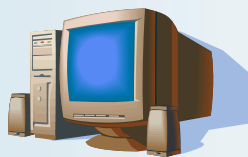


# PHẦN 7: ADO.NET

- ADO.NET
- Quá trình phát triển
- Đặc điểm ADO.NET
- .NET Data Provider
- DataSet
- Data Source Configuration Wizard

# GIỚI THIỆU ADO.NET

- ActiveX Data Object .NET (ADO.NET)
  - Công nghệ của MS
  - Phát triển từ nền tảng ADO
  - Cung cấp các **lớp đối tượng** và **hàm thư viện** phục vụ cho việc kết nối và xử lý dữ liệu



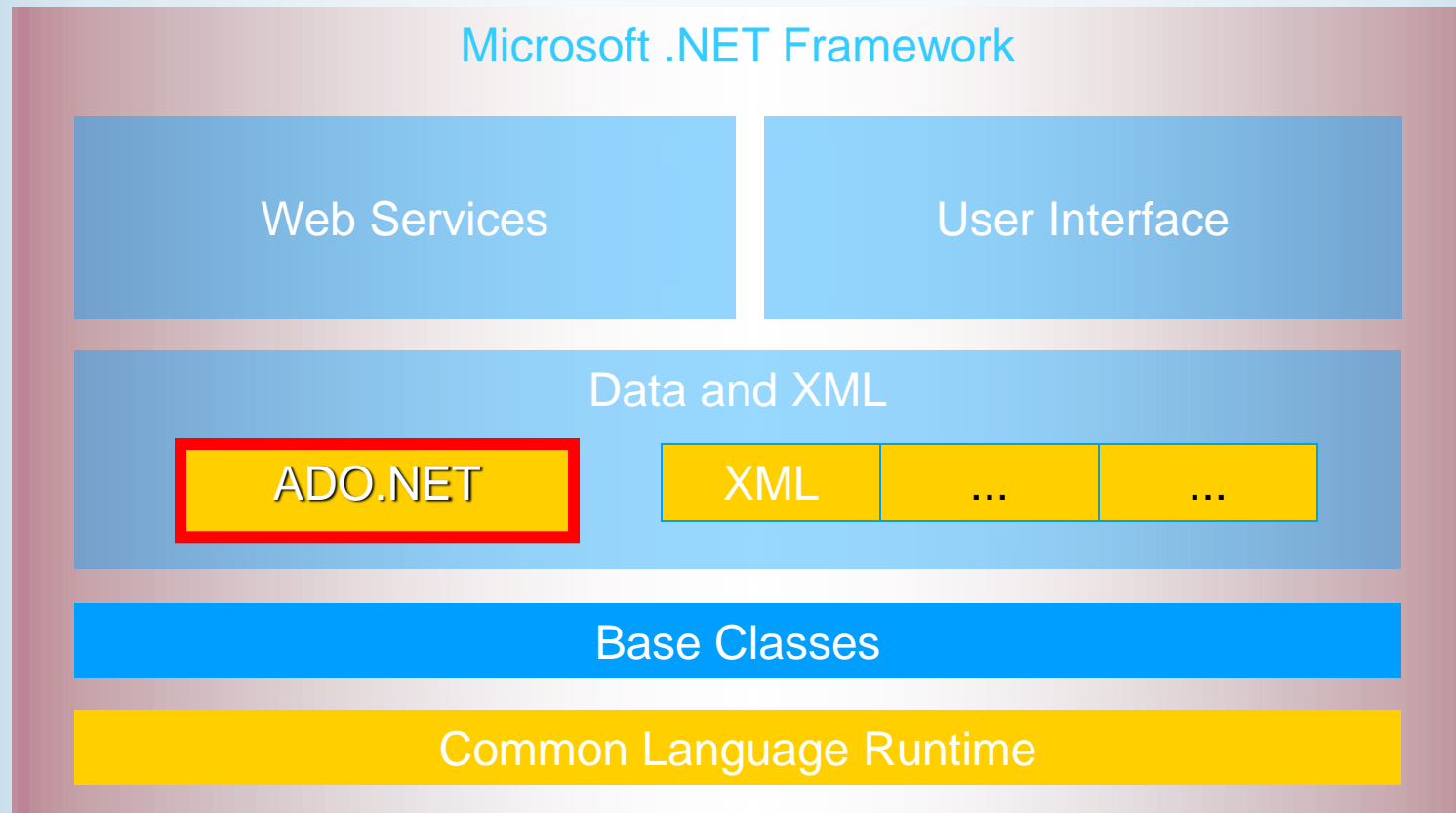
.NET Application

ADO.NET



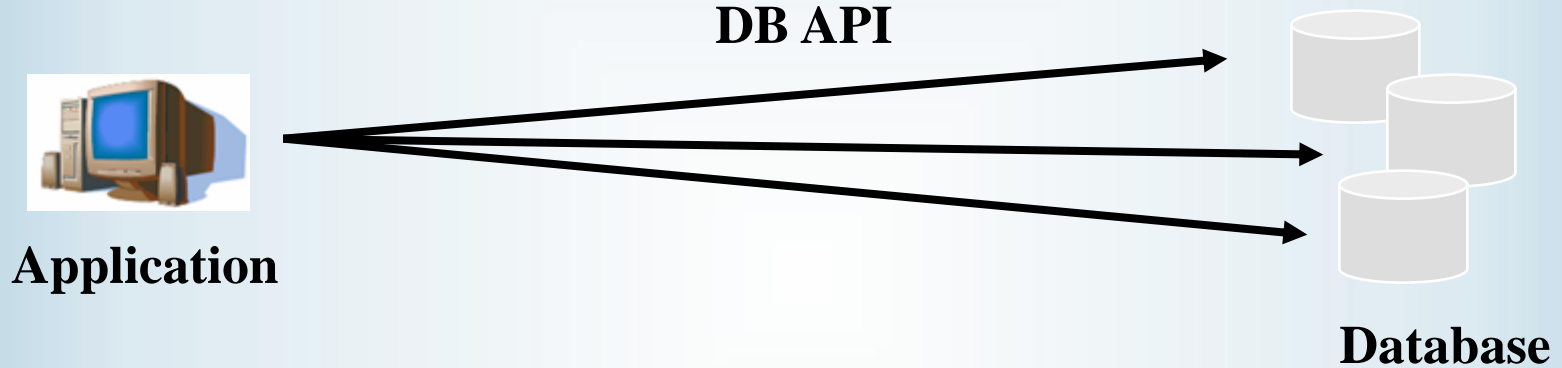
# GIỚI THIỆU ADO.NET

- Mô hình .NET Framework

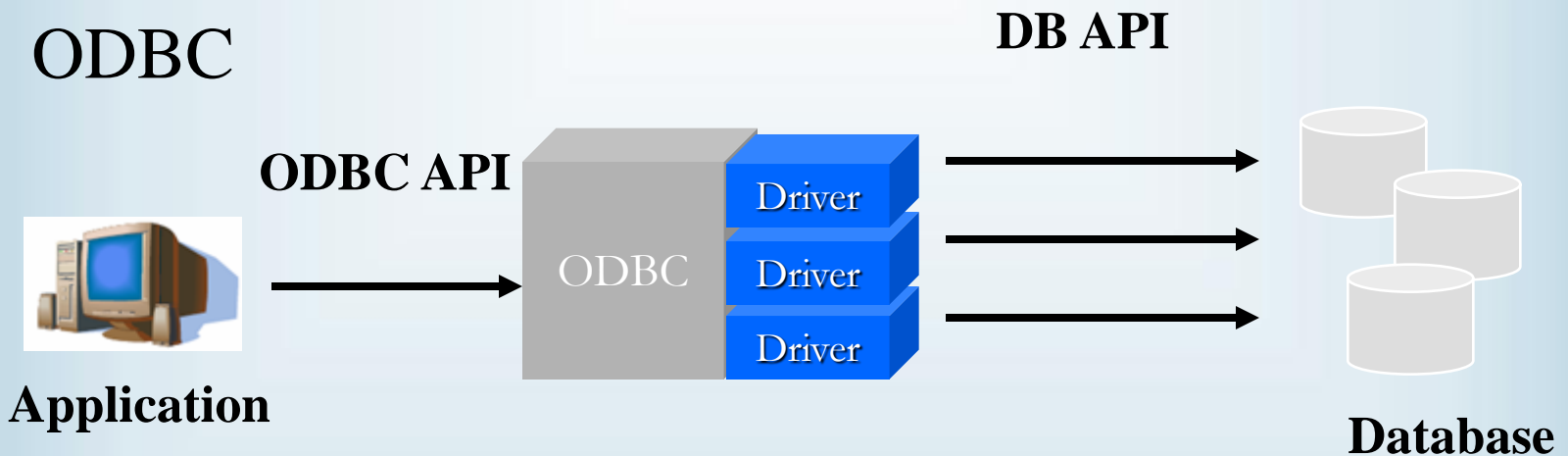


# QUÁ TRÌNH PHÁT TRIỂN

- Native API



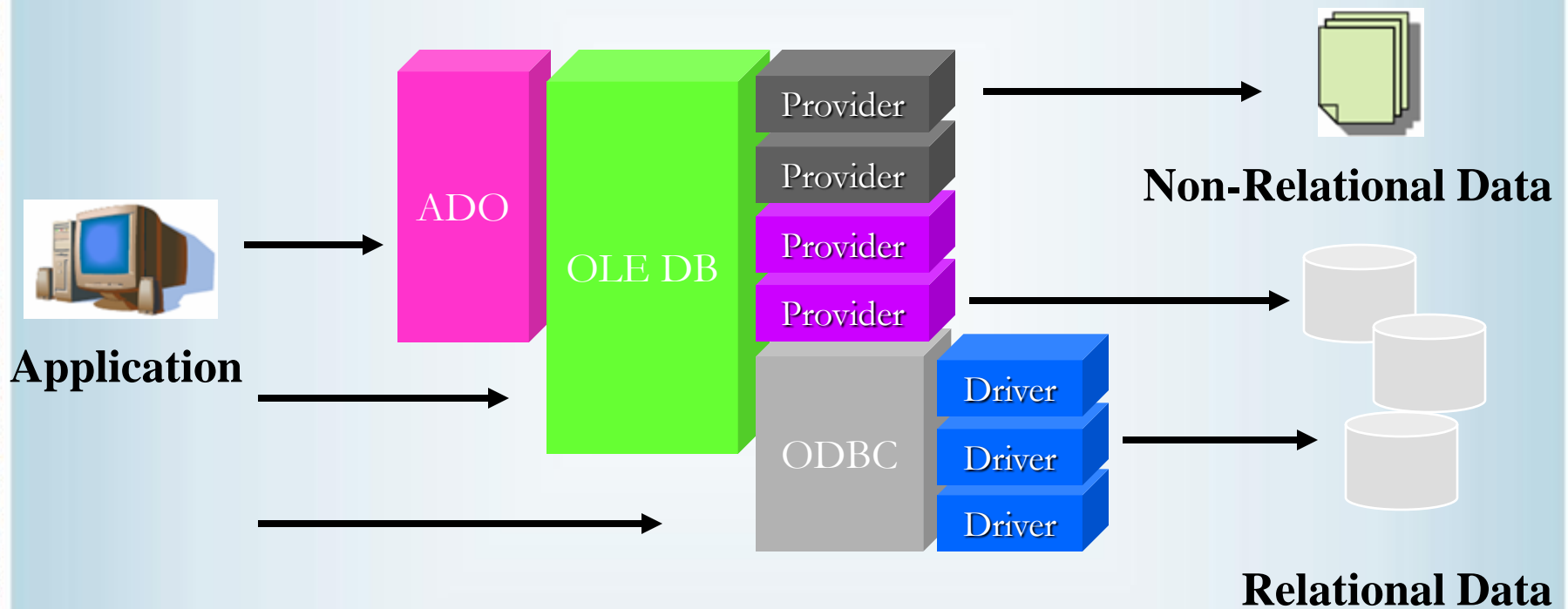
- ODBC





# QUÁ TRÌNH PHÁT TRIỂN

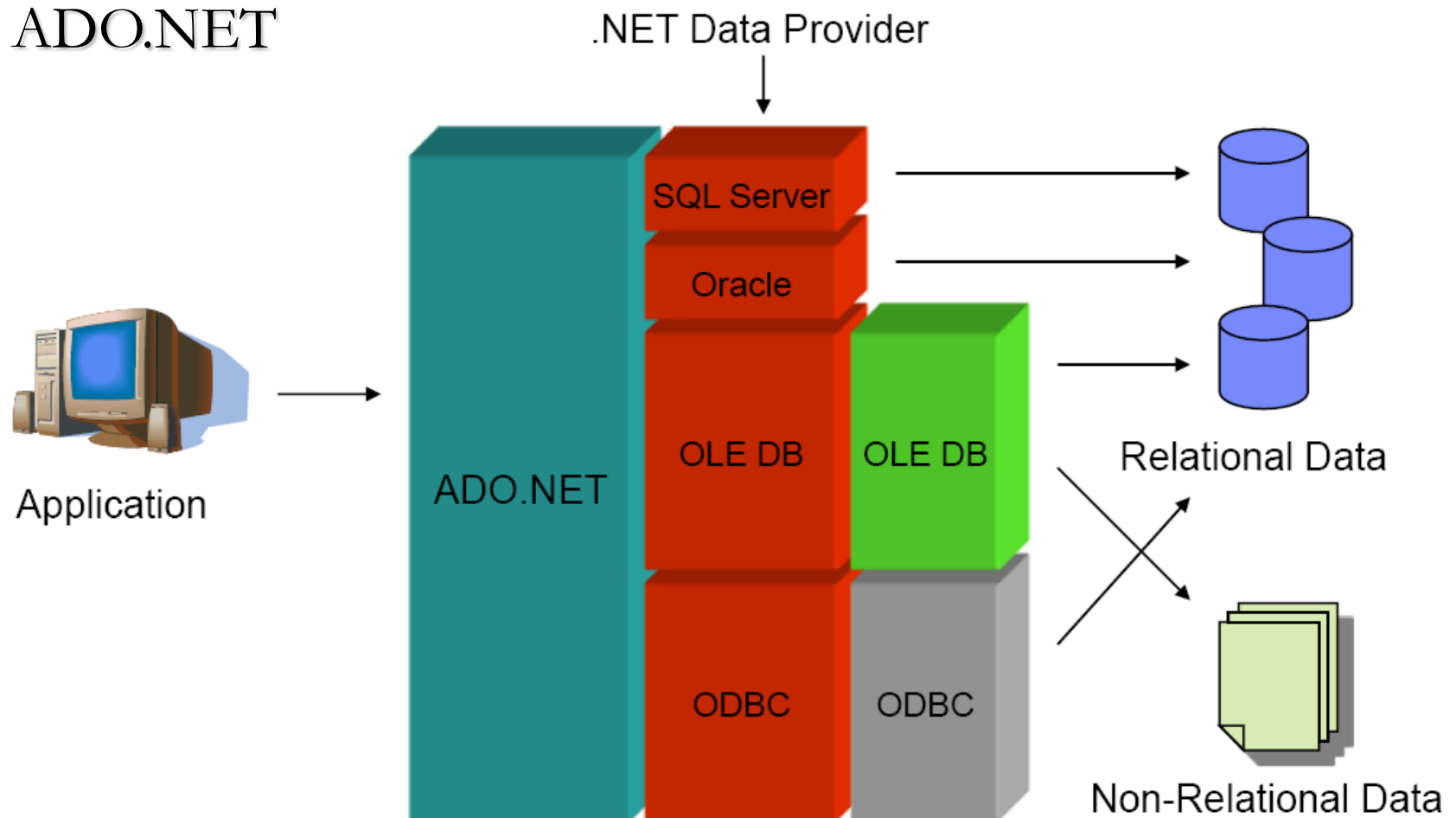
- ADO, OLE DB



*OLE: Object Linking and Embedding*

# QUÁ TRÌNH PHÁT TRIỂN

ADO.NET



# ADO.NET

- ADO.NET là một phần của .NET Framework
  - Thư viện lớp có chức năng **xử lý dữ liệu** trong ngôn ngữ MS.NET
- ADO.NET là dạng “*Disconnected*”
  - Cho phép lấy cả một cấu trúc phức tạp của dữ liệu từ CSDL, sau đó ngắt kết nối rồi mới thực hiện thao tác xử lý
  - Trước đây ADO luôn phải duy trì kết nối trong quá trình thiết kế

# ADO.NET

- ADO.NET mạnh mẽ
  - Kế thừa các ưu điểm của ADO
  - Kết hợp với ý tưởng thiết kế hoàn toàn mới
- Đặc điểm nổi bật
  - Thiết kế hoàn toàn dựa vào XML
    - ✓ Chuẩn giao tiếp dữ liệu tốt nhất trên môi trường Internet hiện nay
  - Thiết kế hoàn toàn hướng đối tượng
    - ✓ Đặc trưng của thư viện .NET Framework



