

LẬP TRÌNH HƯỚNG SỰ KIỆN

Giảng viên: ThS. Phan Thanh Toàn

BÀI 7

LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET

Giảng viên: ThS. Phan Thanh Toàn

MỤC TIÊU BÀI HỌC

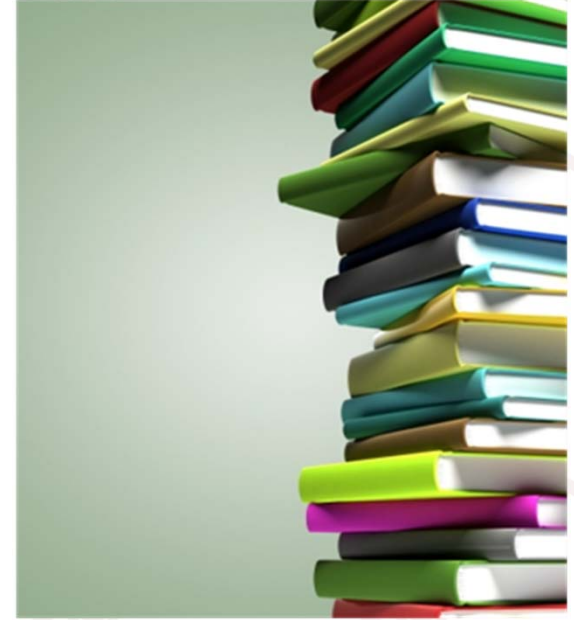
- Trình bày được các đặc trưng cơ bản của ADO.NET.
- Liệt kê được các thành phần cơ bản của ADO.NET.
- Liệt kê được các đối tượng thao tác cơ sở dữ liệu (CSDL) cơ bản trong ADO.NET.
- Vận dụng được ngôn ngữ lập trình C# và các đối tượng ADO.NET để xây dựng các ứng dụng thao tác với CSDL.



CÁC KIẾN THỨC CẦN CÓ

Để học được môn học này, sinh viên phải học xong các môn học:

- Cơ sở lập trình;
- Lập trình hướng đối tượng;
- Cơ sở dữ liệu;
- Hệ quản trị cơ sở dữ liệu SQL Server.



HƯỚNG DẪN HỌC

- Đọc tài liệu tham khảo.
- Thảo luận với giáo viên và các sinh viên khác về những vấn đề chưa hiểu rõ.
- Trả lời các câu hỏi của bài học.



