

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---



**An Toàn Ứng Dụng Web Và CSDL**  
**Lab04 Nhóm**

**Nhóm 28:**

**N18DCAT058: Hồ Minh Phong**

**N18DCAT100: Trần Quốc Trượng**

**N18DCAT102: Huỳnh Tiến Vĩ**

**TP.HCM - 2021**

### a) Viết script tạo Database có tên QLSVNhom

```
--/
MASV: N18DCAT100, N18DCAT058, N18DCAT102
HO TEN CAC THANH VIEN NHOM: Trần Quốc Trọng, Hồ Minh Phong, Huỳnh Tiến Vĩ
LAB: 03 - NHOM
NGAY: 11/3/2021
-----*/
--CAU LENH TAO DB

CREATE DATABASE QLSVNhom
```

### b) Viết script tạo mới các Table SINHVIEN, NHANVIEN, LOP, HOCPHAN, BANGDIEM

```
CREATE TABLE [dbo].[SINHVIEN](
    [MASV] [varchar](20) NOT NULL,
    [HOTEN] [nvarchar](100) NOT NULL,
    [NGAYSINH] [datetime] NULL,
    [DIACHI] [nvarchar](200) NULL,
    [MALOP] [varchar](20) NULL,
    [TENDN] [nvarchar](100) NOT NULL,
    [MATKHAU] [varbinary](max) NOT NULL,
    CONSTRAINT [PK_SINHVIEN] PRIMARY KEY CLUSTERED
```

```
CREATE TABLE [dbo].[NHANVIEN](
    [MANV] [varchar](20) NOT NULL,
    [HOTEN] [nvarchar](100) NOT NULL,
    [EMAIL] [varchar](20) NULL,
    [LUONG] [varbinary](max) NULL,
    [TENDN] [nvarchar](100) NOT NULL,
    [MATKHAU] [varbinary](max) NOT NULL,
    [PUBKEY] [varchar](20) NOT NULL,
    CONSTRAINT [PK_NHANVIEN] PRIMARY KEY CLUSTERED
(
```

```
CREATE TABLE [dbo].[LOP](
    [MALOP] [varchar](20) NOT NULL,
    [TENLOP] [nvarchar](100) NOT NULL,
    [MANV] [varchar](20) NULL,
    CONSTRAINT [PK_LOP] PRIMARY KEY CLUSTERED
(
```

```

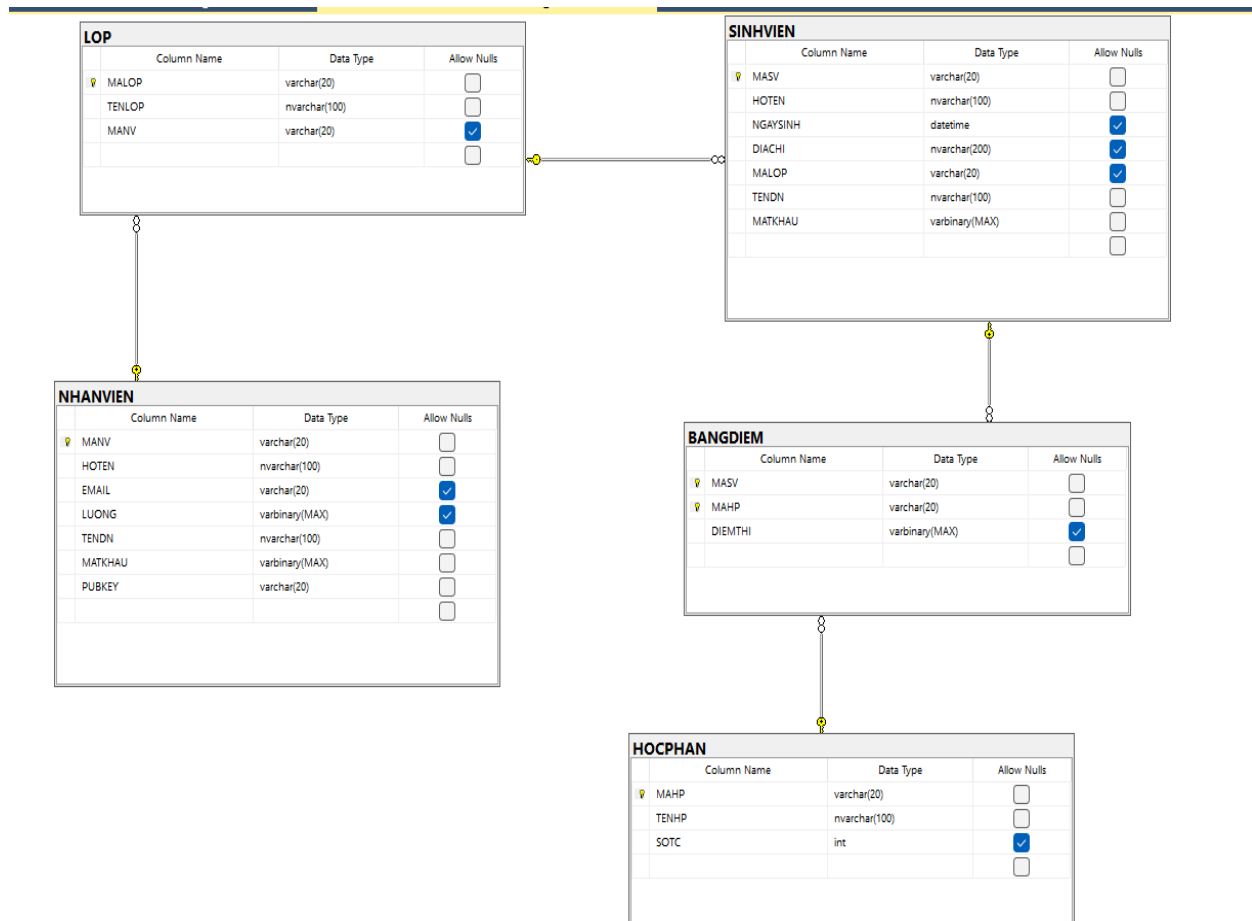
CREATE TABLE [dbo].[HOCPHAN](
    [MAHP] [varchar](20) NOT NULL,
    [TENHP] [nvarchar](100) NOT NULL,
    [SOTC] [int] NULL,
    CONSTRAINT [PK_HOCPHAN] PRIMARY KEY CLU
(

