



Microsoft®
Visual C#®

CHUYÊN ĐỀ NGÔN NGỮ LẬP TRÌNH 1

Người biên soạn: Hồ Quang Thái (MSCB: 2299)

BM. Công Nghệ Phần Mềm, Khoa CNTT&TT

Email: hqthai@cit.ctu.edu.vn

Số tín chỉ: 2 (20 LT + 20TH)



CHƯƠNG 5
ADO.NET

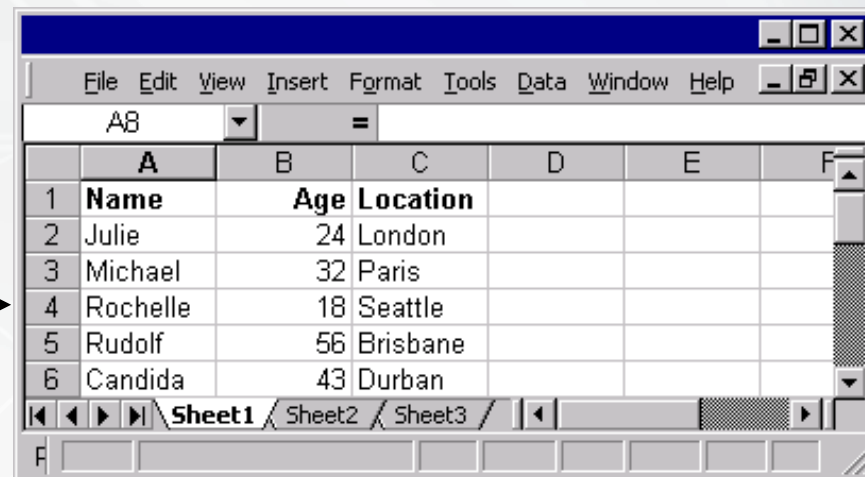
Nội dung

- *Các thành phần truy cập dữ liệu*
- *ADO.NET*
 - *Connection*
 - *Command*
 - *DataReader*
 - *DataAdapter*
 - *DataSet*
- *DataBinding*

Giới thiệu

Data

lưu
trong



	A	B	C	D	E	F
1	Name	Age	Location			
2	Julie	24	London			
3	Michael	32	Paris			
4	Rochelle	18	Seattle			
5	Rudolf	56	Brisbane			
6	Candida	43	Durban			

Cơ sở dữ liệu



Client

Các kỹ thuật
truy cập dữ liệu

Client-Server



Client



Server

Phần mềm Client

Sheet1

Name: Julie 1 of 5

Age: 24

Location: London

New

Delete

Restore

Find Prev

Find Next

Criteria

Close

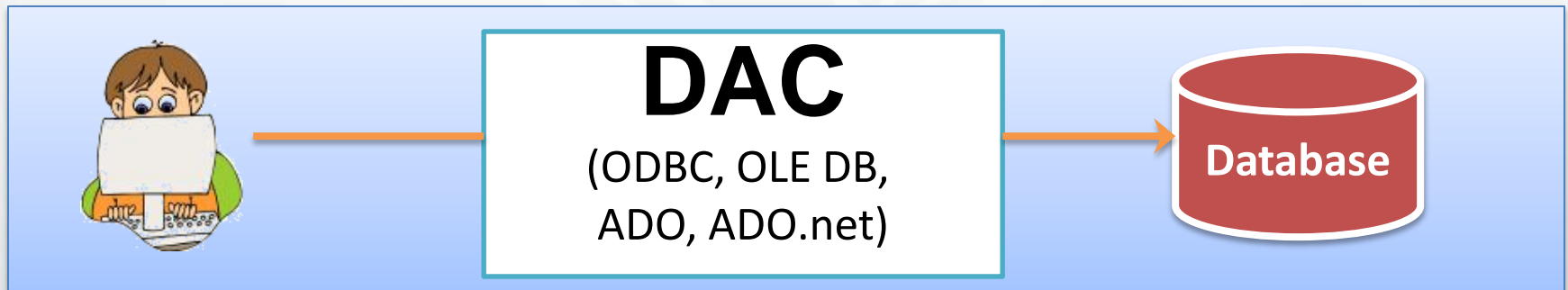
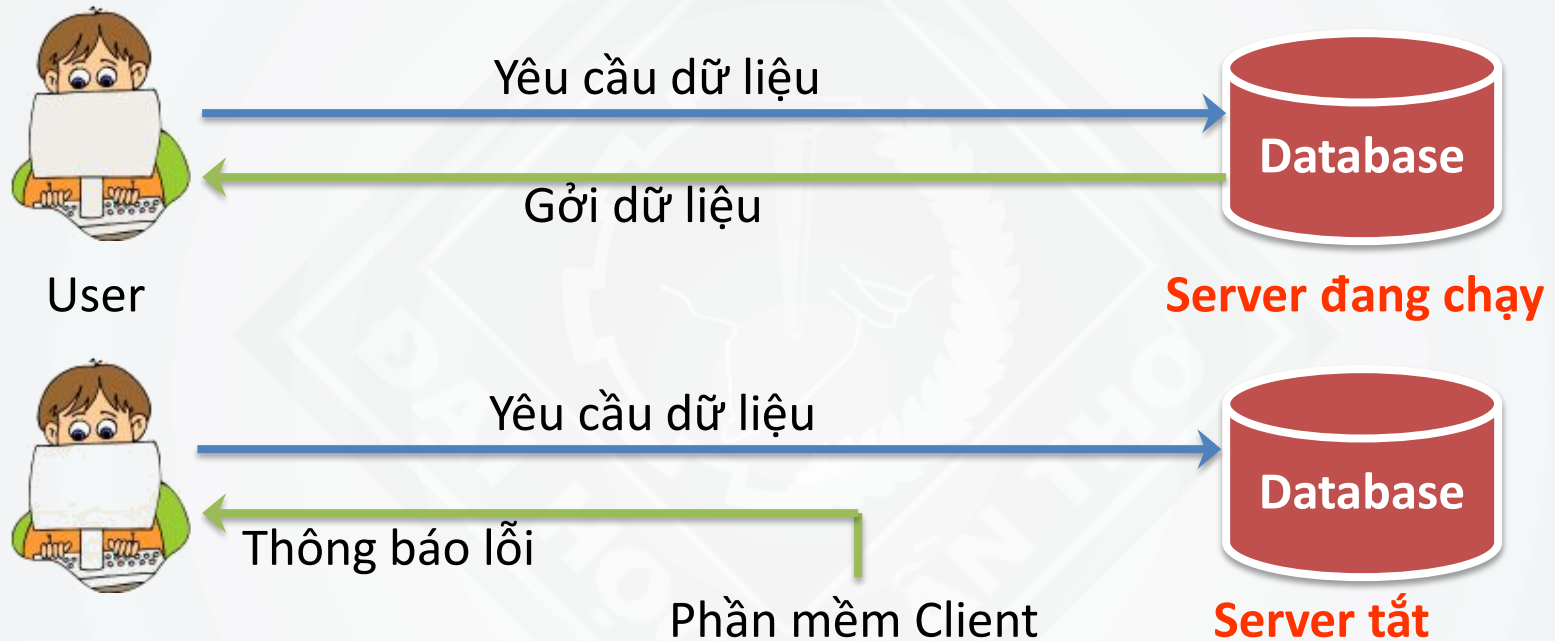
Cơ sở dữ liệu

