



ADO.NET

Nội dung

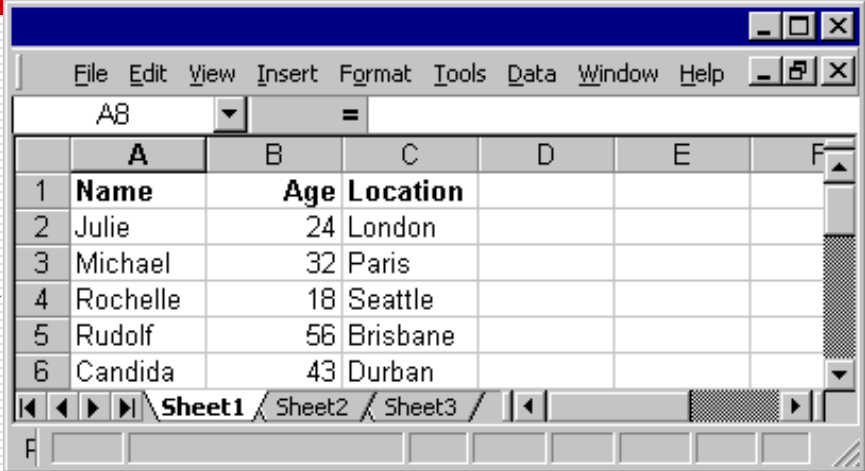
- *Các thành phần truy cập dữ liệu*
- *ADO.NET*
 - *Connection*
 - *Command*
 - *DataReader*
 - *DataAdapter*
 - *DataSet*
- *DataBinding*