```

```

CREATE TABLE [dbo].[BANGDIEM](
    [MASV] [varchar](20) NOT NULL,
    [MAHP] [varchar](20) NOT NULL,
    [DIEMTHI] [varbinary](max) NULL,
    CONSTRAINT [PK_BANGDIEM] PRIMARY KEY CLUSTERED

```



### c) Viết các Stored procedure sau

```
/*-----*/
MASV: N18DCAT100, N18DCAT058, N18DCAT102
HO TEN CAC THANH VIEN NHOM: Trần Quốc Trọng, Hồ Minh Phong, Huỳnh Tiến Vĩ
LAB: 04 - NHOM 28
NGAY: 11/3/2021
/*-----*/

/*i) Stored dùng để thêm mới dữ liệu (Insert) vào table NHANVIEN,
trong đó dữ liệu tham số được mã hóa từ client*/

create proc SP_INS_PUBLIC_ENCRYPT_NHANVIEN
    @MANV varchar(20),
    @HOTEN nvarchar(100),
    @EMAIL varchar(20),
    @LUONG varbinary(MAX),
    @TENDN nvarchar(100),
    @MK varbinary(MAX),
    @PUB varchar(20)
as
insert into NHANVIEN
values(@MANV, @HOTEN, @EMAIL, @LUONG, @TENDN, @MK, @PUB)
go

/* ii) Stored dùng để truy vấn dữ liệu nhân viên (NHANVIEN) */
create proc SP_SEL_PUBLIC_ENCRYPT_NHANVIEN
    @TENDN nvarchar(100),
    @MK varbinary(MAX)
as
select MANV, HOTEN, EMAIL, LUONG
from NHANVIEN
where NHANVIEN.TENDN = @TENDN AND NHANVIEN.MATKHAU = @MK
go
```

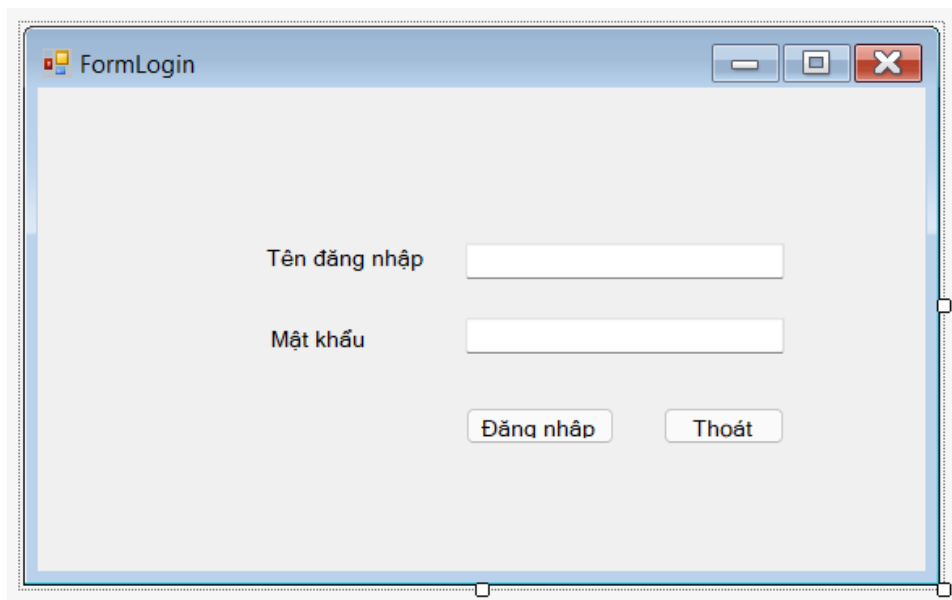
d) Viết các stored procedure và chương trình (sử dụng C#) để thực hiện các yêu cầu sau. Đầu file script ghi chú chi tiết như sau:

## 1. Xây dựng (lập trình) 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)

- Bước 1: viết stored procedure

```
--login
create proc Login @Username varchar(20), @Password varbinary(MAX)
as
select * from NHANVIEN where TENDN = @Username AND MATKHAU = @Password
go
```

- Bước 2: Xây dựng form login



The screenshot shows a Windows application window titled "FormLogin". Inside the window, there are two text input fields. The first field is labeled "Tên đăng nhập" (Username) and the second field is labeled "Mật khẩu" (Password). Below these fields, there are two buttons: "Đăng nhập" (Login) and "Thoát" (Exit). The window has a standard Windows title bar with minimize, maximize, and close buttons.

- Bước 3: viết mã xử lý nút đăng nhập

```

1 reference
private void button_dangNhap_Click(object sender, EventArgs e)
{
    //nút đăng nhập
    string username = textBox_dangNhap.Text.ToUpper();
    string passwd = textBox_Password.Text;
    //mã hóa mật khẩu
    string passwdHash = Cipher.Hash_SHA1(passwd);

    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec Login '{username}', {passwdHash}";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    SqlDataReader read = cmd.ExecuteReader();
    if (read.Read() == true)
    {
        //Đăng nhập thành công thì hiển thị màn hình quản lý
        this.Hide();
        Home home = new Home(username, passwdHash);
        home.ShowDialog();
    }
    else
    {
        MessageBox.Show("Tên đăng nhập hoặc mật khẩu không hợp lệ");
    }

    ketnoi.Close();
}

```

```

22 references
class Cipher
{
    5 references
    public static string Hash_SHA1(string input)
    {
        using (SHA1Managed sha1 = new SHA1Managed())
        {
            var hash = sha1.ComputeHash(Encoding.UTF8.GetBytes(input));
            var sb = new StringBuilder(hash.Length * 2);

            foreach (byte b in hash)
            {
                // can be "x2" if you want lowercase
                sb.Append(b.ToString("X2"));
            }

            return "0x" + sb.ToString();
        }
    }
}

```

## 2. Xây dựng (lập trình) màn hình quản lý nhân viên

- Bước 1: Viết các stored procedure

```
--load table nhan vien
create proc LOAD_NHANVIEN
as
    select MANV, HOTEN, EMAIL, LUONG
    from NHANVIEN
go

--sửa 1 nhân viên
create proc UPDATE_NHANVIEN
    @MANV varchar(20),
    @HOTEN nvarchar(100),
    @EMAIL varchar(20),
    @LUONG varbinary(MAX),
    @TENDN nvarchar(100),
    @MK varbinary(MAX)
as
    update NHANVIEN
    set HOTEN = @HOTEN, EMAIL = @EMAIL, LUONG = @LUONG, TENDN = @TENDN, MATKHAU = @MK
    where MANV = @MANV;
go

-- xóa nhân viên
create proc DELETE_NHANVIEN @MANV varchar(20)
as
    delete from NHANVIEN where MANV = @MANV;
go
```