# ADO.NET VÀ ADO

<b>Đặc Điểm</b>	<b>ADO</b>	<b>ADO.NET</b>
<b>DL xử lý được đưa vào bộ nhớ dưới dạng</b>	<b>Recordset : tương đương 1 bảng dữ liệu trong database</b>	<b>Dataset : tương đương 1 database</b>
<b>Duyệt dữ liệu</b>	<b>Recordset chỉ cho phép duyệt tuần tự, từng dòng một.</b>	<b>Dataset : duyệt “tự do, ngẫu nhiên”, truy cập thẳng tới bảng, dòng ,cột mong muốn.</b>
<b>Dữ liệu ngắt kết nối</b>	<b>Recordset thiên về hướng kết nối, nên việc hỗ trợ ngắt kết nối không mạnh</b>	<b>Dataset hỗ trợ hoàn toàn ngắt kết nối</b>
<b>Trao đổi dữ liệu qua Internet</b>	<b>Khả năng trao đổi dữ liệu ADO qua Internet thường có nhiều hạn chế. Do dùng chuẩn COM</b>	<b>ADO.NET trao đổi dữ liệu qua Internet rất dễ dàng vì ADO.NET được thiết kế theo chuẩn XML, là chuẩn dữ liệu chính được sử dụng để trao đổi trên Internet.</b>

# MÔI TRƯỜNG “CONNECTED”

- Mỗi user có một kết nối cố định tới data source
- Ưu điểm
  - Môi trường được bảo vệ tốt
  - Kiểm soát được sự đồng bộ
  - Dữ liệu luôn được mới
- Nhược
  - Phải có một kết nối mạng cố định
  - Scalability

# MÔI TRƯỜNG “CONNECTED”

- Một tập con của dữ liệu trung tâm được sao chép và bổ sung độc lập, sau đó sẽ được merge lại vào dữ liệu trung tâm.
- Ưu điểm
  - Có thể làm việc bất cứ lúc nào, cũng như có thể kết nối bất kỳ vào Data Source
  - Cho phép user khác có thể kết nối
  - Nâng cao hiệu suất thực hiện của ứng dụng
- Khuyết
  - Dữ liệu không được cập nhật một cách nhanh nhất
  - Sự tranh chấp có thể xuất hiện và phải giải quyết



