

# CƠ SỞ DỮ LIỆU NÂNG CAO – PHÂN TÍCH THIẾT KẾ HTTT

Tháng 5/2014

## LẬP TRÌNH KẾT NỐI CSDL – C#

Tóm tắt nội dung bài thực hành:

- Giới thiệu ADO.NET
- Ứng dụng minh họa

Bộ môn **Hệ thống thông tin**

Khoa Công nghệ thông tin

ĐH Khoa học tự nhiên TP HCM



## MỤC LỤC

1	Mục tiêu và tóm tắt nội dung .....	1
2	Hướng dẫn cụ thể.....	1
2.1	Giới thiệu tổng quan .....	1
2.2	Trình tự thao tác CSDL với ADO.Net.....	2
2.3	Đọc dữ liệu .....	3
2.4	Ghi dữ liệu .....	4
2.5	Ví dụ .....	5
3	Bài tập về nhà .....	11

## 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 Giới thiệu tổng quan

#### ➤ DataSet và DataTable

- DataSet là trung tâm của kiến trúc ADO.NET, mục tiêu là hỗ trợ hiệu quả thao tác trên dữ liệu từ nhiều nguồn, cũng như tương tác dữ liệu trong mô hình ứng dụng nhiều lớp (multiple tier). DataSet có thể được xem như là một cấu trúc dữ liệu để lưu trữ dữ liệu trong bộ nhớ chính.
- DataSet chứa một tập các đối tượng DataTable (có cấu trúc logic tương tự như một bảng trong CSDL), các ràng buộc trên chúng và cả mối quan hệ giữa các bảng này. Một đối tượng DataTable (hoặc một view của nó - thuộc lớp đối DataView), có thể được kết buộc với các control như ComboBox, DataList, DataGrid, ...
- Các lớp đối tượng DataSet, DataTable, DataView, DataColumn,... nằm trong namespace System.Data

#### ➤ DataProvider

- Trong namespace System.Data có 3 namespace tương ứng với 3 loại Data Provider: Data Provider for SQL Server (System.Data.SqlClient), Data Provider for ODBC (System.Data.Odbc) và Data Provider for OLE DB (System.Data.OleDb). Ở đây ta sử dụng SQL Server, nên sẽ sử dụng trực tiếp Data Provider for SQL Server (tất nhiên ta cũng có thể thông qua Provider for ODBC hoặc OLE DB để thao tác với CSDL SQL Server).
- Các lớp đối tượng chính để kết nối và thao tác với CSDL nằm trong namespace System.Data.SqlClient là: SqlConnection, SqlCommand,

SqlDataReader, SqlDataAdapter. (Tương tự đối với hai Provider còn lại).

#### ❖ **SQLConnection**

Một đối tượng thuộc lớp này thể hiện một kết nối đến CSDL. Các thông số để kết nối được chỉ định trong Connection String

Ví dụ:

```
ConnectionString = "Data Source = .; Initial Catalog = Northwind;  
Integrated Security = SSPI" //dùng Window Authentication
```

hoặc :

```
ConnectionString = "Data Source = .; Initial Catalog = Northwind;  
User ID = ws01; Password = "  
(dấu "." chỉ local host)
```

#### ❖ **SQLCommand**

Một đối tượng thuộc lớp này thể hiện một lệnh thực thi trên HTQ CSDL. Có thể thiết lập thuộc tính CommandType của đối tượng Command để chỉ ra lệnh được khai báo ở dạng text hay là tên stored procedure.

#### ❖ **SQLDataReader**

Lớp đối tượng dùng để đọc kết quả truy vấn được từ CSDL. Có thể xem SqlDataReader là một RecordSet chỉ có thể đọc và đọc tuần tự một chiều.

#### ❖ **SQLDataAdapter**

Một đối tượng Data Adapter có thể xem như một cầu nối giữa DataSet và CSDL, để chuyển dữ liệu từ CSDL vào DataSet và cập nhật những thay đổi trên DataSet trở lại vào CSDL

## 2.2 Trình tự thao tác CSDL với ADO.Net

- Tạo lập và thiết lập các thông số cho đối tượng Connection (nếu chưa thiết lập trước đó)
- Mở kết nối bằng phương thức Open của đối tượng Connection

- Thực hiện các công việc đọc/ghi với CSDL vừa kết nối tới
- Đóng kết nối

**Lưu ý:**

- ✓ Không phải luôn luôn mở và đóng kết nối mỗi khi thực hiện một lệnh, có thể mở kết nối một lần và thực hiện nhiều lệnh trước khi đóng nó.
- ✓ Luôn sử dụng try và catch để bắt các lỗi phát sinh từ CSDL khi thực hiện các lệnh mở kết nối hoặc thực thi lệnh trong đối tượng command, nếu có lỗi hiển thị thông điệp lỗi để hiểu cho NSD

### 2.3 Đọc dữ liệu

Trong phần này chỉ mô tả một số điểm cần lưu ý, sinh viên tự tìm hiểu cách thực hiện cụ thể trong các ví dụ và ebook được cung cấp hoặc MSDN.

