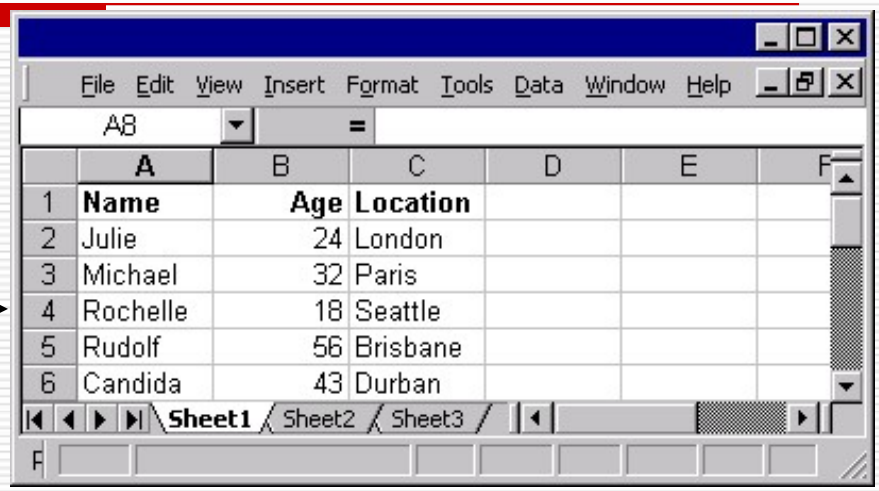
ADO.NET

Agenda

- **Data Access Components**
- ADO.NET
- DataBinding

Introduction

stored in →



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