# ADO.NET

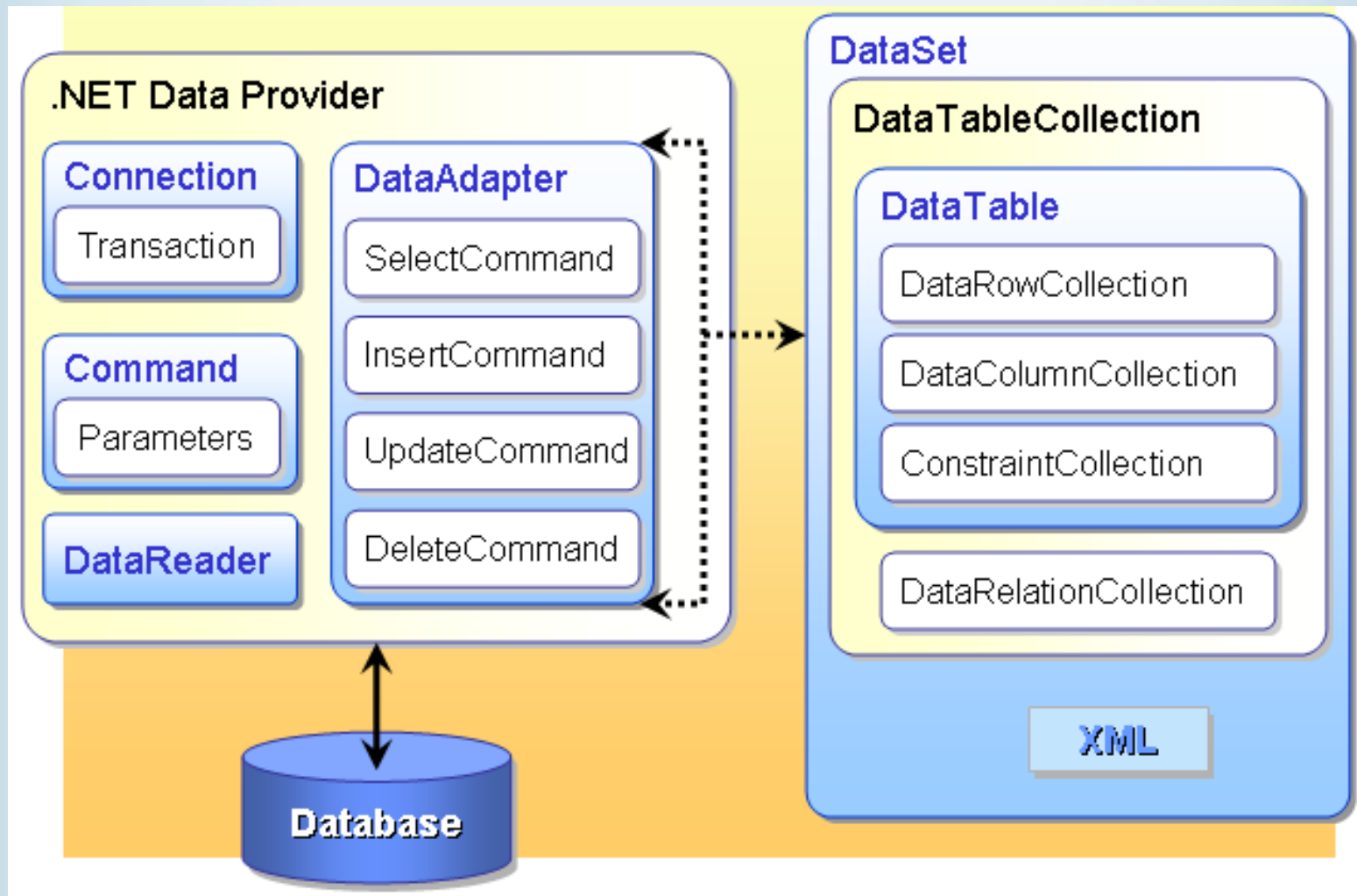
- Kiến trúc của ADO.NET gồm 2 phần chính:
  - Phần kết nối;
  - Phần ngắt kết nối.
- **Phần kết nối:** sử dụng khi kết nối CSDL và thao tác dữ liệu, phải thực hiện kết nối khi thao tác
  - **Connection:** quản lý việc đóng mở DB
    - ???Connection: **SqlConnection**, **OleDbConnection**
  - **Command:** lệnh truy vấn, tương tác dữ liệu khi đang lập kết nối
    - ???Command: **SqlCommand**, **OleDbCommand**
  - **DataReader:** đọc dữ liệu, chỉ xử lý 1 dòng dữ liệu tại một thời điểm
    - ???DataReader: **SqlDataReader**, **OleDbDataReader**
  - **DataAdapter:** cầu nối giữa DB và DataSet



# ADO.NET

- **Phần ngắt kết nối:** là DataSet
  - DataSet không quan tâm đến DB thuộc kiểu gì, và lấy dữ liệu từ DataAdapter để xử lý
  - DataSet xem như một DB trong bộ nhớ: bảng, quan hệ...
  - DataSet có các thành phần con như
    - ✓ DataTable
    - ✓ DataRow
    - ✓ DataColumn
    - ✓ DataRelation
    - ✓ Các đối tượng nhóm: DataTableCollection, DataRowCollection, DataColumnCollection

# MÔ HÌNH ĐỐI TƯỢNG ADO.NET



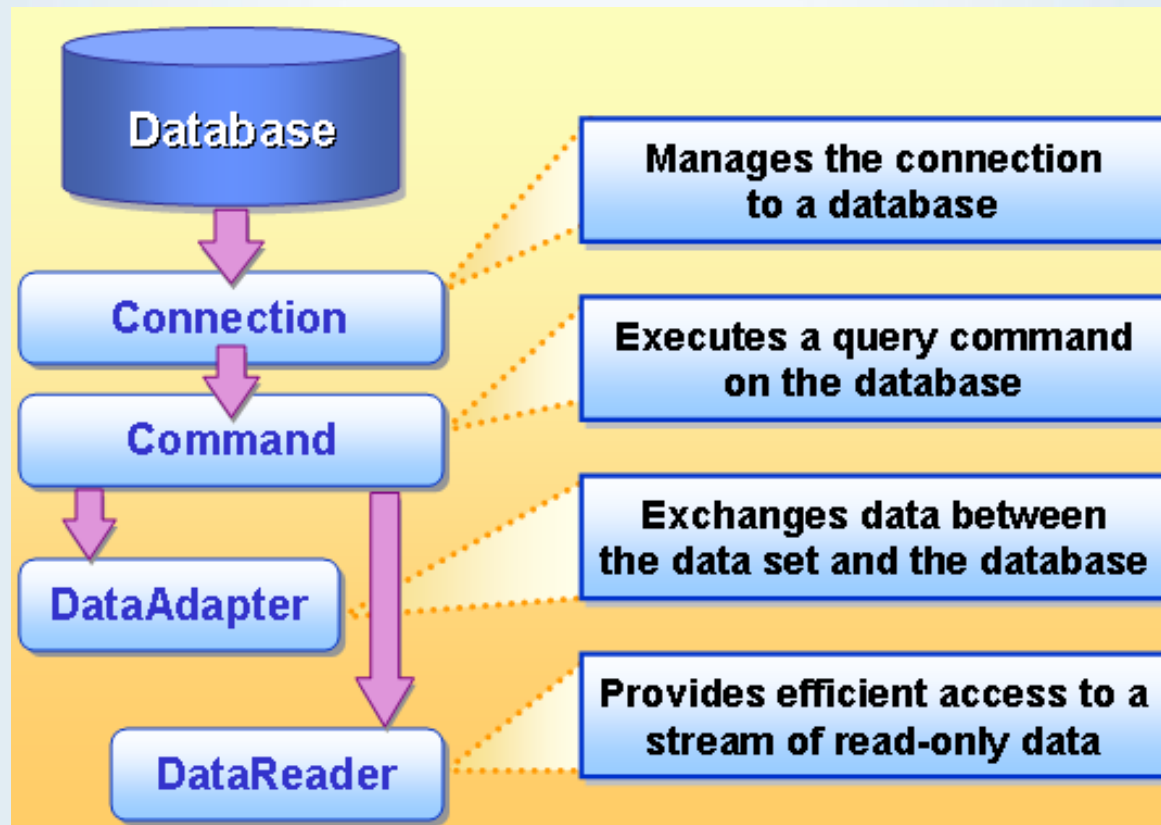
# CÁC LỚP THƯ VIỆN ADO.NET

- System.Data.OleDb: Access, SQL Server, Oracle
- System.Data.SqlClient: SQL Server
- System.Data.OracleClient: Oracle
- Đặc điểm:
  - Cả ba thư viện trên về giao tiếp lập trình là giống nhau
  - Dùng thư viện SqlConnection truy xuất SQL Server nhanh hơn OleDb
  - Tương tự cho OracleClient



# NET DATA PROVIDER

- Mô hình .NET Data Provider





# NET DATA PROVIDER - CONNECTION



**Application**

Connection



**Database**

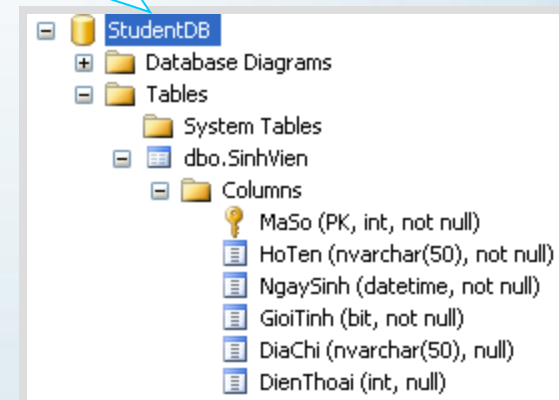
## Thuộc tính & Phương thức

- **ConnectionString**: chuỗi kết nối DataSource
- **Open()**: thiết lập kết nối đến DS
- **Close()**: đóng kết nối với DS

# NET DATA PROVIDER - CONNECTION

- Mô tả CSDL sử dụng minh họa

Đặc điểm	Thông tin
DBMS	MS SQL Server Express 2005
DB 1	CSDL StudentDB
DB 2	CSDL Northwind



# NET DATA PROVIDER - CONNECTION

## SQL Connection


```
SqlConnection cnn = new SqlConnection();  
cnn.ConnectionString = “server=.\\SQLEXPRESS;  
database=Northwind; Trusted_Connection=true”;
```

```
cnn.Open();
```

```
//xử lý trong quá trình kết nối
```

```
...
```

```
cnn.Close();
```



Thay bằng “localhost”  
trong SQL Server 2005

# NET DATA PROVIDER - CONNECTION

Database	ODBC/OLE DB Connection (*)
MS Access	<b>Driver = {Microsoft Access Driver (*.mdb)};</b> <b>DBQ = &lt;đường dẫn file access&gt;</b> <b>Provider=Microsoft.Jet.OLEDB.4.0; Data Source</b> <b>= &lt;đường dẫn file access&gt;</b>
SQL Server	<b>Driver = {SQLServer}; Server = ServerName;</b> <b>Database= DatabaseName; Uid=Username;</b> <b>Pwd=Password;</b> <b>Provider= SQLOLEDB; Data Source=ServerName;</b> <b>Initial Catalog=DatabaseName; UserId=Username;</b> <b>Password=Password</b>

(\*): Xem thêm chuỗi kết nối trong: <http://www.connectionstrings.com/?carrier=sqlserver2005>



# NET DATA PROVIDER - CONNECTION

## OleDb Connection

```
OleDbConnection cnn = new OleDbSqlConnection();  
cnn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;  
Data Source =Sinhvien.mdb";
```