File Edit View Insert Format Tools Data Window Help						
A8		=				
	A	B	C	D	E	F
1	Name	Age	Location			
2	Julie	24	London			
3	Michael	32	Paris			
4	Rochelle	18	Seattle			
5	Rudolf	56	Brisbane			
6	Candida	43	Durban			

Sheet1 Sheet2 Sheet3

Cung cấp dữ liệu
cho Client

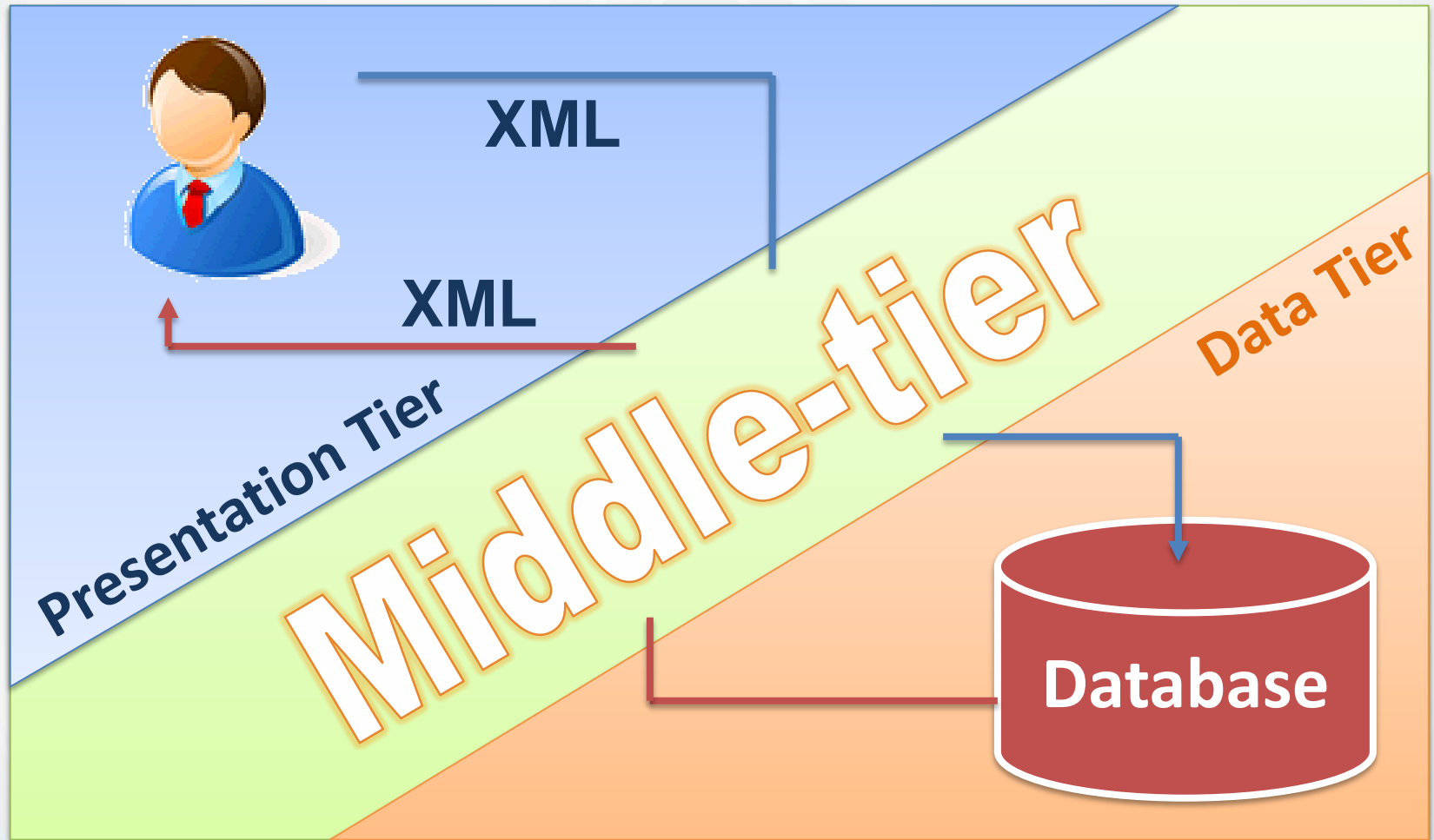
Data Access Components - Các thành phần truy cập dữ liệu