Databases

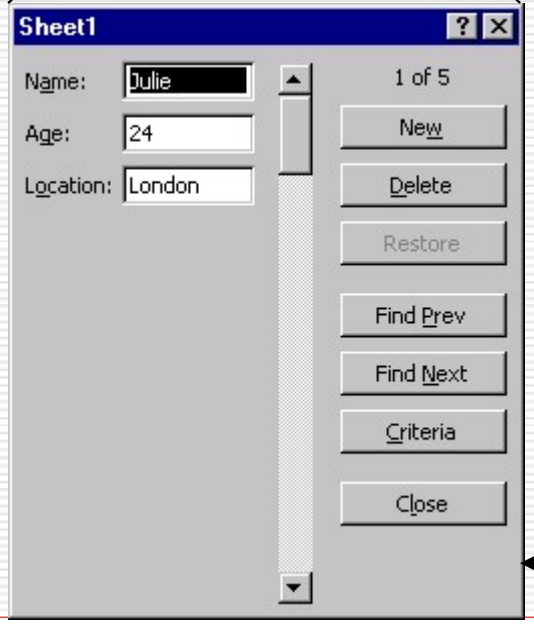
Client

↔ **Data Access Components** ↔ Databases

Client-Server

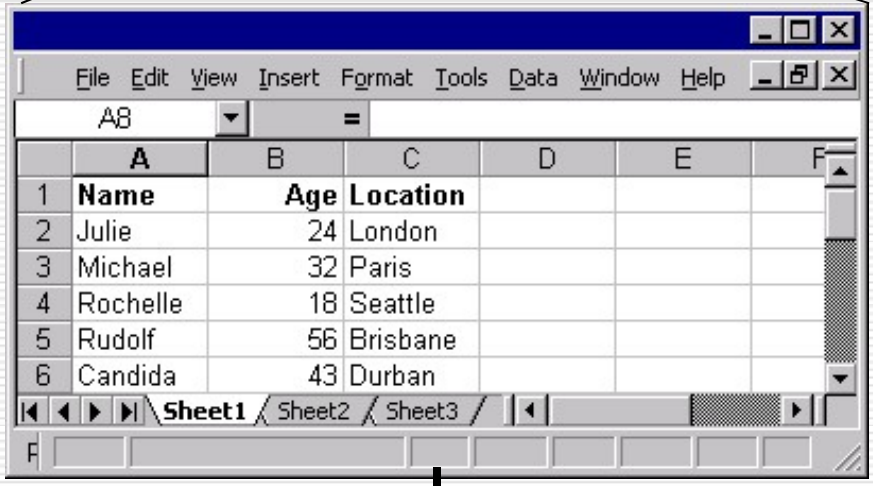
Client

Phần mềm Client