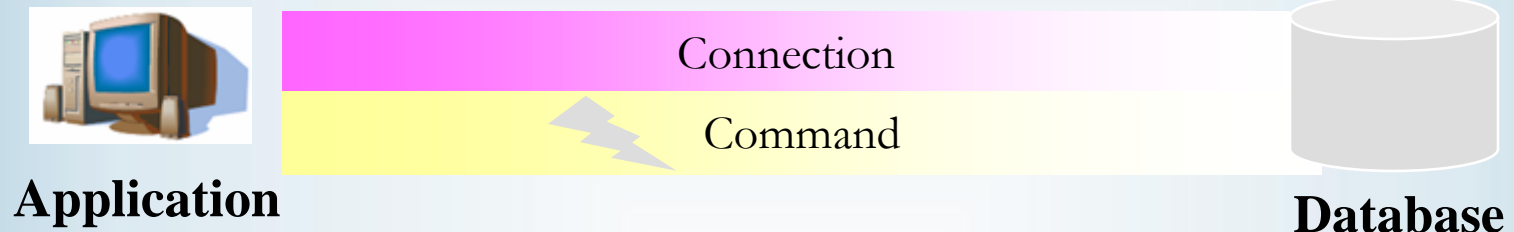
```
cnn.Open();
```

```
//xử lý trong quá trình kết nối
```

```
...
```

```
cnn.Close();
```

# NET DATA PROVIDER - COMMAND



## Thuộc tính & Phương thức

- **Connection:** kết nối để thực hiện lệnh
- **CommandText:** câu lệnh cần thực hiện
- **CommandType:** loại câu lệnh (Text, TableDirect, StoredProc)
- **ExecuteScalar():** thực hiện câu lệnh và trả về giá trị đơn
- **ExecuteNonQuery():** gọi các lệnh SQL, store, trả về số row bị tác động (Insert, Update, Delete...)
- **ExecuteReader():** thực hiện lệnh và trả về DataReader

# NET DATA PROVIDER - COMMAND

## SQL Command

```
SqlConnection cnn = new SqlConnection();  
cnn.ConnectionString = “server=.\SQLEXPRESS;  
database=Northwind; Trusted_connection=true”;
```

```
SqlCommand cmd = new SqlCommand(“”,cnn);  
cmd.CommandText = “Select COUNT(*) From Customers”;
```

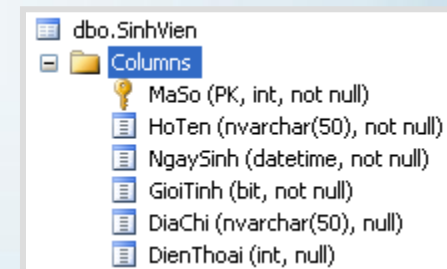
```
cnn.Open();
```

```
int count = (int) cmd.ExecuteScalar();  
cnn.Close();
```

# NET DATA PROVIDER - COMMAND

## SQL Command

```
SqlConnection conn = new SqlConnection();  
conn.ConnectionString = "Data source=.\SQLEXPRESS;" +  
    "Initial Catalog=StudentDB; Integrated security=yes;";  
  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText =  
    "Insert into Sinhvien values(007,'Le Nam','1/1/1980',1,null,null)";  
cmd.Connection = conn;  
  
conn.Open();  
  
cmd.ExecuteNonQuery();  
  
conn.Close();
```



The screenshot shows the 'Columns' tab for the 'dbo.SinhVien' table. It lists six columns with their data types and constraints:

Column Name	Data Type	Constraints
MaSo	int	PK, not null
HoTen	nvarchar(50)	not null
NgaySinh	datetime	not null
GioiTinh	bit	not null
DiaChi	nvarchar(50)	null
DienThoai	int	null



# NET DATA PROVIDER - COMMAND

- Tham số hóa câu lệnh
  - Một câu lệnh được sử dụng nhiều lần

## SQL Command - Param

```
...  
cmd.CommandText = "Insert into Sinhvien  
values(@MS,@HT,@NS,@GT,@DC,@DT)";  
  
cmd.Parameters.Add("@MS", SqlDbType.Int);  
cmd.Parameters.Add("@HT", SqlDbType.NVarChar);  
cmd.Parameters.Add("@NS", SqlDbType.DateTime);  
cmd.Parameters.Add("@GT", SqlDbType.Bit);  
cmd.Parameters.Add("@DC", SqlDbType.NVarChar);  
cmd.Parameters.Add("@DT", SqlDbType.Int);
```

**Câu lệnh  
được tham  
số hóa**

**Khai báo  
tham số**

# NET DATA PROVIDER - COMMAND

## SQL Command - Param

```
...  
cmd.Parameters["@MS"].Value = 999;  
cmd.Parameters["@HT"].Value = "Nguyen Ha Giang";  
cmd.Parameters["@NS"].Value = new DateTime(1978,12,4);  
cmd.Parameters["@GT"].Value = 1;  
cmd.Parameters["@DC"].Value = "Tan Binh";  
cmd.Parameters["@DT"].Value = 5120791;  
  
conn.Open();  
int count = (int) cmd.ExecuteNonQuery();  
  
conn.Close();
```

Truyền các  
giá trị của  
tham số  
tương ứng

# NET DATA PROVIDER - DATAREADER



## Thuộc tính & Phương thức

- **HasRow**: cho biết câu truy vấn có trả về dữ liệu
- **Read()**: đọc một mẫu tin
- **[i]**: truy xuất đến cột i của mẫu tin được đọc
- **Close()**: đóng

Truy xuất tuần tự

Chỉ đọc dữ liệu

Cơ chế kết nối



# NET DATA PROVIDER - DATAREADER

## DataReader

...

```
SqlCommand cmd = new SqlCommand("Select * From  
Sinhvien", conn);
```

```
SqlDataReader reader;
```

```
conn.Open();
```

```
reader = cmd.ExecuteReader();
```

```
while (reader.Read())
```

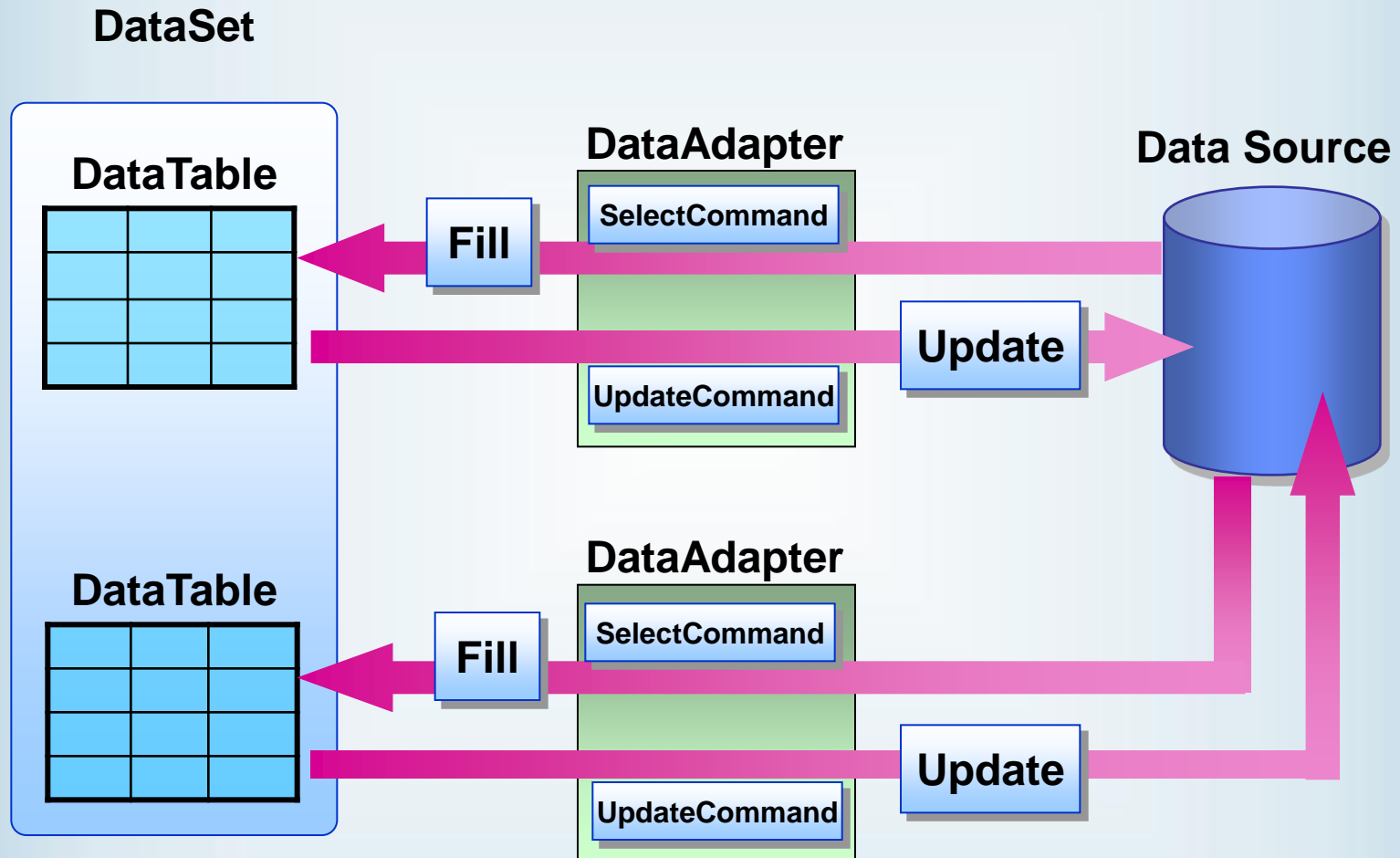
```
    listBox1.Items.Add(reader["Hoten"]);
```

```
reader.Close();
```

```
conn.Close();
```