➤ **DataReader**

- DataReader là một cách tốt (tiết kiệm tài nguyên - bộ nhớ) để đọc dữ liệu trong trường hợp chỉ cần lấy dữ liệu để hiển thị, không cần thao tác phức tạp hay thao tác trên nhiều tập dữ liệu (không cần đến datatable hay dataset)

➤ **DataAdapter**

- Như đã nói ở trên, ta có thể sử dụng DataAdapter như là một cầu nối để lấy dữ liệu từ CSDL vào DataSet. Ta có thể định nghĩa hoặc không định nghĩa trước bảng (đối tượng DataTable) và cấu trúc của bảng sẽ nhận dữ liệu. Nếu lệnh trả về n tập dữ liệu, n bảng tương ứng sẽ được tạo ra trong dataset để chứa các tập dữ liệu này.
- Để lấy dữ liệu, ta sử dụng phương thức Fill của đối tượng DataAdapter. Phương thức này có nhiều overload, hỗ trợ nhiều cách truyền tham số (DataSet, DataTable, DataSet và tên DataTable, ...).
- Phương thức Fill có thể được gọi mà không cần mở kết nối trước, trong trường hợp này, kết nối tương ứng sẽ được mở và đóng lại ngay sau khi thực hiện xong việc lấy dữ liệu.

- Ghi chú: Trong trường hợp muốn dữ liệu đổ vào đối tượng DataTable thỏa ràng buộc khóa chính (tự động loại bỏ dữ liệu trùng trên khóa chính nếu có), có thể thiết lập thuộc tính MissingSchemaAction của đối tượng DataAdapter là AddWithKey. Tuy nhiên, thiết lập lựa chọn này sẽ làm chậm đáng kể quá trình đọc dữ liệu, thay vào đó, nếu có thể nên định nghĩa trước cấu trúc và khóa chính của đối tượng DataTable này, rồi đọc bình thường, tác dụng cũng sẽ tương tự.

➤ **ExecuteScalar**

- Trong trường hợp câu truy vấn chỉ trả về một giá trị, sử dụng phương thức ExecuteScalar của đối tượng Command để thực thi truy vấn và nhận giá trị trả về.
- Ghi chú: Trong trường hợp gọi thực thi một thủ tục (với CommandType là stored procedure), giá trị của các tham số output khai báo trong Parameters của Command sẽ được cập nhật tương ứng

## 2.4 Ghi dữ liệu

➤ **ExecuteNonQuery**

- Cách thông dụng nhất để ghi dữ liệu là đưa trực tiếp các lệnh cập nhật dữ liệu (Insert, Update, Delete, lệnh tạo các đối tượng trong CSDL) (và hầu hết các lệnh T-SQL khác), hoặc tên của stored procedure thực hiện các công việc này vào một đối tượng Command, sau đó gọi phương thức ExecuteNonQuery của đối tượng Command để thực hiện.