Các thành phần truy cập dữ liệu

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			

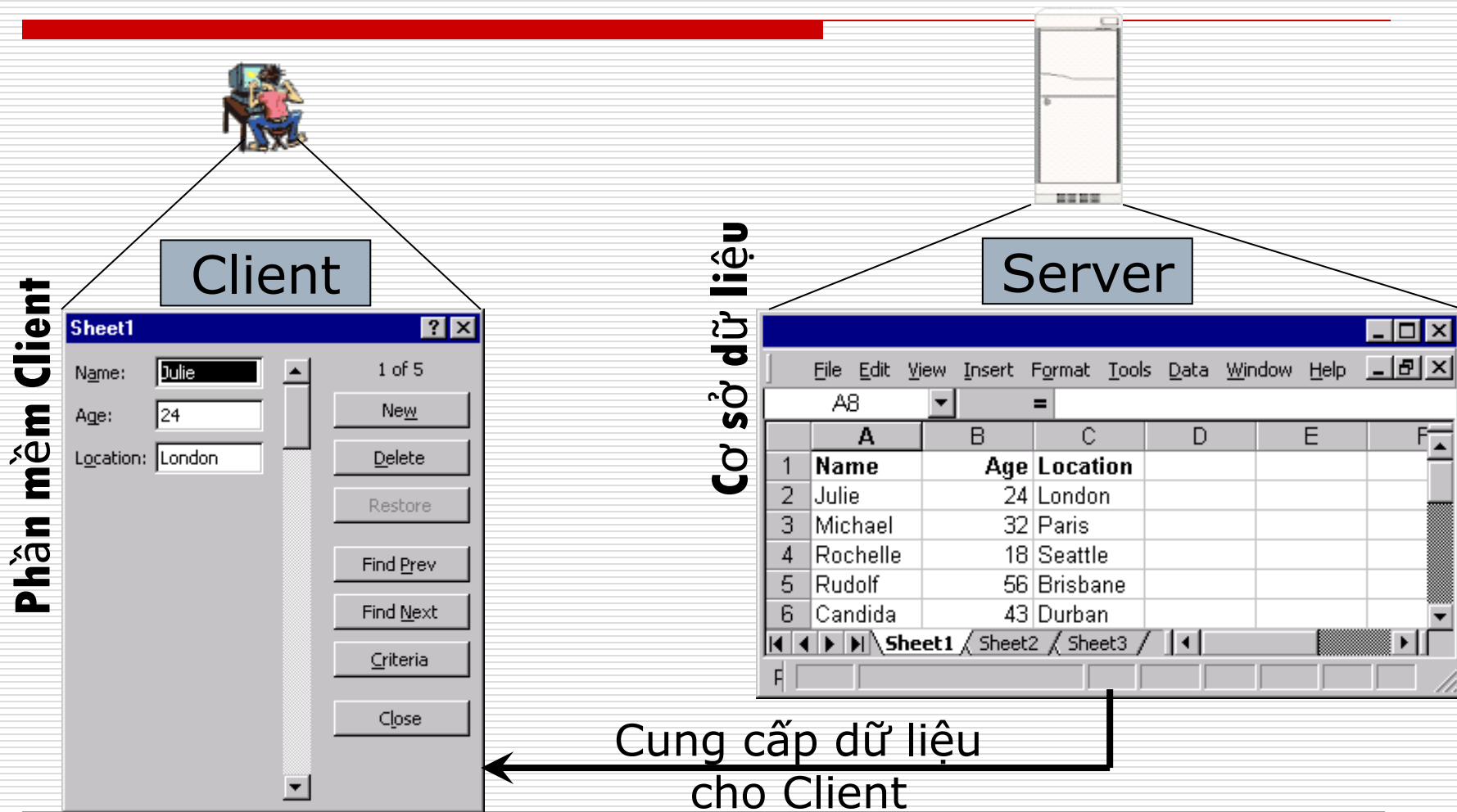


Client

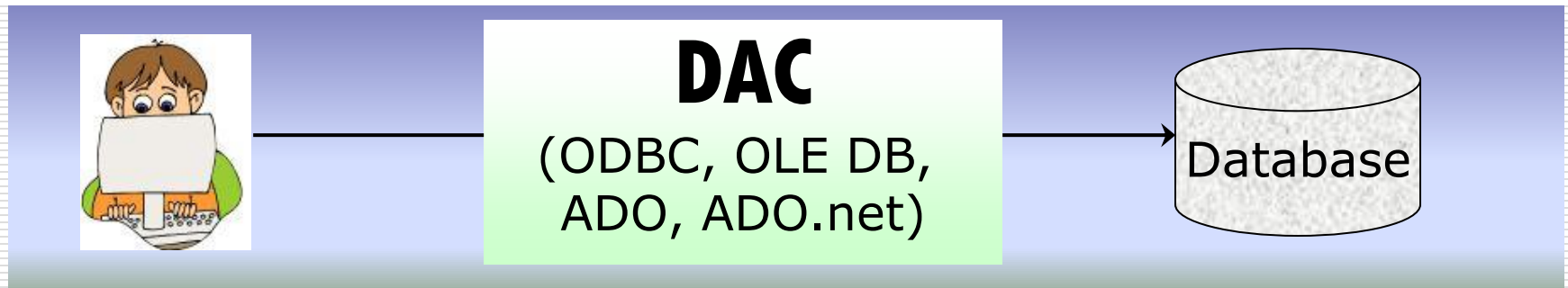
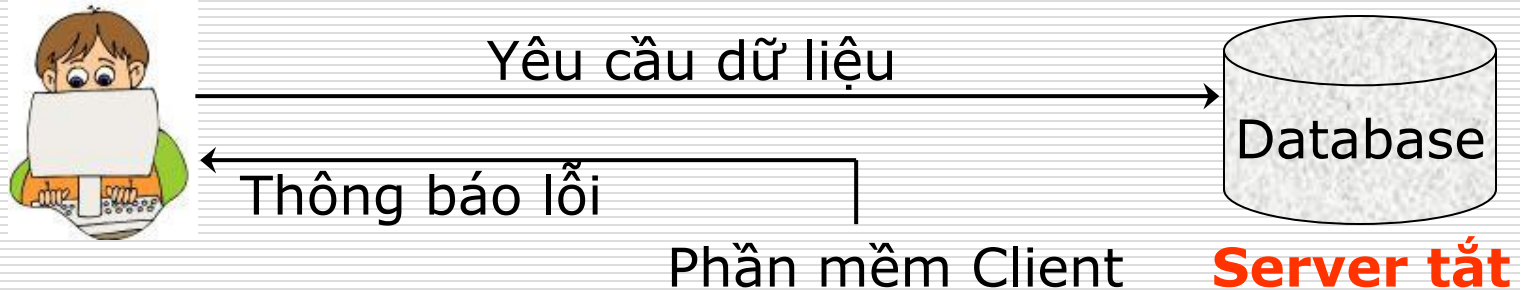
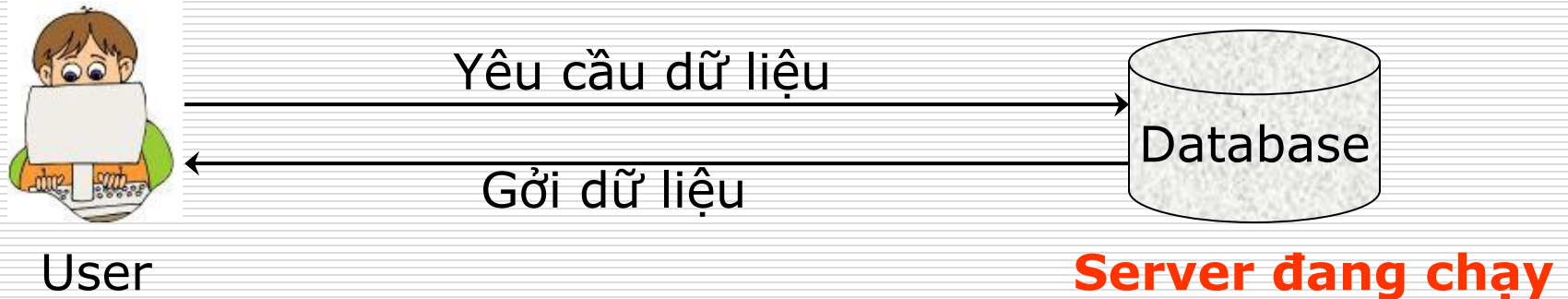
Các kỹ thuật
truy cập dữ liệu