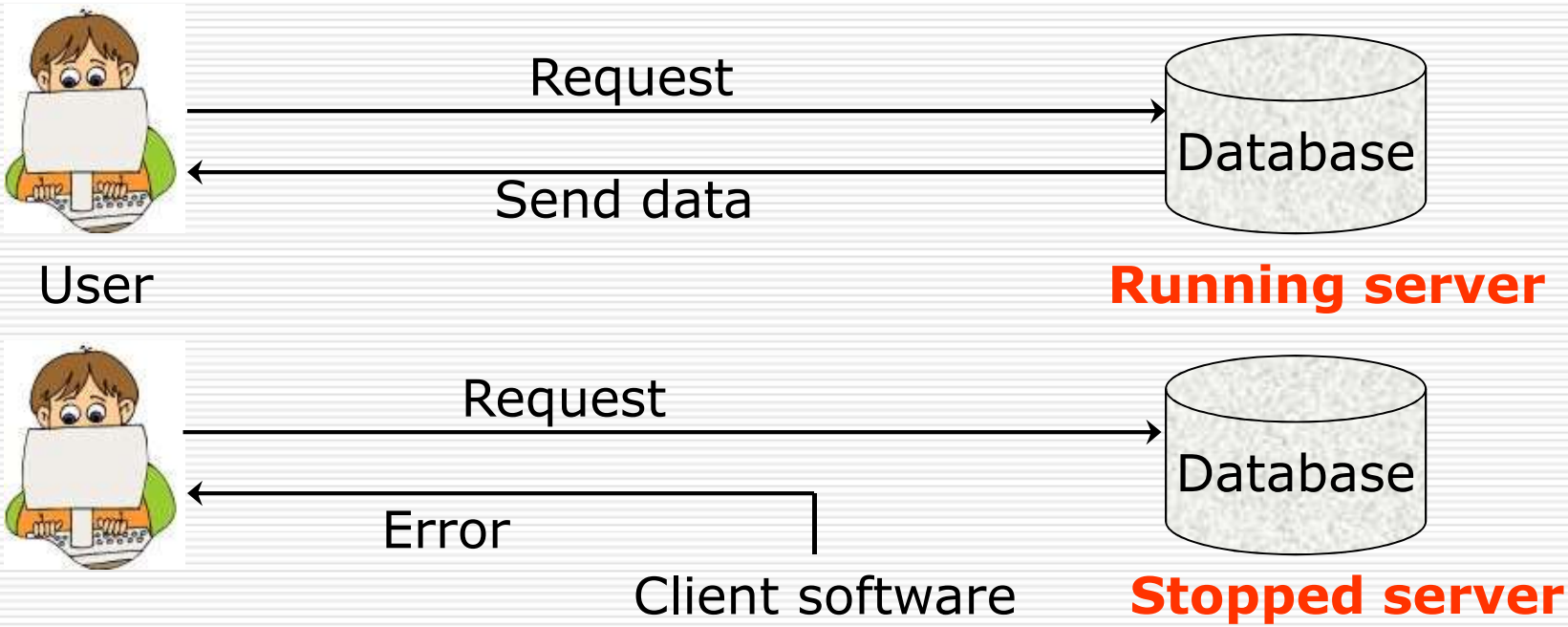
Server

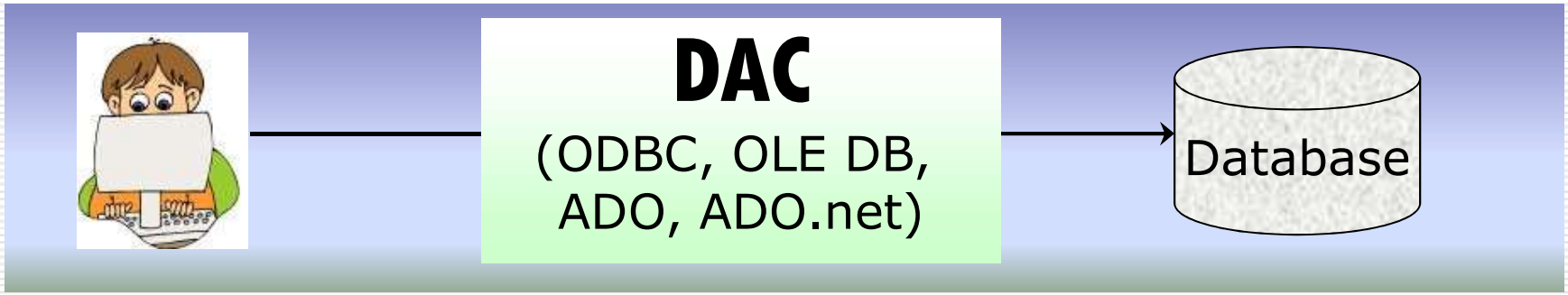
Cơ sở dữ liệu



Cung cấp dữ liệu cho Client

Data Access Components





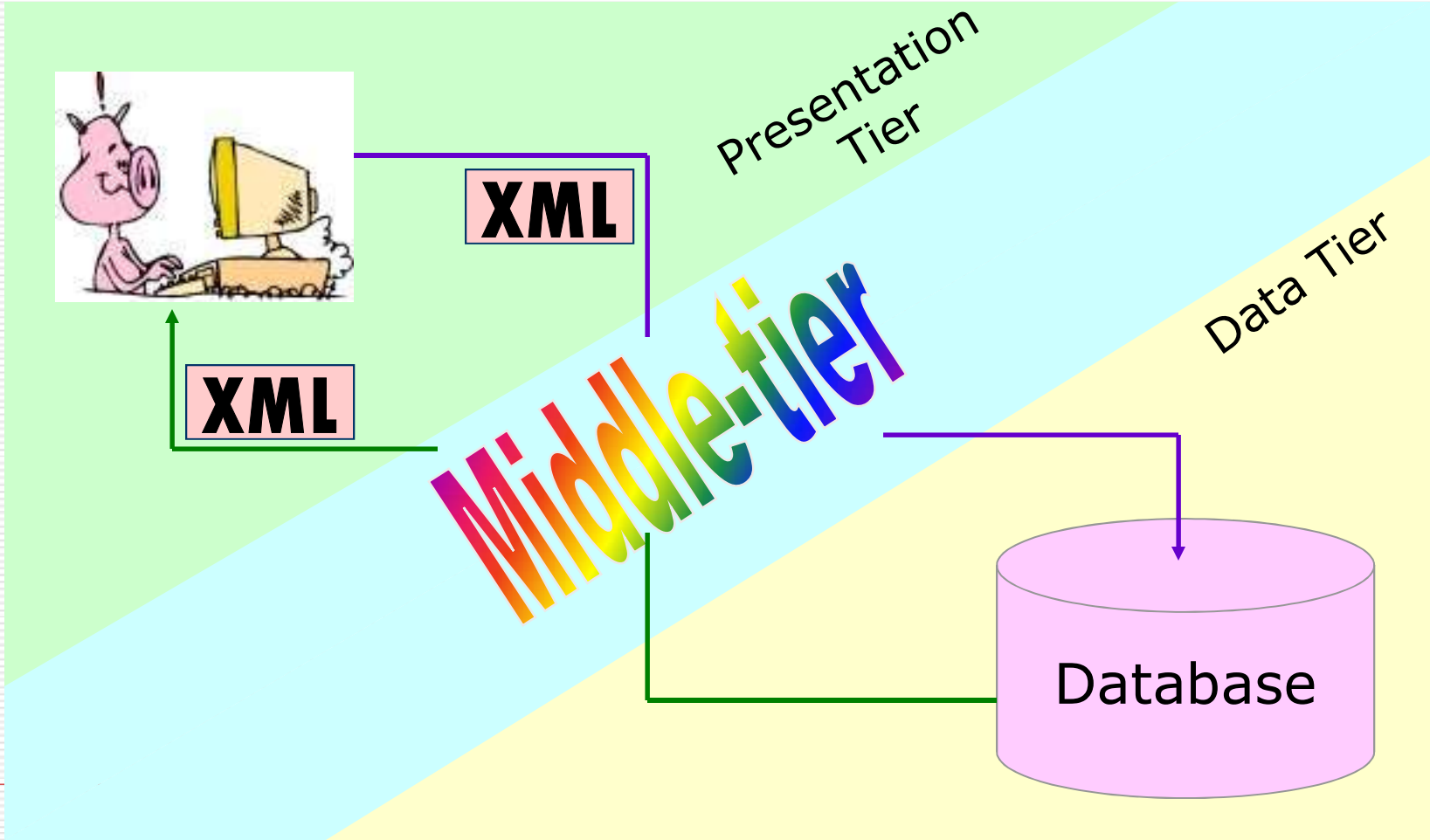
