THỰC HÀNH TUẦN #3-4-5

KẾT NỐI CSDL SỬ DỤNG

STORE PROCEDURE + ADO.NET

GV: Hồ Thị Hoàng Vy

Lê Nguyễn Hoài Nam

Hoàng Anh Tú

Nguyễn Trường Sơn

Phạm Xuân Quang

-

MỤC LỤC

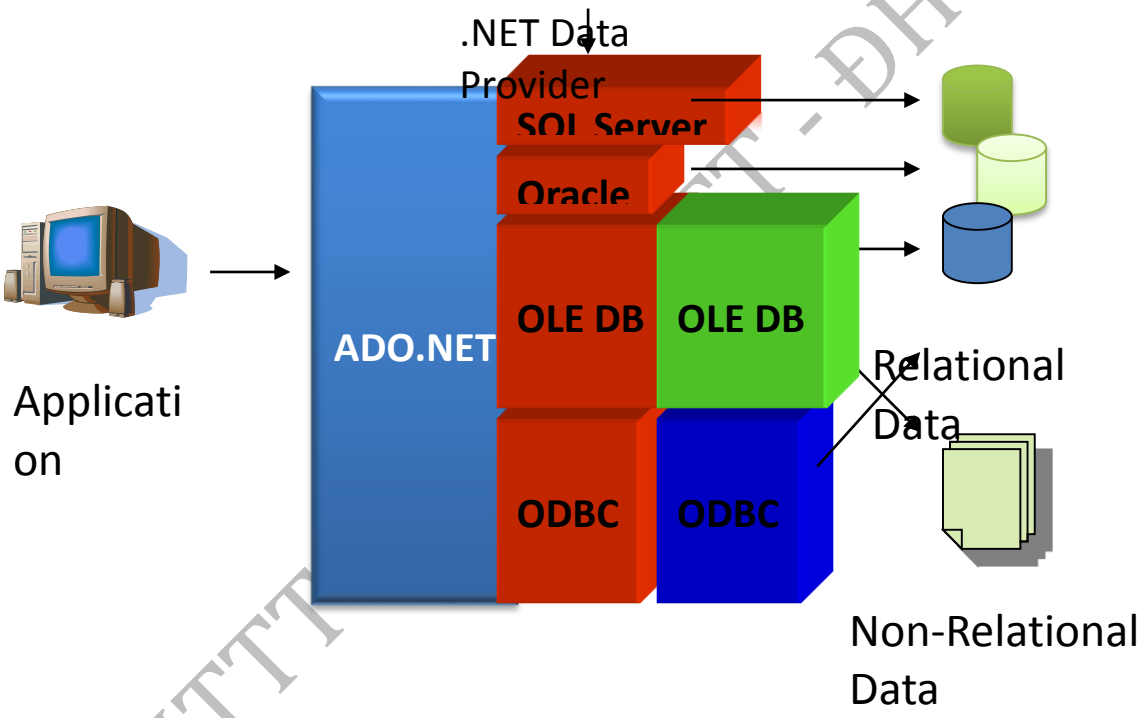
1 Mục tiêu và tóm tắt nội dung

Sau khi hoàn thành bài tập này sinh viên có thể:

- Xây dựng được một ứng dụng (C#) window form kết nối CSDL với các chức năng cơ bản

2 Hướng dẫn cụ thể

2.1 Kiến trúc ADO.Net



ADO.NET là một phần của .NET Framework, nó được xem là "bộ thư viện lớp" chịu trách nhiệm xử lý dữ liệu trong ngôn ngữ MS.NET. ADO.NET được thiết kế với dạng dữ liệu "ngắt kết nối", nghĩa là chúng ta có thể lấy cả một cấu trúc phức tạp của dữ liệu từ database, sau đó ngắt kết nối với database rồi mới thực hiện các thao tác cần thiết. Đây là một sự tiến bộ về mặt thiết kế bởi vì thiết kế ADO trước đây luôn cần duy trì một kết nối trong quá trình thao tác dữ liệu.