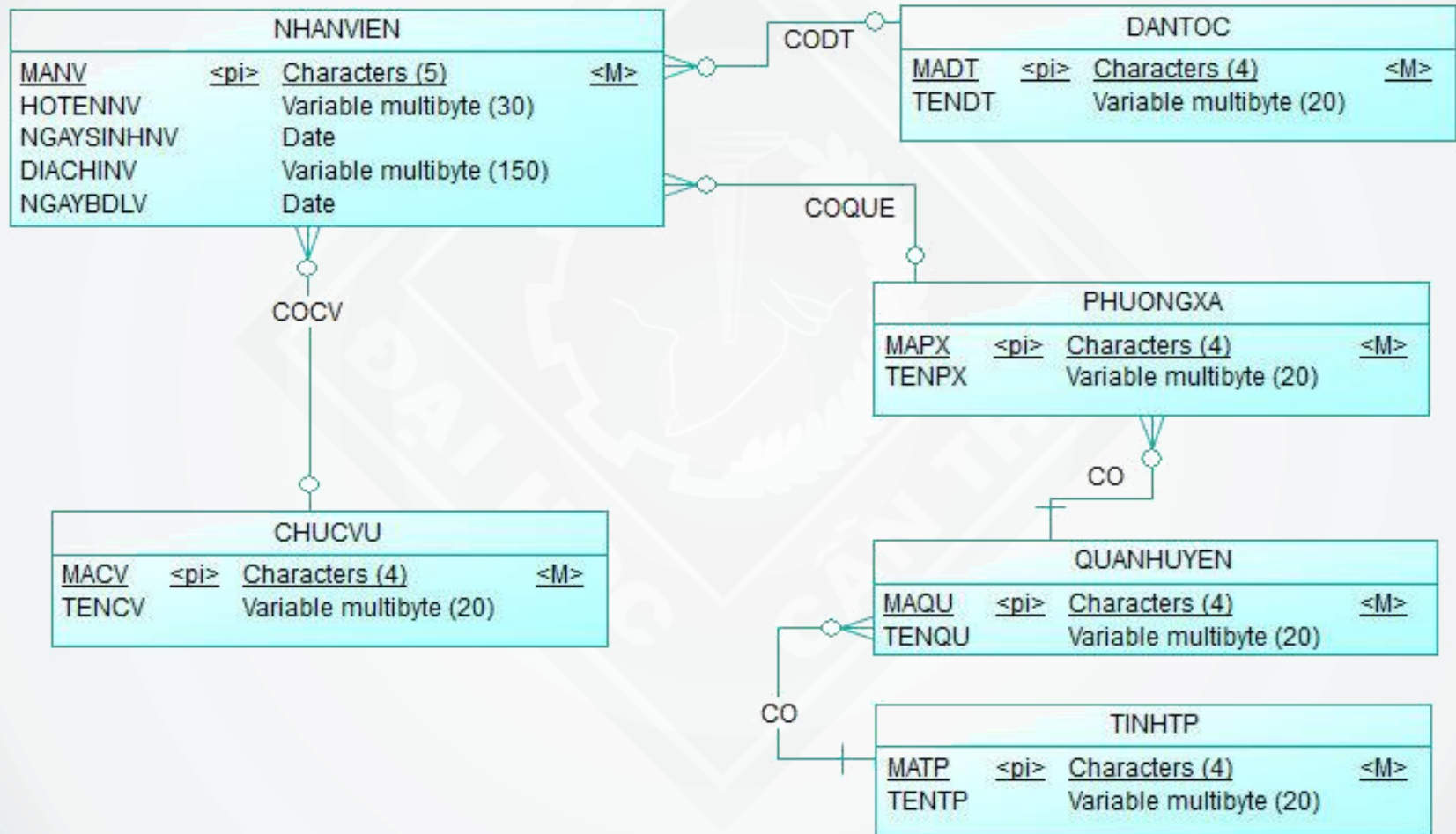
ADO.NET

- Là kỹ thuật truy cập cơ sở dữ liệu được hỗ trợ bởi thư viện lớp cơ sở của ***.NET Framework***.

Kiến trúc ADO.NET



Từ mô hình quan niệm đến CSDL

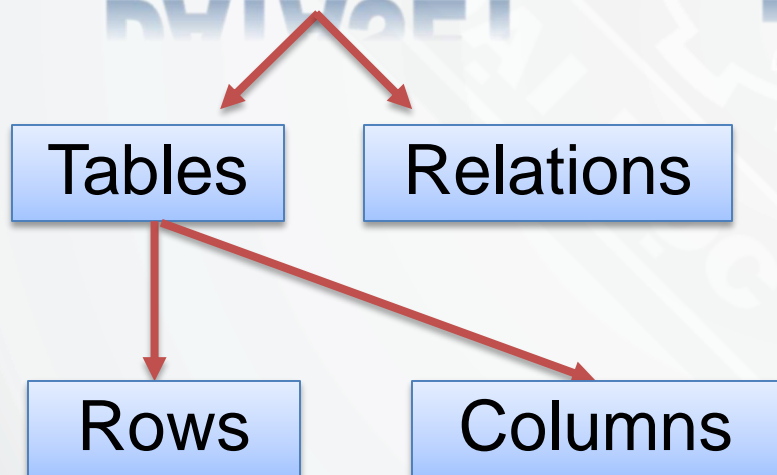


Các thành phần của ADO.NET

ADO.Net

DATASET

.NET DATA PROVIDER



- ✓ Connection
- ✓ Command
- ✓ DataReader
- ✓ DataAdapter

.NET Data Provider

- Trình cung cấp dữ liệu .NET - **.NET Data Provider**:
 - SQL .NET Framework Data Provider
 - OLE DB .NET Framework Data Provider
 - ODBC .NET Framework Data Provider
 - Oracle .NET Framework Data Provider

