



ADO.NET

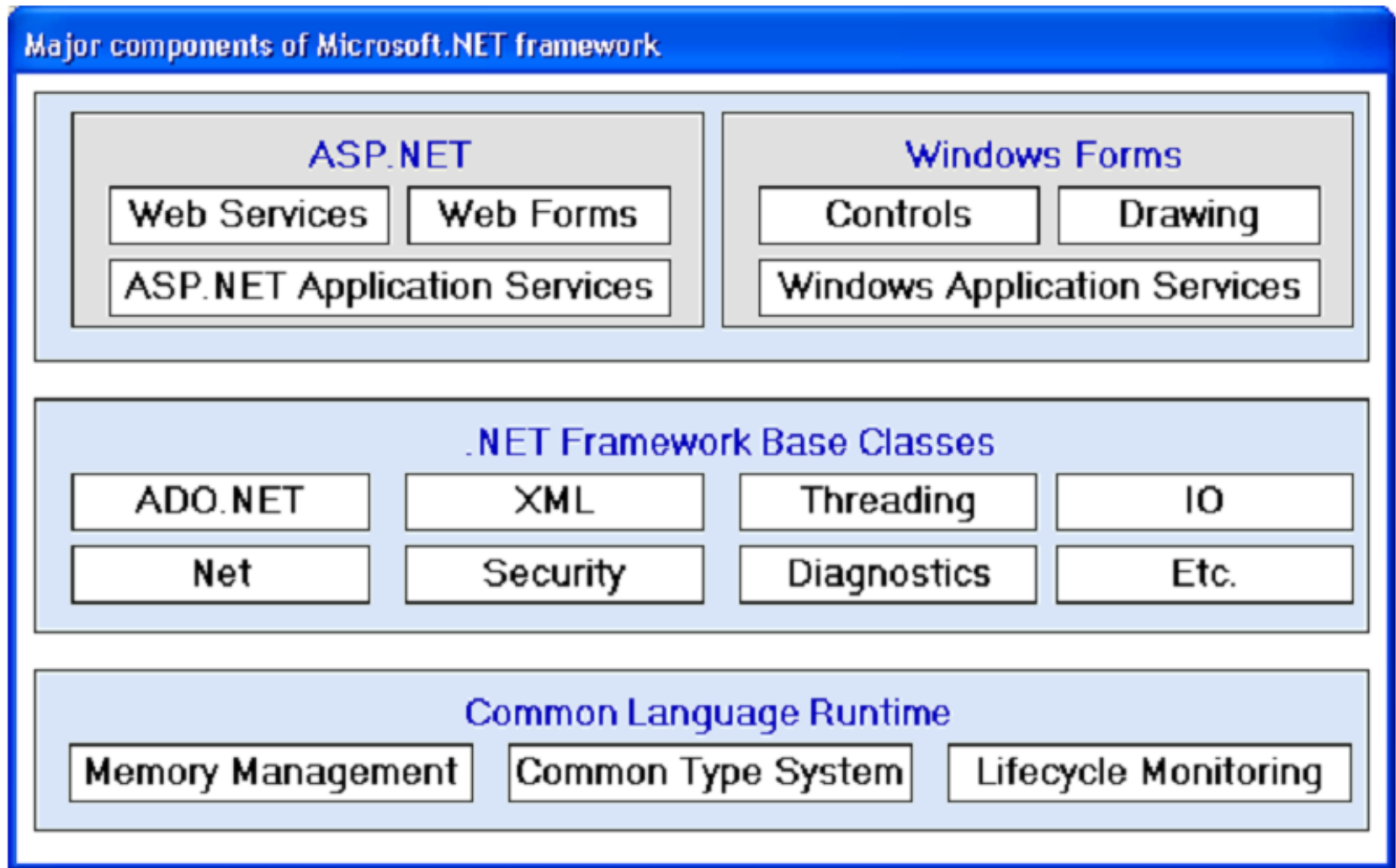
Giảng viên: BÙI NGỌC LÊ

Giới thiệu ADO.NET

ADO.NET là ***một tập các lớp*** nằm trong bộ thư viện lớp cơ sở của .NET Framework, cho phép các ứng dụng Windows (như C#, VB.NET...) hay ứng dụng web (như ASP.NET) thao tác dễ dàng với các nguồn dữ liệu



