

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 13 tháng 3 năm 2025



Mục lục

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

1 Phân công

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

Bảng 1: Bảng phân công công việc

2 Viết các stored procedure

Hình 1: Tao master key và certificate

Báo cáo Lab 3

```

-- Stored procedure để thêm mới dữ liệu (Insert) vào table NHANVIEN
CREATE OR ALTER PROCEDURE SP_INS_PUBLIC_NHANVIEN
    @MANV NVARCHAR(20),
    @HOTEN NVARCHAR(100),
    @EMAIL NVARCHAR(20),
    @LUONGCB INT, -- Lương trước khi mã hóa
    @TENDIN NVARCHAR(100),
    @MK NVARCHAR(100) -- Mật khẩu trước khi mã hóa
AS
BEGIN
    -- Mã hóa mật khẩu bằng SHA1
    DECLARE @MATKHAU VARBINARY(MAX);
    SET @MATKHAU = HASHBYTES('SHA1', @MK);

    -- Tạo Asymmetric Key cho nhân viên này (nếu chưa có)
    IF NOT EXISTS (SELECT * FROM sys.asymmetric_keys WHERE name = @MANV)
    BEGIN
        EXEC('CREATE ASYMMETRIC KEY ' + @MANV +
              ' WITH ALGORITHM = RSA_512 ENCRYPTION BY PASSWORD = ' + @MK + '***');
    END;

    -- Mã hóa lương bằng RSA_512
    DECLARE @LUONG VARBINARY(MAX);
    SET @LUONG = EncryptByAsymKey(AsymKey_ID(@MANV), CAST(@LUONGCB AS VARBINARY));

    -- Kiểm tra nếu mã hóa thất bại
    IF @LUONG IS NULL
    BEGIN
        PRINT N'Lỗi: Mã hóa lương thất bại. Kiểm tra Asymmetric Key.';
        RETURN;
    END;

    -- Thêm nhân viên vào bảng NHANVIEN
    INSERT INTO NHANVIEN (MANV, HOTEN, EMAIL, LUONG, TENDIN, MATKHAU, PUBKEY)
    VALUES (@MANV, @HOTEN, @EMAIL, @LUONG, @TENDIN, @MATKHAU, @MK);

    PRINT N'Nhân viên đã được thêm thành công!';
END;
GO

```

Results Messages

Query executed successfully.

Hình 2: Stored dùng để thêm mới dữ liệu (Insert) vào table NHANVIEN

```

EXEC SP_INS_PUBLIC_NHANVIEN 'NV01', 'NGUYEN VAN A', 'NVA@', 3000000, 'NVA', 'abcd12'
go
EXEC SP_INS_PUBLIC_NHANVIEN 'NV02', 'NGUYEN VAN B', 'NVB@', 4000000, 'NVB', 'abcd123'
go

SELECT * FROM NHANVIEN
GO

```

Results Messages

	MANV	HOTEN	EMAIL	LUONG	TENDIN	MATKHAU	PUBKEY
1	NV01	NGUYEN VAN A	NVA@	0x8490F2A389F802A2DE38E1034620F4309C71F4894...	NVA	0x2F30042F077C1100241B001FE9E59465701C1	NV01
2	NV02	NGUYEN VAN B	NVB@	0x84283C3C85E234C2502007A51742F0E3CBA42D3...	NVB	0x7975896E024E728B102C09E5E76038662E9...	NV02

Query executed successfully.

Hình 3: Kết quả thêm mới dữ liệu

Báo cáo Lab 3

```

-- Stored dùng để truy vấn dữ liệu nhân viên (NHANVIEN)
CREATE OR ALTER PROCEDURE SP_SEL_PUBLIC_NHANVIEN
    @TENDN NVARCHAR(100),
    @MK NVARCHAR(100) -- Mật khẩu để mở khóa
AS
BEGIN
    -- Kiểm tra nếu nhân viên tồn tại
    IF NOT EXISTS (SELECT 1 FROM NHANVIEN WHERE TENDN = @TENDN)
    BEGIN
        PRINT N'Lỗi: TENDN không tồn tại!';
        RETURN;
    END;

    -- Lấy giá trị lương đã mã hóa
    DECLARE @LUONG_GIAIMA VARBINARY(MAX);
    DECLARE @PUB_KEY NVARCHAR(100);
    SELECT @LUONG_GIAIMA = LUONG, @PUB_KEY = PUBKEY FROM NHANVIEN WHERE TENDN = @TENDN;

    -- Giải mã lương
    DECLARE @LUONG_DEC INT;
    SET @LUONG_DEC = CAST(DecryptByAsymKey(AsymKey_ID(@PUB_KEY), @LUONG_GIAIMA, @MK) AS INT);

    -- Hiển thị lương gốc
    PRINT 'Lương sau giải mã: ' + CAST(@LUONG_DEC AS NVARCHAR);
    SELECT @LUONG_DEC AS LUONG_GOC;
END;
GO

```

Query executed successfully.

Hình 4: Stored dùng để truy vấn dữ liệu nhân viên (NHANVIEN)

```

EXEC SP_SEL_PUBLIC_NHANVIEN 'NVA', 'abcd12';
go
EXEC SP_SEL_PUBLIC_NHANVIEN 'NVB', 'abcd123';
go