Có thể coi ADO.NET là một thế hệ tiếp theo của ADO. ADO.NET kế thừa tất cả những ưu điểm của ADO, đồng thời với ý tưởng thiết kế hoàn toàn mới ADO.NET

có một diện mạo khác hẳn so với tiền thân của nó. Một vài đặc điểm nổi bật của ADO.NET mà ADO không có như sau:

ADO.NET được thiết kế hoàn toàn dựa vào XML vì XML là chuẩn trao đổi dữ liệu tiến bộ và tốt nhất trên môi trường Internet hiện nay. ADO.NET được thiết kế hoàn toàn hướng đối tượng : đây là đặc điểm chi phối toàn bộ các sản phẩm Microsoft .NET.

2.2 DataSet

Là thành phần chính cho đặc trưng kết nối không liên tục (ngắt kết nối) của kiến trúc ADO.NET.

DataSet được thiết kế để thích ứng với bất kì nguồn dữ liệu nào.

Nhiệm vụ của DataSet là nhận dữ liệu về từ DataAdapter và xử lý nó.

Lưu trữ dữ liệu của DataBase trong bộ nhớ.

Mọi thao tác thay đổi dữ liệu được thực hiện trên DataSet, không làm ảnh hưởng đến DataBase.

Theo vết các thay đổi trên dữ liệu và có thể cập nhật dữ liệu ngược vào DataBase thông qua SqlDataAdapter.

Gồm các đối tượng : DataTable, DataRelationship, Constraint.

Việc sử dụng DataSet là một tiến bộ lớn của kiến trúc ADO.NET tuy nhiên với các ứng dụng Web việc sử dụng DataSet không được khuyến khích vì đối tượng DataSet được xem là quá lớn, nặng nề khó thích hợp cho đường truyền trên web vốn rất hạn chế.

2.2.1 Các thuộc tính của DataSet

STT	Tên Thuộc Tính	Ý Nghĩa
1	DataSetName	Tên của dataset
2	Relations	

3	Tables

2.2.2 Các phương thức chính của DataSet

STT	Tên Phương Thức	Ý Nghĩa
1	GetChange	
2	RejectChanges	
3	AcceptChanges	
4	GetXML	
5	ReadXML	
6	WriteXML	

2.3 SQL Server Data Provider

Các lớp chính của SQL Server Data Provider

Tên Lớp	Ý Nghĩa
SqlCommand	Thực thi SQL queries, câu lệnh hoặc lưu trữ thủ tục
SqlConnection	Tạo kết nối tới SQL Server
SqlDataAdapter	Cầu nối trung gian giữa dataset và data source
SqlReader	Cung cấp một data stream tới kết quả
SqlError	Lưu trữ thông tin về lỗi và cảnh cáo (warning)
SqlException	Các ngoại lệ trong trường hợp SQL Server lỗi và cảnh báo
SqlParameter	Tham sô biên command
SqlTransaction	Transaction của SQL Server

2.3.1 Lớp đối tượng SqlConnection

Dùng để tạo kết nối đến các CSDL Sql Server

2.3.1.1 Các thuộc tính của SqlConnection

STT	Tên Thuộc Tính	Ý Nghĩa
1	ConnectionString	Chuỗi kết nối database
2	ConnectionTimeout	Thời gian chờ trước khi ngắt kết nối với database
3	Container	Icontainer chứa các Component
4	Database	Tên của database hiện tại sau khi kết nối

5	DataSource	Tên của server, tên file chứa dữ liệu
6	Provider	Tên của OLEDB provider
7	ServerVersion	Version của server
8	Site	Isite của Component
9	State	Trạng thái của liên kết

2.3.1.2 Các phương thức chính của SqlConnection

STT	Tên Phương Thức	Ý Nghĩa
1	SqlConnection	Phương thức khởi tạo
2	Open	Mở kết nối tới database
3	Close	Đóng kết nối với database
4	Dispose	Hủy đối tượng
5	BeginTransaction	Bắt đầu 1 transaction
6	Commit	Kết thúc 1 transaction

Cách tạo một kết nối:

Cách 1:

- Tạo chuỗi kết nối
- Tạo đối tượng kết nối SqlConnection, truyền tham số chuỗi kết nối vào.