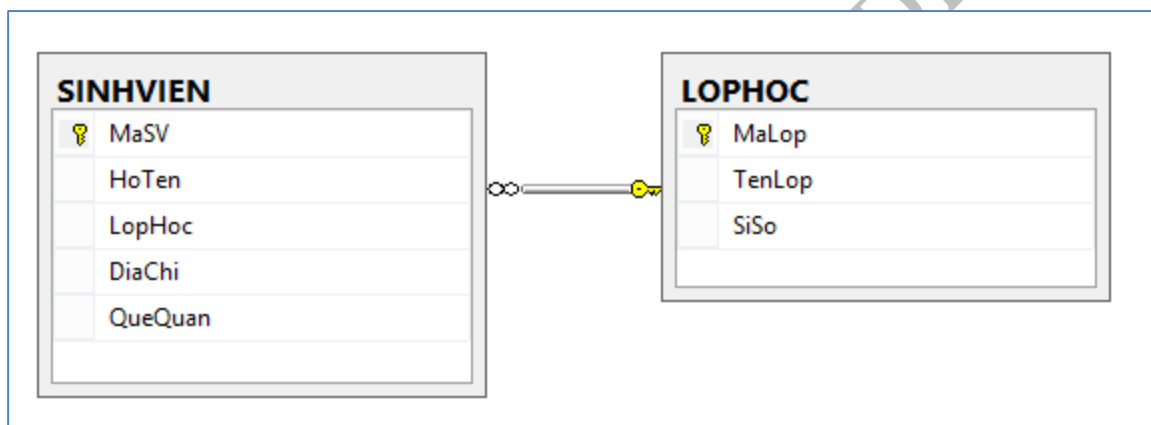
➤ **DataAdapter**

- DataAdapter thực hiện cập nhật CSDL theo cách ánh xạ những thay đổi trên DataSet vào CSDL (thành các thao tác insert, update, delete tương ứng)
- Cơ sở của việc ánh xạ này là việc quản lý tình trạng của các dòng (DataRow) trong DataTable, thể hiện qua thuộc tính RowState. Khi một dòng trong DataTable được thêm, xóa, cập nhật, nó sẽ có trạng thái tương ứng là Inserted, Deleted, và Updated, sau khi phương thức AcceptChanges được gọi, nó trở lại trạng thái UnChanged.

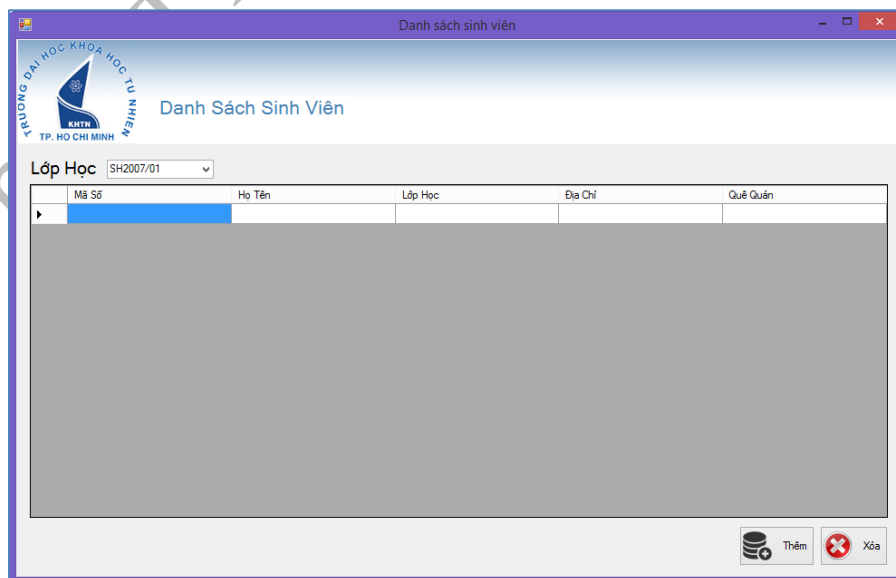
- Để thực hiện cập nhật ta gán InsertCommand, DeleteCommand và UpdateCommand cho đối tượng DataAdapter, sau đó gọi phương thức Update của nó. Khi đó, tất cả những thay đổi trên DataSet sẽ được ánh xạ về CSDL theo cách mà được chỉ định trong các Commands.
- Phương thức Update của DataAdapter cũng có nhiều overload, cho phép ta lựa chọn Update một dòng, một table hay toàn bộ DataSet.

## 2.5 Ví dụ

Cho CSDL QLSV với 2 bảng LOPHOC và SINHVIEN như hình:



Thiết kế form Danh sách sinh viên gồm combobox Lớp học và datagridview danh sách sinh viên như hình dưới:



Xây dựng ứng dụng QLSV trên .Net Framework 4.5 với các chức năng: xem danh sách sinh viên của mỗi lớp học, thêm/xóa sinh viên khỏi lớp học. Ứng dụng được thiết kế theo kiến trúc 3 lớp (GUI, BUS, DAO).