```

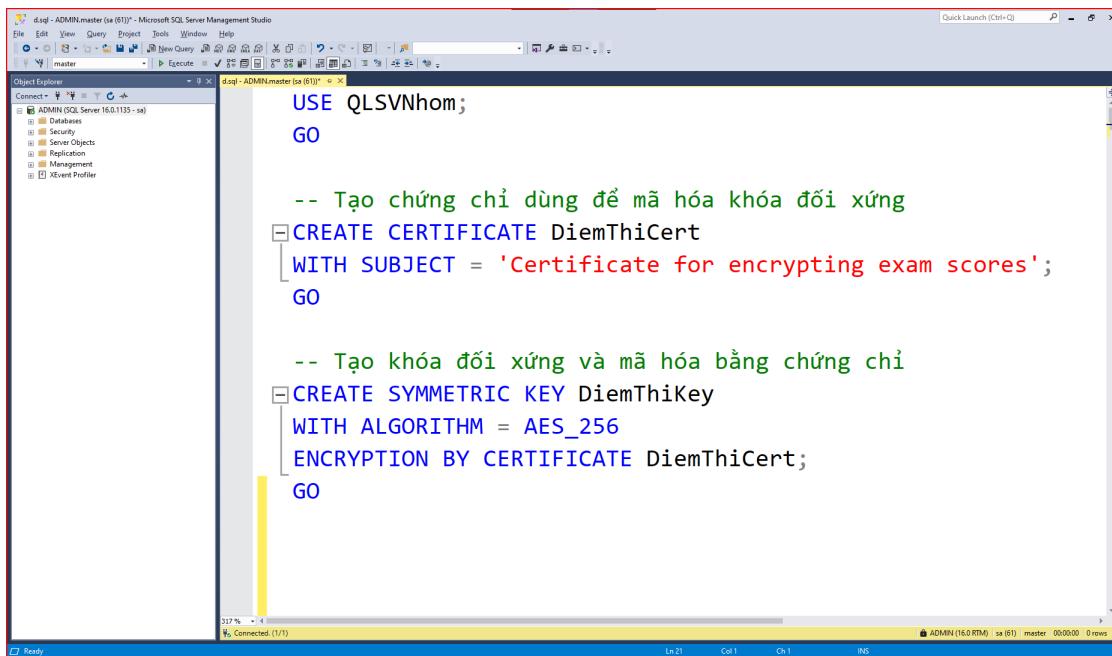
LUONG_GOC
300000

LUONG_GOC
400000

Query executed successfully.

Hình 5: Kết quả truy vấn đã được mã hoá

3 Viết các stored procedure và chương trình



```

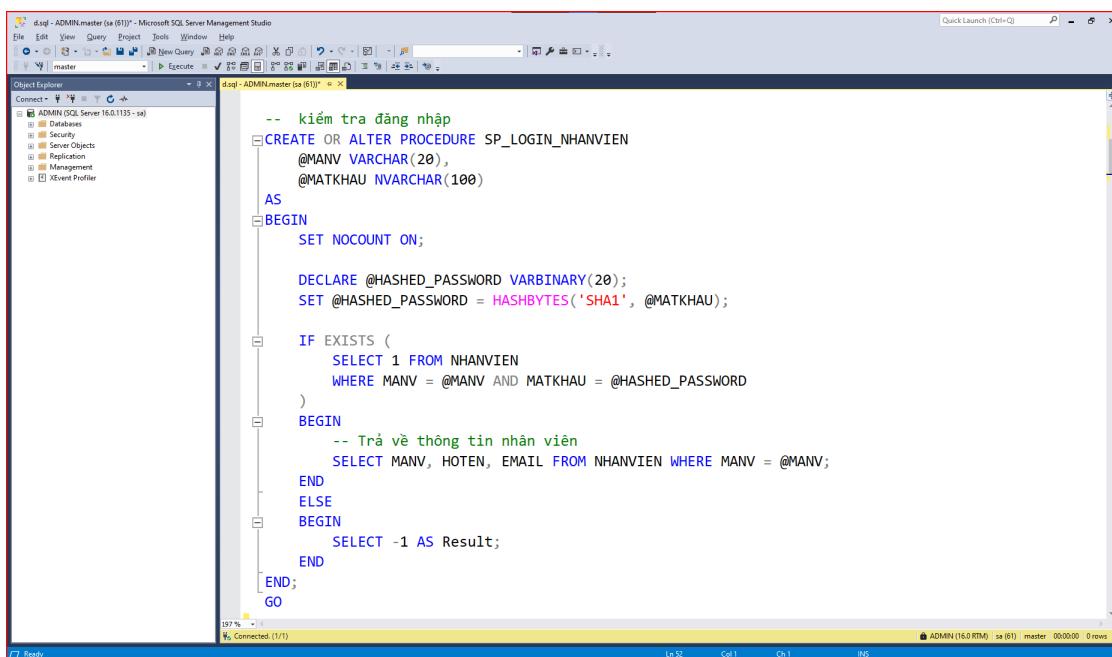
USE QLSVNhom;
GO

-- Tạo chứng chỉ dùng để mã hóa khóa đối xứng
CREATE CERTIFICATE DiemThiCert
    WITH SUBJECT = 'Certificate for encrypting exam scores';
GO

-- Tạo khóa đối xứng và mã hóa bằng chứng chỉ
CREATE SYMMETRIC KEY DiemThiKey
    WITH ALGORITHM = AES_256
    ENCRYPTION BY CERTIFICATE DiemThiCert;
GO

```

Hình 6: Tạo certificate và symmetric key



```

-- kiểm tra đăng nhập
CREATE OR ALTER PROCEDURE SP_LOGIN_NHANVIEN
    @MANV VARCHAR(20),
    @MATKHAU NVARCHAR(100)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @HASHED_PASSWORD VARBINARY(20);
    SET @HASHED_PASSWORD = HASHBYTES('SHA1', @MATKHAU);

    IF EXISTS (
        SELECT 1 FROM NHANVIEN
        WHERE MANV = @MANV AND MATKHAU = @HASHED_PASSWORD
    )
    BEGIN
        -- Trả về thông tin nhân viên
        SELECT MANV, HOTEN, EMAIL FROM NHANVIEN WHERE MANV = @MANV;
    END
    ELSE
    BEGIN
        SELECT -1 AS Result;
    END
END;
GO

```

Hình 7: Stored procedure kiểm tra 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 8: Stored procedure thêm lớp mới

```
-- 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 9: Stored procedure cập nhật thông tin lớp học

Báo cáo Lab 3

```
-- xóa 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 tất cả sinh viên thuộc lớp trước
    DELETE FROM SINHVIEN WHERE MALOP = @MALOP;