Ví dụ:

```
string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection(CnStr);
```

Cách 2:

- Tạo chuỗi kết nối
- Tạo đối tượng kết nối SqlConnection, không truyền tham số
- Trỏ thuộc tính ConnectionString của đối tượng SqlConnection đến chuỗi kết nối.

Ví dụ:

```
string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection();
Cn.ConnectionString = CnStr;
Cn.Open();
Cn.Close();
```

Các tham số của SqlConnection

Tham số	Mô tả
Data Source	Tên máy hoặc IP
Initial Catalog	Tên Database
Integrated Security	Sử dụng SSPI
User ID	Tên User kết nối
Password	Mật khẩu kết nối

Lưu ý:

- Windows Authenticate: có nghĩa là sử dụng quyền hạn của Windows Account để truy cập SQl Server. Khi kết nối ở mode này, thì không cần truyền userID, password.
- Để kết nối bằng Window Authencation, trong chuỗi kết nối cho thêm thuộc tính: "Integrated Security=true" hoặc là: "Integrated Security=SSPI"
- Ví dụ:

  Data Source=localhost;Initial Catalog=myDB;Integrated Security=SSPI;
- Sql server account: truyền thông tin account (username, password) vào chuỗi kết nối
- Ví dụ:

  string CnStr = "Server=.;Database=TestDB;UID=sa;PWD=sa";

2.3.2 Lớp đối tượng SqlCommand

Đối tượng thực hiện các câu lệnh tương tác truy vấn, rút trích dữ liệu từ database khi đã thiết lập kết nối tới dữ liệu và trả về kết quả.

Kết quả trả về được lưu trữ dưới dạng luồng thông qua 2 đối tượng :

- DataReader
- DataSet thông qua một đối tượng SqlDataAdapter

Các hàm khởi tạo

```
new SqlCommand()
new SqlCommand(cmdText)
new SqlCommand(cmdText, connection)
new SqlCommand(cmdText, connection, transaction)
```

Ví dụ 1:

String sql = “Select * from SinhVien”

SqlCommand cmd = new SqlCommand();

Cmd.CommandText = sql;

Ví dụ 2:

String sql = “Select * from SinhVien”

SqlCommand cmd = new SqlCommand(sql);

Ví dụ 3:

String CnStr = “Server=.;Database=”SinhVienDB”;uid=sa;pwd=sa;”;

SqlConnection cn = new SqlConnection(CnStr);

String sql = “Select * from SinhVien”

SqlCommand cmd = new SqlCommand(sql,cn);

Ví dụ 4:

String CnStr = “Server=.;Database=”SinhVienDB”;uid=sa;pwd=sa;”;

SqlConnection cn = new SqlConnection(CnStr);

String sql = “Select * from SinhVien”

SqlCommand cmd = new SqlCommand(sql,cn,null);

2.3.2.1 Các thuộc tính của SqlCommand

STT	Tên Thuộc Tính	Ý Nghĩa
1	CommandText	Câu lệnh SQL hay stored procedure kết nối data source
2	CommandTimeout	Thời gian chờ trước khi ngắt kết nối
3	CommandType	Giá trị mô tả hoạt động của CommandText
4	Connection	Thiết lập SqlConnection
5	Container	IContainer chứa Component
6	DesignTimeVisible	Giá trị mô tả đối tượng command xuất hiện trong Designer
7	Parameters	Lấy các tham số
8	Site	ISite của Component
9	Transaction	SqlTransaction khi SqlCommand thực thi
10	UpdatedRowSource	Update DataRow