Agenda

- Data Access Components
- **ADO.NET**
- DataBinding

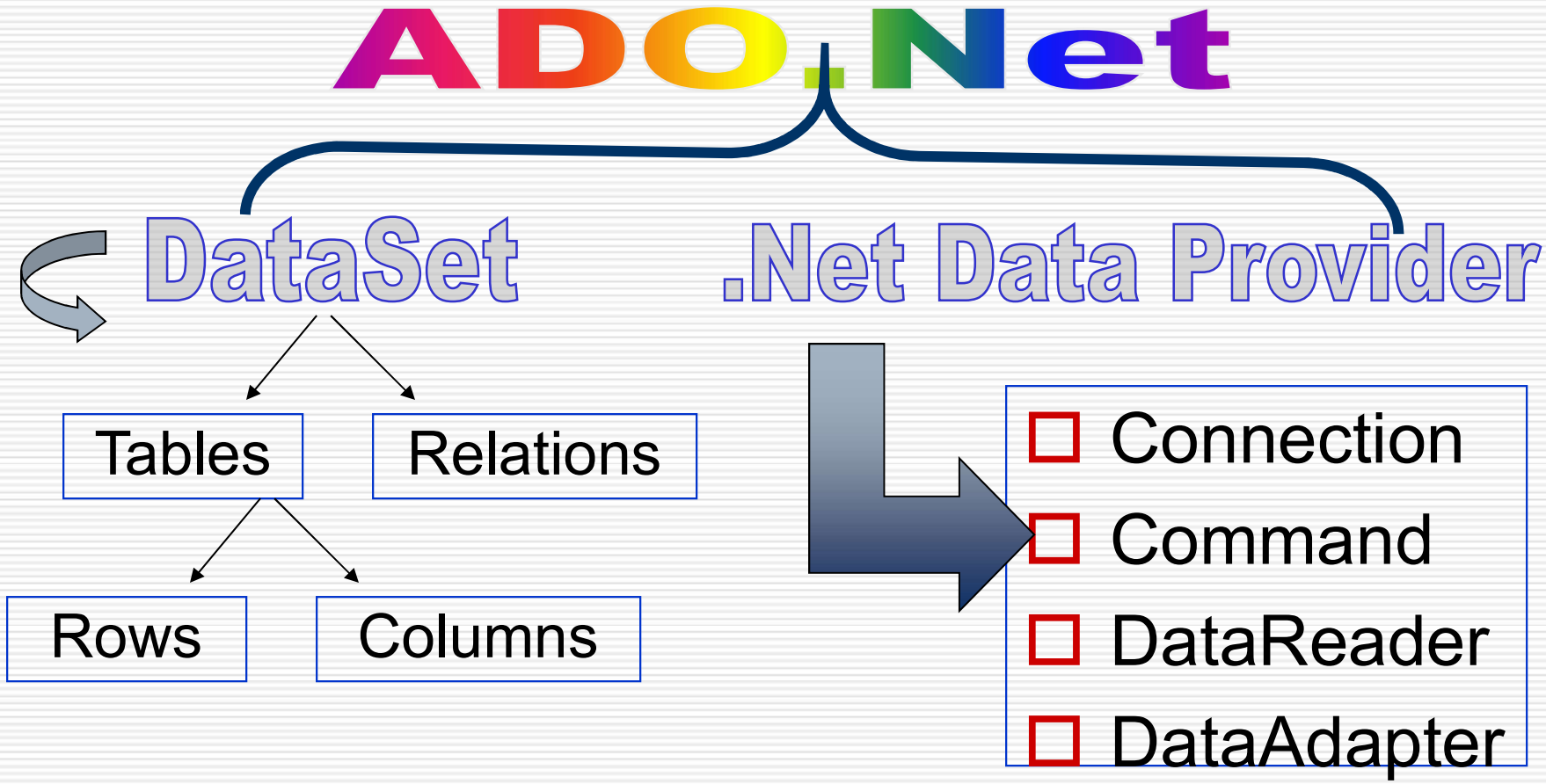
ADO.NET

- A DAC supported by BCL of .NET Framework
- Able to access varieties of databases