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

Hình 10: 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 11: Stored procedure xoá danh sách lớp

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
    All Objects
    Replication
    Management
    XEvent Profiler
dsql - ADMIN.master (sa (8)) * >
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 12: Stored procedure thêm sinh viên vào lớp

```

dsql - ADMINmaster (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)
    Databases
    Security
    All Objects
    Replication
    Management
    XEvent Profiler
dsql - ADMINmaster (sa (8)) * >
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 13: Stored procedure cập nhật thông tin 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 điểm của sinh viên
    DELETE FROM BANGDIEM WHERE MASV = @MASV

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

    SELECT 1 AS Result;
END;
```

Hình 14: Stored procedure xoá sinh viên

```
-- Lưu điểm thi
CREATE OR ALTER PROCEDURE SP_INS_BANGDIEM
    @MASV VARCHAR(20),
    @MAHP VARCHAR(20),
    @DIEMTHI FLOAT,
    @MANV VARCHAR(20) -- Nhân viên nhập điểm
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRANSACTION;

    -- Kiểm tra quyền nhập điểm
    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
        ROLLBACK TRANSACTION;
        SELECT -1 AS Result;
        RETURN;
    END

    -- Kiểm tra mã môn học có tồn tại
    IF NOT EXISTS (SELECT 1 FROM HOCPHAN WHERE MAHP = @MAHP)
    BEGIN
        ROLLBACK TRANSACTION;
        SELECT -1 AS Result;
        RETURN;
    END
```

Hình 15: Stored procedure lưu điểm thi

Báo cáo Lab 3

```
-- Kiểm tra mã môn học có tồn tại
IF NOT EXISTS (SELECT 1 FROM HOCPHAN WHERE MAHP = @MAHP)
BEGIN
    ROLLBACK TRANSACTION;
    SELECT -1 AS Result;
    RETURN;
END

-- Lấy Public Key của nhân viên
DECLARE @PUBKEY VARCHAR(50);
SELECT @PUBKEY = PUBKEY FROM NHANVIEN WHERE MANV = @MANV;

-- Mở khóa trước khi mã hóa
OPEN SYMMETRIC KEY DiemThiKey DECRYPTION BY CERTIFICATE DiemThiCert;

-- Kiểm tra xem khóa đã mở chưa
IF NOT EXISTS (SELECT 1 FROM sys.openkeys WHERE key_name = 'DiemThiKey')
BEGIN
    ROLLBACK TRANSACTION;
    SELECT -4 AS Result;
    RETURN;
END
```

Hình 16: Stored procedure lưu điểm thi

```
-- Mã hóa điểm
DECLARE @ENCRYPTED_SCORE VARBINARY(MAX);
SET @ENCRYPTED_SCORE = EncryptByKey(Key_GUID('DiemThiKey'), CAST(@DIEMTHI AS VARCHAR(10)));

-- Kiểm tra nếu mã hóa thất bại
IF @ENCRYPTED_SCORE IS NULL
BEGIN
    CLOSE SYMMETRIC KEY DiemThiKey;
    ROLLBACK TRANSACTION;
    SELECT -5 AS Result;
    RETURN;
END

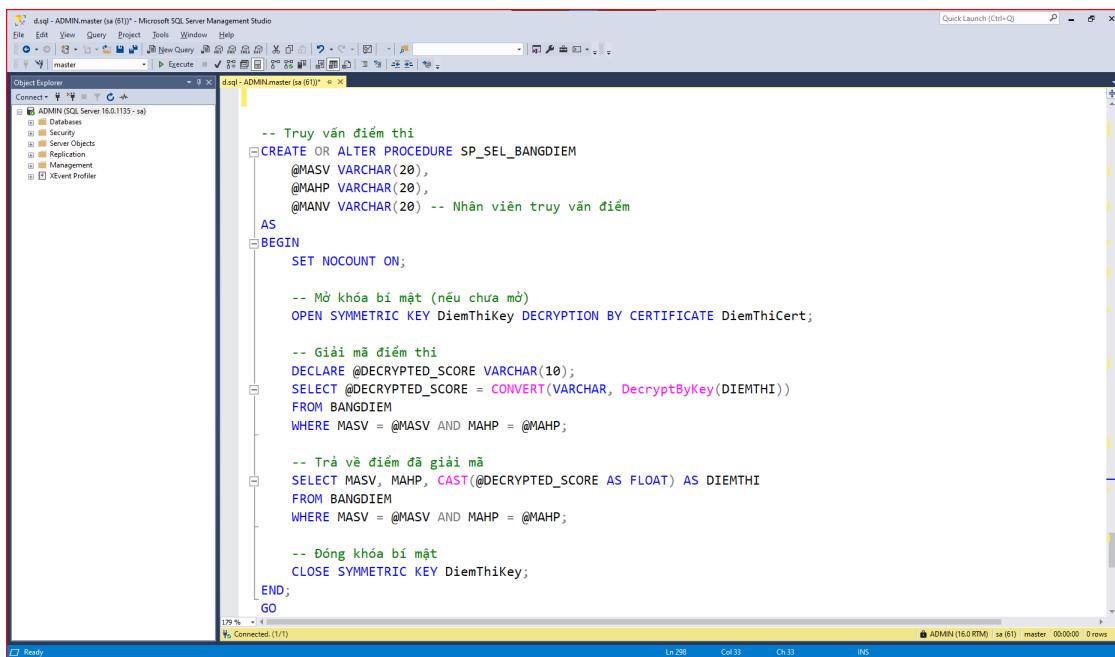
-- Lưu vào bảng điểm
INSERT INTO BANGDIEM (MASV, MAHP, DIEMTHI)
VALUES (@MASV, @MAHP, @ENCRYPTED_SCORE);

-- Đóng khóa và commit transaction
CLOSE SYMMETRIC KEY DiemThiKey;
COMMIT TRANSACTION;

SELECT 1 AS Result;
```