7.1

Tổng quan về ADO.NET

7.2

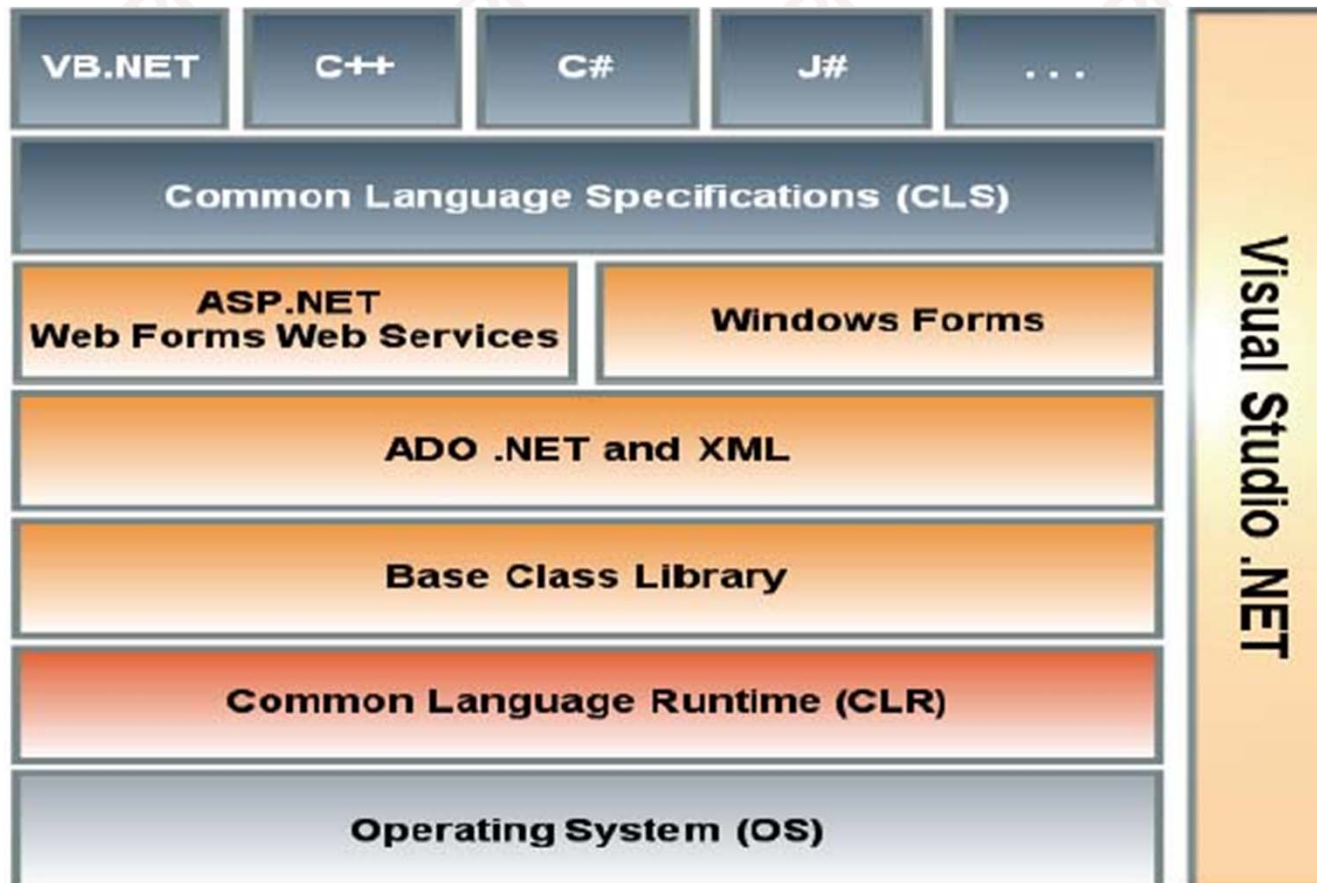
Các thành phần cơ bản của ADO.NET

7.3

Lập trình cơ sở dữ liệu với ADO.NET trong C#

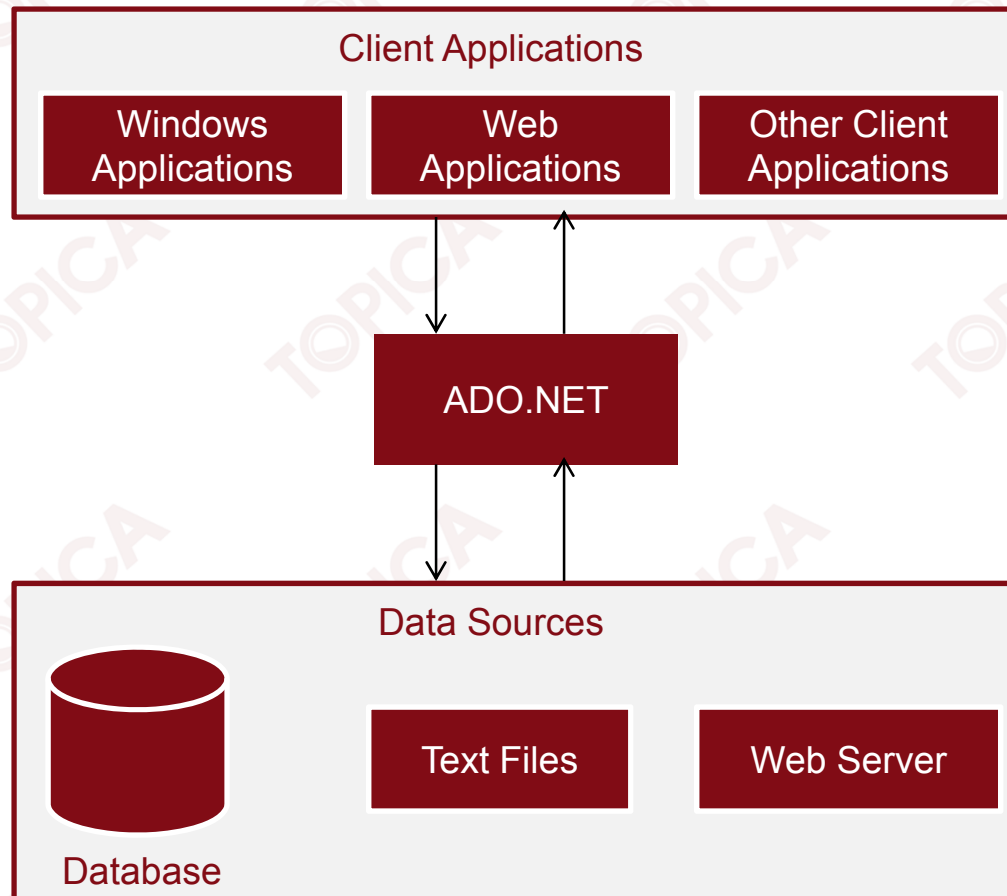
7.1. TỔNG QUAN VỀ ADO.NET

- ADO.NET (ActiveX Data Objects.NET) là một thành phần trong .NET FRAMEWORK đảm nhận vai trò thao tác với CSDL.
- Cung cấp các lớp đối tượng và hàm thư viện phục vụ kết nối và xử lý dữ liệu.



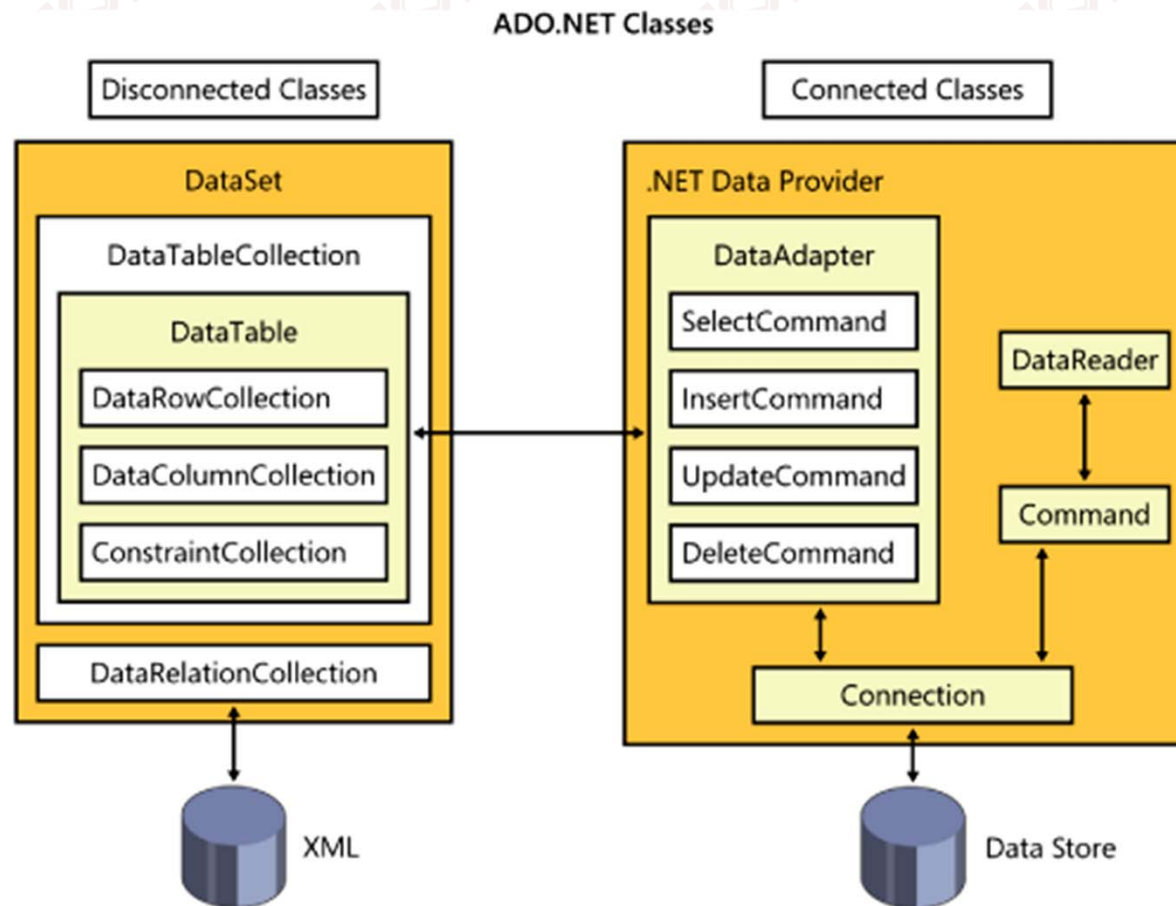
7.1. TỔNG QUAN VỀ ADO.NET (tiếp theo)

- ADO.NET là cầu nối giữa ứng dụng và CSDL.
- ADO.NET hỗ trợ việc kết nối và truy cập CSDL đối với nhiều hệ quản trị CSDL khác nhau như: MS SQL Server, Oracle, Access...