ADO.NET architecture



ADO.NET components



.NET Data Provider

- .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

Connection

Connection classes

- Create an object representing a connection to a database
- Include:
  - SqlConnection (SQL .Net Framework Data Provider)
  - MySqlConnection (MySQL .Net Data Provider)

Connection classes

- **Properties**
  - ConnectionString: hat specifies information about a data source and the means of connecting to it
- **Methods**
  - Open(): open a connection
  - Close(): Close a connection
  - CreateCommand(): create a Command object with a query input

SqlConnection

- Construction
  - ConnectionString)
    - Server: Server name
    - Database: database name
    - uid, pwd
    - …

    Separated by semicolon (;)
- Example:

```
SqlConnection sqlcon = new SqlConnection("Server=SQLDB;
      uid=mylogin;pwd=mylogin;Database=Tennis");
```

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);
        }
    }
}
```

Command

Command classes

- Represents a SQL statement or stored procedure to execute against a server database
- Common classes
  - SqlCommand: SQL .Net Framework Data Provider.
  - MySqlCommand: MySql Data Provider.

Command classes

- **Properties**
  - CommandText: a string representing a query, a stored procedure name or a table name
  - CommandType: either Text, StoredProcedure or TableDirect
  - Connection: a Connection object.
- **Methods**
  - ExecuteNonQuery(): Executes a SQL statement against the connection and returns the number of rows affected.
  - ExecuteReader(): Sends the CommandText to the Connection and builds a DataReader.
  - ExecuteScalar(): Executes the query, and returns the first column of the first row in the result set returned by the query. Additional columns or rows are ignored.

Example 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);
}
```

Example 2

```
try
{
    //Tim 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);
}
```

Pass values to a query

- **Problem**: A query is executed based on input values
- **Solutions**
  - Direct embed
  - Parameter classes

Direct embed

```
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);
}
```

Parameter classes

- Represents a parameter to a Command and optionally its mapping to DataSet columns
- Some classes
  - SqlParameter: SQL .Net Framework Data Provider
  - MySqlParameter: MySql Data Provider

Parameter classes

- **Properties**
  - ParameterName: name
  - DbType: Type
  - Value: Value
- **Constructor**
  - SqlParameter(string, SqlDbType)
  - MySqlParameter(string, MySqlDbType)
- Parameters: a collection of parameters of a Command object

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);
}
```

Stored procedure

- Use Command classes
- Steps
  - CommandText: stored procedure name
  - CommandType: StoredProcedure
  - Define parameters (Parameter classes, optionally)
  - Add parameters to the Parameters collection
  - Execute

Stored procedure

- A stored procedure update player birthday

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

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);
}
```

DataReader

DataReader

- Provides a way of reading a forward-only stream of rows from a server database
- A DataReader is created as the ExecuteReader() of a Command object is executed.
- Some classes
  - SqlDataReader: SQL .Net Framework Provider
  - MySqlDataReader: MySql Data Provider

DataReader

- **Properties**
  - HasRows: Gets a value that indicates whether the SqlDataReader contains one or more rows.
- **Phương thức**
  - Close: close a DataReader
  - Read: Advances the DataReader to the next record.
  - GetBoolean: Gets the value of the specified column as a boolean object.
  - GetDateTime: Gets the value of the specified column as a datetime object.
  - GetInt32: Gets the value of the specified column as an int object.
  - GetString: Gets the value of the specified column as a string object.