Vị trí của ADO.NET



Giới thiệu ADO.NET

ADO.NET bao gồm 2 Provider (2 bộ thư viện) (thường dùng) để thao tác với các CSDL là

- OleDb Provider (nằm trong System.Data.OleDb) dùng để truy xuất đến bất kỳ CSDL nào có hỗ trợ OleDb
- SQL Provider (nằm trong System.Data.SqlClient) chuyên dùng để truy xuất đến CSDL SQL Server



Giới thiệu ADO.NET

ADO.NET có 5 thành phần chính:

- OleDbConnection
- OleDbCommand
- OleDbDataReader
- OleDbDataAdapter
- DataSet



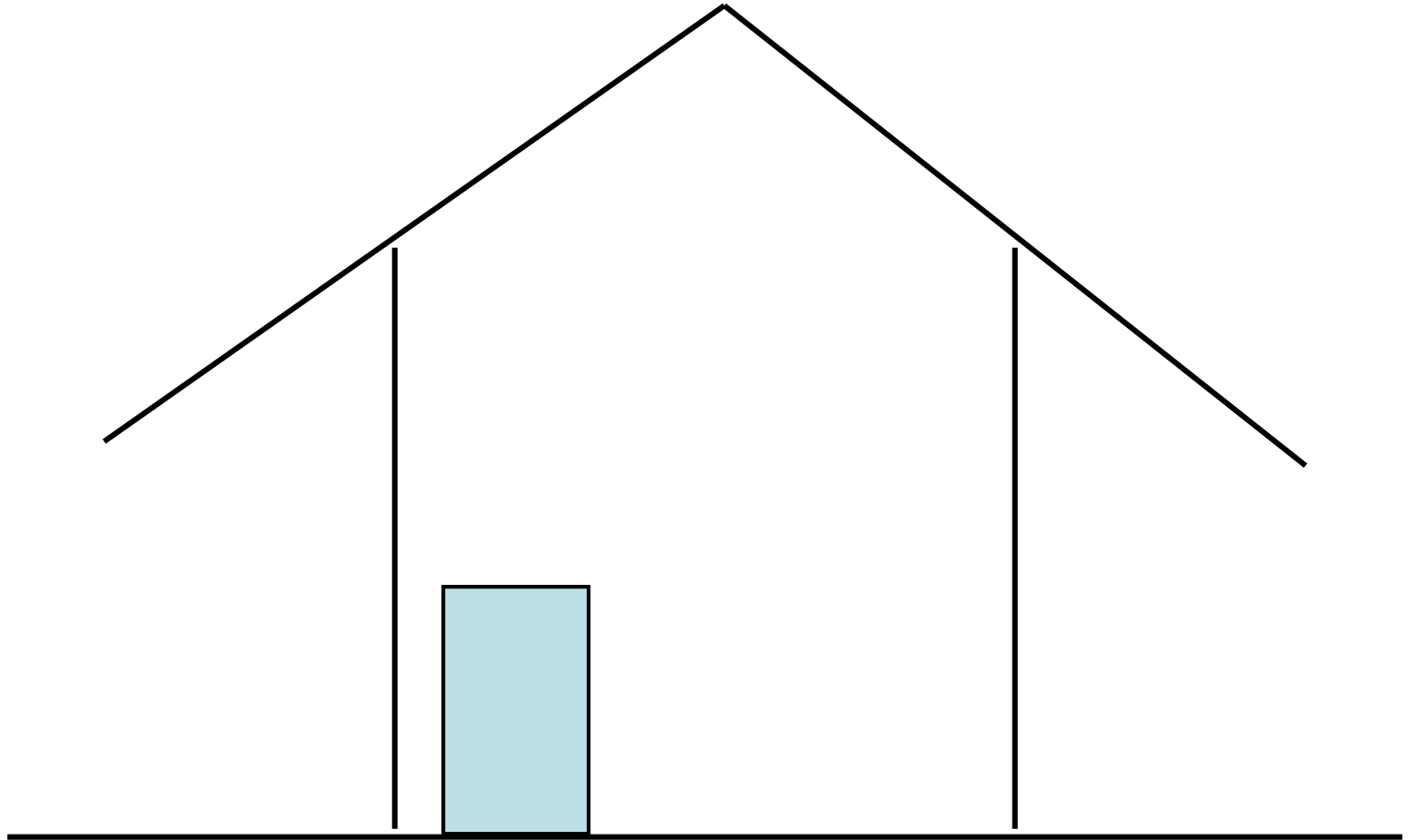
Giới thiệu ADO.NET

ADO.NET thao tác với CSDL theo hai mô hình:

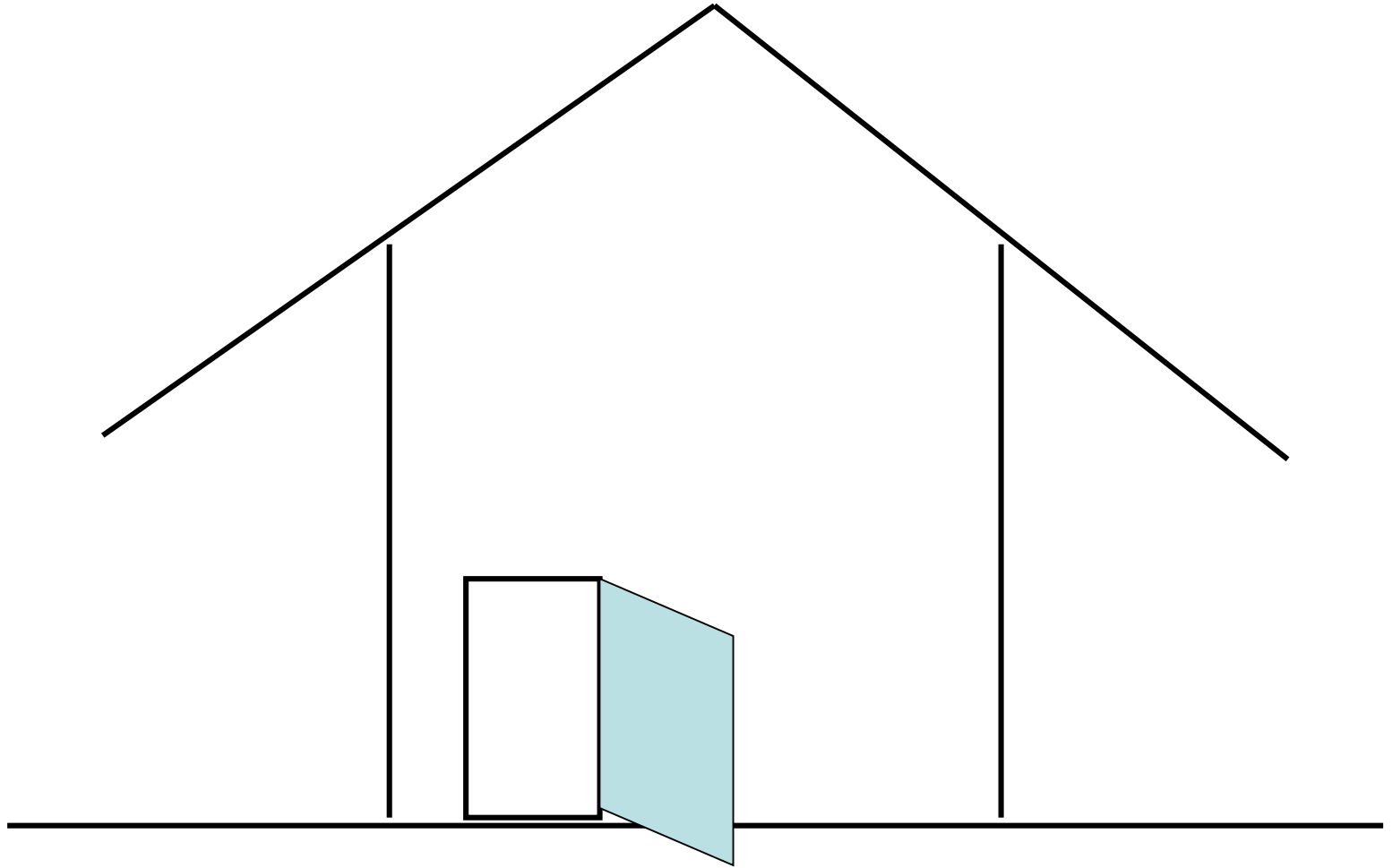
- Kết nối (Connected data).
- Phi kết nối (Disconnected data)