SQL .NET Framework Data Provider	OLE DB .NET Framework Data Provider
System.Data	System.Data
System.Data.SqlClient	System.Data.OleDb

Các lớp Connection

- Một đối tượng **Connection** thể hiện 1 kết nối đến cơ sở dữ liệu.
- Gồm:
 - Lớp **SqlConnection** (*SQL .Net Framework Data Provider*)
 - Lớp **OleDbConnection** (*OLEDB .Net Framework Data Provider*)

Các lớp Connection

- Thuộc tính
 - *ConnectionString*: chuỗi nối kết chứa thông tin về cơ sở dữ liệu.
- Phương thức
 - *Open()*: Mở nối kết
 - *Close()*: Đóng nối kết
 - *CreateCommand()*: Tạo một đối tượng Command nhờ một câu truy vấn.

Lớp SqlConnection

- Sử dụng phương thức xây dựng có tham số của lớp **SqlConnection**.
 - Tham số này là chuỗi kết nối (**ConnectionString**):
 - Tên Server lưu cơ sở dữ liệu (*Server*)
 - Tên cơ sở dữ liệu (*Database*)
 - Tài khoản, Mật khẩu (*uid, pwd*)
 - ...Mỗi thuộc tính cách nhau dấu ;
- Ví dụ:

```
SqlConnection sqlcon = new  
SqlConnection("Server=serverName;uid=myUser;pwd=myPass;Database=myDB");
```

Lớp SqlConnection

```
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DBdemo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                // Khai báo và mở kết nối
                SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;pwd=123Abc;Database=demo");
                con.Open();
                // Thông báo khi kết nối thành công
                MessageBox.Show("Kết nối thành công !", "Thông báo");
            }
            catch (SqlException ex)
            {
                MessageBox.Show(ex.Message, "Thông báo");
            }
        }
    }
}
```

Lớp OleDbConnection

- Sử dụng phương thức xây dựng có tham số của lớp **OleDbConnection**.
 - Tham số này là chuỗi kết nối (**ConnectionString**):
 - Tên trình cung cấp (*Provider*)
 - Tên Server lưu cơ sở dữ liệu (*Server, Data Source*)
 - Tên cơ sở dữ liệu (*Database*)
 - Tài khoản, Mật khẩu (*uid, pwd*)
 - ...
- Mỗi thuộc tính cách nhau dấu ;

Lớp OleDbConnection

```
OleDbConnection con=new OleDbConnection  
    ("Provider=SQLOLEDB;Server=SQLDB;  
    Database=Tennis;uid=sa;pwd=password");
```

```
OleDbConnection con=new OleDbConnection  
    ("Provider=Microsoft.Jet.OLEDB.4.0;  
    Data Source=D:\\Tennis.mdb");
```

- Một số Provider:
 - ***SQLOLEDB***: Provider của SQL Server
 - ***Microsoft.Jet.OLEDB.4.0***: Provider của Access

Các lớp Command

- Một đối tượng **Command** cho phép truy xuất hoặc thao tác dữ liệu trong cơ sở dữ liệu (thông qua câu truy vấn dạng chuỗi).
- Đối tượng **Command** được khởi tạo sau khi 1 đối tượng **Connection** được thiết lập.
- Gồm:
 - **SqlCommand**: cho phép thực thi câu truy vấn với SQL .Net Framework Data Provider.
 - **OleDbCommand**: cho phép thực thi câu truy vấn với OleDb .Net Framework Data Provider.

Các lớp Command

- Thuộc tính

- **CommandText**: là chuỗi thể hiện câu truy vấn hoặc tên của 1 stored procedure hay tên 1 bảng.
- **CommandType**: loại của đối tượng Command (StoredProcedure, TableDirect, Text).
- **Connection**: đối tượng Connection đến 1 cơ sở dữ liệu.

- Phương thức

- **ExecuteNonQuery()** : thực thi câu truy vấn hành động (Insert, Update, Delete)
- **ExecuteReader()** : thực thi câu truy vấn dạng Select; kết quả trả về là 1 đối tượng DataReader.
- **ExecuteScalar()** : thực thi câu truy vấn dạng Select với kết quả của câu truy vấn là 1 giá trị đơn.

Thực thi câu truy vấn hành động

```
try
{
    // Khai báo và mở kết nối
    SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
        "pwd=123Abc;Database=demo");
    con.Open();
    // Thông báo khi kết nối thành công
    MessageBox.Show("Kết nối thành công !", "Thông báo");
    string sql = "INSERT INTO [demo].[dbo].[CHUCVU]" +
        "([MACV],[TENCV]) VALUES " +
        "('CV06','Lao công')";
    SqlCommand com = new SqlCommand(sql, con);
    com.ExecuteNonQuery();
}
catch (SqlException ex)
{
    MessageBox.Show(ex.Message, "Thông báo");
}
```

Câu truy vấn với kết quả trả về giá trị

```
try
{
    // Khai báo và mở kết nối
    SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
        "pwd=123Abc;Database=demo");
    con.Open();
    // Thông báo khi kết nối thành công
    MessageBox.Show("Kết nối thành công !", "Thông báo");
    string sql = "SELECT COUNT(MANV) FROM NHANVIEN";
    SqlCommand com = new SqlCommand(sql, con);
    int soNV = Convert.ToInt32(com.ExecuteScalar());
    MessageBox.Show("Tổng số NV: " + soNV.ToString(), "Thông báo");
}
catch (SqlException ex)
{
    MessageBox.Show(ex.Message, "Thông báo");
}
```