Hàm KetNoiCSDL (AbstractDAO.cs) sử dụng đối tượng SqlConnection với chuỗi kết nối được lưu ở file App.config là: "Data Source=.; Initial Catalog=QLSV; Trusted\_Connection=True"

```
protected SqlConnection KetNoiCSDL()
{
    return new SqlConnection()
    {
        ConnectionString = ConfigurationManager.ConnectionStrings["QuanLySinhVien"].ConnectionString
    };
}
```

➤ **Load danh sách lớp học vào combobox**

- Sự kiện load form Danh sách sinh viên gọi hàm HienThiDanhSachLopHoc để load danh sách các lớp học trong CSDL vào combobox Lớp học

```
private void fm_DSSV_Load(object sender, EventArgs e)
{
    HienThiDanhSachLopHoc();
}

private void HienThiDanhSachLopHoc()
{
    cb_DSlop.DisplayMember = "TenLop";
    cb_DSlop.ValueMember = "MaLop";
    LopHocBUS bus = new LopHocBUS();
    IList<LopHocDTO> dsLopHoc = bus.LayDanhSachLopHoc();
    cb_DSlop.Items.AddRange(dsLopHoc.ToArray<LopHocDTO>());
    if (cb_DSlop.Items.Count > 0)
    {
        cb_DSlop.SelectedIndex = 0;
    }
}
```

- Hàm LayDanhSachLopHoc (LopHocDAO.cs) sử dụng đối tượng SqlDataAdapter để đổ dữ liệu trong bảng LOPHOC vào đối tượng DataSet



```

public IList<LopHocDTO> LayDanhSachLopHop()
{
    List<LopHocDTO> dsLopHoc = new List<LopHocDTO>();
    string sql = @"SELECT * FROM LOPHOC";
    using (SqlConnection cnn = KetNoiCSDL())
    {
        cnn.Open();
        SqlCommand command = new SqlCommand(sql, cnn);
        SqlDataAdapter adapter = new SqlDataAdapter(command);
        DataSet dataset = new DataSet();
        adapter.Fill(dataset, "LOPHOC");
        foreach (DataRow r in dataset.Tables["LOPHOC"].Rows)
        {
            dsLopHoc.Add(new LopHocDTO()
            {
                MaLop = (string)r["MaLop"],
                TenLop = (string)r["TenLop"],
                SiSo = (int)r["SiSo"]
            });
        }
        cnn.Close();
    }
    return dsLopHoc;
}

```

- Danh sách các lớp học trong CSDL được load vào combobox Lớp học khi form Danh sách sinh viên được load lên

- Load danh sách sinh viên của mỗi lớp học

- Sự kiện click chọn một lớp trong danh sách lớp ở combobox Lớp học gọi hàm HienThiDanhSachSinhVien để load danh sách các sinh viên thuộc lớp học đó vào datagridview

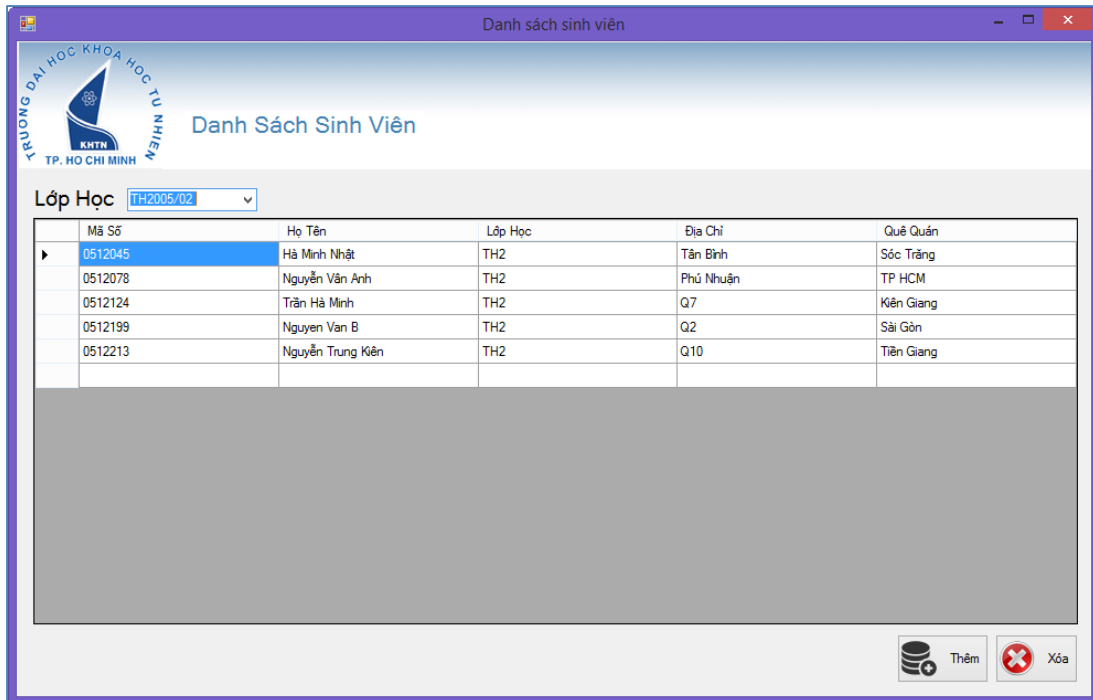
```
private void cb_DSlop_SelectedIndexChanged(object sender, EventArgs e)
{
    LopHocDTO lopHoc = (LopHocDTO)cb_DSlop.SelectedItem;
    grid_DSSV.Rows.Clear();
    HienThiDanhSachSinhVien(lopHoc.MaLop);
    grid_DSSV.Rows.Add();
}

private void HienThiDanhSachSinhVien(string maLop)
{
    SinhVienBUS bus = new SinhVienBUS();
    IList<SinhVienDTO> dsSinhVien = bus.LayDanhSachSinhVienTheoLop(maLop);
    foreach (SinhVienDTO sv in dsSinhVien)
    {
        grid_DSSV.Rows.Add(sv.MaSV, sv.HoTen, sv.LopHoc, sv.DiaChi, sv.QueQuan);
    }
}
```