Cơ sở dữ liệu

Client-Server



Data Access Components - Các thành phần truy cập dữ liệu

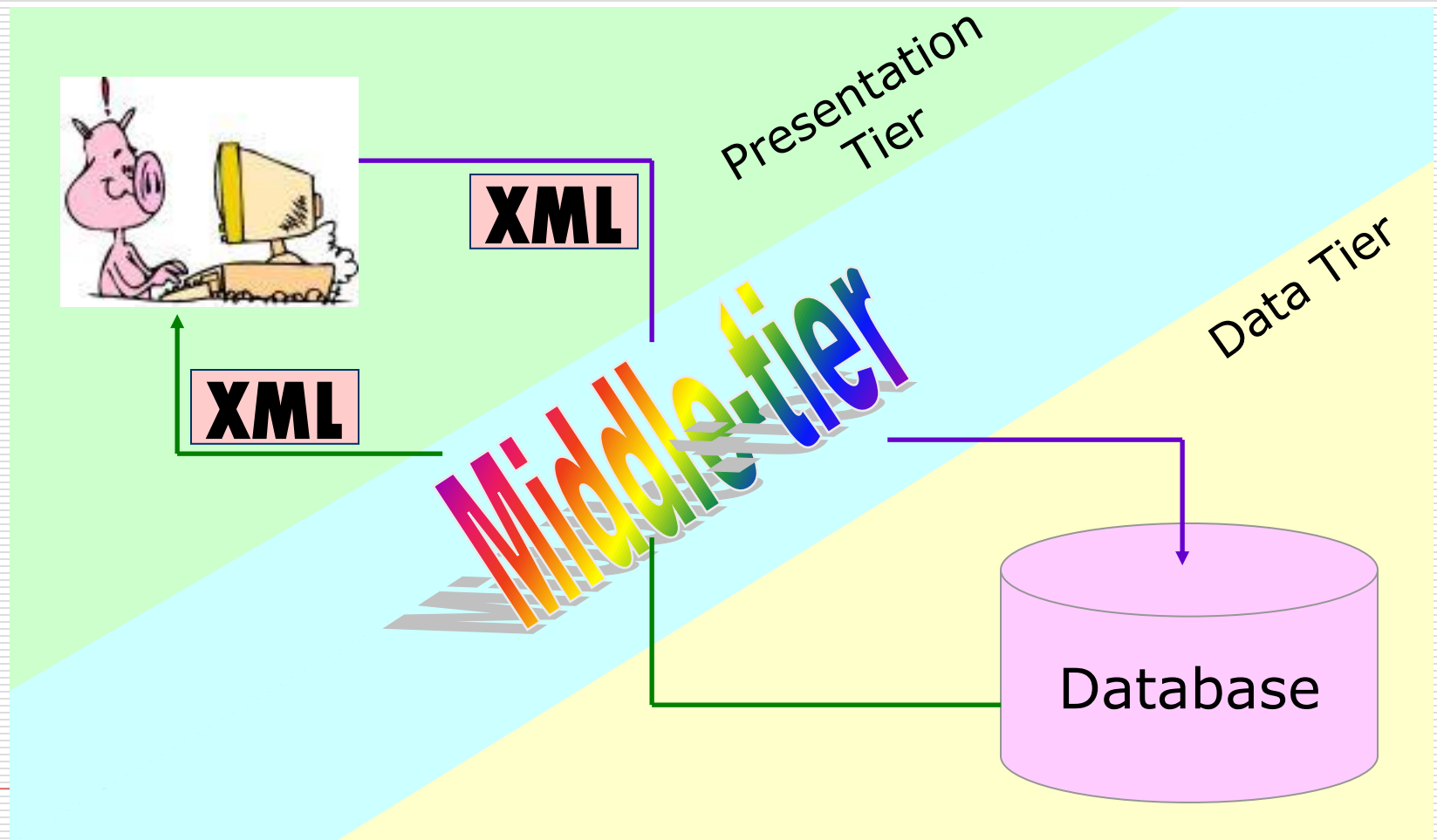


Giới thiệu về ADO.NET

ADO.NET

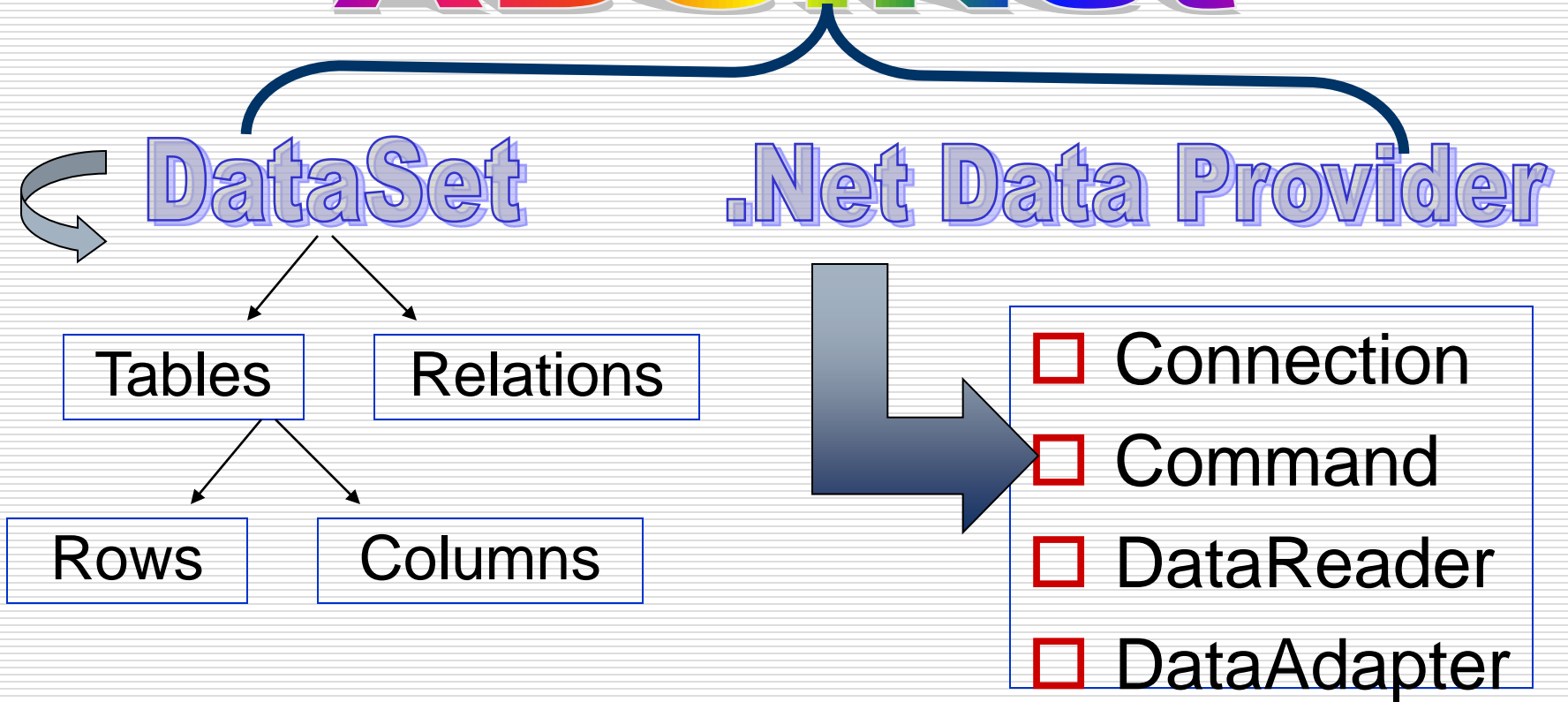
- ❑ Là kỹ thuật truy cập cơ sở dữ liệu được hỗ trợ bởi thư viện lớp cơ sở của .NET Framework.