Hình 17: Stored procedure lưu điểm thi

Báo cáo Lab 3



```
-- Truy vấn điểm thi
CREATE OR ALTER PROCEDURE SP_SEL_BANGDIEM
    @MASV VARCHAR(20),
    @MAHP VARCHAR(20),
    @MANV VARCHAR(20) -- Nhân viên truy vấn điểm
AS
BEGIN
    SET NOCOUNT ON;

    -- Mở khóa bí mật (nếu chưa mở)
    OPEN SYMMETRIC KEY DiemThiKey DECRYPTION BY CERTIFICATE DiemThiCert;

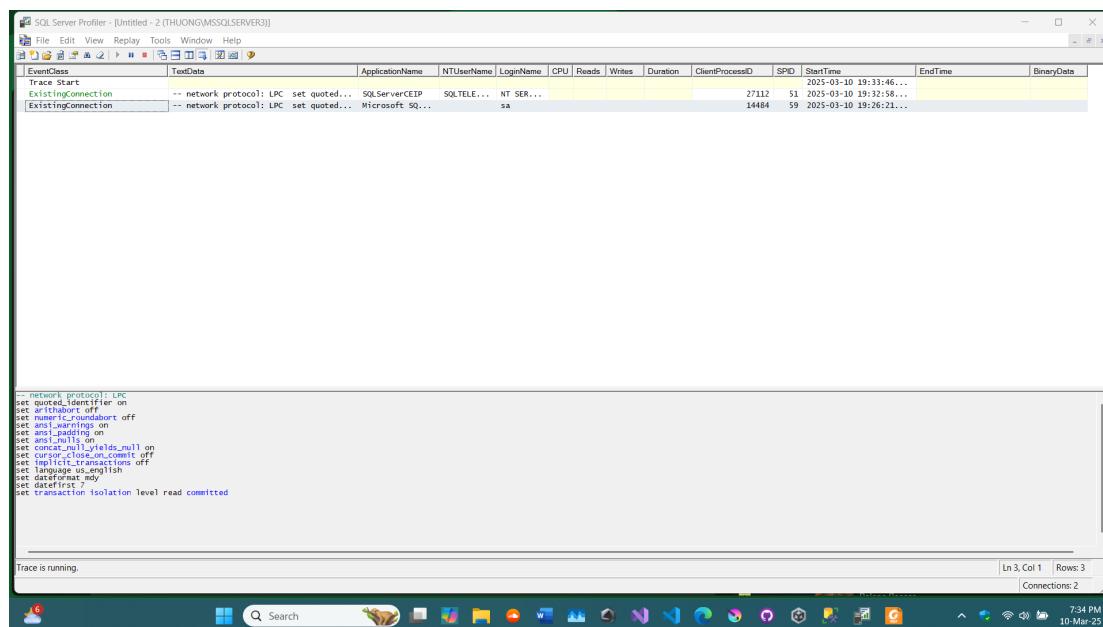
    -- Giải mã điểm thi
    DECLARE @DECRYPTED_SCORE VARCHAR(10);
    SELECT @DECRYPTED_SCORE = CONVERT(VARCHAR, DecryptByKey(DIEMTHI))
    FROM BANGDIEM
    WHERE MASV = @MASV AND MAHP = @MAHP;

    -- Trả về điểm đã giải mã
    SELECT MASV, MAHP, CAST(@DECRYPTED_SCORE AS FLOAT) AS DIEMTHI
    FROM BANGDIEM
    WHERE MASV = @MASV AND MAHP = @MAHP;

    -- Đóng khóa bí mật
    CLOSE SYMMETRIC KEY DiemThiKey;
END;
GO
```

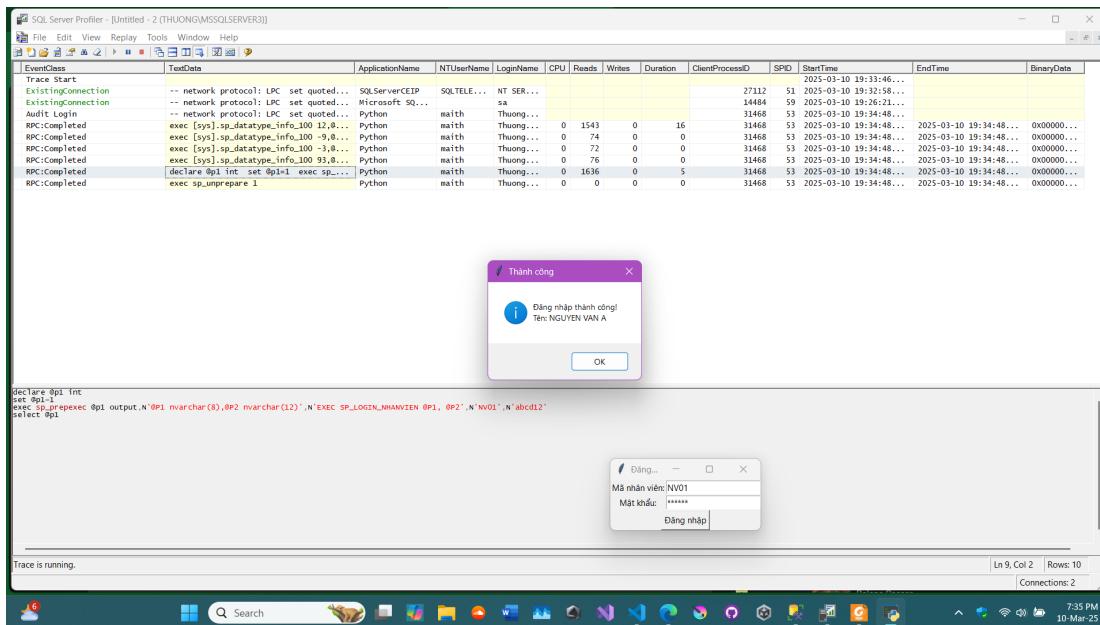
Hình 18: Stored procedure truy vấn điểm thi