7.1. TỔNG QUAN VỀ ADO.NET (tiếp theo)

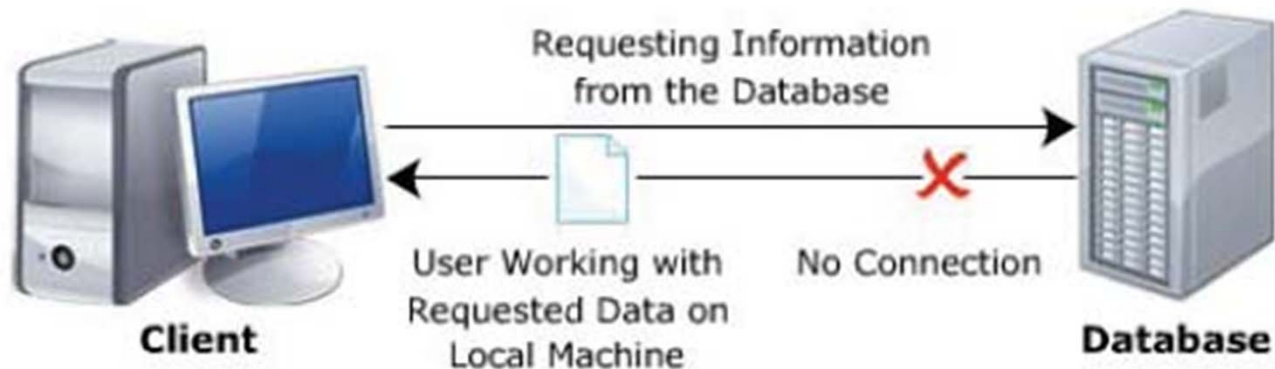
- ADO.NET hoạt động theo 2 kiến trúc kết nối và ngắt kết nối.
- Mô hình ngắt kết nối: Cho phép lấy cả một cấu trúc dữ liệu phức tạp từ CSDL sau đó ngắt kết nối với CSDL và thực hiện xử lý dữ liệu.
- Mô hình kết nối: Luôn phải duy trì kết nối trong suốt quá trình xử lý dữ liệu.



7.1. TỔNG QUAN VỀ ADO.NET (tiếp theo)

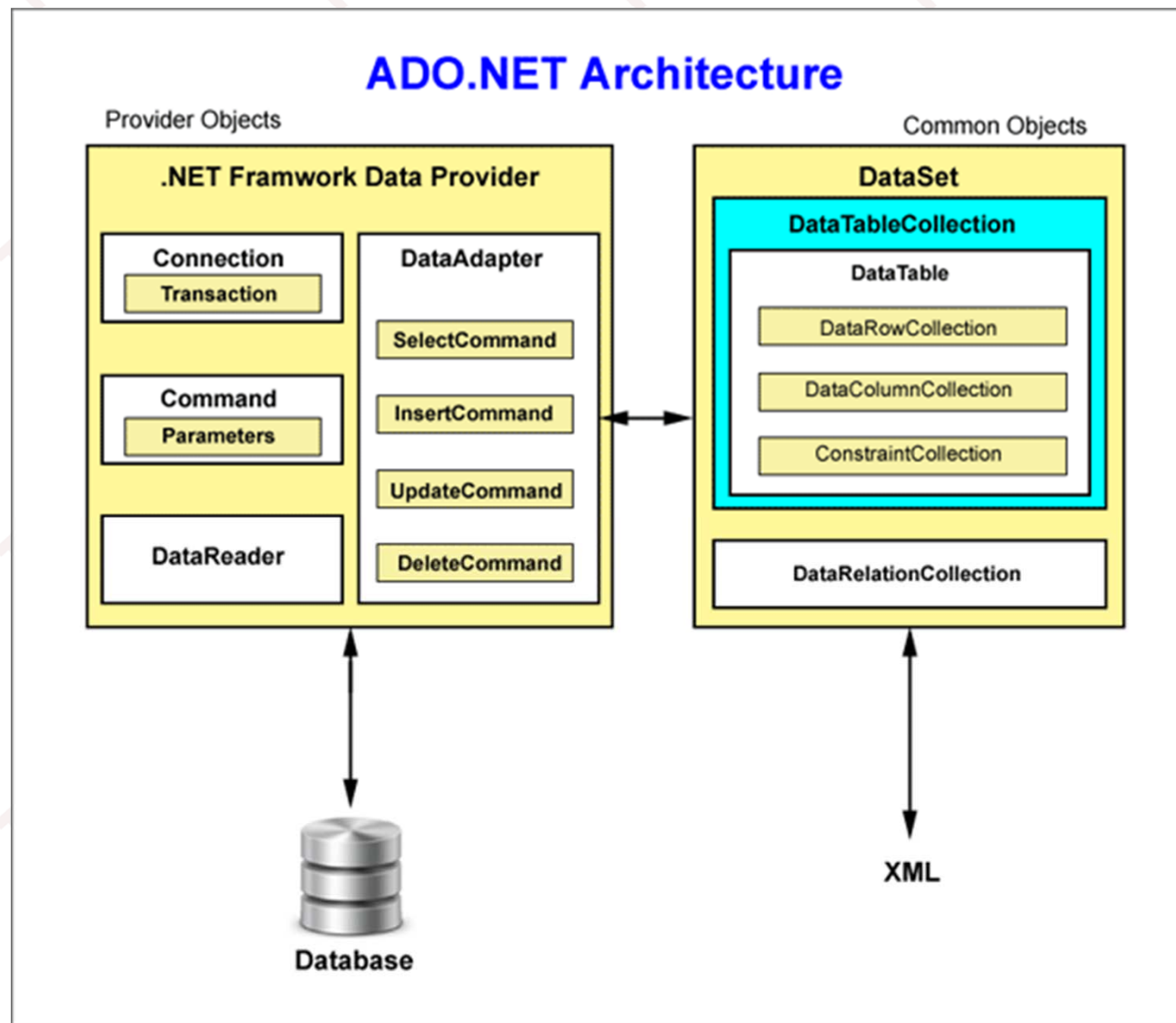
Các đặc trưng của ADO.NET:

- Hỗ trợ lập trình:
 - Cung cấp các lớp thao tác với CSDL giúp lập trình nhanh hơn và giảm lỗi;
 - Cung cấp các công cụ để thao tác với CSDL ngay trên phần Designer giúp thao tác với CSDL dễ dàng hơn.
- Khả năng mở rộng: Sử dụng kiến trúc ngắt kết nối giúp giảm tải cho server, hỗ trợ nhiều người sử dụng truy cập CSDL đồng thời tốt hơn.
- Khả năng tích hợp:
 - ADO.NET có thể gửi dữ liệu cho bất cứ loại ứng dụng nào;
 - Hỗ trợ XML.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET

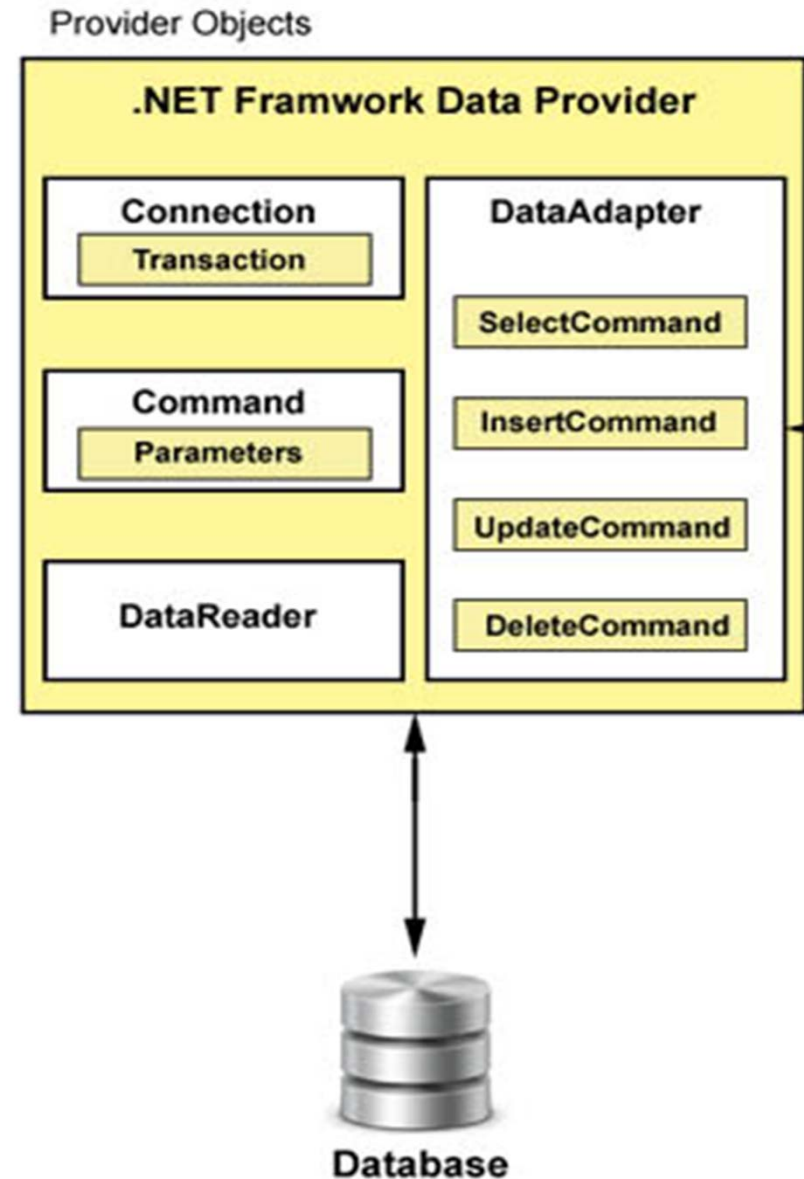
- Kiến trúc ADO.NET gồm 2 thành phần chính:



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

➤ Phần kết nối: Sử dụng khi kết nối CSDL và thực hiện các thao tác xử lý dữ liệu. Bao gồm các thành phần chính sau:

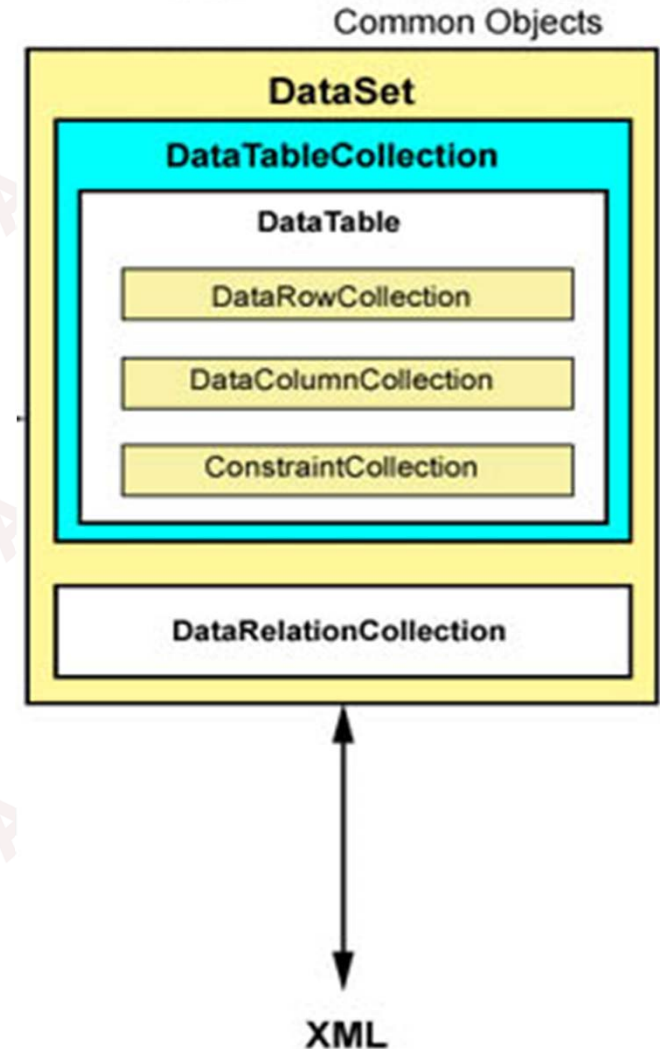
- Connection: quản lý việc mở, đóng CSDL;
- Command: lệnh truy vấn, tương tác dữ liệu khi đang lập kết nối;
- DataReader: đọc dữ liệu, chỉ xử lý một dòng dữ liệu tại một thời điểm;
- DataAdapter: Cầu nối giữa CSDL với DataSet.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- Phần ngắt kết nối DataSet:
 - DataSet không quan tâm đến kiểu của CSDL;
 - Lấy dữ liệu từ DataAdapter;
 - DataSet xem như một CSDL trong bộ nhớ;
 - DataSet gồm các thành phần con như: DataTable, DataRow, DataColumn, DataRelation...