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);
}
```

DataAdapter

DataAdapter classes

- Represents a set of SQL commands and a database connection that are used to fill the DataSet and update the data source.
- Common classes
  - SqlDataAdapter: SQL .Net Framework Data Provider
  - MySqlDataAdapter: MySql .Net Data Provider

DataAdapter classes

Properties	Description
SelectCommand	A Command to retrieve data
InsertCommand	Update databases due to Dataset changes (via 3 Commands to Insert, Update, Delete)
UpdateCommand	
DeleteCommand	

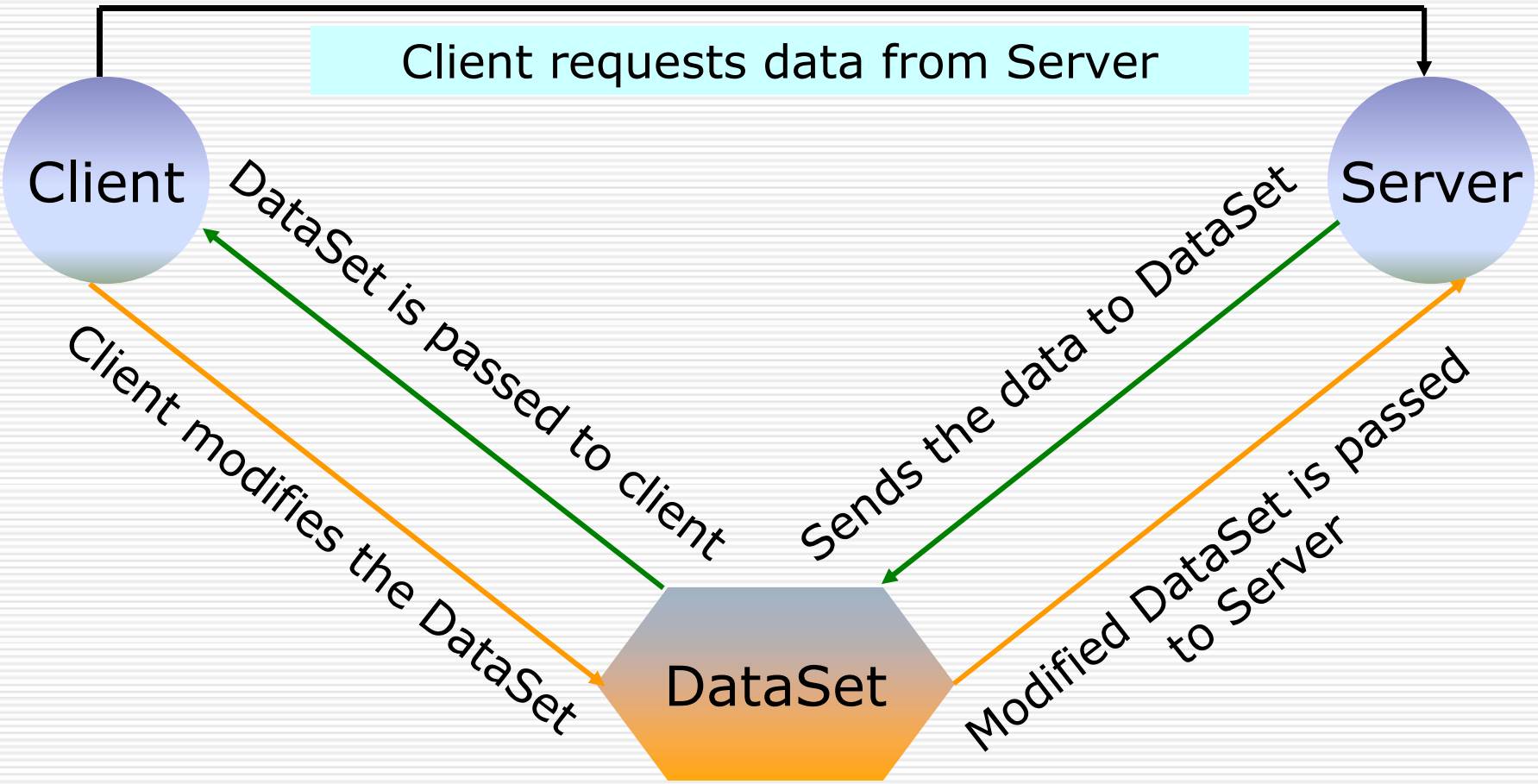
Method	Description
Fill	Adds or refreshes rows in the DataSet to match those in the data source.

DataSet

DataSet

- Represents an in-memory cache of data.
- The structure of a dataset is similar to that of a relational database: DataTable, DataRow, DataColumn, DataRelation, etc.

DataSet


Client requests data from Server
Client
Server
DataSet is passed to client
Sends the data to DataSet
Client modifies the DataSet
Modified DataSet is passed to Server
DataSet

Tables collection of DataSet

Properties

Item

Methods

Add

Remove

RemoveAt

DataTable

- Represents one table of in-memory data