2.3.2.2 Các phương thức chính của SqlCommand

STT	Tên Phương Thức	Ý Nghĩa
1	SqlCommand	Phương thức khởi tạo
2	ExecuteReader	Gửi CommandText đến Kết nối để tạo ra SqlDataReader
3	ExecuteNonQuery	Trả về số lượng dòng bị ảnh hưởng trên CSDL
4	ExecuteScalar	Trả về 1 giá trị đầu tiên (VD: giá trị tính tổng)
5	ExecuteXMLReader	Trả về 1 XMLReader
6	Dispose	Hủy đối tượng

2.3.3 Lớp đối tượng SqlParameter

Cần thực hiện câu lệnh nhiều lần với các giá trị khác nhau.

Đối tượng tham số truyền vào cho đối tượng SqlCommand

Có các thuộc tính sau :

STT	Tên Thuộc Tính	Ý Nghĩa
1	ParameterName	Tên tham số
2	SqlDbType	Kiểu dữ liệu của tham số tương ứng với kiểu dữ liệu của SqlServer
3	Direction	Input, Output, InputOutput, ReturnValue, …
4	Size	Kích thước tối đa của dữ liệu
5	Value	Giá trị của tham số (input / Output)

Để truyền tham số cho command:

- Khai báo đối tượng command với tham số:

```
SqlCommand cmd = new SqlCommand ( "select * from Sach where SachID = @SachID ", conn);
```

- Định nghĩa các tham số được dùng ở đối tượng command:

```
SqlParameter param = new SqlParameter();
param.ParameterName = "@SachID ";
param.Value = sachID;
```

- Thêm mới tham số vào đối tượng command:

```
cmd.Parameters.Add(param);
```

Ví dụ: truyền nhiều tham số

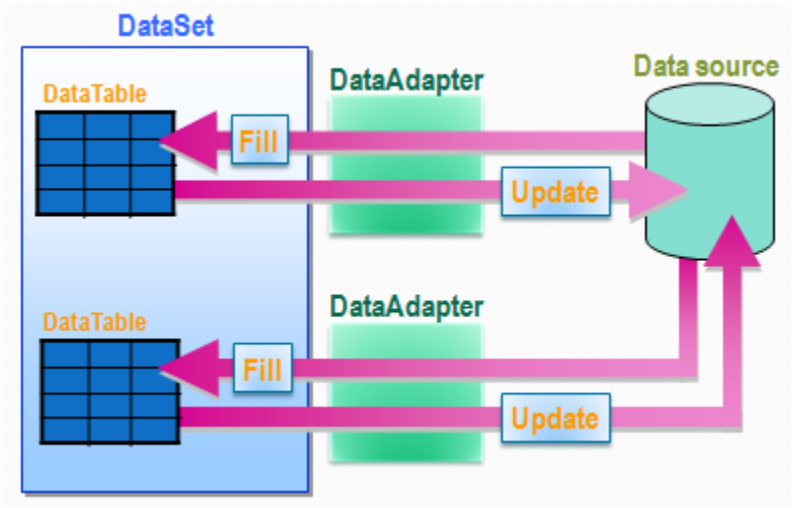
```
String CnStr = "Server=localhost;Database=DBTest;UID=sa;PWD=sa";
SqlConnection = new SqlConnection(CnStr);
String cmdText = "INSERT INTO Student(Name,Address, Age) VALUES(@Name, @Address, @Age)";
SqlCommand cmd = new SqlCommand(cmdText,Cn);

cmd.Parameters.Add("@Name", SqlDbType.NVarChar);
cmd.Parameters.Add("@Address", SqlDbType.NVarChar);
cmd.Parameters.Add("@Age", SqlDbType.Int)

cmd.Parameters["@Name"].Value = txtName.Text;
cmd.Parameters["@Address"].Value = txtAddress.Text;
cmd.Parameters["@Age"].Value = ConVert.ToInt32(txtAge.Text);

Cn.Open();
– //Thực thi câu lệnh và đóng kết nối.
cmd.ExecuteNonQuery();
Cn.Close();
```