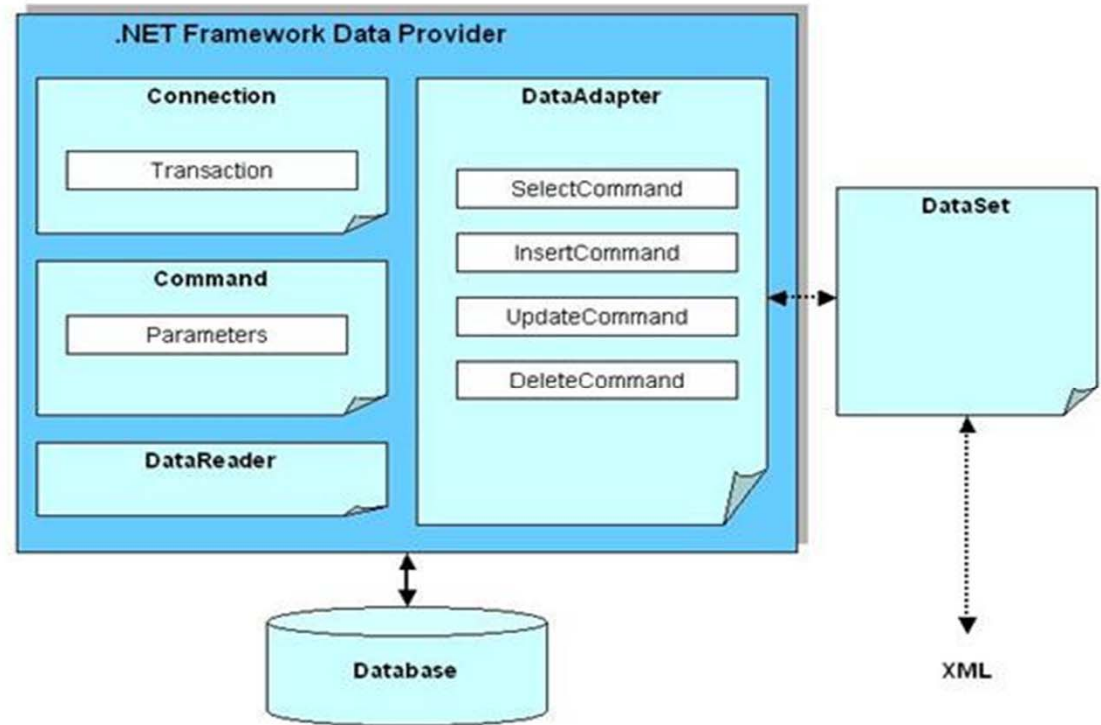
- ADO.NET có một số lớp thư viện:
 - System.Data.OleDb: Làm việc với CSDL Access, SQL Server, Oracle;
 - System.Data.SqlClient: Làm việc với CSDL SQL Server;
 - System.Data.OracleClient: Làm việc với CSDL Oracle.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- Mô hình .NET Data Provider:

- Connection;
- Command;
- DataReader;
- DataAdapter.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Đối tượng Connection

- Tạo kết nối với CSDL.
- Các thuộc tính và phương thức của Connection:
 - `ConnectionString`: Thuộc tính xâu kết nối với CSDL DataSource;
 - `Open()`: Phương thức mở kết nối với CSDL;
 - `Close()`: Phương thức đóng kết nối.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- Tạo kết nối tới CSDL SQL Server:

- Bước 1: Khai báo đối tượng SqlConnection

```
SqlConnection conn = null;
```

- Bước 2: Tạo chuỗi kết nối tới máy chủ SQL Server

```
string connectionString = "Data Source=Server; Initial  
Catalog=AdventureWorks; UID=sa; PassWord=' '";
```

- Bước 3: Tạo đối tượng Connection

```
conn = new SqlConnection(connectionString);
```

- Bước 4: Mở kết nối qua đối tượng Connection

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

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Đối tượng Command

- Xử lý dữ liệu qua các câu lệnh sql, với một kết nối Connection đang mở.
- Thuộc tính và phương thức của đối tượng Command:
 - Connection: Đối tượng Connection, xác định kết nối thực hiện lệnh;
 - CommandText: Câu lệnh sql 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, StoredProc trả về số hàng bị tác động (Insert, Update, Delete);
 - ExecuteReader(): Thực hiện lệnh và trả về DataReader.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Ví dụ: Sử dụng đối tượng Command để đếm số người trong bảng Person.Contact.

- Tạo đối tượng SqlConnection và thực hiện mở kết nối tới CSDL AdventureWorks.
- Tạo đối tượng Command và sử dụng phương thức ExecuteScalar để đếm số bản ghi trong bảng Person.Contact như sau:

```
SqlCommand command = new SqlCommand("", conn);  
command.CommandText = "Select COUNT(*) From Person.Contact";  
int TotalPerson =(int) command.ExecuteScalar();  
MessageBox.Show("Total: " + TotalPerson);
```


7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Đối tượng DataReader

- Được sử dụng để đọc dữ liệu từ CSDL:
 - Truy xuất dữ liệu tuần tự;
 - Chỉ đọc dữ liệu;
 - Cơ chế kết nối.
- Thuộc tính và phương thức của đối tượng DataReader:
 - HasRow: Thuộc tính cho biết truy vấn có trả về dữ liệu hay không;
 - Read(): Phương thức đọc một bản ghi dữ liệu;
 - [i]: Truy xuất đến trường thứ i trong bản ghi đọc được;
 - Close(): Phương thức đóng đối tượng.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Ví dụ: Chương trình sử dụng đối tượng DataReader đọc dữ liệu từ bảng Sales.Currency trong CSDL AdventureWorks và nạp mã CurrencyCode vào một ListBox

Bước 1: Khai báo đối tượng SqlConnection

```
public SqlConnection conn = null;
```

Bước 2: Viết thủ tục OpenConnection()

```
private void OpenConnection()
{
    string connectionString = "Data Source=Server; Initial
Catalog=AdventureWorks; UID=sa; PassWord=' '";
    conn = new SqlConnection(connectionString);
    conn.Open();
}
```

Bước 3: Gọi thủ tục OpenConnection() trong sự kiện Form_Load

```
private void Form2_Load(object sender, EventArgs e)
{
    OpenConnection();
}
```