Kiến trúc ADO.NET



Các thành phần của ADO.NET

ADO.NET



.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

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=SQLDB;  
uid=mylogin;pwd=mylogin;Database=Tennis");
```

Lớp SqlConnection

```
public partial class frmPlayer : Form
{
    SqlConnection con;
    public frmPlayer()

    private void frmPlayer_Load(object sender, EventArgs e)
    {
        try
        {
            //Khoi tao doi tuong SqlConnection
            con = new SqlConnection("Server=HoaiBao\\HoaiBao;Database=Tennis;" +
                                   "uid=mylogin5;pwd=mylogin5");
            con.Open();//Mo noi ket
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
```


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

Lớp OleDbConnection

```
public partial class frmPlayerOledb : Form
{
    OleDbConnection con;
    public frmPlayerOledb()...
    |
    private void frmPlayerOledb_Load(object sender, EventArgs e)
    {
        try
        {
            //Khoi tao doi tuong SqlConnection
            con = new OleDbConnection("Provider=SQLOLEDB;Server=HoaiBao\\HoaiBao;" +
                                     "Database=Tennis;uid=mylogin5;pwd=mylogin5");
            con.Open();//Mo noi ket
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
```

Các lớp Command

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 [1]

```
try
{
    string str = "Insert Into Players(PlayerNo, [Name], Initials, Birth_Date, " +
        "Sex, Joined, Street, HouseNo, PostCode, Town, PhoneNo, LeagueNo) " +
        "Values (77, 'Thomas', 'M', '02/02/1989', 'M', 2008, 'Hognikamp', " +
        "2, '6302CD', 'Arnhem', '070-123444', '3455')";
    SqlCommand com = new SqlCommand(str, con);
    com.ExecuteNonQuery();
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Thực thi câu truy vấn hành động [2]

```
try
{
    string str = "Insert Into Players(PlayerNo, [Name], Initials, Birth_Date, " +
        "Sex, Joined, Street, HouseNo, PostCode, Town, PhoneNo, LeagueNo) " +
        "Values (77, 'Thomas', 'M', '02/02/1989', 'M', 2008, 'Hognikamp', " +
        "2, '6302CD', 'Arnhem', '070-123444', '3455')";
    OleDbCommand com = new OleDbCommand(str, con);
    com.ExecuteNonQuery();
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```


Câu truy vấn với kết quả trả về là 1 giá trị [1]

```
try
{
    //Tìm nam sinh lon nhat
    string str = "select Max(Year(Birth_Date)) from players";
    SqlCommand com = new SqlCommand(str, con);
    //Thuc thi cau truy van, ket qua tra ve la 1 gia tri
    int year = Convert.ToInt32(com.ExecuteScalar());
    MessageBox.Show(year.ToString());
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Câu truy vấn với kết quả trả về là 1 giá trị [2]

```
try
{
    //Tìm nam sinh lon nhat
    string str = "select Max(Year(Birth_Date)) from players";
    OleDbCommand com = new OleDbCommand(str, con);
    //Thuc thi cau truy van, ket qua tra ve la 1 gia tri
    int year = Convert.ToInt32(com.ExecuteScalar());
    MessageBox.Show(year.ToString());
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

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
{
    //Xoa cau thu co ma cau thu nhap tu TextBox txtPlayerNo
    string str = "Delete From Players Where PlayerNo='" + txtPlayerNo.Text + "'";
    SqlCommand com = new SqlCommand(str, con);
    com.ExecuteNonQuery();
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

```
try
{
    //Xoa cau thu co ma cau thu nhap tu TextBox txtPlayerNo
    string str = "Delete From Players Where PlayerNo='" + txtPlayerNo.Text + "'";
    OleDbCommand com = new OleDbCommand(str, con);
    com.ExecuteNonQuery();
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

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, DbType)

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
{
    //Xoa cau thu co ma cau thu nhap tu TextBox txtPlayerNo
    //Cau truy van chi ra tham so ten @No
    string str = "Delete From Players Where PlayerNo=@No";

    //Dinh nghia 1 tham so ten @No co kieu Int
    SqlParameter par = new SqlParameter("@No", SqlDbType.Int);
    //Gia tri tham so nhap tu TextBox txtPlayerNo
    par.Value = Convert.ToInt32(txtPlayerNo.Text);