Truyền giá trị khi thực hiện câu truy vấn

- **Vấn đề:** Câu truy vấn được thực hiện dựa trên những giá trị được nhập từ bàn phím.
- **Giải pháp:**
 - Viết câu truy vấn dạng chuỗi trực tiếp
 - Sử dụng lớp **Parameter**

Câu truy vấn dạng chuỗi trực tiếp

```
try
{
    // Khai báo và mở kết nối
    SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
        "pwd=123Abc;Database=demo");
    con.Open();
    // Thông báo khi kết nối thành công
    string sql = "DELETE FROM NHANVIEN WHERE MANV='" +
        maNV.Text + "'";
    SqlCommand com = new SqlCommand(sql, con);
    com.ExecuteNonQuery();
}
catch (SqlException ex)
{
    MessageBox.Show(ex.Message, "Thông báo");
}
```

Các lớp Parameter

- 1 đối tượng **Parameter** là 1 tham số được truyền vào khi thực hiện 1 câu truy vấn hoặc 1 **stored procedure** thông qua 1 đối tượng Command.
- Gồm:
 - Lớp **SqlParameter**: tham số với *SQL .Net Framework Data Provider*.
 - Lớp **OleDbParameter**: tham số với *OleDb .Net Framework Data Provider*

Các lớp Parameter

- Thuộc tính

- *ParameterName*: tên tham số.
- *DbType*: Kiểu tham số
- *Value*: Giá trị của tham số

- Phương thức xây dựng

- *SqlParameter*(string, SqlDbType)
- *OleDbParameter*(string, OleDbType)

Định nghĩa 1 tham số với tên và kiểu thích hợp

- Một tham số sau khi định nghĩa sẽ được thêm vào tập hợp **Parameters** của 1 đối tượng **Command**

Lớp SqlParameter

```
try
{
    // Khai báo và mở kết nối
    SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
        "pwd=123Abc;Database=demo");
    con.Open();
    // Thông báo khi kết nối thành công
    string sql = "DELETE FROM CHUCVU WHERE MACV=@maCV";
    // Định nghĩa một biến kiểu SqlParameter
    SqlParameter par = new SqlParameter("@maCV", SqlDbType.VarChar);
    par.Value = "CV06";
    SqlCommand com = new SqlCommand(sql, con);
    // Thêm tham số khi thực thi
    com.Parameters.Add(par);
    com.ExecuteNonQuery();
}
catch (SqlException ex)
{
    MessageBox.Show(ex.Message, "Thông báo");
}
```

Thực thi stored procedure

- Thuộc tính **CommandText** của đối tượng Command là tên của stored procedure.
- Thuộc tính **CommandType** là **StoredProcedure**
- Dùng lớp **Parameter** để định nghĩa các tham số.
- Thêm các tham số vào tập hợp **Parameters** của đối tượng **Command**.
- Thực thi câu truy vấn

Thực thi stored procedure

- Giả sử ta có một **stored procedure** cho phép cập nhật ngày sinh của các cầu thủ:

```
-- Phương thức để cập nhật ngày sinh của một nhân viên
CREATE PROCEDURE upNgaySinhNV
    @maNV varchar, @ngay datetime
AS
BEGIN
    UPDATE [demo].[dbo].[NHANVIEN]
    SET [NGAYSINHNV] = @ngay
    WHERE [MANV] = @maNV
END
```

Thực thi stored procedure

```
// Khai báo và mở kết nối
SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
    "pwd=123Abc;Database=demo");
con.Open();
// Định nghĩa biến kiểu SqlParameter
SqlParameter parMANV = new SqlParameter("@maNV", SqlDbType.VarChar);
parMANV.Value = maNV.Text;
SqlParameter parNGAYSINH = new SqlParameter("@ngay", SqlDbType.DateTime);
parNGAYSINH.Value = dtpNGAYSINH.Value;
SqlCommand com = new SqlCommand("upNgaySinhNV", con);
com.CommandType = CommandType.StoredProcedure;
// Thêm tham số khi thực thi
com.Parameters.Add(parMANV);
com.Parameters.Add(parNGAYSINH);
com.ExecuteNonQuery();
```

Các lớp DataReader

- Đối tượng thuộc lớp **DataReader** chứa luồng dữ liệu chỉ đọc là kết quả của việc thực thi câu truy vấn dạng Select.
 - Luồng dữ liệu này chỉ cho phép truy xuất 1 chiều (từ đầu đến cuối - *forward only*).
- 1 đối tượng **DataReader** được tạo ra khi phương thức **ExecuteReader()** của 1 đối tượng **Command** được gọi thực hiện.
- Gồm:
 - Lớp **SqlDataReader**: đọc dữ liệu với *SQL .Net Framework Provider*
 - Lớp **OleDbDataReader**: đọc dữ liệu với *OleDb .Net Framework Provider*

Các lớp DataReader

- **Thuộc tính**

- **HasRows**: xác định đối tượng DataReader còn trả về kết quả nữa (true) hay không (false)?

- **Phương thức**

- **Close()**: đóng lại đối tượng DataReader
- **Read()**: di chuyển đến mẫu tin kế tiếp, kết quả trả về là true nếu di chuyển thành công, ngược lại false
- **GetBoolean()**: trả về giá trị của cột chỉ định như là 1 giá trị kiểu bool.
- **GetDateTime()**: trả về giá trị của cột chỉ định như là 1 giá trị kiểu DateTime
- **GetInt32()**: trả về giá trị của cột chỉ định như là 1 giá trị kiểu int
- **GetString()**: trả về giá trị của cột chỉ định như là 1 giá trị kiểu string
- **GetValue()**: trả về giá trị của cột chỉ định