- Hàm LayDanhSachSinhVienTheoLop (SinhVienDAO.cs) cũng sử dụng đối tượng SqlDataAdapter để đổ dữ liệu trả về vào đối tượng DataSet

```
public IList<SinhVienDTO> LayDanhSachSinhVienTheoLop(string maLop)
{
    List<SinhVienDTO> dsSinhVien = new List<SinhVienDTO>();
    string sql = string.Format("SELECT * FROM SINHVIEN WHERE [LopHoc]='{0}'", maLop);
    using (SqlConnection cnn = KetNoiCSDL())
    {
        cnn.Open();
        SqlCommand command = new SqlCommand(sql, cnn);
        SqlDataAdapter adapter = new SqlDataAdapter(command);
        DataSet dataset = new DataSet();
        adapter.Fill(dataset, "SINHVIEN");
        foreach (DataRow r in dataset.Tables["SINHVIEN"].Rows)
        {
            dsSinhVien.Add(new SinhVienDTO
            {
                MaSV = (string)r["MaSV"],
                HoTen = (string)r["HoTen"],
                LopHoc = (string)r["LopHoc"],
                DiaChi = (string)r["DiaChi"],
                QueQuan = (string)r["QueQuan"]
            });
        }
        cnn.Close();
    }
    return dsSinhVien;
}
```

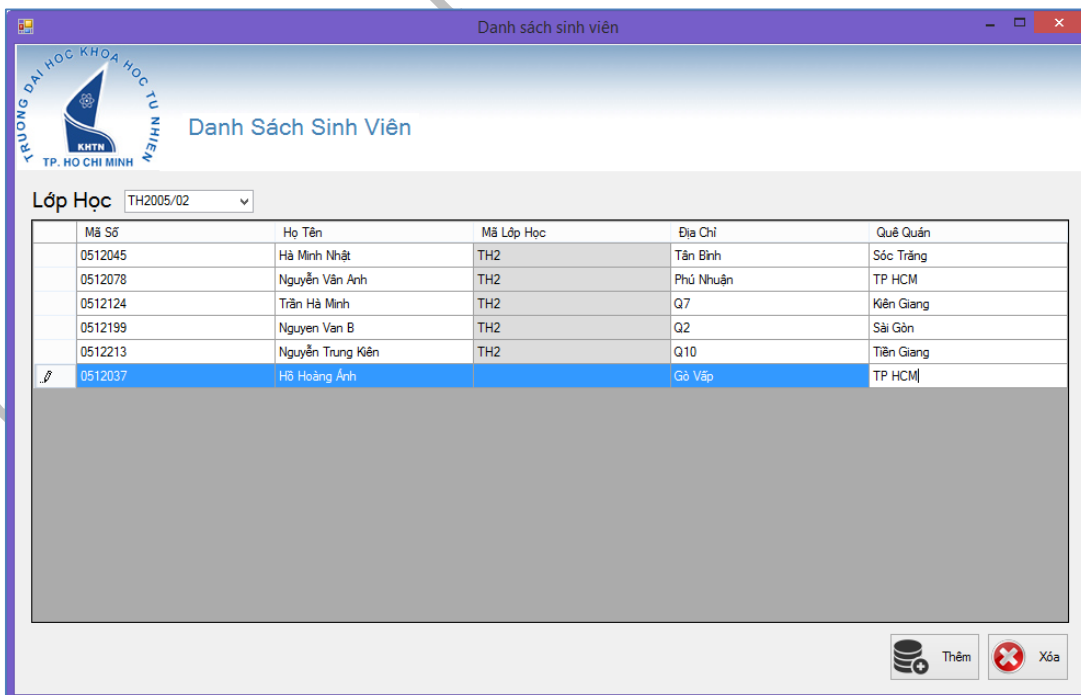
- Danh sách các sinh viên của mỗi lớp học được load vào datagridview