    SqlCommand com = new SqlCommand(str, con);
    com.Parameters.Add(par); //Them vao tap hop Parameters

    com.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Lớp OleDbParameter

```
try
{
    //Xoa cau thu co ma cau thu nhap tu TextBox txtPlayerNo
    //Cau truy van chi ra tham so ten @No
    string str = "Delete From Players Where PlayerNo=@No";

    //Dinh nghia 1 tham so ten @No co kieu Int
    OleDbParameter par = new OleDbParameter("@No",OleDbType.Integer);
    //Gia tri tham so nhap tu TextBox txtPlayerNo
    par.Value = Convert.ToInt32(txtPlayerNo.Text);

    OleDbCommand com = new OleDbCommand(str,con);
    com.Parameters.Add(par); //Them vao tap hop Parameters

    com.ExecuteNonQuery();
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```


Thực thi 1 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 1 stored procedure

- ❑ Giả sử ta có 1 stored procedure cho phép cập nhật ngày sinh của các cầu thủ:

```
Create Procedure spUpdate_Players  
    @No int, @date DateTime  
As  
    Update Players  
    Set Birth_Date = @date  
    Where PlayerNo=@No
```

Thực thi 1 stored procedure

```
try
{
    SqlParameter parNo = new SqlParameter("@No", SqlDbType.Int);
    parNo.Value = Convert.ToInt32(txtPlayerNo.Text);

    SqlParameter parBirthDate = new SqlParameter("@date", SqlDbType.DateTime);
    parBirthDate.Value = dtpBirth.Value;

    SqlCommand com = new SqlCommand("spUpdate_Players", con);
    com.CommandType = CommandType.StoredProcedure;
    com.Parameters.Add(parNo); //Them vao tap hop Parameters
    com.Parameters.Add(parBirthDate);

    com.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Thực thi 1 stored procedure

```
try
{
    OleDbParameter parNo = new OleDbParameter("@No",OleDbType.Integer);
    parNo.Value = Convert.ToInt32(txtPlayerNo.Text);

    OleDbParameter parBirthDate = new OleDbParameter("@date",OleDbType.Date);
    parBirthDate.Value=dtpBirth.Value;

    OleDbCommand com = new OleDbCommand("spUpdate_Players",con);
    com.CommandType=CommandType.StoredProcedure;
    com.Parameters.Add(parNo); //Them vao tap hop Parameters
    com.Parameters.Add(parBirthDate);

    com.ExecuteNonQuery();
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Các lớp DataReader

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

```
try
{
    string str = "Select * From Penalties";

    SqlCommand com = new SqlCommand(str, con);
    SqlDataReader dr = com.ExecuteReader();

    string strResult="";
    while (dr.Read())
    {
        strResult = strResult + dr.GetInt32(0).ToString() + "\t" +
            dr.GetInt16(1).ToString() + "\t" +
            dr.GetDateTime(2).ToString() + "\t" +
            dr.GetDecimal(3).ToString() + "\n";
    }
    MessageBox.Show(srResult);
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```