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

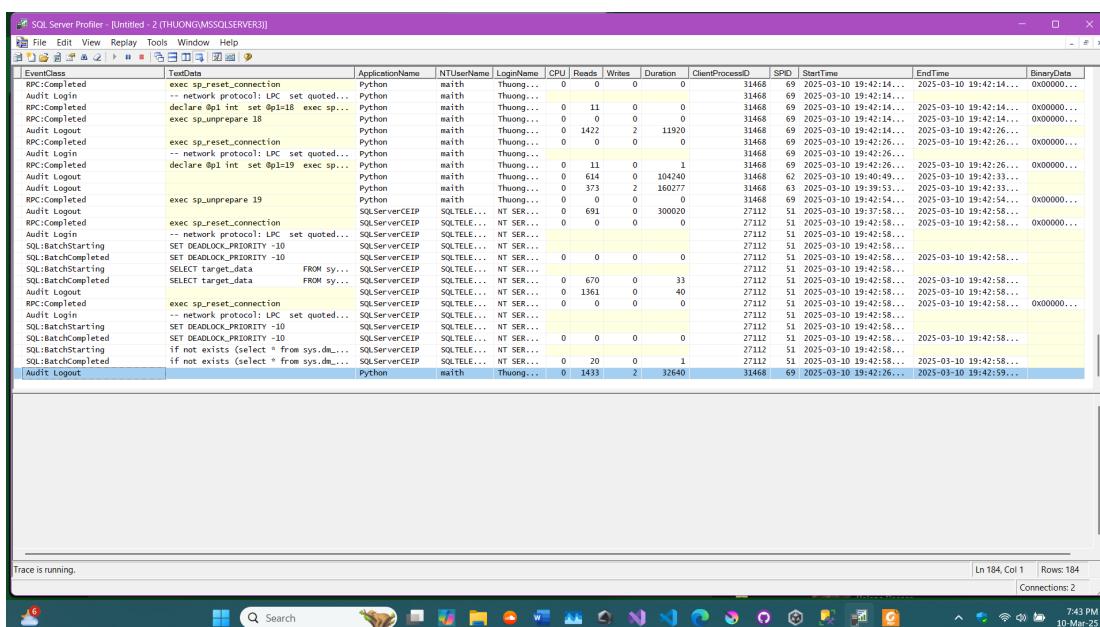


Hình 19: Vào thành công SQL Profiler

Báo cáo Lab 3

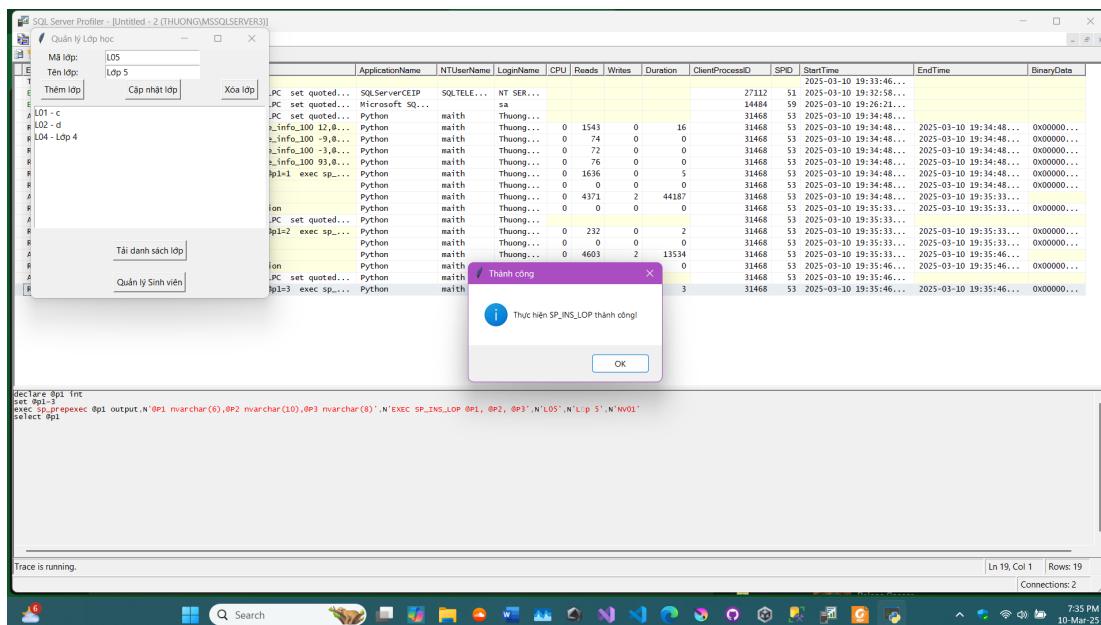


Hình 20: Đăng nhập

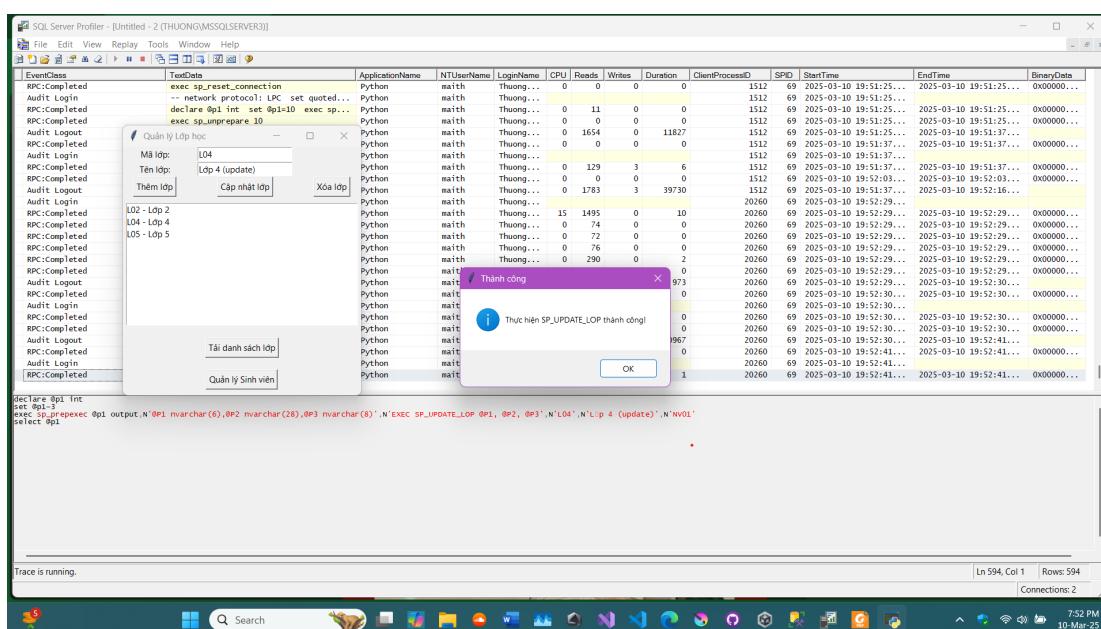