- Bước 2: Xây dựng form QL Nhan Vien

Home

User Info QL NV QL Lop Hoc QL SV QL Diem QL Hoc Phan

Thông tin nhân viên

Mã NV  Họ tên

Email  Lương

Tên đăng nhập  Mật khẩu

Thêm Xóa Sửa Exit

	Mã Nhân Viên	Họ Tên	Email	Lương
*				

- Bước 3: Viết code xử lý các nút chức năng (chi tiết xem trong source code)

```
1 reference
private void button_insert_Click(object sender, EventArgs e)
{
    //thêm nhân viên
    //lay input
    string maNv = this.textBox1_manv.Text;
    string hoTen = this.textBox33_hoten.Text;
    string email = this.textBox22_email.Text;
    string luong = textBox_luong.Text;
    string tenDn = textBox_tenDangNhap.Text;
    string matKhai = textBox_matkhau.Text;
    string pubKeyName = "PubKey" + maNv;
    //mã hóa input
    string passEncrypted = Cipher.Hash_SHA1(matKhai);
    byte[] luongToEncrypt = ByteConverter.GetBytes(luong);
    //tạo cặp khóa
    RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(512);
    RSAParameters publicKey = RSA.ExportParameters(false);
    RSAParameters privateKey = RSA.ExportParameters(true);
    string pubKeyString = Cipher.KeyToString(publicKey);
    string priKeyString = Cipher.KeyToString(privateKey);
    //luu public key, private key vào dictionary
    listPublicKey.Add(pubKeyName, pubKeyString);
    listPrivateKey.Add(maNv, priKeyString);
    //luu dictionary vào file
    updateFileKey();
    loadKey();
    //mã hóa lương RSA
    byte[] encryptedLuong = Cipher.RSAEncrypt(luongToEncrypt, publicKey, false);
    string encryptedLuongString = "0x" + Cipher.ByteArrayToString(encryptedLuong);
    //insert nhanvien to database
    themNhanVien(maNv, hoTen, email, encryptedLuongString, tenDn, passEncrypted, pubKeyName);
    //loadTableNhanVien();
    refresh();
}
```

Figure 1. Thêm Nhân Viên

```
private void button_delete_Click(object sender, EventArgs e)
{
    string maNv = this.textBox1_manv.Text;
    string pubKeyName = "PubKey" + maNv;
    //xóa nhân viên khỏi DB
    xoaNhanVien(maNv);

    //xóa key của nhân viên đó
    listPrivateKey.Remove(maNv);
    listPublicKey.Remove(pubKeyName);
    updateFileKey();
    loadKey();
    refresh();
}
```

Figure 2. Xóa Nhân Viên



1 reference

```
private void button_update_Click(object sender, EventArgs e)
{
    //sửa 1 nhân viên
    string maNv = this.textBox1_manv.Text;
    string hoTen = this.textBox33_hoten.Text;
    string email = this.textBox22_email.Text;
    string luong = textBox_luong.Text;
    string tenDn = textBox_tenDangNhap.Text;
    string matKhai = textBox_matkhau.Text;
    string pubKeyName = "PubKey" + maNv;

    //mã hóa input
    string passEncrypted = Cipher.Hash_SHA1(matKhai);
    byte[] luongToEncrypt = ByteConverter.GetBytes(luong);
    //lấy public key
    RSAParameters PUBLICKEY = Cipher.StringToKey(listPublicKey[pubKeyName]);
    //mã hóa lương RSA
    byte[] encryptedLuong = Cipher.RSAEncrypt(luongToEncrypt, PUBLICKEY, false);
    string encryptedLuongString = "0x" + Cipher.ByteArrayToString(encryptedLuong);
    //sửa thông tin nhân viên
    suaNhanVien(maNv, hoTen, email, encryptedLuongString, tenDn, passEncrypted);

    refresh();
}
```

Figure 3. Sửa Nhân Viên

1 reference

```
private void loadTableNhanVien()
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = @"exec LOAD_NHANVIEN";

    SqlCommand scmd = new SqlCommand(sql, ketnoi);
    SqlDataReader sdr = scmd.ExecuteReader();

    DataTable dt = new DataTable();

    for (int i = 0; i < table_NhanVien.ColumnCount; ++i)
    {
        dt.Columns.Add(new DataColumn(table_NhanVien.Columns[i].Name));
        table_NhanVien.Columns[i].DataPropertyName = table_NhanVien.Columns[i].Name;
    }

    while (sdr.Read())
    {
        string manv = sdr.GetString(0); //MaNV
        string hoTen = sdr.GetString(1); //HoTen
        string email = sdr.GetString(2); //Email
        byte[] luongEncrypted = (byte[])sdr.GetValue(3); //Luong

        //giải mã lương nhân viên
        RSAParameters PRIVATEKEY = Cipher.StringToKey(listPrivateKey[manv]);
        byte[] LuongDecrypted = Cipher.RSADecrypt(luongEncrypted, PRIVATEKEY, false);
        string luong = ByteConverter.GetString(LuongDecrypted);

        dt.Rows.Add(manv, hoTen, email, luong);
    }
    //table_NhanVien.Columns.Clear();
    table_NhanVien.DataSource = dt;

    ketnoi.Close();
}
```

### 3. Xây dựng (lập trình) màn hình quản lý lớp học

- Bước 1: Viết các stored procedure

```
--load table LOP
create proc LOAD_LOPHOC
as
    select * from LOP
go

-- thêm lớp học
create proc INSERT_LOPHOC
    @MALOP varchar(20),
    @TENLOP nvarchar(100),
    @MANV varchar(20)
as
    insert into LOP
    values(@MALOP, @TENLOP, @MANV)
go

-- xóa lớp học
create proc DELETE_LOPHOC @MALOP varchar(20)
as
    delete from LOP where MALOP = @MALOP;
go

-- sửa thông tin lớp học
create proc MODIFY_LOPHOC
    @MALOP varchar(20),
    @TENLOP nvarchar(100),
    @MANV varchar(20)
as
    update LOP
    set TENLOP = @TENLOP, MANV = @MANV
    where MALOP = @MALOP;
go
```