Mã Số	Họ Tên	Lớp Học	Địa Chỉ	Quê Quấn
0512045	Hà Minh Nhật	TH2	Tân Bình	Sóc Trăng
0512078	Nguyễn Văn Anh	TH2	Phú Nhuận	TP HCM
0512124	Trần Hà Minh	TH2	Q7	Kiên Giang
0512199	Nguyễn Văn B	TH2	Q2	Sài Gòn
0512213	Nguyễn Trung Kiên	TH2	Q10	Tiền Giang

#### ➤ Thêm sinh viên vào lớp học

Thêm sinh viên tên Hồ Hoàng Ánh vào lớp TH2005. Sự kiện click button Thêm sẽ gọi hàm ThemSinhVien (SinhVienDAO.cs)



Mã Số	Họ Tên	Mã Lớp Học	Địa Chỉ	Quê Quấn
0512045	Hà Minh Nhật	TH2	Tân Bình	Sóc Trăng
0512078	Nguyễn Văn Anh	TH2	Phú Nhuận	TP HCM
0512124	Trần Hà Minh	TH2	Q7	Kiên Giang
0512199	Nguyễn Văn B	TH2	Q2	Sài Gòn
0512213	Nguyễn Trung Kiên	TH2	Q10	Tiền Giang
0512037	Hồ Hoàng Ánh		Gò Vấp	TP HCM

```

public bool ThemSinhVien(SinhVienDTO sinhVien)
{
    string sql = string.Format("INSERT INTO SINHVIENT (MaSV, HoTen, LopHoc, DiaChi, QueQuan) VALUES (N'{0}', N'{1}', N'{2}', N'{3}', N'{4}']",
        sinhVien.MaSV, sinhVien.HoTen, sinhVien.LopHoc, sinhVien.DiaChi, sinhVien.QueQuan);
    try
    {
        using (SqlConnection cnn = KetNoiCSDL())
        {
            cnn.Open();
            SqlCommand command = new SqlCommand(sql, cnn);
            command.ExecuteNonQuery();
            cnn.Close();
        }
    }
    catch (Exception ex)
    {
        return false;
    }
    return true;
}

```

### ➤ Xóa sinh viên khỏi lớp học

Chọn sinh viên tên Nguyen Van B và xóa khỏi lớp TH2005/02. Sự kiện click button Xóa sẽ gọi hàm XoaSinhVien (SinhVienDAO.cs)

Danh sách sinh viên

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
TP. HO CHI MINH

Danh Sách Sinh Viên

Lớp Học: TH2005/02

Mã Số	Họ Tên	Mã Lớp Học	Địa Chỉ	Quê Quán
0512045	Hà Minh Nhật	TH2	Tân Bình	Sóc Trăng
0512078	Nguyễn Văn Anh	TH2	Phú Nhuận	TP HCM
0512124	Trần Hà Minh	TH2	Q7	Kiên Giang
0512199	Nguyễn Văn B	TH2	Q2	Sài Gòn
0512213	Nguyễn Trung Kiên	TH2	Q10	Tiền Giang
0512037	Hồ Hoàng Ánh	TH2	Gò Vấp	TP HCM

Thêm Xóa

```

public bool XoaSinhVien(SinhVienDTO sinhVien)
{
    string sql = string.Format("DELETE FROM SINHVIEN WHERE [MaSV] = '{0}'", sinhVien.MaSV);
    try
    {
        using (SqlConnection cnn = KetNoiCSDL())
        {
            cnn.Open();
            SqlCommand command = new SqlCommand(sql, cnn);
            command.ExecuteNonQuery();
            cnn.Close();
        }
    }
    catch (Exception ex)
    {
        return false;
    }
    return true;
}

```

### 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