2.3.4 Lớp đối tượng SqlDataAdapter

Đây là đối tượng rất quan trọng của ADO.NET, nó là cầu nối trung gian của database và dataset (dataset là đối tượng ngắt kết nối), bởi vì đối tượng "ngắt kết nối" dataset không thể liên lạc trực tiếp với database nên nó cần một đối tượng trung gian lấy dữ liệu từ database cho nó

Cung cấp các phương thức và thuộc tính để lấy và lưu dữ liệu giữa DataSet và CSDL

Sử dụng DataSet để lưu trữ dữ liệu, đồng thời, cho cập nhật dữ liệu ngược lại vào Database



Các phương thức chính :

STT	Tên Phương Thức	Ý Nghĩa
1	Fill	Lấy dữ liệu từ data source
2	Update	Cập nhật dữ liệu vào data source

Sử dụng Stored Procedure để truy xuất dữ liệu:

- Trước đây khi dùng Query Analyzer chúng ta có thể đặt tên và save các nhóm câu lệnh SQL vào một file dưới dạng script để có thể sử dụng trở lại sau này. Tuy nhiên thay vì save vào text file ta có thể save vào trong SQL Server dưới dạng Stored Procedure.
- Stored Procedure là một nhóm câu lệnh Transact-SQL đã được compiled (biên dịch) và chứa trong SQL Server dưới một tên nào đó và được xử lý như một đơn vị (chứ không phải nhiều câu SQL riêng lẻ).

Cách dùng:

- Tạo stored procedure trong sqlServer
- Tạo Sqlcommand với giá trị command text là tên Stored procedure
    - SqlCommand cmd = new SqlCommand("Ten stored",cn)

❖ Gán commandType:

- cmd.CommandType = CommandType.StoredProcedure

❖ Truyền các tham số vào command nếu StoredProcedure có yêu cầu

❖ Thực thi command

Ví dụ 1:

Ví dụ 1: Stored procedure trả về kết quả của lệnh Select

```
string procName = "sp_LoadAccType";
string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection(CnStr);

SqlDataAdapter da = new SqlDataAdapter(procName, Cn);
DataTable dt = new DataTable();
da.Fill(dt);
return dt;
```

Ví dụ 2:

Ví dụ 2: Stored procedure trả về kết quả của lệnh Insert/Delete/Update

```
string CnStr = "Server=.;Database=AgriBankDB;UID=sa;PWD=hoangvy";
        SqlConnection Cn = new SqlConnection(CnStr);
        string procName = "sp_InsertAccTypes";
        SqlCommand cmd = new SqlCommand(procName, Cn);
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("@AccTypeName", SqlDbType.NVarChar);
        cmd.Parameters.Add("@Note", SqlDbType.NVarChar);

        cmd.Parameters["@AccTypeName"].Value = txtAccTypeName.Text;
        cmd.Parameters["@Note"].Value = txtNote.Text;

        cmd.Connection.Open();
        cmd.ExecuteNonQuery();
        cmd.Connection.Close();
```

Ví dụ 3:

Ví dụ 3: Stored procedure có dùng tham số output

```
CREATE PROC [dbo].[sp_InsertAccTypes]
        @AccTypeName nvarchar(50),
        @Note nvarchar(255),
        @AccID numeric output,
        @err nvarchar(255) output
AS
BEGIN TRAN

INSERT INTO AccTypes(AccTypeName,Note)
VALUES(@AccTypeName,@Note)
SET @AccID = @@IDENTITY
SET @err = ''
COMMIT TRAN
```