- Bước 2: Xây dựng form QL Lop Hoc

- Bước 3: Viết code xử lý các nút chức năng (chi tiết xem source code)

```
1reference
private void button_insertLop_Click(object sender, EventArgs e)
{
    //get input
    string MALOP = this.textBox_maLop.Text;
    string TENLOP = this.textBox_tenLop.Text;
    string MANV = this.comboBox_maNV.SelectedItem.ToString();
    //inser to DB
    themLopHoc(MALOP, TENLOP, MANV);
    refresh();
}
```

Figure 4. Thêm Lớp Học

```

private void xoaLopHoc(string MALOP)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec DELETE_LOPHOC '{MALOP}'";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}

1 reference
private void button2_deleteLop_Click(object sender, EventArgs e)
{
    string MALOP = this.textBox_maLop.Text;

    xoaLopHoc(MALOP);
    refresh();
}

```

Figure 5. Xóa Lớp Học

```

1 reference
private void suaLopHoc(string MALOP, string TENLOP, string MANV)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec MODIFY_LOPHOC '{MALOP}', '{TENLOP}', '{MANV}'";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}

1 reference
private void button3_ModifyLop_Click(object sender, EventArgs e)
{
    string MALOP = this.textBox_maLop.Text;
    string TENLOP = this.textBox_tenLop.Text;
    string MANV = this.comboBox_maNV.SelectedItem.ToString();

    suaLopHoc(MALOP, TENLOP, MANV);
    refresh();
}

```

Figure 6. Sửa Lớp Học

```

1 reference
private void loadLopHoc()
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = @"exec LOAD_LOPHOC";

    SqlCommand scmd = new SqlCommand(sql, ketnoi);
    SqlDataReader sdr = scmd.ExecuteReader();

    DataTable dt = new DataTable();

    for (int i = 0; i < table_LOPHOC.ColumnCount; ++i)
    {
        dt.Columns.Add(new DataColumn(table_LOPHOC.Columns[i].Name));
        table_LOPHOC.Columns[i].DataPropertyName = table_LOPHOC.Columns[i].Name;
    }

    while (sdr.Read())
    {
        string maLop = sdr.GetString(0);
        string tenLop = sdr.GetString(1);
        string maNv = sdr.GetString(2);

        dt.Rows.Add(maLop, tenLop, maNv);
    }
    table_LOPHOC.DataSource = dt;

    ketnoi.Close();
}

```

#### 4. Xây dựng (lập trình) 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)

- Bước 1: Viết các stored procedure

```

-- xóa sinh viên
create proc DELETE_SINHVIEN @MASV varchar(20)
as
    delete from SINHVIEN where MASV = @MASV;
go

-- sửa thông tin sinh viên
create proc MODIFY_SINHVIEN
    @MASV varchar(20),
    @HOTEN nvarchar(100),
    @NGAYSINH datetime,
    @DIACHI nvarchar(200),
    @MALOP varchar(20),
    @TENDN nvarchar(100),
    @MATKHAU varbinary(MAX)
as
    update SINHVIEN
    set HOTEN = @HOTEN, NGAYSINH = @NGAYSINH, DIACHI = @DIACHI, MALOP = @MALOP, TENDN = @TENDN, MATKHAU = @MATKHAU
    where MASV = @MASV;
go

```

```

-- load table SINHVIEN
create proc LOAD_ALL_SINHVIEN
as
    select * from SINHVIEN
go

create proc LOAD_SINHVIEN @MALOP varchar(20)
as
    select * from SINHVIEN
    where MALOP = @MALOP;
go

-- thêm sinh viên
create proc INSERT_SV
    @MASV varchar(20),
    @HOTEN nvarchar(100),
    @NGAYSINH datetime,
    @DIACHI nvarchar(200),
    @MALOP varchar(20),
    @TENDN nvarchar(100),
    @MATKHAU varbinary(MAX)
as
    insert into SINHVIEN
    values(@MASV, @HOTEN, @NGAYSINH, @DIACHI, @MALOP, @TENDN, @MATKHAU)
go

```

- Bước 2: Xây dựng form QL Sinh Viên

The screenshot shows a Windows application window titled "Home" with a menu bar containing "User Info", "QL NV", "QL Lop Hoc", "QL SV", "QL Diem", and "QL Hoc Phan". The "QL SV" menu item is selected. The main form contains the following fields and controls:

- MSSV**: Text input field.
- Ngày Sinh**: Date input field.
- Tên DN**: Text input field.
- Họ Tên**: Text input field.
- Địa Chỉ**: Text input field.
- Mật Khẩu**: Text input field.
- Mã Lớp**: Dropdown menu.
- Buttons**: "Thêm", "Xóa", "Sửa", and "Exit".
- Chọn Lớp Học**: Dropdown menu.
- Filter**: Button.