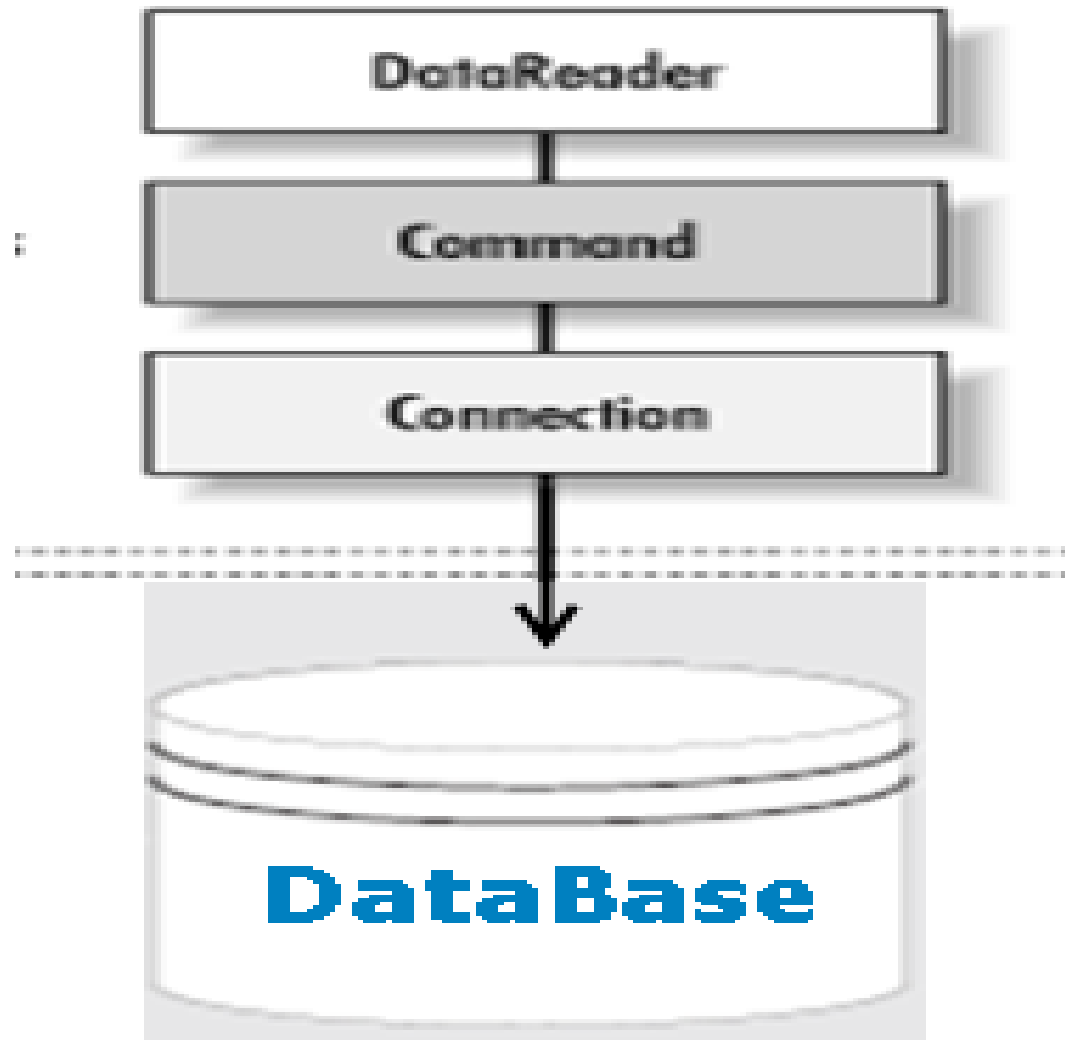
Mô hình kết nối



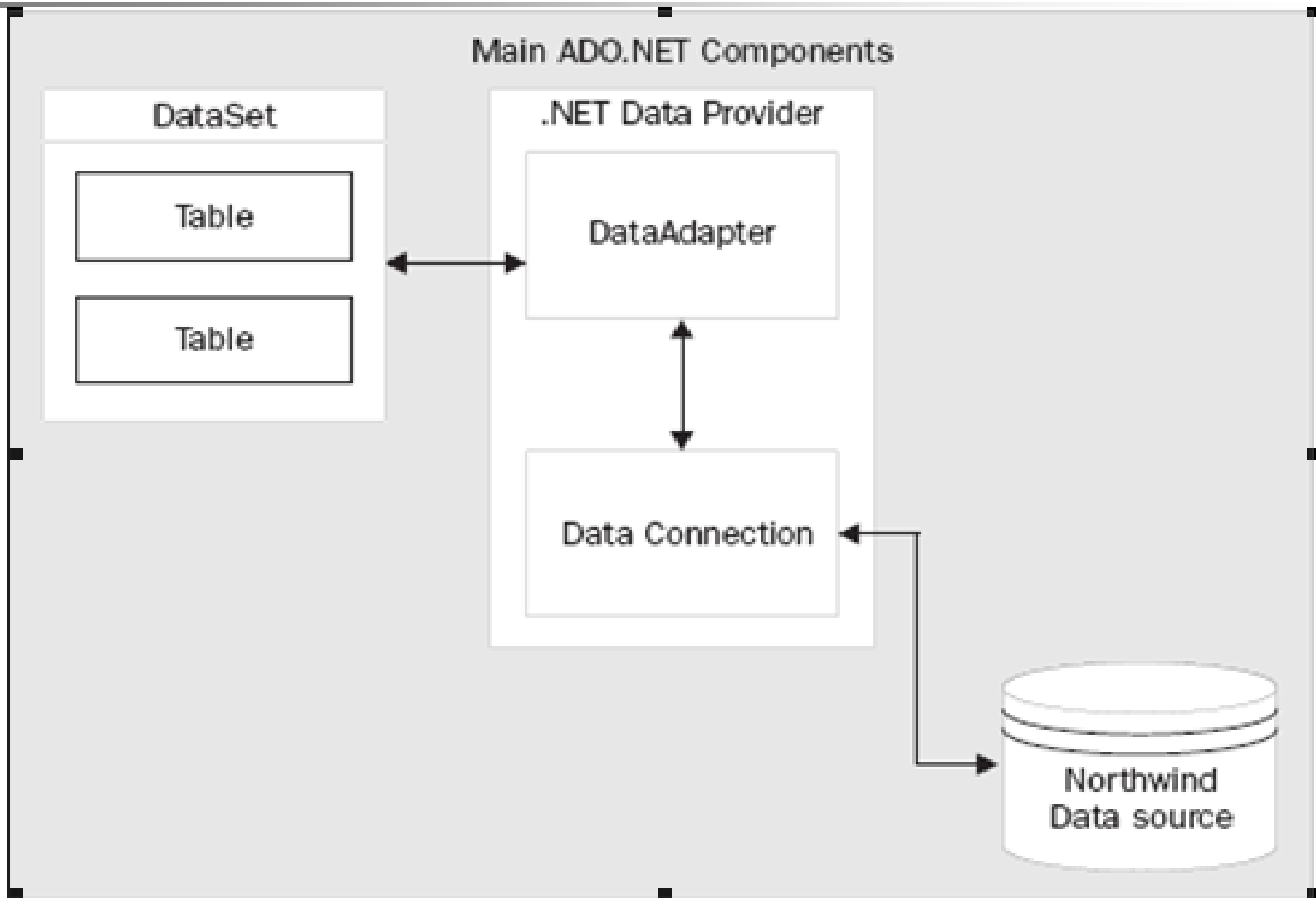
Mô hình kết nối



Mô hình kết nối



Mô hình phi kết nối



Đối tượng Connection

1. Chức năng:

- Đối tượng Connection có nhiệm vụ thực hiện kết nối đến Cơ sở dữ liệu để các đối tượng như Command thao tác với CSDL thông qua Connection này

2. Phương thức

- Open(): Mở kết nối tới CSDL
- Close(): Đóng kết nối



Đối tượng Connection

3. Khai báo đối tượng Connection

→ 1 vài cách

VD: `OleDbConnection con = new OleDbConnection(connectstring);`
`SqlConnection con = new SqlConnection(connectstring);`

4. Khai báo chuỗi kết nối:

- Cách 1:
- Cách 2:
- Cách 3:



Đối tượng Connection

Vd : Kết nối tới CSDL Access:

Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\mydatabase.mdb;User Id=admin;Password=;

Vd: Kết nối tới CSDL SQL Server:

Data Source=myServerAddress;Initial Catalog=myDataBase;User
Id=myUsername;Password=myPassword;