Gọi stored procedure và có truyền tham số:

```
string CnStr = "Server=.;Database=AgriBankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection(CnStr);
string procName = "sp_InsertAccTypes";
SqlCommand cmd = new SqlCommand(procName, Cn);
cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@AccTypeName", SqlDbType.NVarChar);
cmd.Parameters.Add("@Note", SqlDbType.NVarChar);
cmd.Parameters.Add("@AccID", SqlDbType.Int).Direction = ParameterDirection.Output;

cmd.Parameters["@AccTypeName"].Value = txtAccTypeName.Text;
cmd.Parameters["@Note"].Value = txtNote.Text;
cmd.Parameters["@AccID"].Value = -1

cmd.Connection.Open();
cmd.ExecuteNonQuery();
cmd.Connection.Close();
string err = cmd.Parameters["@err"].Value.ToString();
int AccId = (int)cmd.Parameters["@AccId"].Value.;
```

Ví dụ 4:

Ví dụ 4: Stored procedure có giá trị trả về

```
Create Proc TinhTong2
        @soA int,
        @soB int
AS
BEGIN
declare @c int
        set @c= @soA + @SoB
return @c
END
```

```
string procName = "TinhTong2";
        string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
        SqlConnection Cn = new SqlConnection(CnStr);
        SqlCommand cmd = new SqlCommand(procName, Cn);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.Add("@SoA", SqlDbType.Int);
        cmd.Parameters.Add("@SoB", SqlDbType.Int);
        cmd.Parameters["@SoA"].Value = 2;
        cmd.Parameters["@SoB"].Value = 3;

        SqlParameter p = new SqlParameter("@c", SqlDbType.Int);
        p.Direction = ParameterDirection.ReturnValue;
        cmd.Parameters.Add(p);

        Cn.Open();
        cmd.ExecuteNonQuery();
        Cn.Close();

        MessageBox.Show(cmd.Parameters["@c"].Value.ToString());
```