Hình 21: Đăng xuất

Báo cáo Lab 3

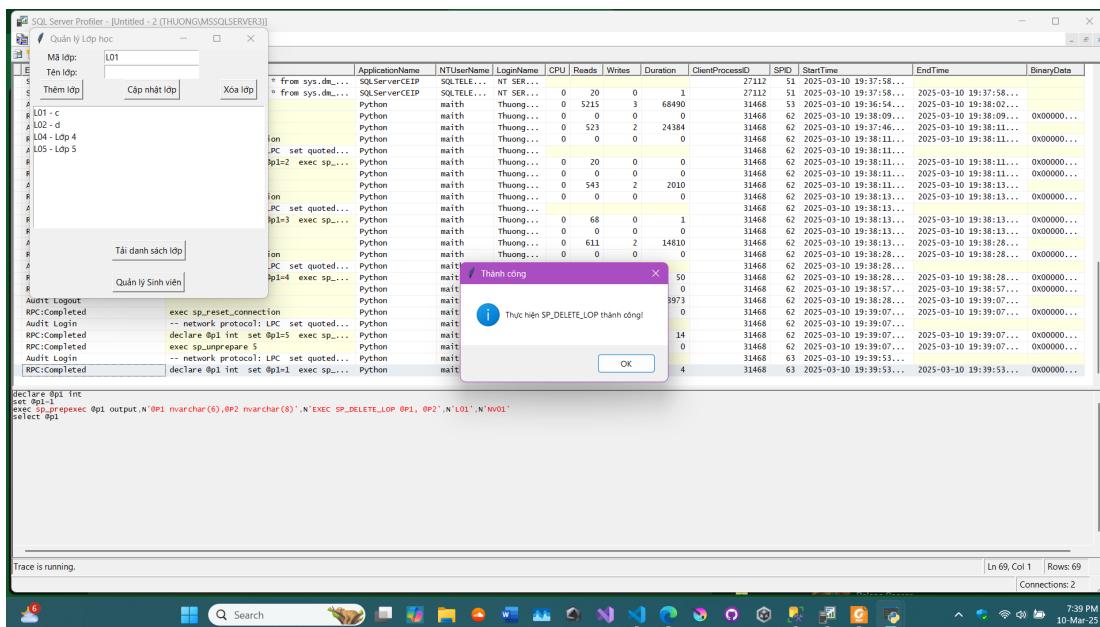


Hình 22: Thêm lớp

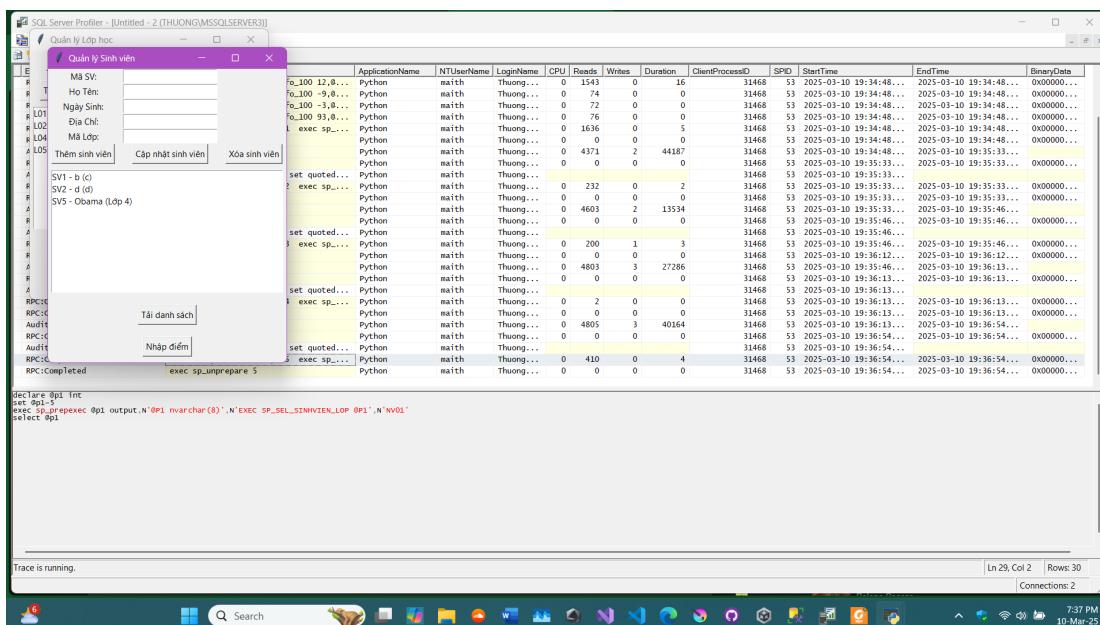


Hình 23: Cập nhật thông tin lớp

Báo cáo Lab 3

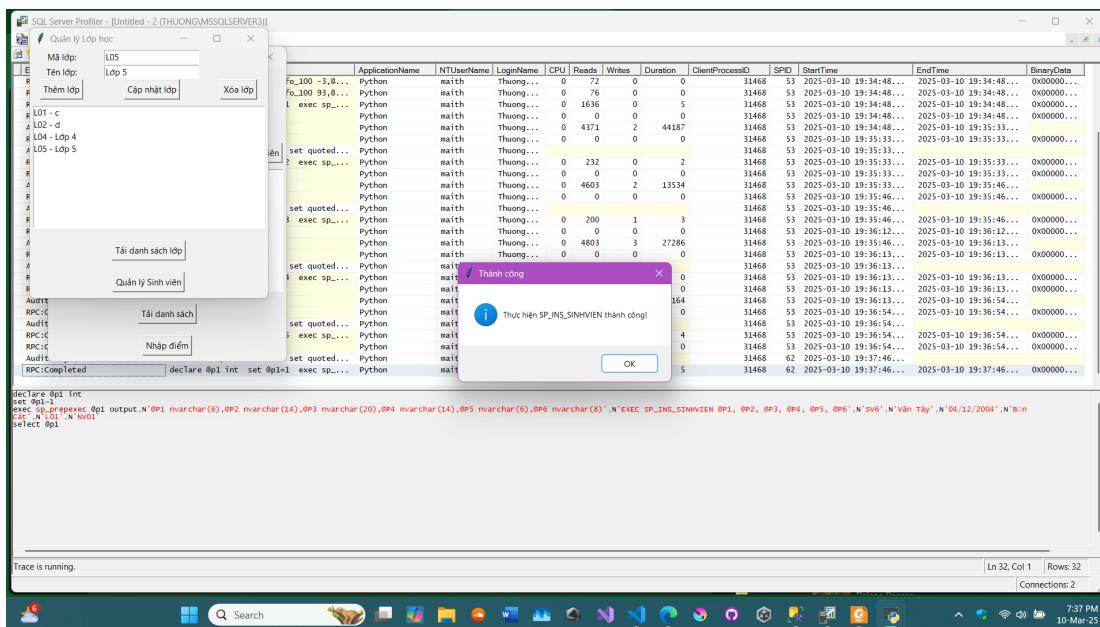


Hình 24: Xoá lớp