Vd: Kết nối tới **“CSDL Excel”**:

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\MyExcel.xls;
Extended Properties="Excel 8.0;HDR=Yes; IMEX=1";



Đối tượng Connection

Làm việc với CSDL Excel

1. Chuỗi kết nối tới CSDL Excel

```
string strConnectExcel = "Provider=Microsoft.Jet.OLEDB.4.0;  
Data Source=DataBaseExcel.xls;Extended Properties=Excel 8.0;";
```

2. Tạo bảng trong Excel



Đối tượng Command

1. Chức năng

- Dùng để thực thi các câu lệnh SQL thao tác với CSDL như : Insert, Update, Select, Delete...
- Trước khi thực thi câu lệnh SQL bằng đối tượng Command thì bắt buộc phải **mở kết nối** tới CSDL

2. Khai báo đối tượng Command

```
OleDbCommand com = new OleDbCommand(sqlquery, oledbconnection);
```

```
SqlCommand com = new SqlCommand(sqlquery, sqlconnection);
```



Đối tượng Command

3. Các phương thức của đối tượng Command

- **ExecuteScalar():** Thực hiện câu lệnh mà kết quả trả về chỉ có 1 ô (Ví dụ câu lệnh `Select Count(*)...`).
- **ExecuteReader():** Thực hiện câu lệnh `Select` và trả về một `DataReader`
- **ExecuteNonQuery():** Thực hiện câu lệnh SQL (`Delete`, `Update`, `Insert ...`).
- **ExecuteXmlReader():** Tạo một bộ đọc từ file XML. Phương thức này không có trong `OleDbCommand`, chỉ có trong `SqlCommand`.



Đối tượng Command

4. Xử lý gởi tham số vào trong đt Command

- VD: Tạo đt cập nhật tt SV

```
SqlCommand command = new SqlCommand("UPDATE SINHVIEN  
SET LOP = @LOP, TENSX = @TENSX " +  
"WHERE MASX = @MASX", con);
```

```
command.Parameters.Add("@LOP", SqlDbType.NVarChar, 50,  
"LOP");
```

```
command.Parameters.Add("@TENSX", SqlDbType.NVarChar, 200,  
"TENSX");
```

```
command.Parameters.Add("@MASX", SqlDbType.NVarChar, 50,  
"MASX");
```



Đối tượng DataReader

1. Chức năng:

- Dùng để đón nhận kết quả (dữ liệu) trả về từ phương thức `ExecuteReader` của đối tượng `Command`, dữ liệu là `Readonly` và chỉ đọc theo chiều tiến.

2. Khai báo

- Khai báo: `OleDbDataReader Biến_DataReader;` //Không có `New`
- **Lấy kết quả từ Command**
`<Biến_DataReader> = <Biến_Command>.ExecuteReader();`



Đối tượng DataAdapter

1. Chức năng:

- Có chức năng như một chiếc cầu nối giữa nguồn (tệp) dữ liệu và các bảng được cached trong bộ nhớ (đối tượng DataSet.). DataAdapter sử dụng phương thức **Fill()** để nhận dữ liệu từ nguồn dữ liệu vào một dataset.

2. Sử dụng

BiếnAdapter.Fill(<Biến dataset>, “Tên cho bảng”)



Đối tượng DataAdapter

3. Sử dụng Data Adapter để đọc và ghi dữ liệu

- SelectCommand: đối tượng Command để truy xuất dữ liệu
- InsertCommand: đối tượng Command để insert dữ liệu
- UpdateCommand: đối tượng Command để update dữ liệu
- DeleteCommand: đối tượng Command để deletedữ liệu



Đối tượng DataSet

1. Chức năng:

- DataSet là một lớp dùng để thao tác với dữ liệu theo mô hình dữ liệu phi kết nối. Nó được coi như là một kho chứa các bảng (table). Người dùng có thể thay đổi dữ liệu trong các bảng này và khi thực sự muốn cập nhật vào Cơ sở dữ liệu thì DataSet sẽ thực hiện cập nhật thông qua lớp DataAdapter
- Các bảng trong DataSet có thể do DataAdapter Fill vào hoặc cũng có thể là các bảng được tạo thành từ lớp DataTable

2. Khai báo

Adapt.Fill(<Biến dataset>, “Tên cho bảng”)



Đối tượng DataSet