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

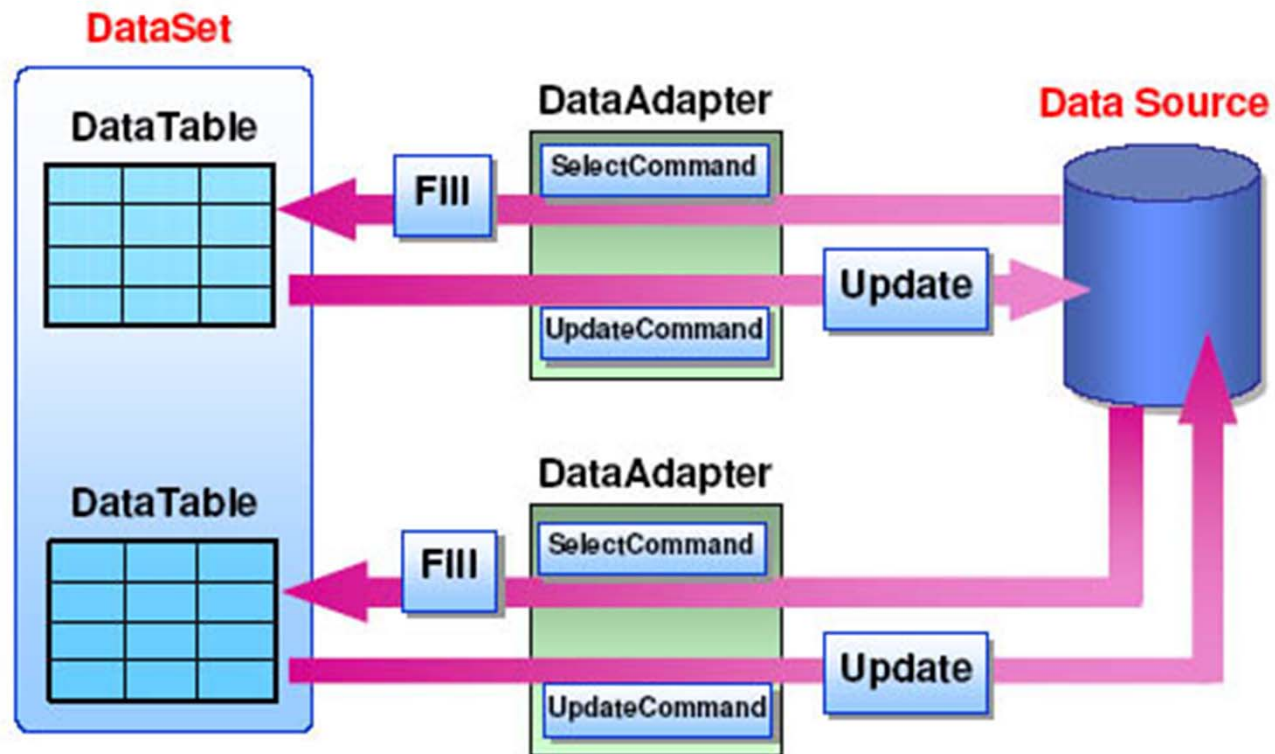
- Đọc dữ liệu qua DataReader và đưa mã CurrencyCode vào ListBox

```
private void LoadCurrentCy()  
{  
    string strSQL="Select * From Sales.Currency";  
    SqlCommand command = new SqlCommand(strSQL, conn);  
    SqlDataReader reader = command.ExecuteReader();  
    while (reader.Read())  
    {  
        lstCurrency.Items.Add(reader["CurrencyCode"]);  
    }  
    reader.Close();  
    command.Dispose();  
}
```

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Đối tượng DataAdapter

- Là cầu nối giữa CSDL và DataSet, thường sử dụng DataAdapter để truy cập vào các loại CSDL khác nhau như SQL Server, Oracle... Và sẽ đổ toàn bộ dữ liệu vào DataSet được lưu trên bộ nhớ, sau đó sẽ sử dụng dữ liệu trên DataSet.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- Thuộc tính và phương thức của DataAdapter:
 - Fill (DataSet): Sử dụng SelectCommand để lấy dữ liệu từ CSDL và đổ vào DataSet;
 - Update(DataSet): InsertCommand, UpdateCommand, DeleteCommand, cập nhật dữ liệu trong DataSet vào cơ sở dữ liệu qua DataAdapter.
- Chú ý: Các thao tác với DataSet là thao tác với dữ liệu trên bộ nhớ máy tính, chưa tác động đến dữ liệu trong CSDL, để thực hiện các thao tác cập nhật, xóa,... dữ liệu trên CSDL ta phải sử dụng qua đối tượng DataAdapter.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Ví dụ: Chương trình đọc dữ liệu từ bảng Sales.Customer qua DataAdapter và đổ vào DataSet

Bước 1: Viết thủ tục OpenConnection như phần 7.2.

Bước 2: Tạo DataAdapter và đọc dữ liệu từ CSDL sau đó đổ vào DataSet

```
//Lấy dữ liệu qua mệnh đề SQL
string strSQL = "Select * From Sales.Customer";

//Tạo DataAdapter và lấy dữ liệu từ bảng Sales.Customer
SqlDataAdapter adapter = new SqlDataAdapter(strSQL, conn);
OpenConnection();

DataSet ds = new DataSet();

//Đổ dữ liệu từ DataAdapter và DataSet
adapter.Fill(ds);
conn.Close();
```