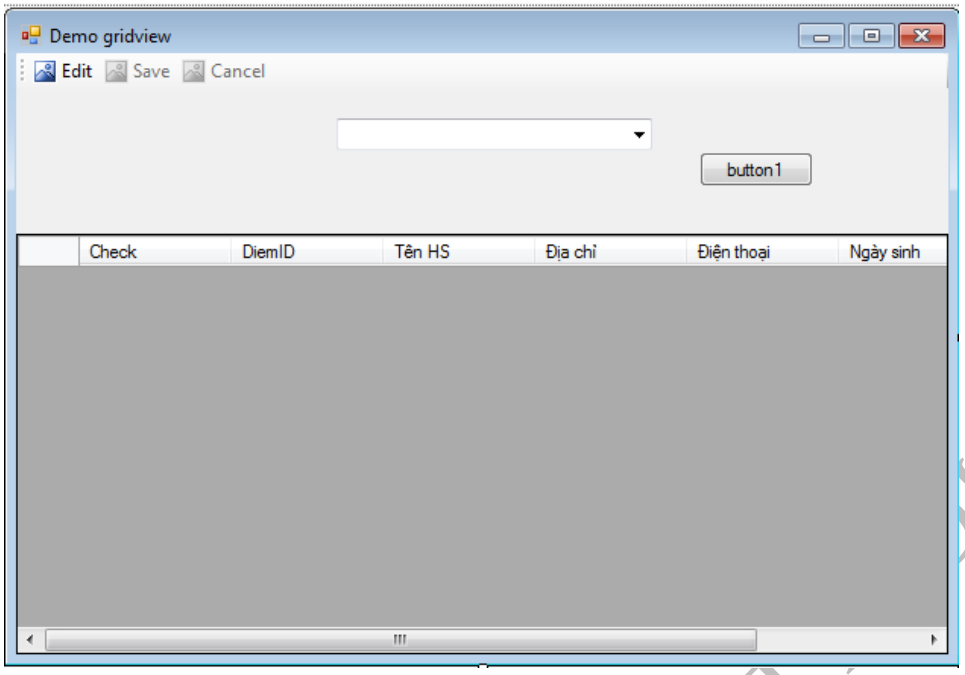
2.4 Bài tập lớp

Chuẩn bị csdl như sau:

```
drop proc sp_InsertHocSinh
go
create proc sp_InsertHocSinh
        @TenHS nvarchar(255),
        @DiaChi nvarchar(255),
        @DienThoai varchar(11),
        @NgaySinh datetime,
        @MaLop int,
        @MaHS int output
as
begin
        Insert into HocSinh(TenHS,DiaChi,DienThoai,NgaySinh,MaLop)
Values(@TenHS,@DiaChi,@DienThoai,@NgaySinh,@MaLop)
        Select @MaHS = @@Identity
end
GO

drop proc sp_DeleteHocSinh
```

```
GO

create proc sp_DeleteHocSinh
	@MaHS int
as
begin
	Delete From HocSinh where MaHS=@MaHS
end
GO

drop proc sp_UpdateHocSinh
go

create proc sp_UpdateHocSinh
	@MaHS int,
	@TenHS nvarchar(255),
	@DiaChi nvarchar(255),
	@DienThoai varchar(11),
	@NgaySinh datetime,
	@MaLop int

as
begin
	Update HocSinh set
TenHS=@TenHS,DiaChi=@DiaChi,DienThoai=@DienThoai,NgaySinh=@NgaySinh
,MaLop=@MaLop where MaHS=@MaHS
end
GO
```

Thiết kế giao diện:



Code behind:

```
public partial class frmMain : Form
    {
        const string cnStr = "Server=.; Database=QLSV; Integrated
Security=SSPI;";
        DataTable dt = new DataTable();

        public frmMain()
        {
            InitializeComponent();
        }

        private void frmMain_Load(object sender, EventArgs e)
        {
            SqlConnection cn = new SqlConnection(cnStr);

            SqlDataAdapter da = new SqlDataAdapter("select * from
Lop", cn);
            DataTable dtLop = new DataTable();
            da.Fill(dtLop);

            colMaLop.DataSource = dtLop;
            colMaLop.DisplayMember = "TenLop";
            colMaLop.ValueMember = "MaLop";
```

```
            cboLop.DataSource = dtLop;
            cboLop.DisplayMember = "TenLop";
            cboLop.ValueMember = "MaLop";


            da = new SqlDataAdapter("select * from Diem", cn);
            DataTable dtDiem = new DataTable();
            da.Fill(dtDiem);

            colID.DataSource = dtDiem;
            colID.DisplayMember = "DiemThi";
            colID.ValueMember = "id";


            //da = new SqlDataAdapter("select * from HocSinh",
cn);
            da.SelectCommand.CommandText = "select * from
HocSinh";
            da.Fill(dt);

            grd.DataSource = dt;
        }

        private void tsbEdit_Click(object sender, EventArgs e)
        {
            ButtonStatus(false);
        }

        private void ButtonStatus(bool status)
        {
            tsbEdit.Enabled = status;
            tsbSave.Enabled = tsbCancel.Enabled = !status;

            grd.AllowUserToAddRows = !status;
            grd.ReadOnly = status;
            grd.AllowUserToDeleteRows = !status;
        }

        private void tsbSave_Click(object sender, EventArgs e)
        {
            SqlConnection cn = new SqlConnection(cnStr);
            cn.Open();

            foreach (DataRow row in dt.Rows)
```

```
            {
                switch (row.RowState)
                {
                    case DataRowState.Added:
                        InsertHocSinh(row, cn);
                        break;
                    case DataRowState.Modified:
                        UpdateHocSinh(row, cn);
                        break;
                    case DataRowState.Deleted:
                        DeleteHocSinh(row, cn);
                        break;
                    default: break;
                }
            }

            cn.Close();
            dt.AcceptChanges();

            ButtonStatus(true);
        }

        private void InsertHocSinh(DataRow row, SqlConnection cn)
        {
            string TenHS = row["TenHS"].ToString();
            string DiaChi = row["DiaChi"].ToString();
            string DienThoai = row["DienThoai"].ToString();
            DateTime                    NgaySinh                    =
Convert.ToDateTime(row["NgaySinh"]);
            int MaLop = Convert.ToInt32(row["MaLop"]);

            string procName = "sp_InsertHocSinh";
            SqlCommand cmd = new SqlCommand(procName, cn);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.Add("@TenHS", SqlDbType.NVarChar);
            cmd.Parameters.Add("@DiaChi", SqlDbType.NVarChar);
            cmd.Parameters.Add("@DienThoai", SqlDbType.VarChar);
            cmd.Parameters.Add("@NgaySinh", SqlDbType.DateTime);
            cmd.Parameters.Add("@MaLop", SqlDbType.Int);
            cmd.Parameters.Add("@MaHS",   SqlDbType.Int).Direction
= ParameterDirection.Output;

            cmd.Parameters["@TenHS"].Value = TenHS;
            cmd.Parameters["@DiaChi"].Value = DiaChi;
```

```
            cmd.Parameters["@DienThoai"].Value = DienThoai;
            cmd.Parameters["@NgaySinh"].Value = NgaySinh;
            cmd.Parameters["@MaLop"].Value = MaLop;

            cmd.ExecuteNonQuery();

            int                         MaHS                        =
Convert.ToInt32(cmd.Parameters["@MaHS"].Value);
            row["MaHS"] = MaHS;
        }

        private void DeleteHocSinh(DataRow row, SqlConnection cn)
        {
            int      MaHS      =      Convert.ToInt32(row["MaHS",
DataRowVersion.Original]);//dong da xoa roi ko con nua, phai lay
lai phien ban cu~
            string procName = "sp_DeleteHocSinh";

            SqlCommand cmd = new SqlCommand(procName, cn);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.Add("@MaHS", SqlDbType.Int);

            cmd.Parameters["@MaHS"].Value = MaHS;
            cmd.ExecuteNonQuery();
        }

        private void UpdateHocSinh(DataRow row,SqlConnection cn)
        {
            int MaHS = Convert.ToInt32(row["MaHS"]);
            string TenHS = row["TenHS"].ToString();
            string DiaChi = row["DiaChi"].ToString();
            string DienThoai = row["DienThoai"].ToString();
            DateTime                    NgaySinh                    =
Convert.ToDateTime(row["NgaySinh"]);
            int MaLop = Convert.ToInt32(row["MaLop"]);

            string procName = "sp_UpdateHocSinh";
            SqlCommand cmd = new SqlCommand(procName, cn);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.Add("@MaHS", SqlDbType.Int);
            cmd.Parameters.Add("@TenHS", SqlDbType.NVarChar);
            cmd.Parameters.Add("@DiaChi", SqlDbType.NVarChar);
            cmd.Parameters.Add("@DienThoai", SqlDbType.VarChar);
```

```
            cmd.Parameters.Add("@NgaySinh", SqlDbType.DateTime);
            cmd.Parameters.Add("@MaLop", SqlDbType.Int);

            cmd.Parameters["@MaHS"].Value = MaHS;
            cmd.Parameters["@TenHS"].Value = TenHS;
            cmd.Parameters["@DiaChi"].Value = DiaChi;
            cmd.Parameters["@DienThoai"].Value = DienThoai;
            cmd.Parameters["@NgaySinh"].Value = NgaySinh;
            cmd.Parameters["@MaLop"].Value = MaLop;

            cmd.ExecuteNonQuery();
        }

        private void tsbCancel_Click(object sender, EventArgs e)
        {
            dt.RejectChanges();
            ButtonStatus(true);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //int                          i                          =
(int)grd.SelectedRows[0].Cells["colCheck"].Value;
            //string                        s                         =
grd.SelectedRows[0].Cells["colCheck"].Value.ToString();
            MessageBox.Show("aaa");

        }
```

3 Bài tập về nhà

Sinh viên tự xây dựng lại ứng dụng QLSV với các chức năng trong bài hướng dẫn, suy nghĩ và phát triển một số chức năng khác cho ứng dụng.

HẾT