Lớp OleDbDataReader

```
try
{
    string str = "Select * From Penalties";

    OleDbCommand com = new OleDbCommand(str, con);
    OleDbDataReader dr = com.ExecuteReader();

    string strResult="";
    while (dr.Read())
    {
        strResult = strResult + dr.GetInt32(0).ToString() + "\t" +
            dr.GetInt16(1).ToString() + "\t" +
            dr.GetDateTime(2).ToString() + "\t" +
            dr.GetDecimal(3).ToString() + "\n";
    }
    MessageBox.Show(strResult);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Các lớp DataAdapter

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

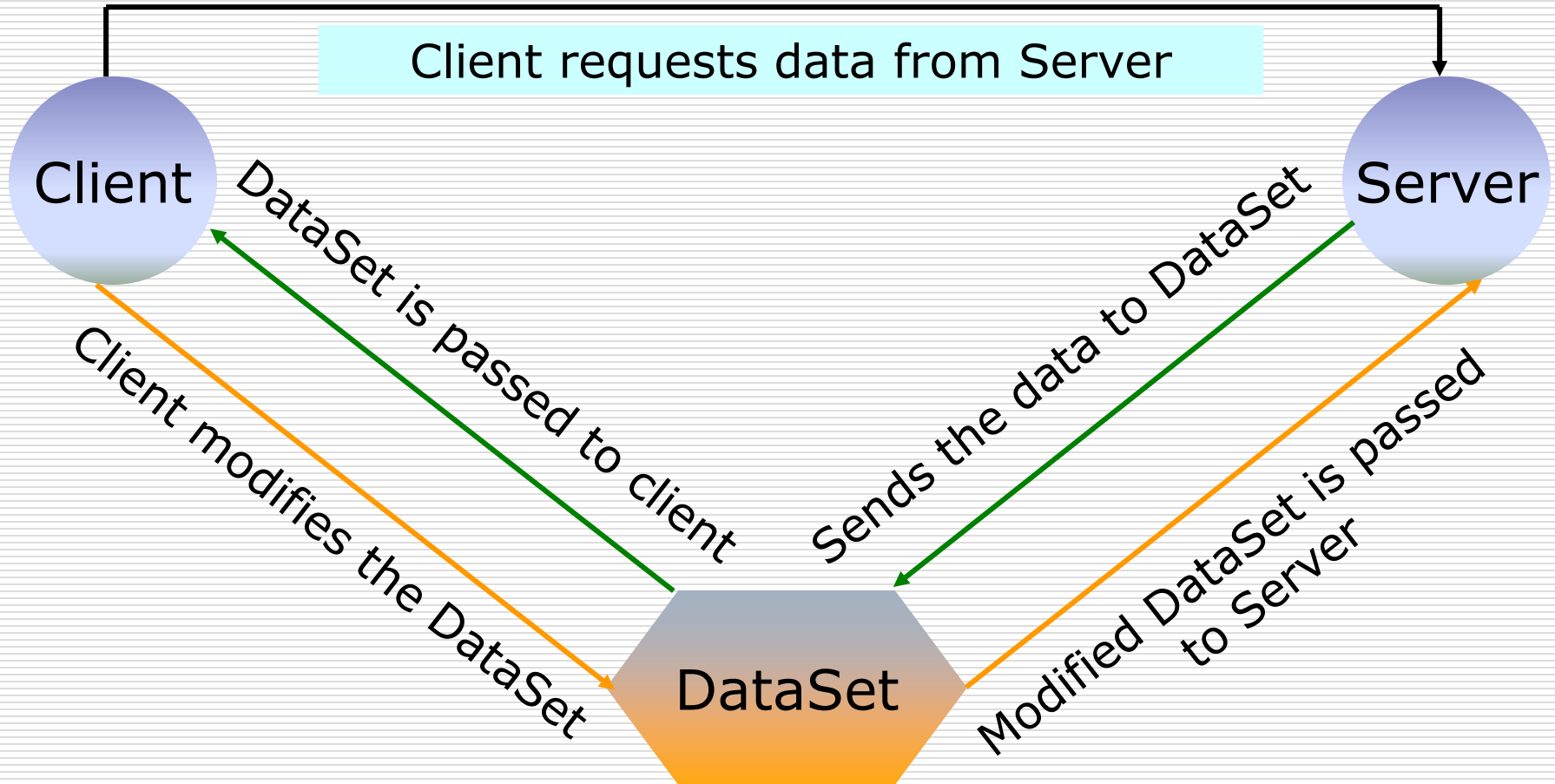


DataSet

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 1 DataSet tương tự như cấu trúc một cơ sở dữ liệu quan hệ gồm các bảng (DataTable), dòng (DataRow), cột (DataColumn), quan hệ (DataRelation), etc.

Giới thiệu DataSet

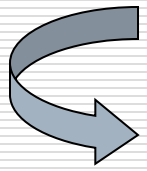


Lớp DataSet

- ❑ 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.

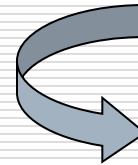
Tập hợp Tables của DataSet

Properties



Item

Methods



Add

Remove

RemoveAt

Lớp DataTable

- ❑ Một DataTable là một bảng trong 1 DataSet (~ 1 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

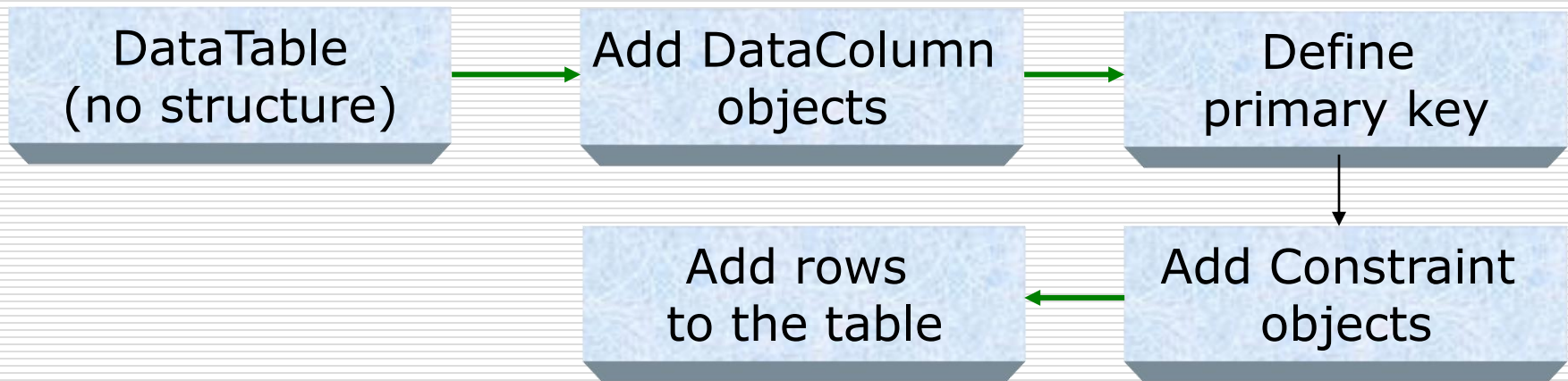
- ❑ Một DataColumn là 1 cột của 1 DataTable trong 1 DataSet.
- ❑ **Thuộc tính**
 - ColumnName: tên của cột
 - DataType: kiểu dữ liệu

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.



```
File Edit View Insert Format Help
DataSet empDS = new DataSet();
DataTable empTable = empDS.Tables.Add("EmpTable");
```

The code creates an instance of a DataTable by the name EmpTable and then adding it to the Tables collection of a empDS DataSet.

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.

```
DataTable empTable = new DataTable("Emp");  
  
DataColumn empCol = empTable.Columns.Add("EmpNo",  
typeof(Int32));  
empCol.AllowDBNull = false;  
empCol.Unique = true;  
  
empTable.Columns.Add("EmpName", typeof(String));  
empTable.Columns.Add("Sal", typeof(Double));
```

Khóa chính của DataTable

□ Sử dụng thuộc tính PrimaryKey

Khóa chính là 1 cột

```
empTable.PrimaryKey=new DataColumn[]  
{empTable.Columns["EmpNo"]};
```

hay

```
DataColumn[]col_arr=new DataColumn[1];  
col_arr[0]=empTable.Columns["EmpNo"];  
empTable.PrimaryKey = col_arr;
```

**Khóa chính là khóa
tổ hợp**

```
empTable.PrimaryKey=new DataColumn[]  
{empTable.Columns["EmpNo"],  
empTable.Columns["EmpName"]};
```

hay

```
DataColumn[]pkey=new DataColumn[2];  
pkey[0] = empTable.Columns["EmpNo"];  
pkey[1] = empTable.Columns["EmpName"];  
empTable.PrimaryKey = myKey;
```


Thêm dòng mới vào DataTable

- ❑ Sử dụng phương thức NewRow của lớp DataTable.