Below the form is a table with the following columns: MSSV, Họ Tên, Ngày Sinh, Địa Chỉ, and Mã Lớp. The table is currently empty, with a single row containing an asterisk (\*) in the first column.

- Bước 3: Viết code các nút chức năng (chi tiết xem source code)

```
1 reference
private void themSV(string MASV, string HOTEN, string NGAYSINH, string DIACHI, string MALOP, string TENDN, string MATKHAU)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec INSERT_SV '{MASV}', '{HOTEN}', '{NGAYSINH}', '{DIACHI}', '{MALOP}', '{TENDN}', {MATKHAU}";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}

1 reference
private void xoaSV(string MASV)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec DELETE_SINHVIEN '{MASV}'";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}

1 reference
private void suaSV(string MASV, string HOTEN, string NGAYSINH, string DIACHI, string MALOP, string TENDN, string MATKHAU)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec MODIFY_SINHVIEN '{MASV}', '{HOTEN}', '{NGAYSINH}', '{DIACHI}', '{MALOP}', '{TENDN}', {MATKHAU}";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}
```

Figure 7. Thêm, Xóa Sửa SV

```
1 reference
private void filterSV(string MALOP)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec LOAD_SINHVIEN '{MALOP}'";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    SqlDataReader rd = cmd.ExecuteReader();
    DataTable dt = new DataTable();

    for (int i = 0; i < table_SinhVien.ColumnCount; ++i)
    {
        dt.Columns.Add(new DataColumn(table_SinhVien.Columns[i].Name));
        table_SinhVien.Columns[i].DataPropertyName = table_SinhVien.Columns[i].Name;
    }

    while (rd.Read())
    {
        string maSv = rd.GetString(0);
        string hoTen = rd.GetString(1);
        string ngaySinh = rd.GetDateTime(2).ToString();
        string diaChi = rd.GetString(3);
        string maLop = rd.GetString(4);

        dt.Rows.Add(maSv, hoTen, ngaySinh, diaChi, maLop);
    }
    table_SinhVien.DataSource = dt;

    ketnoi.Close();
}
```

Figure 8. Lọc sinh viên theo lớp

2 references

```
private void loadAllSV()  
{  
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();  
    string sql = @"exec LOAD_ALL_SINHVIEN";  
    SqlCommand cmd = new SqlCommand(sql, ketnoi);  
    SqlDataReader rd = cmd.ExecuteReader();  
  
    DataTable dt = new DataTable();  
  
    for (int i = 0; i < table_SinhVien.ColumnCount; ++i)  
    {  
        dt.Columns.Add(new DataColumn(table_SinhVien.Columns[i].Name));  
        table_SinhVien.Columns[i].DataPropertyName = table_SinhVien.Columns[i].Name;  
    }  
  
    while (rd.Read())  
    {  
        string maSv = rd.GetString(0);  
        string hoTen = rd.GetString(1);  
        string ngaySinh = rd.GetDateTime(2).ToString();  
        string diaChi = rd.GetString(3);  
        string maLop = rd.GetString(4);  
  
        dt.Rows.Add(maSv, hoTen, ngaySinh, diaChi, maLop);  
    }  
    table_SinhVien.DataSource = dt;  
  
    ketnoi.Close();  
}
```

## 5. Xây dựng (lập trình) 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)

- Bước 1: Viết các stored procedure

```
--load table học phần  
create proc LOAD_HOCPHAN  
as  
select * from HOCPHAN  
go  
--insert học phần  
create proc INSERT_HOCPHAN  
    @MAHP varchar(20),  
    @TENHP nvarchar(100),  
    @SOTC int  
as  
insert into HOCPHAN  
values(@MAHP, @TENHP, @SOTC)  
go
```



```

-- sửa học phần
create proc UPDATE_HOCPHAN
    @MAHP varchar(20),
    @TENHP nvarchar(100),
    @SOTC int
as
    update HOCPHAN
    set TENHP = @TENHP, SOTC = @SOTC
    where MAHP = @MAHP;
go

-- xóa học phần
create proc DELETE_HOCPHAN @MAHP varchar(20)
as
    delete from HOCPHAN where MAHP = @MAHP;
go