Lớp SqlDataReader

```
// Khai báo và mở kết nối
SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
    "pwd=123Abc;Database=demo");
con.Open();
// Định nghĩa biến kiểu SqlParameter
string sql = "SELECT * FROM NHANVIEN";
SqlCommand com = new SqlCommand(sql, con);
SqlDataReader dr = com.ExecuteReader();
// Xử lý và hiển thị dữ liệu
string rs = "";
while (dr.Read())
{
    rs += dr.GetString(0) + "\t" +
        dr.GetString(4) + "\t" +
        dr.GetSqlDateTime(5).Value.ToShortDateString() + " " +
        dr.GetString(6) + "\n";
}
MessageBox.Show(rs);
```


Các lớp DataAdapter

- Đối tượng **DataAdapter** cho phép lấy dữ liệu từ cơ sở dữ liệu đưa vào **DataSet** và ngược lại cho phép cập nhật cơ sở dữ liệu với dữ liệu trong **DataSet**.
- Gồm:
 - Lớp **SqlDataAdapter**: *SQL .Net Framework Data Provider*
 - Lớp **OleDbDataAdapter**: *OleDb .Net Framework Data Provider*

Các lớp DataAdapter

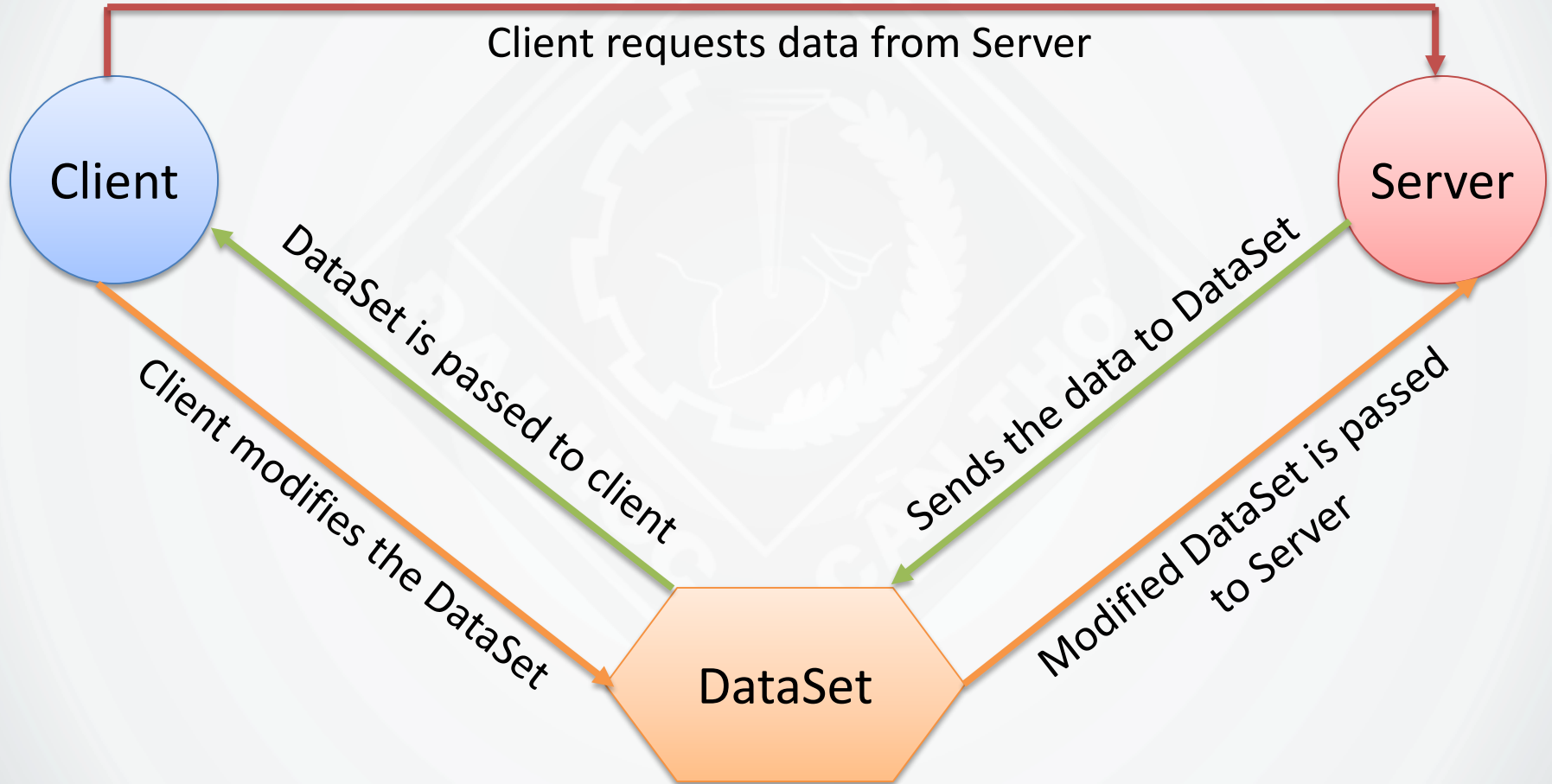
Thuộc tính	Ý nghĩa
SelectCommand	Đối tượng Command để lấy dữ liệu từ CSDL
InsertCommand	Cập nhật CSDL dựa vào thay đổi trong Dataset (thông qua 3 đối tượng Command cho phép Insert, Update, Delete)
UpdateCommand	
DeleteCommand	

Phương thức	Ý nghĩa
Fill	Đưa (Cập nhật) dữ liệu vào DataSet (để khớp với nguồn dữ liệu) nhờ thực thi câu truy vấn trong SelectCommand

Giới thiệu DataSet

- **DataSet** là các đối tượng chứa các bảng tạm thời lưu dữ liệu trong ứng dụng.
- Có thể thao tác với dữ liệu trong **DataSet** ngay cả khi ứng dụng ngắt kết nối với cơ sở dữ liệu (*disconnected architecture*) .
- Cấu trúc của một **DataSet** tương tự như cấu trúc một cơ sở dữ liệu quan hệ gồm các bảng (*DataTable*), dòng (*DataRow*), cột (*DataColumn*), quan hệ (*DataRelation*), ...