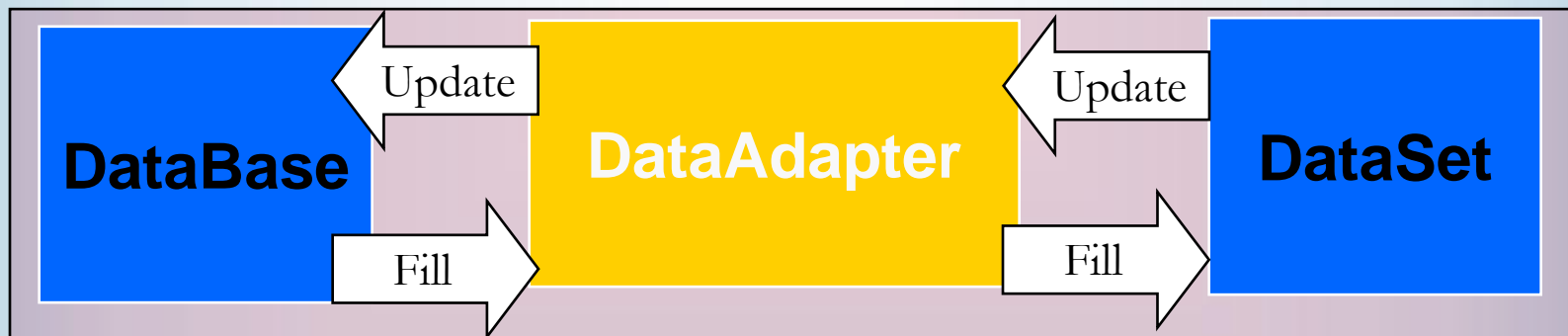
# NET DATA PROVIDER - DATAADAPTER



# NET DATA PROVIDER - DATAADAPTER

## Thuộc tính & Phương thức

- **Fill**(DataSet): sử dụng SelectCommand lấy dữ liệu từ Data Source đổ vào Data Set
- **Update**(DataSet): InsertCommand, UpdateCommand, DeleteCommand cập nhật dữ liệu trong DataSet vào DataSource



# NET DATA PROVIDER - DATAADAPTER

## DataAdapter

```
string strConn="Server=.\SQLEXPRESS; Database=StudentDB;  
Trusted_connection=true";
```

```
SqlDataAdapter adapter = new  
    SqlDataAdapter("Select * From Sinhvien",strConn);
```

```
DataSet ds = new DataSet();
```

```
adapter.Fill(ds);  
// thao tác trên dataset
```

```
...  
adapter.Update(ds);
```

# DATASET

- **DataSet**: là phần cơ sở dữ liệu được lưu trữ trong bộ nhớ (in-memory database)
- Cơ chế không kết nối
- Nhờ đối tượng DataAdapter làm trung gian
- Hỗ trợ đầy đủ đặc tính XML
- Thao tác được với tất cả mô hình lưu trữ hiện tại: flat, relational, hierarchical

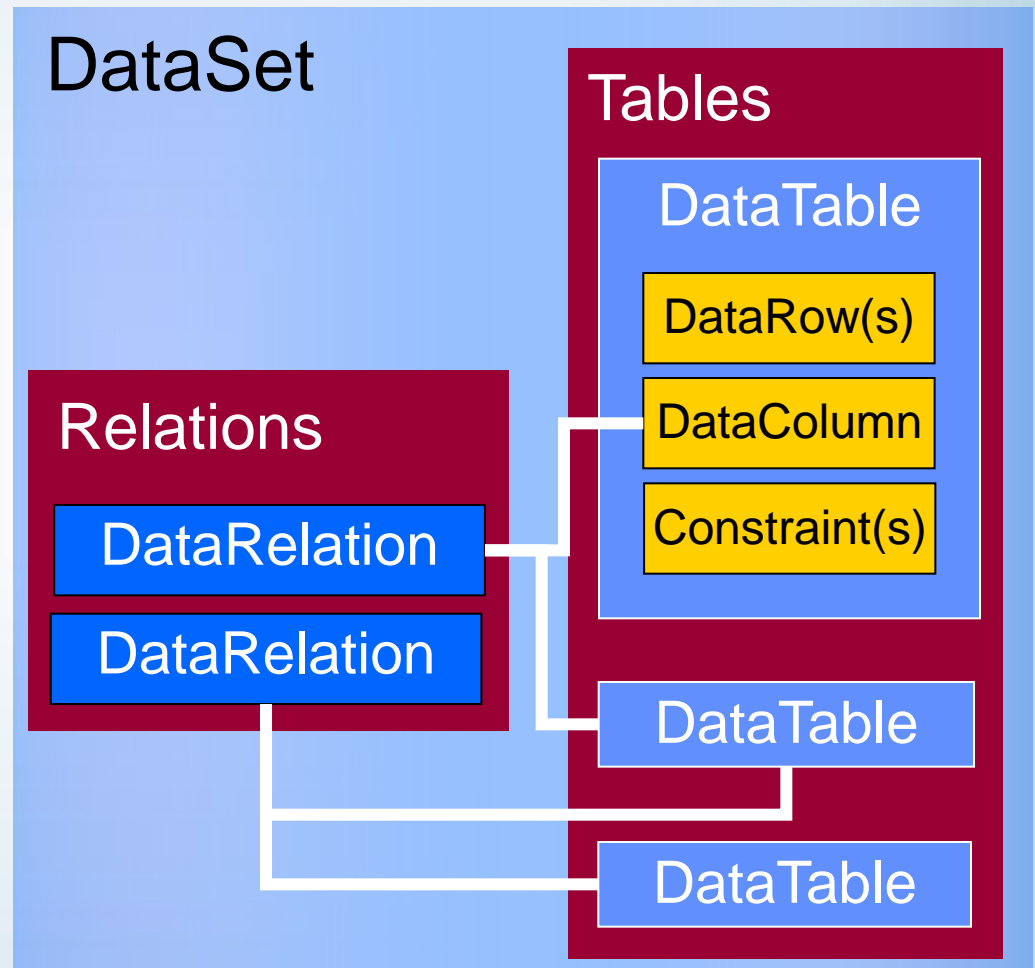


# **DATASET VS. DATAREADER**

<b>DataSet</b>	<b>DataReader</b>
<b>Read/write access to data</b>	<b>Read-only</b>
<b>Includes multiple tables from different databases</b>	<b>Based on one SQL statement from one database</b>
<b>Disconnected</b>	<b>Connected</b>
<b>Bind to multiple controls</b>	<b>Bind to one control only</b>
<b>Forward and backward scanning of data</b>	<b>Forward-only</b>
<b>Slower access</b>	<b>Faster access</b>
<b>Supported by Visual Studio .NET tools</b>	<b>Manually coded</b>