```
DataRow empRow=empTable.NewRow();  
empRow ["EmpName"] = "Smith";  
empRow [1] = "Smith";  
empTable.Rows.Add(empRow);  
empTable.Rows.Add(new Object[] {1, "Smith"});
```

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
{
    SqlDataAdapter da = new SqlDataAdapter("Select * From Players", con);
    DataSet ds = new DataSet();
    da.Fill(ds, "CauThu");
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Đưa dữ liệu từ cơ sở dữ liệu vào DataSet

```
try
{
    OleDbDataAdapter da = new OleDbDataAdapter("Select * From Players", con);
    DataSet ds = new DataSet();
    da.Fill(ds, "CauThu");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

DataBinding

Giới thiệu DataBinding

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

Winforms controls

Examples

TextBox	Object Oriented Programming
ComboBox	Software Engineering

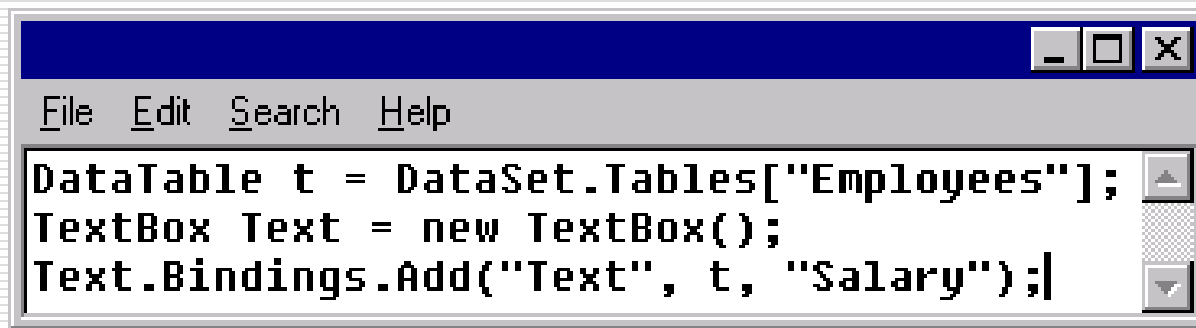
ADO.Net objects

Examples

DataSet
DataTable
DataView

Simple Binding

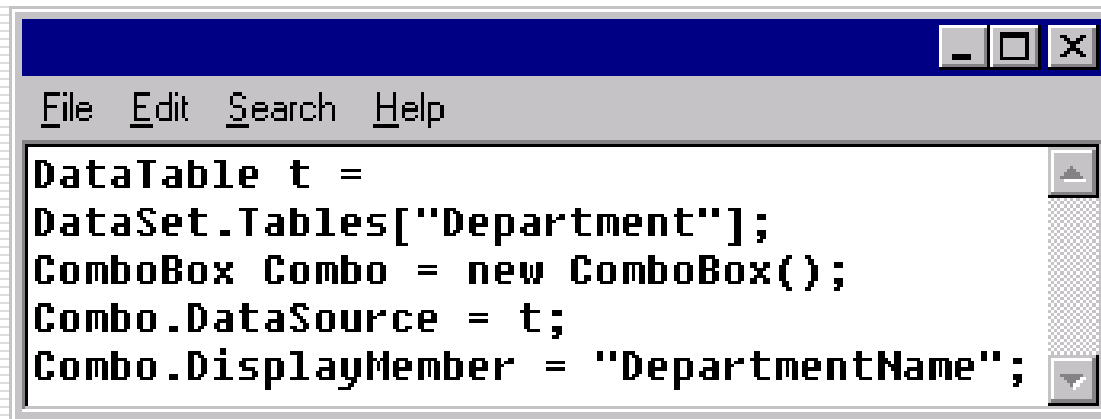
- ❑ Một cột của 1 DataTable ràng buộc với 1 controls bất kỳ của Winforms.
- ❑ **Thí dụ:** Cột Salary của DataTable t sẽ được ràng buộc với thuộc tính Text của TextBox Text



```
File Edit Search Help
DataTable t = DataSet.Tables["Employees"];
TextBox Text = new TextBox();
Text.Bindings.Add("Text", t, "Salary");
```


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 DepartmentName của DataTable t sẽ ràng buộc với Combobox Combo



```
File Edit Search Help
DataTable t =
DataSet.Tables["Department"];
ComboBox Combo = new ComboBox();
Combo.DataSource = t;
Combo.DisplayMember = "DepartmentName";
```