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

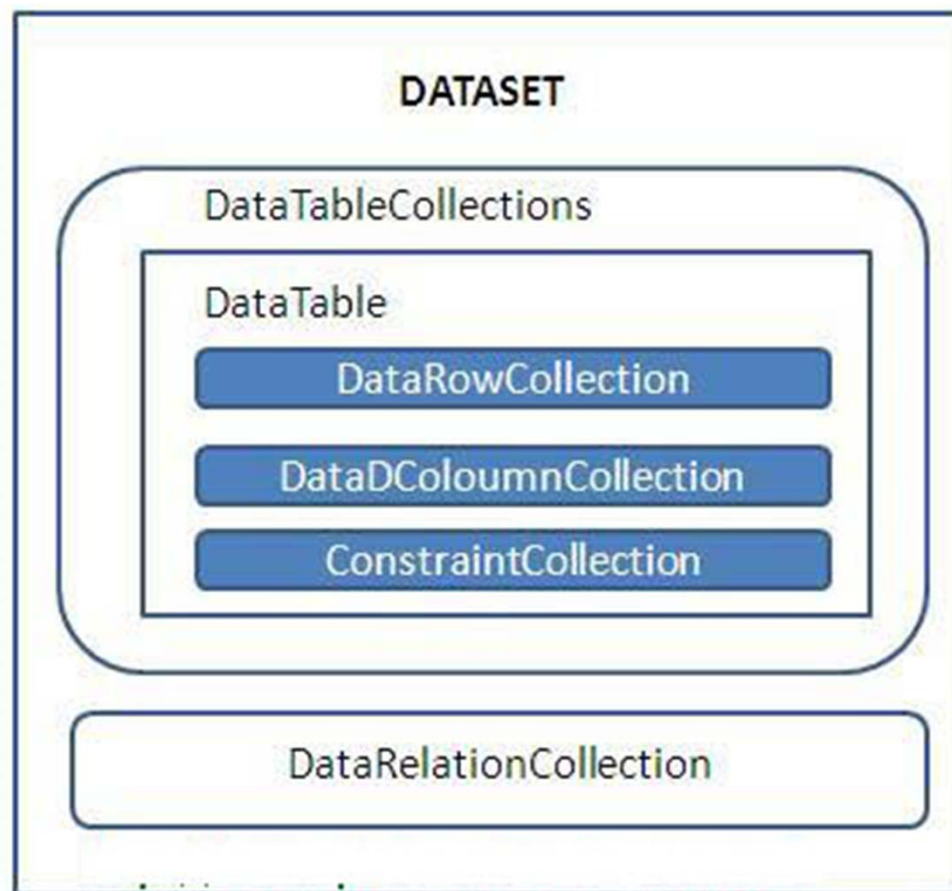
Đối tượng DataSet

- Là phần cơ sở dữ liệu được lưu trữ trong bộ nhớ:
 - Cơ chế ngắt kết nối;
 - Thao tác với CSDL qua cầu nối là DataAdapter;
 - Hỗ trợ đầy đủ đặc tính XML;
 - Thao tác được với tất cả các mô hình lưu trữ dữ liệu: Flat file database, Relational database, Hierarchical.
- DataSet và DataReader đều sử dụng để lưu trữ dữ liệu, sự khác nhau giữa DataSet và DataReader:

DataSet	DataReader
Có thể đọc/ghi dữ liệu	Chỉ đọc dữ liệu
Bao gồm nhiều bảng từ nhiều CSDL	Dựa trên một câu lệnh SQL từ một CSDL
Ngắt kết nối	Kết nối
Di chuyển qua các bản ghi theo hai chiều	Chỉ di chuyển qua các bản ghi theo một chiều (chiều tiến)
Truy cập chậm	Truy cập nhanh

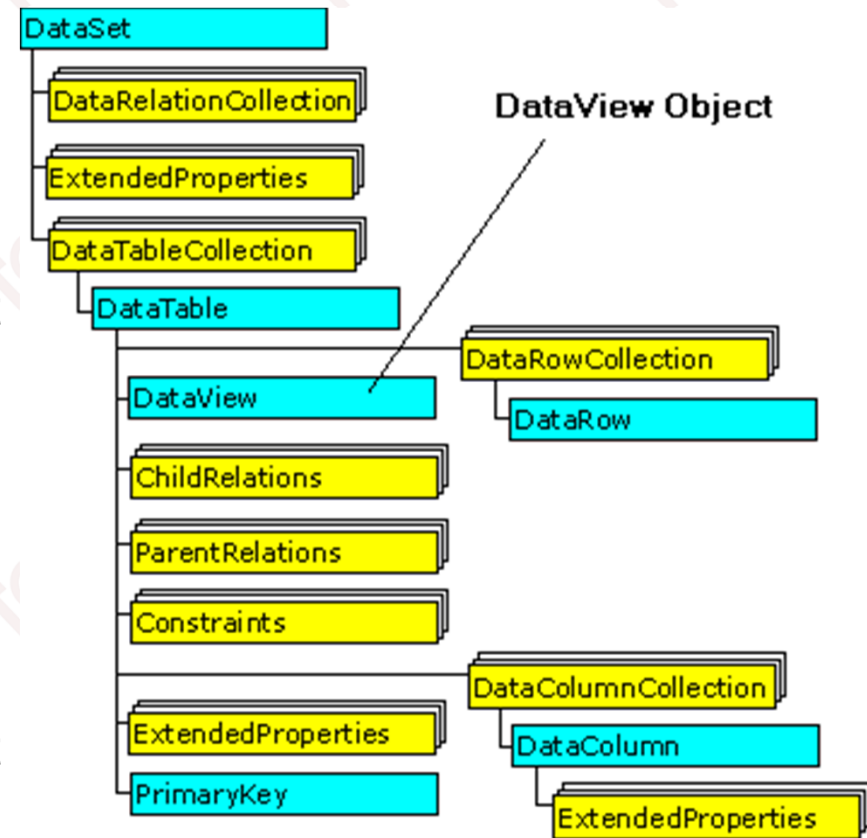
7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- Các thành phần chính của DataSet:
 - Tables: Tập các bảng;
 - Relations: Tập các quan hệ CSDL.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- DataTable:
 - Thể hiện một bảng trong CSDL.
 - Thuộc tính và phương thức:
 - TableName: Tên bảng;
 - Columns: Danh sách các cột (trường) trong bảng;
 - Rows: danh sách các hàng (bản ghi) trong bảng;
 - PrimaryKey: Danh sách các cột là khóa chính của bảng;
 - NewRow(): Phương thức tạo một hàng mới.



7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- DataColumn: Đại diện cho một cột trong bảng
 - ColumnName: Tên cột;
 - DataType: Kiểu dữ liệu của cột;

	ID ▾	Order No. ▾	Order Date ▾	Shipped? ▾
▶	75084	SO75084	31, Jul, 2004	<input checked="" type="checkbox"/>
	75061	SO75061	30, Jul, 2004	<input type="checkbox"/>
	75030	SO75030	29, Jul, 2004	<input checked="" type="checkbox"/>
	74998	SO74998	28, Jul, 2004	<input type="checkbox"/>
	74969	SO74969	27, Jul, 2004	<input checked="" type="checkbox"/>
	74937	SO74937	26, Jul, 2004	<input type="checkbox"/>
	74913	SO74913	25, Jul, 2004	<input checked="" type="checkbox"/>
	74873	SO74873	24, Jul, 2004	<input type="checkbox"/>

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

- DataRow: Đại diện cho một dòng (bản ghi)
 - RowState: Trạng thái Added, Modified, Deleted...
 - [i]: Truy xuất đến hàng có thứ tự i;
 - Delete(): Phương thức xóa bản ghi.