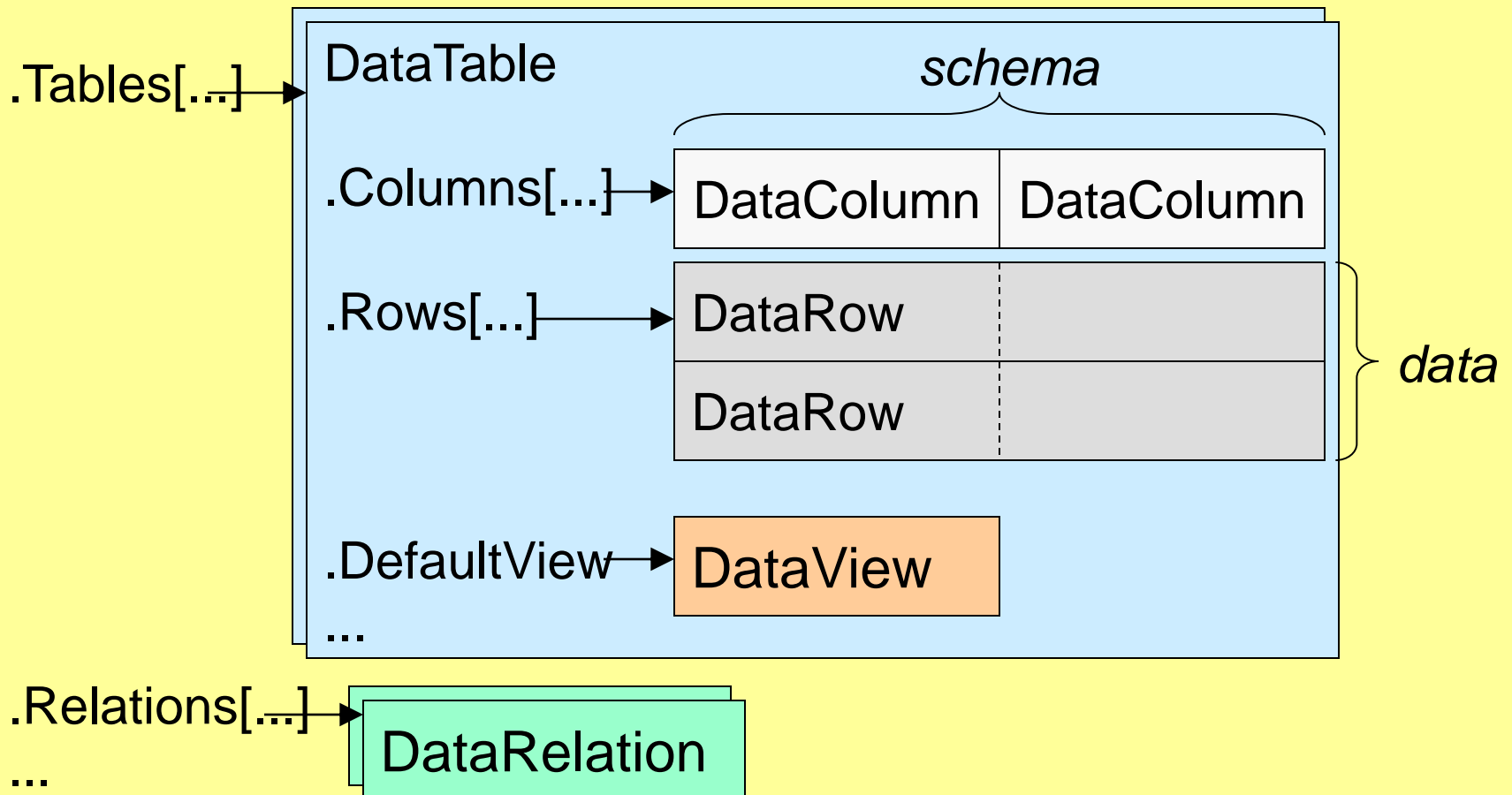
# DATASET

- Các thành phần
  - Tables
  - Relations



# DATASET

DataSet



# DATASET - DATATABLE

- DataTable: thể hiện một bảng trong CSDL

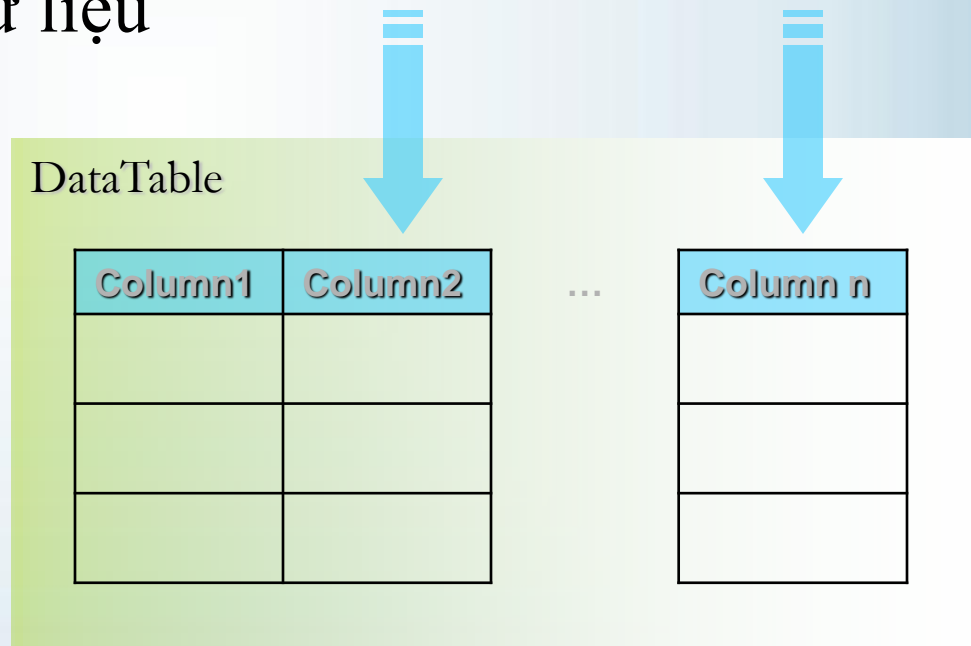
## Thuộc tính & Phương thức

- **TableName:** tên của bảng dữ liệu
- **Columns:** danh sách các cột
- **Rows:** danh sách các mẫu tin
- **PrimaryKey:** danh sách các cột là khóa chính
- **NewRow():** tạo một mẫu tin mới



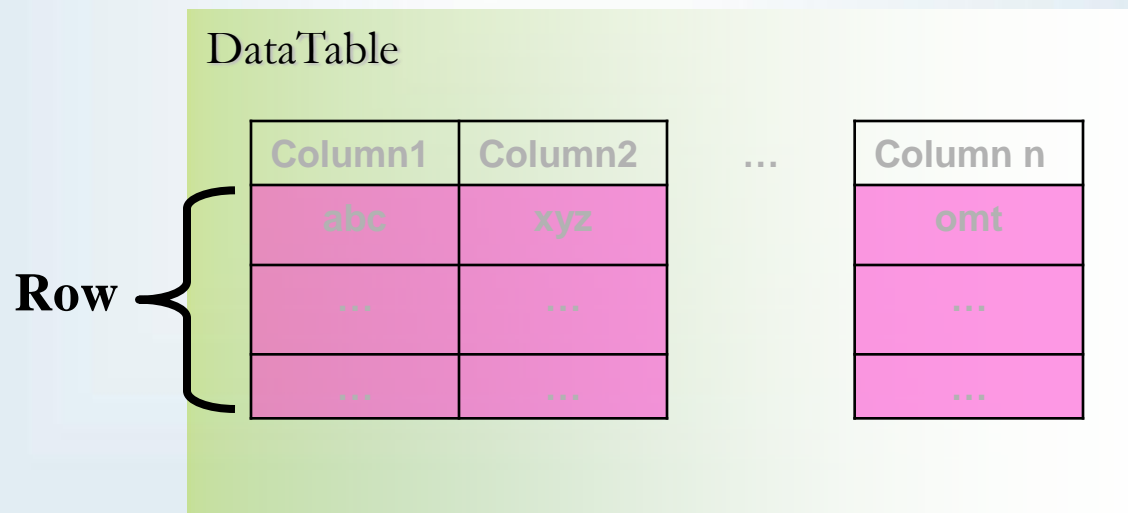
# DATASET - DATATABLE

- DataColumn: đại diện cho một cột trong bảng
  - ColumnName: tên cột
  - DataType: kiểu dữ liệu



# DATASET - DATAROW

- DataRow: đại diện cho mẫu tin trong bảng
  - RowState: trạng thái Added, Modified, Deleted,...
  - [i]: truy xuất đến cột i
  - Delete(): đánh dấu xóa mẫu tin



# DATASET

## DataSet – Update Row

```
string strConn = "data source=.\SQLEXPRESS; Initial  
Catalog=StudentDB; integrated security=yes;";  
string strCmd = "Select * From Sinhvien";  
SqlDataAdapter da = new SqlDataAdapter(strCmd, strConn);  
  
SqlCommandBuilder builder = new SqlCommandBuilder(da);  
DataSet ds = new DataSet();  
  
da.Fill(ds);  
  
foreach (DataRow dr in ds.Tables[0].Rows)  
    dr["Ngaysinh"] = DateTime.Now;  
  
da.Update(ds);
```

# DATASET

## DataSet – Delete Row

...

```
SqlDataAdapter da = new SqlDataAdapter(strCmd, strConn);  
SqlCommandBuilder builder = new SqlCommandBuilder(da);
```

```
DataSet ds = new DataSet();  
da.Fill(ds);
```

```
DataTable table = ds.Tables[0];
```

```
DataRow[] rows = table.Select("'Ngaysinh<'1/1/1980'");
```

```
foreach (DataRow r in rows)  
    r.Delete();  
da.Update(ds);
```

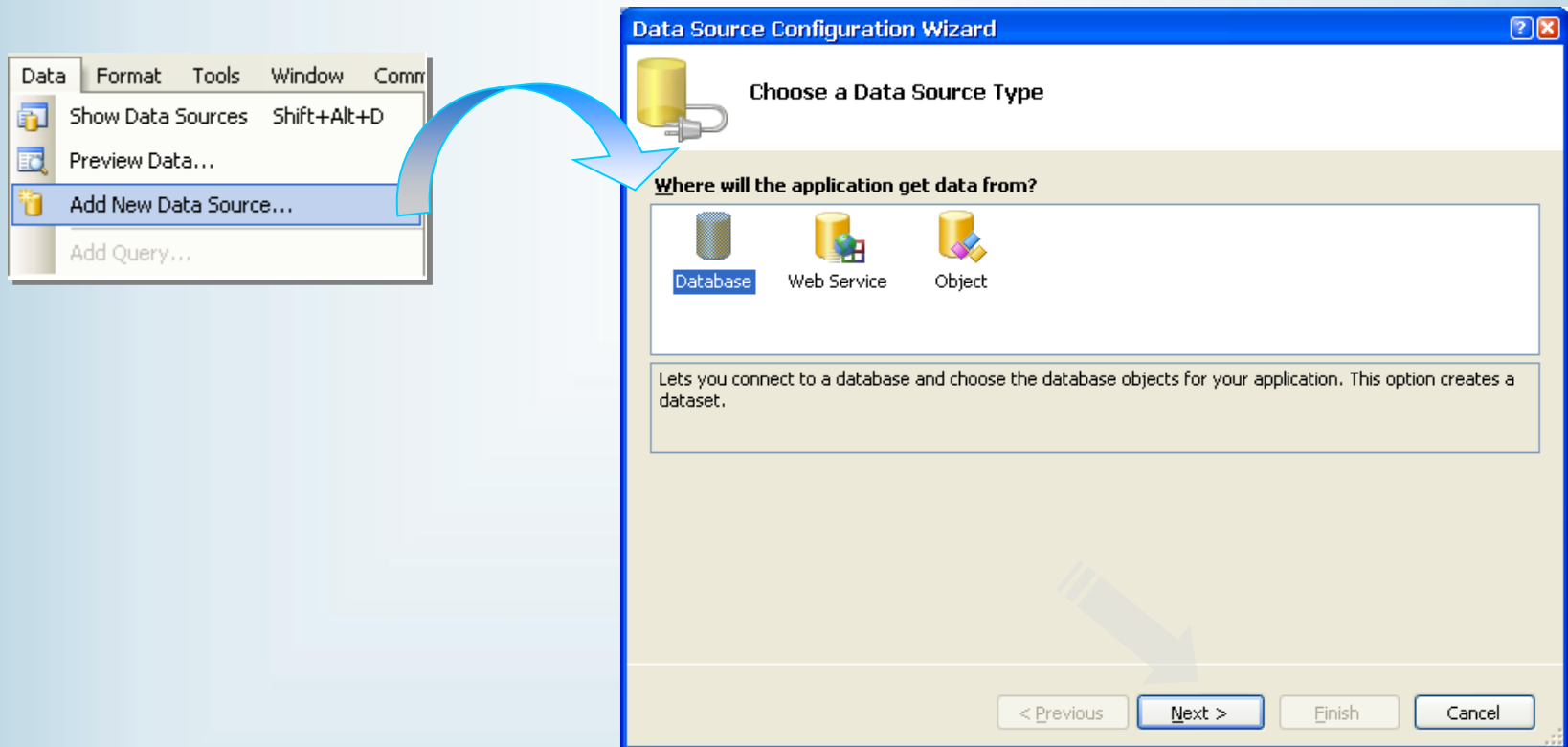


# DATA SOURCE CONFIGURATION WIZARD

- Trong VS.NET 2005 có chức năng Data Source Configuration Wizard
  - Nhanh chóng thiết lập Data Source cho project
  - Xây dựng form hiển thị và thao tác dữ liệu
    - ✓ Thông qua thao tác kéo thả từ Data Source

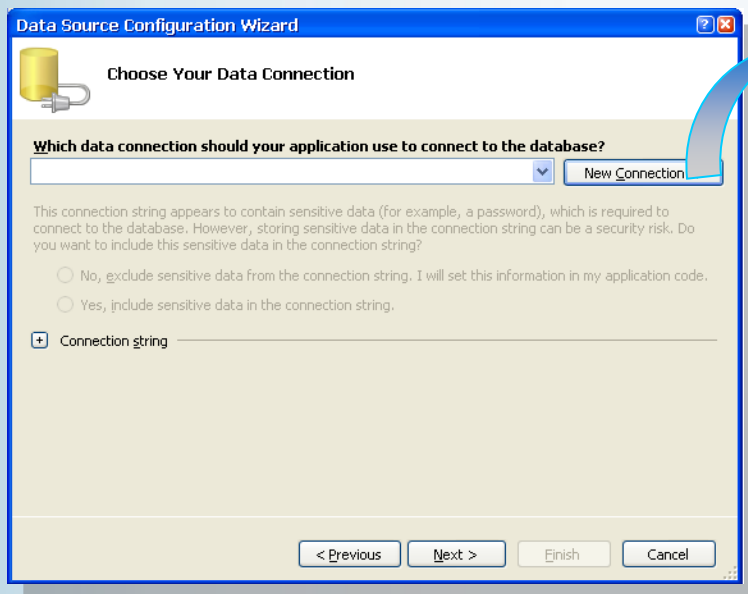
# DATA SOURCE CONFIGURATION WIZARD

- Tạo một ứng dụng Windows Application
- Trong menu Data | Add New Data Source...