```

- Bước 2: Xây dựng form QL Học Phần

The screenshot shows a Windows application window titled "Home" with a menu bar containing "User Info", "QL NV", "QL Lop Hoc", "QL SV", "QL Diem", and "QL Hoc Phan". The "QL Hoc Phan" menu item is selected. The main form area contains three input fields: "Mã Học Phần", "Tên Học Phần", and "Số TC". Below these fields are four buttons: "Thêm", "Xóa", "Sửa", and "Exit". At the bottom of the form is a table with the following structure:

	Mã Học Phần	Tên Học Phần	Số Tín Chỉ
*			

- Bước 3: Viết code xử lý các nút chức năng (chi tiết xem source code)

1 reference

```
private void loadHocPhan()
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = @"exec LOAD_HOCPHAN";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    SqlDataReader rd = cmd.ExecuteReader();

    DataTable dt = new DataTable();

    for (int i = 0; i < table_HocPhan.ColumnCount; ++i)
    {
        dt.Columns.Add(new DataColumn(table_HocPhan.Columns[i].Name));
        table_HocPhan.Columns[i].DataPropertyName = table_HocPhan.Columns[i].Name;
    }

    while (rd.Read())
    {
        string MAHP = rd.GetString(0);
        string TENHP = rd.GetString(1);
        int SOTC = rd.GetInt32(2);

        dt.Rows.Add(MAHP, TENHP, SOTC);
    }
    table_HocPhan.DataSource = dt;

    ketnoi.Close();
}
```

1 reference

```
private void themHocPhan(string MAHP, string TENHP, string SOTC)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec INSERT_HOCPHAN '{MAHP}', '{TENHP}', {SOTC}";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}
```

1 reference

```
private void suaHocPhan(string MAHP, string TENHP, string SOTC)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec UPDATE_HOCPHAN '{MAHP}', '{TENHP}', {SOTC}";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}
```

1 reference

```
private void xoaHocPhan(string MAHP)
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = $"exec DELETE_HOCPHAN '{MAHP}'";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    cmd.ExecuteNonQuery();
    ketnoi.Close();
}
```

- Bước 4: Viết các stored procedure sau

```
--load bảng điểm
create proc LOAD_DIEM
as
    select MASV, BANGDIEM.MAHP, TENHP, DIEMTHI
    from BANGDIEM
    inner join HOCPHAN on BANGDIEM.MAHP = HOCPHAN.MAHP;
go

-- insert điểm
create proc INSERT_DIEM
    @MASV varchar(20),
    @MAHP varchar(20),
    @DIEMTHI varbinary(MAX)
as
    insert into BANGDIEM
    values(@MASV, @MAHP, @DIEMTHI);
go

-- sửa điểm
create proc UPDATE_DIEM
    @MASV varchar(20),
    @MAHP varchar(20),
    @DIEMTHI varbinary(MAX)
as
    update BANGDIEM
    set DIEMTHI = @DIEMTHI
    where MASV = @MASV and MAHP = @MAHP;
go

-- xóa điểm
create proc DELETE_DIEM @MASV varchar(20), @MAHP varchar(20)
as
    delete from BANGDIEM where MASV = @MASV and MAHP = @MAHP;
go
```

- Bước 5: Xây dựng form QL DIEM

- Bước 6: Viết code xử lý các nút chức năng (chi tiết xem source code)

```
1 reference
private void button2_insertDIEM_Click(object sender, EventArgs e)
{
    string MASV = textBox1_MSSV_Diem.Text.ToUpper();
    string MAHP = comboBox1_MAHP_DIEM.SelectedItem.ToString();
    string[] mang = MAHP.Split(' ');
    MAHP = mang[0];
    string DIEMTHI = textBox2_DIEMTHI.Text;

    //mã hóa điểm thi bằng public key của nhân viên đang đăng nhập
    byte[] diemToEncrypt = ByteConverter.GetBytes(DIEMTHI);
    RSAParameters publicKey = Cipher.StringToKey(listPublicKey[PUBKEYNAME]);
    byte[] encryptedDiem = Cipher.RSAEncrypt(diemToEncrypt, publicKey, false);
    string encryptedDiemString = "0x" + Cipher.ByteArrayToString(encryptedDiem);
    //insert DB
    nhapDiem(MASV, MAHP, encryptedDiemString);
    //lưu lại mã nhân viên nhập điểm
    whoEncryptedDiem.Add(MASV + MAHP, MANV_LOGIN);
    FileClass.WriteFile(whoEncryptedDiem, "log.bin");

    refresh();
}
```

Figure 9. Nhập Điểm

```
1 reference
private void button3_deleteDIEM_Click(object sender, EventArgs e)
{
    string MASV = textBox1_MSSV_Diem.Text.ToUpper();
    string MAHP = comboBox1_MAHP_DIEM.SelectedItem.ToString();
    string[] mang = MAHP.Split(' ');
    MAHP = mang[0];
    //xóa điểm đã nhập trong DB
    xoaDiem(MASV, MAHP);
    //xóa mã nhân viên đã nhập điểm đó
    whoEncryptedDiem.Remove(MASV + MAHP);
    FileClass.WriteFile(whoEncryptedDiem, "log.bin");

    refresh();
}
```