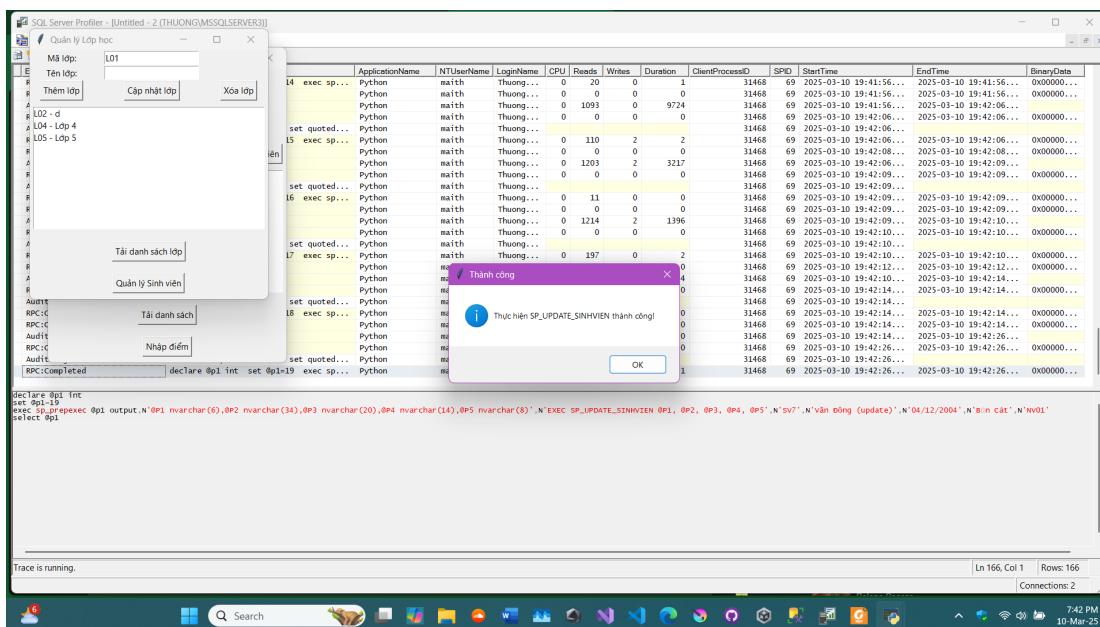


Hình 25: Mở quản lí sinh viên

Báo cáo Lab 3

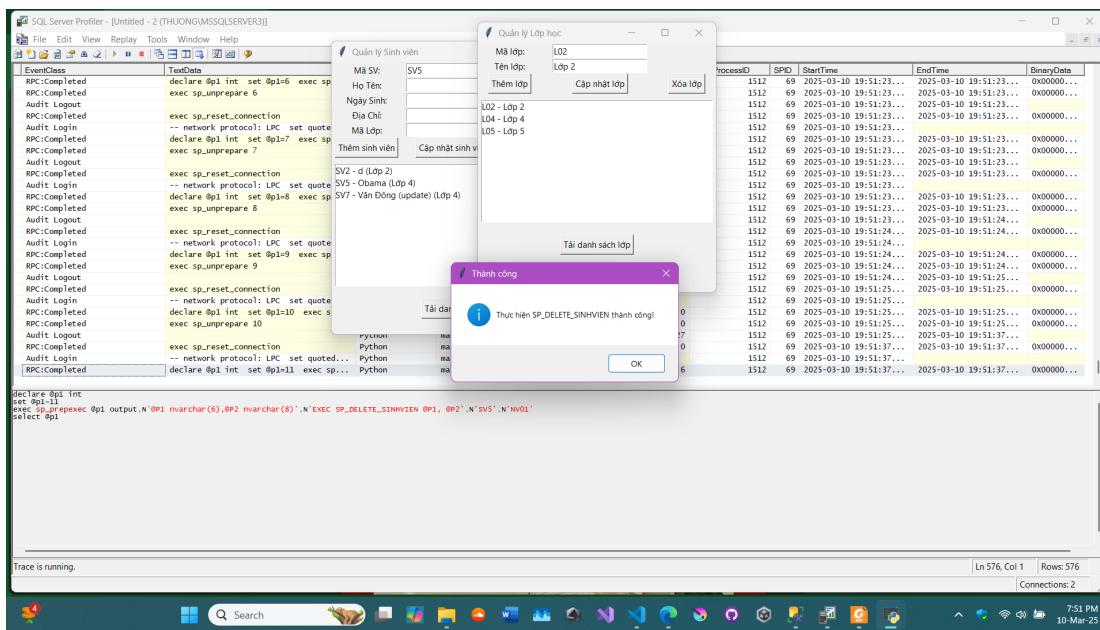


Hình 26: Thêm sinh viên

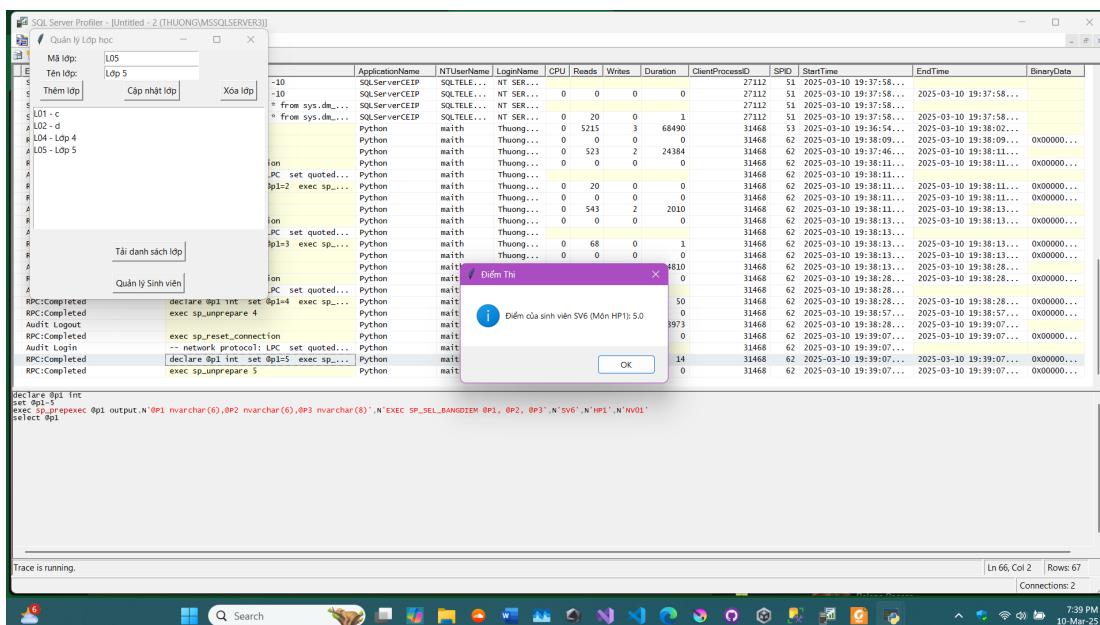


Hình 27: Cập nhật thông tin sinh viên

Báo cáo Lab 3