# DATA SOURCE CONFIGURATION WIZARD

## NEW CONNECTION



**Data Source Configuration Wizard**

**Choose Your Data Connection**

Which data connection should your application use to connect to the database?

**New Connection**

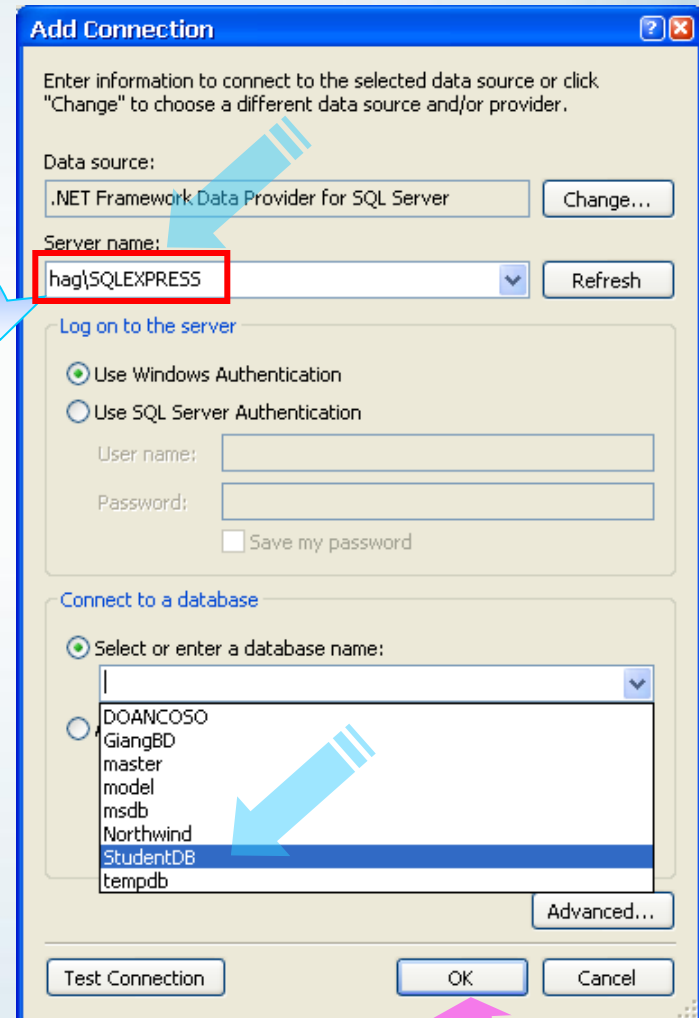
This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set this information in my application code.

☐ Yes, include sensitive data in the connection string.

☒ Connection string

**< Previous   Next >   Finish   Cancel**



**Add Connection**

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:  
 **Change...**

Server name:  
 **Refresh**

**Log on to the server**

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

☐ Save my password

**Connect to a database**

☒ Select or enter a database name:

☐ DOANCOSO

☐ GiangBD

☐ master

☐ model

☐ msdb

☐ Northwind

☒ StudentDB

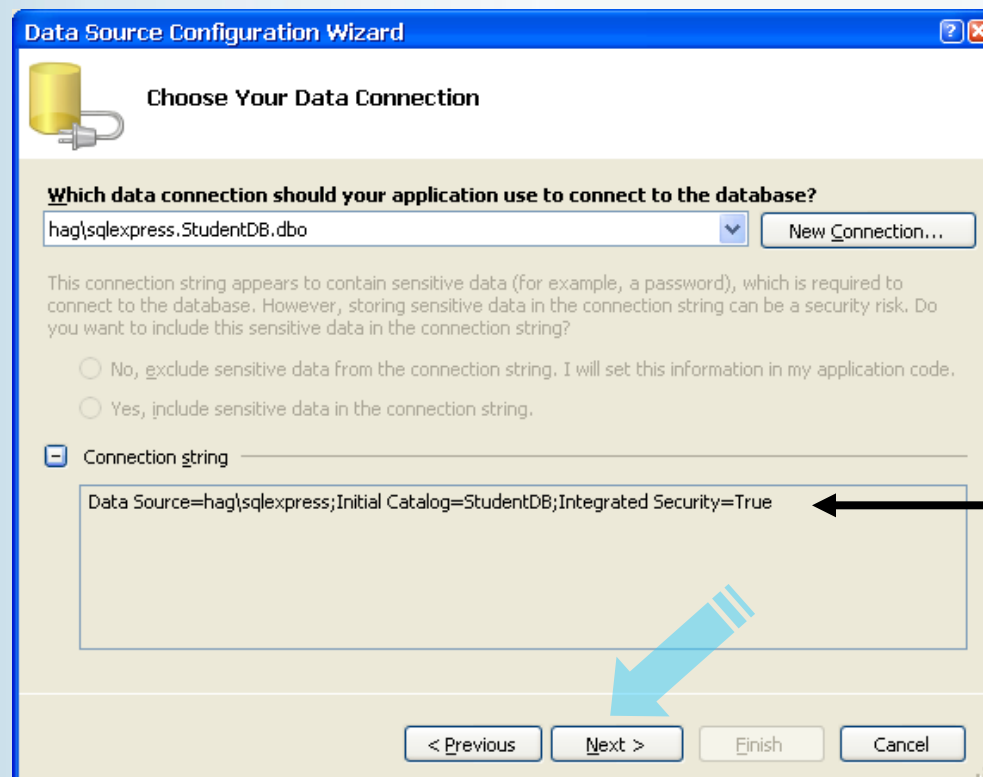
☐ tempdb

**Advanced...**

**Test Connection   OK   Cancel**

# DATA SOURCE CONFIGURATION WIZARD

- Hoàn tất khai báo Data Source



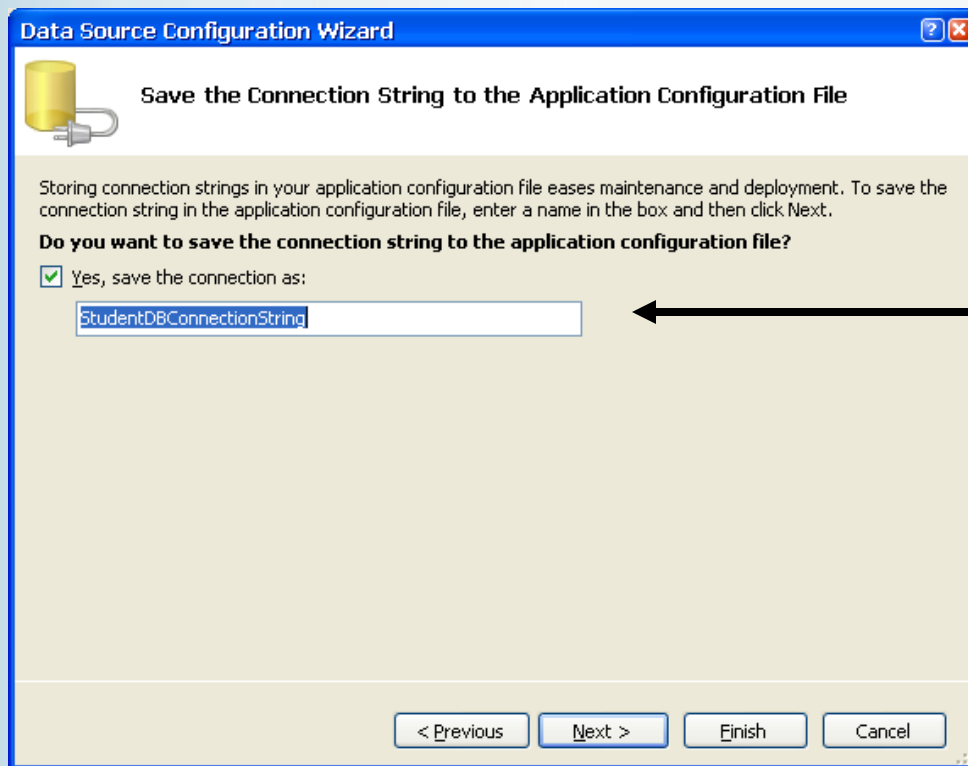
The screenshot shows the 'Data Source Configuration Wizard' window. The title bar reads 'Data Source Configuration Wizard'. Below the title bar is a yellow cylinder icon and the text 'Choose Your Data Connection'. The main area has a question: 'Which data connection should your application use to connect to the database?'. Below this is a dropdown menu showing 'hag\sqlexpress.StudentDB.dbo' and a 'New Connection...' button. A paragraph of text explains that the connection string might contain sensitive data like a password and asks if it should be included. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set this information in my application code.' (selected) and 'Yes, include sensitive data in the connection string.' Below the radio buttons is a checkbox labeled 'Connection string' which is checked. Underneath the checkbox is a text box containing the connection string: 'Data Source=hag\sqlexpress;Initial Catalog=StudentDB;Integrated Security=True'. At the bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'. A large blue arrow points to the 'Next >' button.