Figure 10. Xóa Điểm

1 reference

```
private void button4_updateDIEM_Click(object sender, EventArgs e)
{
    string MASV = textBox1_MSSV_Diem.Text.ToUpper();
    string MAHP = comboBox1_MAHP_DIEM.SelectedItem.ToString();
    string[] mang = MAHP.Split(' ');
    MAHP = mang[0];
    string DIEMTHI = textBox2_DIEMTHI.Text;

    //mã hóa điểm thi bằng public key của nhân viên đang đăng nhập
    byte[] diemToEncrypt = ByteConverter.GetBytes(DIEMTHI);

    RSAParameters publicKey = Cipher.StringToKey(listPublicKey[PUBKEYNAME]);

    byte[] encryptedDiem = Cipher.RSAEncrypt(diemToEncrypt, publicKey, false);
    string encryptedDiemString = "0x" + Cipher.ByteArrayToString(encryptedDiem);

    //update DB
    suaDiem(MASV, MAHP, encryptedDiemString);

    whoEncryptedDiem[MASV + MAHP] = MANV_LOGIN;
    FileClass.WriteFile(whoEncryptedDiem, "log.bin");

    refresh();
}
```

Figure 11. Sửa điểm thi

```
private void loadBangDiem()
{
    SqlConnection ketnoi = ConnectionSQL.LayKetNoi();
    string sql = @"exec LOAD_DIEM";
    SqlCommand cmd = new SqlCommand(sql, ketnoi);
    SqlDataReader rd = cmd.ExecuteReader();

    DataTable dt = new DataTable();
    for (int i = 0; i < table_DIEM.ColumnCount; ++i)
    {
        dt.Columns.Add(new DataColumn(table_DIEM.Columns[i].Name));
        table_DIEM.Columns[i].DataPropertyName = table_DIEM.Columns[i].Name;
    }

    while (rd.Read())
    {
        string MASV = rd.GetString(0);
        string MAHP = rd.GetString(1);
        string TENHP = rd.GetString(2);
        byte[] DIEMTHI = (byte[])rd.GetValue(3); //Điểm thi bị mã hóa
        // lấy mã nhân viên đã mã hóa điểm sinh viên này trong dictionary whoEncryptedDiem
        string MANV = whoEncryptedDiem[MASV + MAHP];
        // tìm PRIVATE KEY trong listPrivateKey
        string privateKeyString = listPrivateKey[MANV];
        // convert PRIVATEKEY
        RSAParameters PRIVATEKEY = Cipher.StringToKey(privateKeyString);
        //giải mã điểm thi
        byte[] DiemDecrypted = Cipher.RSADecrypt(DIEMTHI, PRIVATEKEY, false);
        string diem = ByteConverter.GetString(DiemDecrypted);

        dt.Rows.Add(MASV, MAHP, TENHP, diem);
    }
    table_DIEM.DataSource = dt;

    ketnoi.Close();
}
```

## e) Sử dụng công cụ SQL Profile để theo dõi thao tác trong màn hình nhập điểm sinh viên và cho nhận xét

- Thực hiện nhập điểm cho 1 sinh viên

The screenshot displays the SQL Profiler interface with a trace running. The 'TextData' column shows the execution of an 'INSERT\_DIEM' stored procedure. Simultaneously, a 'Home' window is open, showing a data entry form for a student's score. The form includes fields for 'MSSV' (SV002), 'Mã HP' (HP02 - Mang May Tinh), and 'Điểm Thi' (8). A table on the right lists students and their scores.

MSSV	MAHP	Tên Học Phần	Điểm Thi
SV001	HP01	Kien Truc May Tinh	10
SV002	HP02	Mang May Tinh	8

- Chỉnh sửa điểm cho sinh viên

The screenshot shows the SQL Profiler interface with a trace running. The 'TextData' column shows the execution of an 'UPDATE\_DIEM' stored procedure. The 'Home' window is also open, showing the same data entry form, but the 'Điểm Thi' field is now set to 9. The table on the right shows the updated score for SV002.

MSSV	MAHP	Tên Học Phần	Điểm Thi
SV001	HP01	Kien Truc May Tinh	10
SV002	HP02	Mang May Tinh	9

- Nhận xét: điểm thi đã được mã hóa bằng public key từ client nên dữ liệu sẽ được giữ bí mật trong khi truyền lên server. Chỉ Client có private key mới có thể giải mã được dữ liệu điểm.