Giới thiệu DataSet



Lớp DataSet

- Lớp **DataSet** được định nghĩa trong không gian tên ***System.Data***.
- **Thuộc tính**
 - ***Tables***: tập hợp các DataTable của DataSet
- **Phương thức**
 - Phương thức xây dựng không tham số: tạo 1 đối tượng **DataSet** với tên mặc định.
 - Phương thức xây dựng có tham số: tạo 1 đối tượng **DataSet** với tên được chỉ định.

Tập hợp Tables của DataSet

PROPERTIES
Item



METHODS
Add
Remove
RemoveAt



Lớp DataTable

- Một **DataTable** là một bảng trong một DataSet (tương đương với một bảng trong bộ nhớ của cơ sở dữ liệu quan hệ).

DataSet = Collection of many tables

DataTable = One of the tables in DataSet

Lớp DataTable

- **Thuộc tính**

- ***Columns***: tập hợp các cột (DataColumn) của 1 *DataTable*.
- ***Constraints***: tập hợp các *constraints* của 1 *DataTable*.
- ***PrimaryKey***: tập hợp các trường tạo nên khóa chính.
- ***Rows***: tập hợp các dòng (DataRow) của 1 *DataTable*

Lớp DataColumn

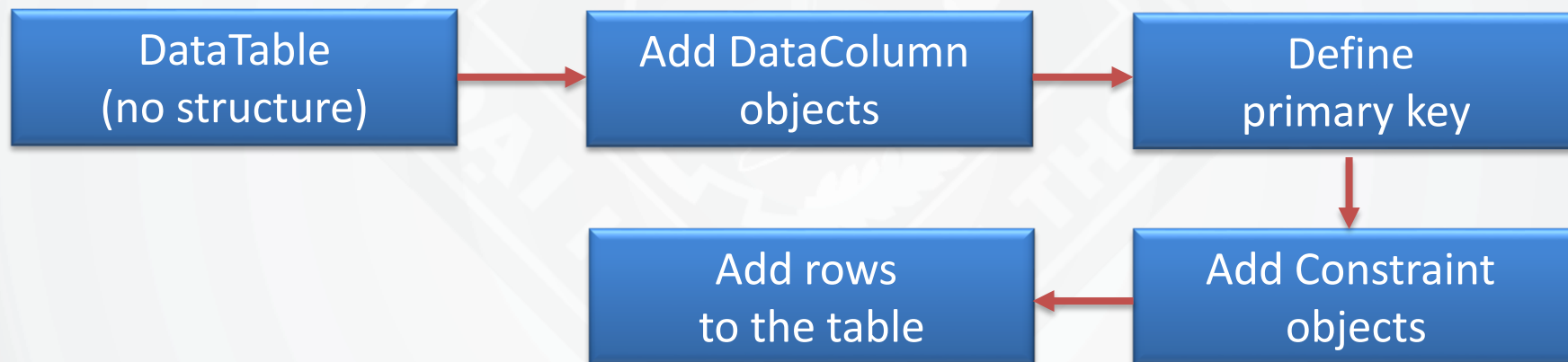
- Một ***DataColumn*** là 1 cột của 1 DataTable trong 1 DataSet.
- **Thuộc tính**
 - ***ColumnName***: tên của cột
 - ***DataType***: kiểu dữ liệu

Lớp DataRow

- Một DataRow là 1 dòng của 1 DataTable trong 1 DataSet.

Tạo DataTable bằng mã lệnh

- Tạo DataTable thông qua phương thức xây dựng của lớp DataTable.



```
DataSet ds = new DataSet();  
DataTable dt = ds.Tables.Add("TNHANVIEN");
```

Thêm cột vào DataTable

- Dùng phương thức xây dựng của lớp **DataColumn**.
- Sử dụng tập hợp **Columns** của đối tượng **DataTable** để thêm cột mới vào.

```
DataSet ds = new DataSet();
DataTable dt = ds.Tables.Add("TNHANVIEN");
...
DataColumn dc = dt.Columns.Add("MANV", typeof(String));
dc.AllowDBNull = false;
dc.Unique = true;
...
dt.Columns.Add("HOTENNV", typeof(String));
dt.Columns.Add("NGAYSINHNV", typeof(DateTime));
dt.Columns.Add("LUONG", typeof(Double));
```

Khóa chính của DataTable

- Sử dụng thuộc tính PrimaryKey

Khóa chính là 1 cột

```
dt.PrimaryKey = new DataColumn[]  
{ dt.Columns["MANV"] };
```

hay

```
DataColumn[] dts = new DataColumn[1];  
dts[0] = dt.Columns["MANV"];  
dt.PrimaryKey = dts;
```

Khóa chính là khóa tổ hợp

```
dt.PrimaryKey = new DataColumn[]  
{ dt.Columns["MANV"],  
  dt.Columns["HOTENNV"] };
```

hay

```
DataColumn[] dts = new DataColumn[2];  
dts[0] = dt.Columns["MANV"];  
dts[1] = dt.Columns["HOTENNV"];  
dt.PrimaryKey = dts;
```

Thêm dòng mới vào DataTable

- Sử dụng phương thức ***NewRow*** của lớp DataTable.

```
DataRow row = dt.NewRow();  
row["HOTENNV"] = "Nguyễn Văn A";  
row[1] = "Nguyễn Văn A";  
dt.Rows.Add(row);  
dt.Rows.Add(new Object[] { 1, "Nguyễn Văn A" });
```