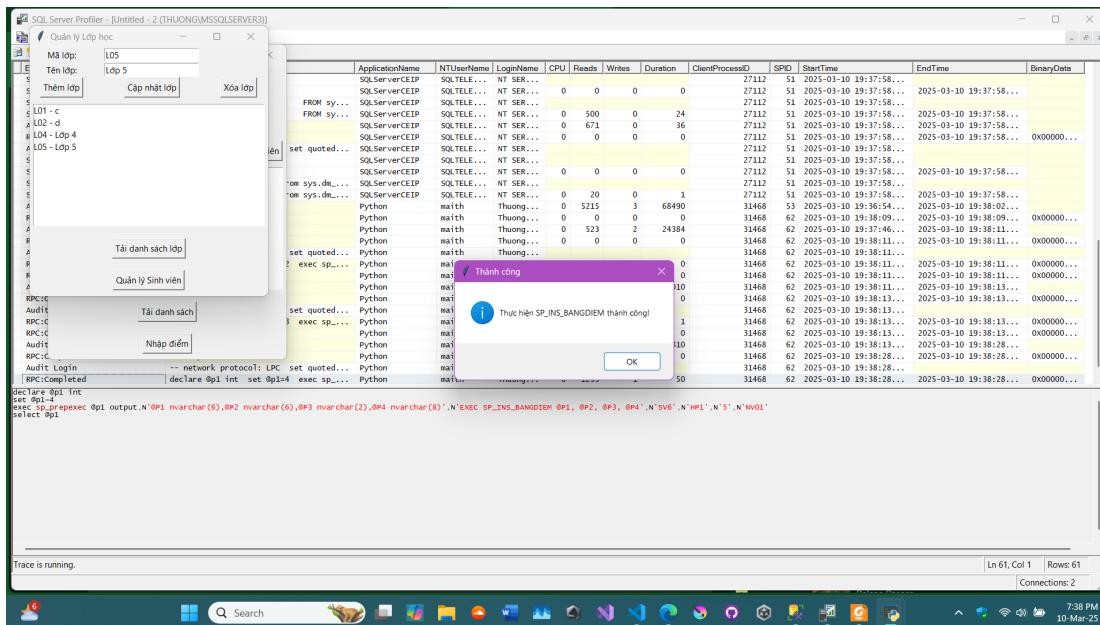


Hình 28: Xoá sinh viên



Hình 29: Xem điểm

Báo cáo Lab 3



Hình 30: Nhập điểm

Tài liệu

[1] SQL Server Certificates and Asymmetric Keys

[2] ENCRYPTBYASYMKEY (Transact-SQL)

[3] DECRYPTBYASYMKEY (Transact-SQL)