**Chuỗi kết nối**



# DATA SOURCE CONFIGURATION WIZARD

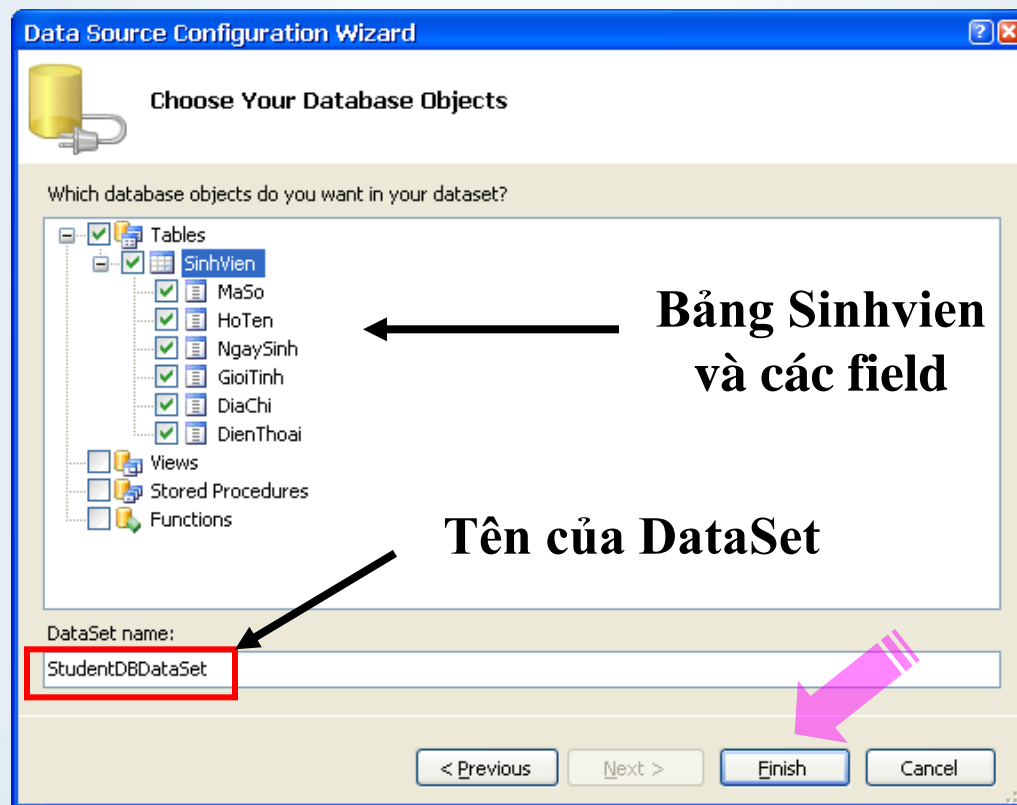
- Lưu chuỗi kết nối trong file cấu hình



Tên của chuỗi kết nối

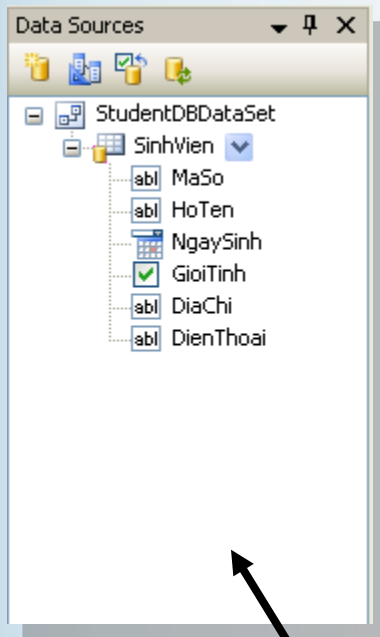
# DATA SOURCE CONFIGURATION WIZARD

- Chọn bảng dữ liệu

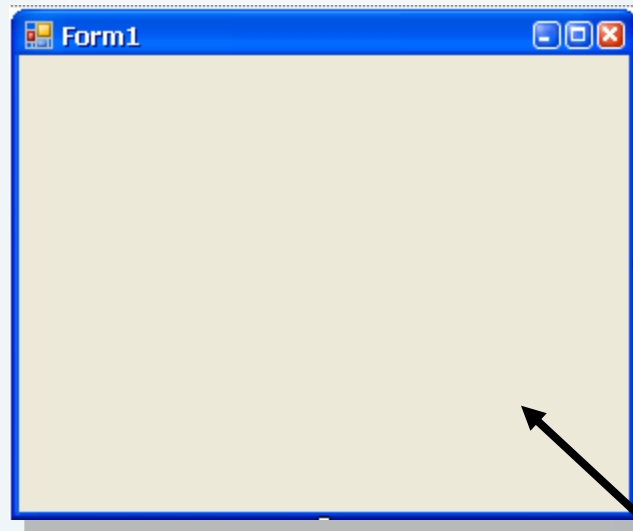


# DATA SOURCE CONFIGURATION WIZARD

- Wizard sẽ tạo ứng dụng với Data Source



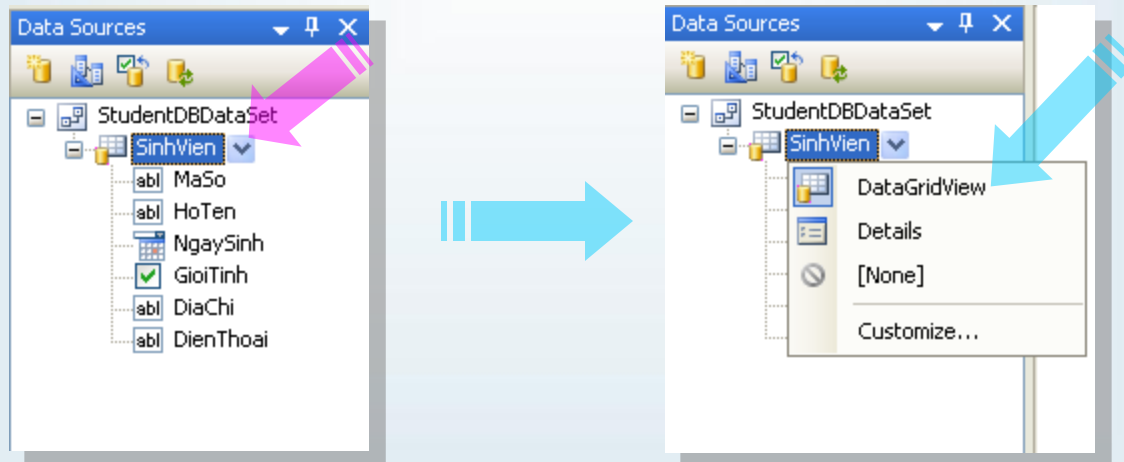
**Data Sources Windows**



**Form in Design View**

# DATA SOURCE CONFIGURATION WIZARD

- Kéo thả binding control vào Form
  - Trong cửa sổ Data Source
    - ✓ Chọn bảng cần sử dụng
      - Thiết lập view là DataGridView hay Details
        - ❖ Kích chọn vào dấu mũi tên xuống sau tên bảng

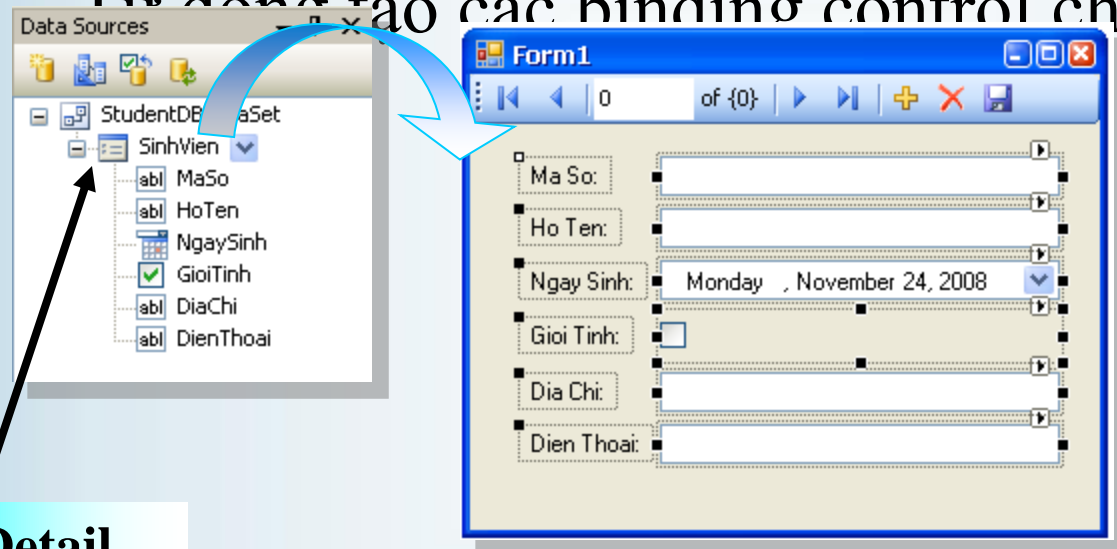




# Data Source Configuration Wizard

- Kéo Table thả vào Form

Tự động tạo các binding control cho table

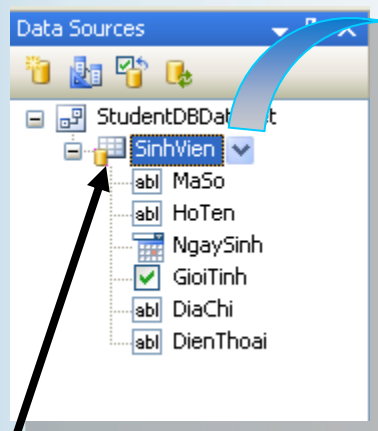


**Detail**



# DATA SOURCE CONFIGURATION WIZARD

- Bổ sung DataGridView cho Form
  - Thay đổi Table sang DataGrid
  - Kéo Table trong Data Source thả vào Form



**DataGridView**

The form 'Form1' contains several input fields at the top: 'Ma So:', 'Ho Ten:', 'Ngay Sinh:' (with a date picker set to 'Monday, November 24, 2008'), 'Gioi Tinh:' (with a checkbox), 'Dia Chi:', and 'Dien Thoai:'. Below these fields is a DataGridView. The DataGridView has a header row with columns: 'MaSo', 'HoTen', 'NgaySinh', 'GioiTinh', 'DiaChi', and 'DienThoai'. The first row of data contains an asterisk (\*) in the 'MaSo' column. The rest of the grid is empty.

studentDBDataSet      sinhVienTableAdapter  
sinhVienBindingSource      sinhVienBindingNavigator

# DATA SOURCE CONFIGURATION WIZARD

- Kết quả ứng dụng

**Form1**

1 of 4

Ma So:  Gioi Tinh: ☒

Ho Ten:  Dia Chi:

Ngay Sinh:  Dien Thoai:

	MaSo	HoTen	NgaySinh	GioiTinh	DiaChi	DienThoai
▶	1	Nguyen Ha Giang	12/4/1978	<input checked="" type="checkbox"/>	Hoang Hoa Tham	5120391
	2	Nguyen Ha Nam	1/1/2010	<input checked="" type="checkbox"/>	Le Van Sy	
	3	Nguyen Ha MyTien	2/10/1999	<input type="checkbox"/>	Hoa Hao	9318146
	4	Nguyen Ha Thanh Tung	8/26/2000	<input checked="" type="checkbox"/>	Le Van Sy	
*				<input type="checkbox"/>		

Chưa viết code!

**THANK YOU**