DatagridView control

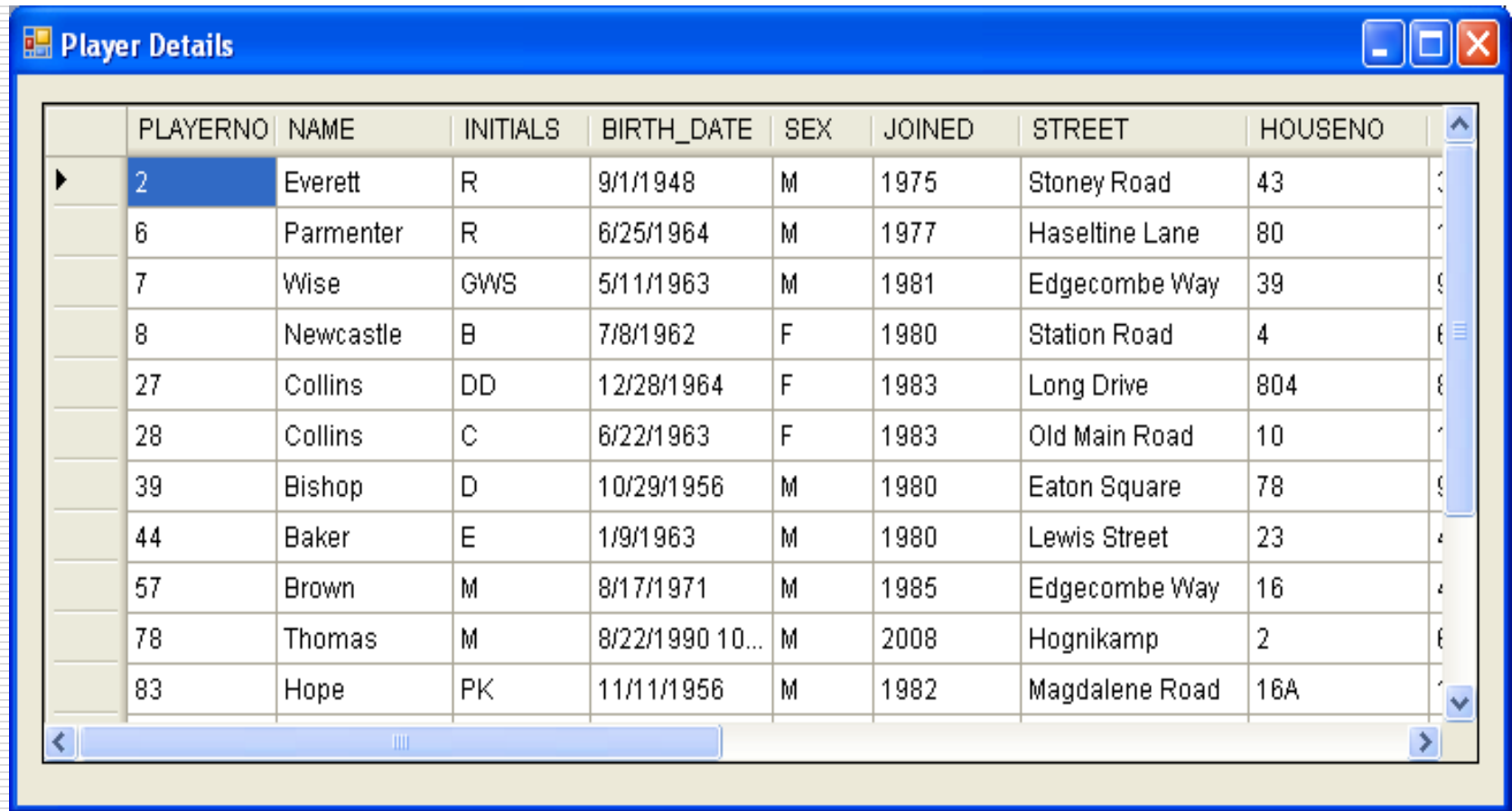
- ❑ Cho phép hiển thị dữ liệu ở dạng bảng
- ❑ Dữ liệu hiển thị trong DataGridView có thể được thêm, sửa, xóa, sắp xếp, phân trang.
- ❑ **Thuộc tính**
 - DataSource: đối tượng nguồn dữ liệu (DataSet, DataTable, ...) sẽ được thể hiện trên DataGridView
- ❑ **Thí dụ:** Hiển thị chi tiết bảng Players

Thí dụ DataGridView controls [1]

```
private SqlConnection con;
private DataSet ds;
private void frmPlayerDetails_Load(object sender, EventArgs e)
{
    try
    {
        //Khoi tao doi tuong SqlConnection
        con = new SqlConnection("Server=HoaiBao\\HoaiBao;Database=Tennis;" +
                                "uid=mylogin5;pwd=mylogin5");

        con.Open(); //Mo noi ket
        SqlDataAdapter da = new SqlDataAdapter("Select * From Players", con);
        ds = new DataSet();
        da.Fill(ds, "CauThu");
        //Complex binding
        grdPlayer.DataSource=ds.Tables["CauThu"];
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Thí dụ DataGridView controls [2]



The screenshot shows a Windows application window titled "Player Details". Inside the window is a DataGridView control displaying a table of player information. The table has 9 columns: PLAYERNO, NAME, INITIALS, BIRTH_DATE, SEX, JOINED, STREET, and HOUSENO. The first row is selected, showing player 2, Everett, R, born 9/1/1948, male, joined 1975, living at Stoney Road 43. The table contains 12 rows in total. The DataGridView has a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

PLAYERNO	NAME	INITIALS	BIRTH_DATE	SEX	JOINED	STREET	HOUSENO
2	Everett	R	9/1/1948	M	1975	Stoney Road	43
6	Parmenter	R	6/25/1964	M	1977	Haseltine Lane	80
7	Wise	GWS	5/11/1963	M	1981	Edgecombe Way	39
8	Newcastle	B	7/8/1962	F	1980	Station Road	4
27	Collins	DD	12/28/1964	F	1983	Long Drive	804
28	Collins	C	6/22/1963	F	1983	Old Main Road	10
39	Bishop	D	10/29/1956	M	1980	Eaton Square	78
44	Baker	E	1/9/1963	M	1980	Lewis Street	23
57	Brown	M	8/17/1971	M	1985	Edgecombe Way	16
78	Thomas	M	8/22/1990 10...	M	2008	Hognikamp	2
83	Hope	PK	11/11/1956	M	1982	Magdalene Road	16A