Collection of many tables

One of the tables in DataSet

DataTable

- **Properties**
  - Columns: Gets the collection of columns that belong to this table.
  - Constraints: Gets the collection of constraints maintained by this table.
  - PrimaryKey: Gets or sets an array of columns that function as primary keys for the data table.
  - Rows: Gets the collection of rows that belong to this table.

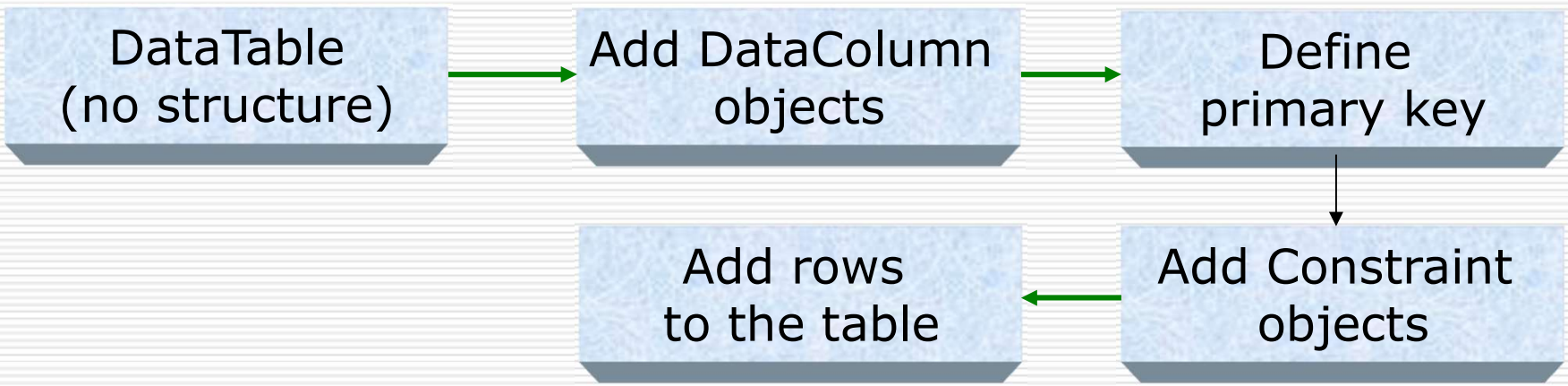
DataColumn

- Represents the schema of a column in a DataTable.
- **Properties**
  - ColumnName: name
  - DataType: data type

DataRow

- Represents a row of data in a DataTable.

Data table definition



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.

Other classes

- DataView
- Constraint
- DataRelation

Fill data to a DataSet

- Connect to a database (Connection)
- Create a DataAdapter
- Provide a query (SelectCommand of DataAdapter)
- Fill data from DataAdapter to a Dataset

Fill data to a dataset

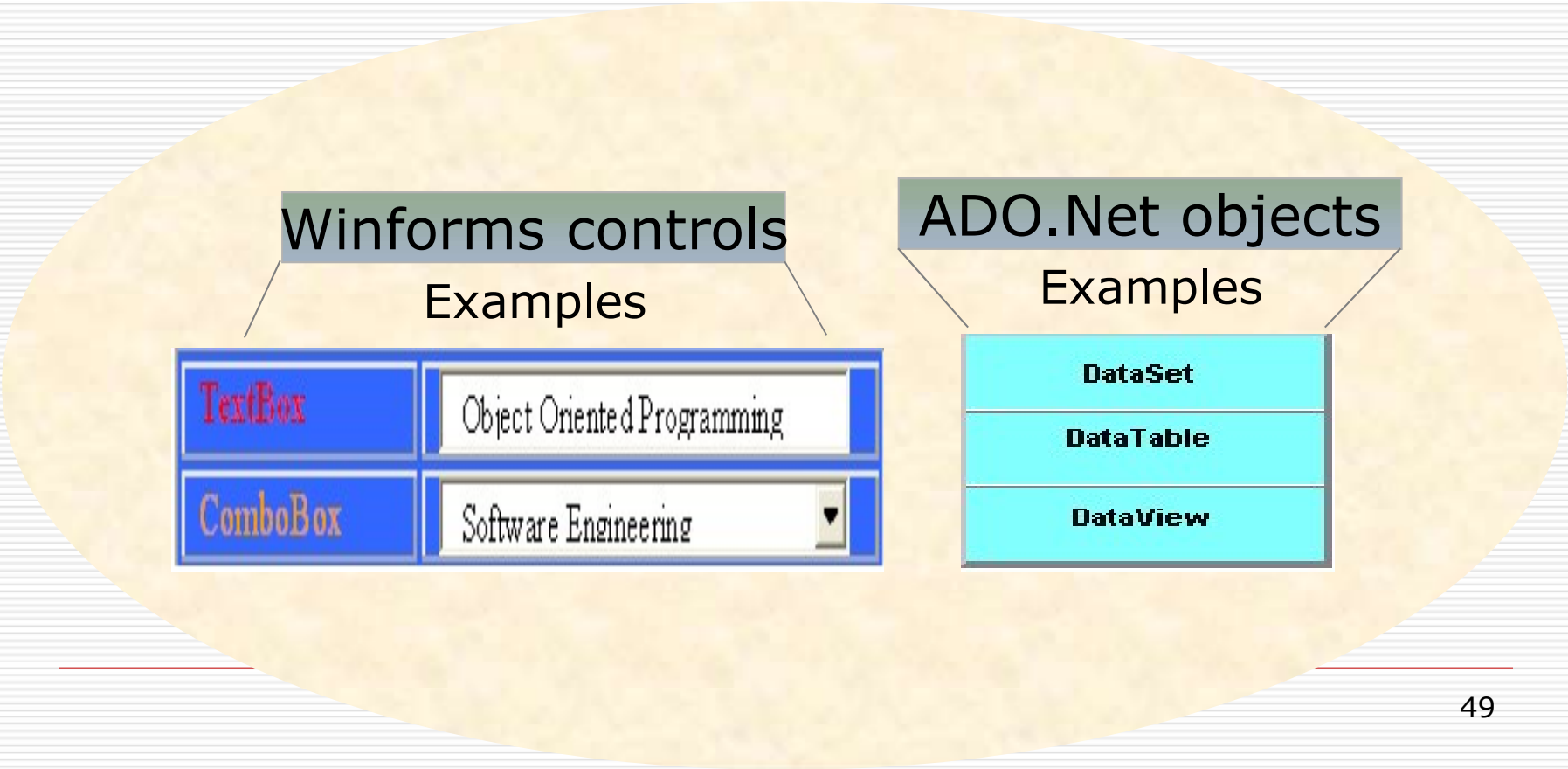
```
try
{
    SqlDataAdapter da = new SqlDataAdapter("Select * From Players",con);
    DataSet ds = new DataSet();
    da.Fill(ds,"CauThu");
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Agenda