Sinh viên đọc thêm

- *DataView*
- *Constraint*
- *DataRelation*

Đưa dữ liệu từ cơ sở dữ liệu vào DataSet

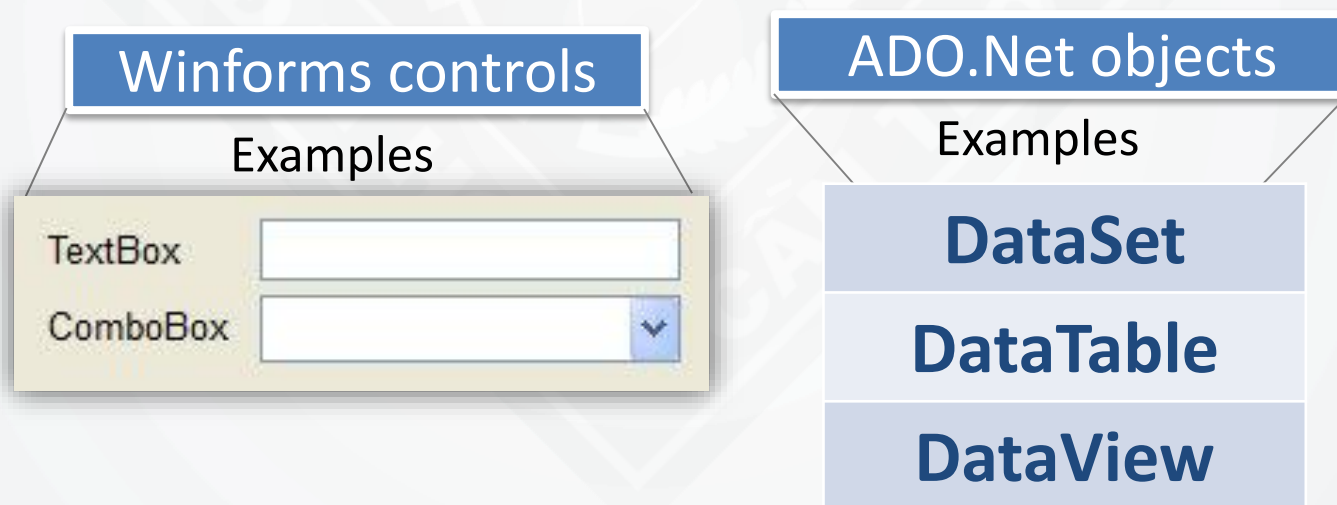
- Nối kết đến CSDL (**Connection**)
- Tạo đối tượng **DataAdapter**
- Chỉ ra câu truy vấn trong thuộc tính **SelectCommand** của **DataAdapter**.
- Điền dữ liệu từ **DataAdapter** vào **Dataset** nhờ phương thức **Fill** của lớp **DataAdapter**

Đưa dữ liệu từ cơ sở dữ liệu vào DataSet

```
try
{
    // Khai báo và mở kết nối
    SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
                                         "pwd=123Abc;Database=demo");
    con.Open();
    // Sử dụng SqlDataAdapter
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM NHANVIEN", con);
    DataSet ds = new DataSet();
    da.Fill(ds);
}
catch (SqlException ex)
{
    MessageBox.Show(ex.Message, "Thông báo");
}
```

Giới thiệu DataBinding

- **DataBinding** là quá trình ràng buộc giá trị trong nguồn dữ liệu (***DataSet***, ***DataTable***, ***DataColumn***,...) với controls của Winforms.



Simple Binding

- Một cột của ***DataTable*** ràng buộc với một controls bất kỳ của Winforms.
- **Thí dụ:** Cột *HOTENNV* của DataTable t sẽ được ràng buộc với thuộc tính Text của TextBox txt

```
DataTable t = ds.Tables["TNHANVIEN"];  
TextBox txt = new TextBox();  
txt.DataBindings.Add("Text", t, "HOTENNV");
```

Complex Binding

- Tất cả các dòng của 1 cột của 1 DataTable (hay toàn bộ DataTable hoặc DataSet) ràng buộc với 1 controls dạng danh sách của Winforms.
- **Thí dụ:** Cột *TENCV*, *MACV* của DataTable t sẽ ràng buộc với Combobox cbo

```
DataTable t = ds.Tables["TCHUCVU"];  
ComboBox cbo = new ComboBox();  
cbo.DataSource = t;  
cbo.DisplayMember = "TENCV";  
cbo.ValueMember = "MACV";
```

DataGridView control

- Cho phép hiển thị dữ liệu ở dạng bảng.
- Dữ liệu hiển thị trong DataGridView có thể được thêm, sửa, xóa, sắp xếp, phân trang.
- **Thuộc tính**
 - ***DataSource***: đối tượng nguồn dữ liệu (***DataSet***, ***DataTable***, ...) sẽ được thể hiện trên DataGridView.
- **Thí dụ**: Hiển thị chi tiết bảng NHANVIEN

Thí dụ DataGridView

```
// Khai báo và mở kết nối
SqlConnection con = new SqlConnection("Server=LIGHTX-LT;uid=sa;" +
    "pwd=123Abc;Database=demo");
con.Open();
// Định nghĩa biến kiểu SqlParameter
SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM NHANVIEN", con);
Dataset ds = new Dataset();
da.Fill(ds, "NHANVIEN");
dgvNHANVIEN.DataSource = ds.Tables["NHANVIEN"];
```


Thí dụ DataGridView

MANV	MADT	MAPX	MACV	HOTENNV	NGAYSINHNV	DIACHINV	NGAYBDLV
NV001	DT01	P001	CV01	Nguyễn Văn A	18/03/1986	25 Hoàng Văn T...	20/05/2010
NV002	DT03	P003	CV03	Hồ Thị Thanh B	13/09/1987	50 Tâm Vu	23/06/2010
NV003	DT01	P002	CV04	Trần Văn C	09/09/1980	12 Nguyễn Trãi	03/06/2009
NV004	DT01	P001	CV05	Lưu Thanh D	23/11/1989	108/95 Nguyễn ...	01/04/2011