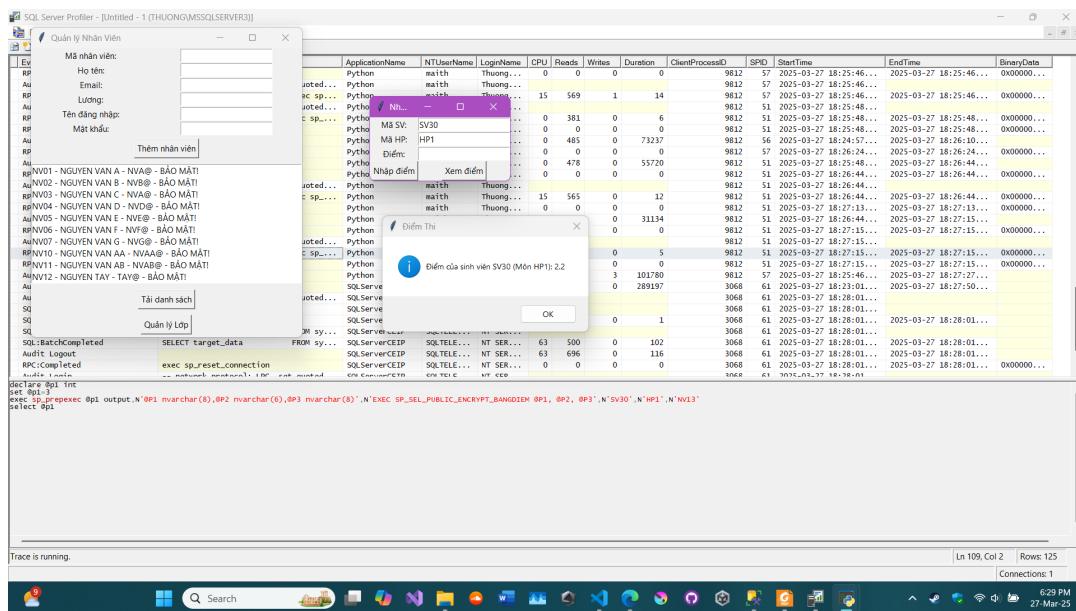


Hình 30: Chương trình giải mã



Hình 31: Nhập điểm cho sinh viên

Báo cáo Lab 3



Hình 32: Xem điểm cho sinh viên

## Tài liệu

- [1] RSA Private & Public Key Encryption in Python
- [2] RSA Encryption From Scratch - Math & Python Code
- [3] PyCryptodome's documentation