- Data Access Components
- ADO.NET
- **DataBinding**

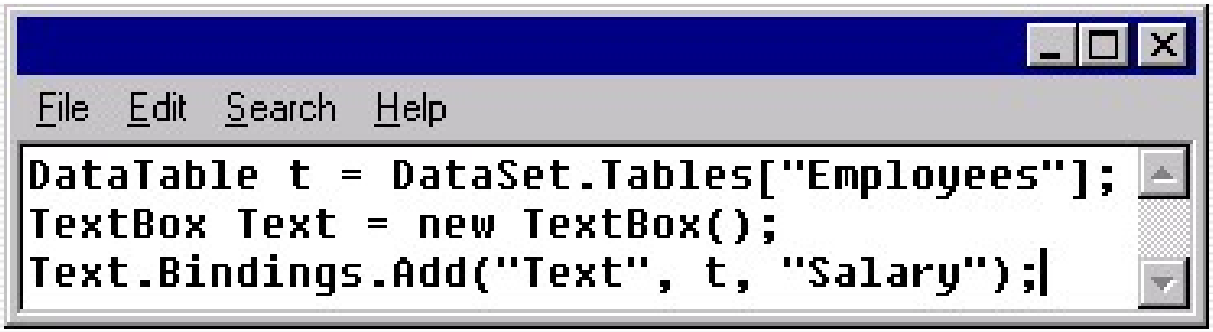
Data binding

- Bound values in data sources (DataSet, DataTable, DataColumn,…) to Winforms controls



Simple Binding

- A column of a DataTable is bounded to a Winforms control
- **Example**:



Complex Binding

- All rows of a column of a data source (or the whole DataTable or DataSet) are bounded to a list controls
- **Example**:

File Edit Search Help

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

DatagridView control

- Represents tabular data
- Able to add, update, delete, sort, page
- **Properties**
  - DataSource: a data source (DataSet, DataTable, …)
- **Example**: Details of Players

DatagridView controls example [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);
    }
}
```

DatagridView controls example [2]

Player Details

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