BookID	BookName	Author
1	Learning ActionScript 3.0, Second Edition	Rich Shupe
2	Flash CS5: The Missing Manual	Chris Grover

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Ví dụ: Chương trình sử dụng DataSet và DataAdapter để cập nhập trường ModifiedDate của bảng ShipMethod trong CSDL AdventureWorks

Bước 1: Khai báo đối tượng SqlConnection

```
SqlConnection conn = null;
```

Bước 2: Viết thủ tục OpenConnection

```
private void OpenConnection()  
{  
    string connectionString = "Data Source=Server; Initial  
        Catalog=AdventureWorks; UID=sa; PassWord='';  
    conn = new SqlConnection(connectionString);  
    conn.Open();  
}
```

7.2. CÁC THÀNH PHẦN CỦA ADO.NET (tiếp theo)

Bước 3: Thay đổi dữ liệu trong DataSet và cập nhật vào CSDL qua DataAdapter

```
//Lấy dữ liệu qua câu lệnh SQL
string strSQL = "Select * From ShipMethod";
//Mở kết nối CSDL
OpenConnection();
//Lấy dữ liệu vào DataAdapter
SqlDataAdapter da = new SqlDataAdapter(strSQL, conn);
DataSet ds = new DataSet();
//Đổ dữ liệu từ DataAdapter vào DataSet
da.Fill(ds, "ShipMethod");
SqlCommandBuilder builder = new SqlCommandBuilder(da);
//Cập nhật dữ liệu trong DataSet
foreach (DataRow dr in ds.Tables[0].Rows)
    dr["ModifiedDate"] = DateTime.Now;
//Cập nhật dữ liệu vào CSDL qua DataAdapter
da.Update(ds.Tables[0]);
```

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C#

- Các bước thực hiện lập trình cơ sở dữ liệu

➤ Bước 1: Khai báo các biến

- `SqlConnection conn;`
- `SqlDataAdapter da;`
- `SqlCommand command;`
- `DataSet ds;`
- `string connectionString ;`

➤ Bước 2: Khởi tạo giá trị cho biến string strConnectionString để trỏ đến CSDL trên máy chủ

```
connectionString = "Data Source=Server; Initial Catalog=AdventureWorks; UID=sa; PassWord='';"
```

➤ Bước 3: Khởi tạo biến SqlConnection và thực hiện mở kết nối

```
conn = new SqlConnection(connectionString);  
conn.Open();
```

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C#

- Bước 4: Lấy dữ liệu từ CSDL đưa vào DataAdapter và đổ vào DataSet qua
`string strSQL = "Mệnh đề SQL truy vấn dữ liệu";`

```
da = new SqlDataAdapter(strSQL, conn);
```

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

```
da.Fill(ds, "Product");
```

- Bước 5:

- Lấy dữ liệu theo từng dòng trong bảng qua DataSet;
- Xử lý cập nhật các dòng dữ liệu qua DataSet;
- Cập nhật lại CSDL qua DataAdapter.

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

- Sử dụng DataView và DataGridView.
 - DataView: Được sử dụng để sắp xếp, lọc và tìm kiếm dữ liệu.
 - Tạo đối tượng DataView:
`DataView <Tên đối tượng> = new DataView(DataTable);`
 - Các thuộc tính của DataView.

Thuộc tính	Ý nghĩa
AllowDelete	Cho phép xóa dữ liệu.
AllowEdit	Cho phép hiệu chỉnh, thay đổi dữ liệu.
AllowNew	Cho phép thêm mới dữ liệu.
Count	Trả về số bản ghi của bảng dữ liệu.
RowFilter	Đặt lọc dữ liệu qua biểu thức lọc.
Sort	Sắp xếp dữ liệu.

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

- Ví dụ: lấy dữ liệu từ DataTable đưa vào DataView

```
DataView dvwCustomer;  
string strSQL = "Select * From Sales.Customer";  
OpenConnection();  
SqlDataAdapter da = new SqlDataAdapter(strSQL, conn);  
DataSet ds = new DataSet();  
da.Fill(ds, "Product");  
dvwCustomer = new DataView(ds.Tables[0]);  
int rowNumber = dvwCustomer.Count;
```

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

- Các phương thức:

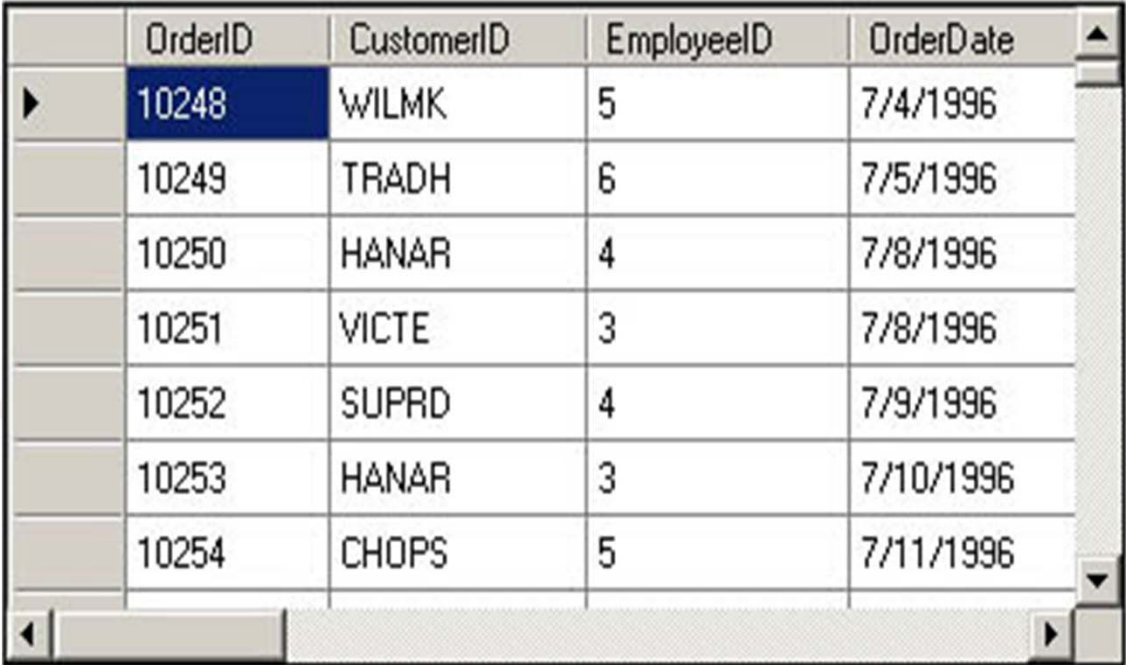
Phương thức	Ý nghĩa
AddNew	Thêm một hàng vào DataView.
Close	Đóng DataView.
Delete	Xóa một hàng khỏi DataView.
Find	Tìm bản ghi theo điều kiện tìm kiếm.
Open	Mở một DataView.

- Ví dụ: Tìm khách hàng có tên là Tomy

```
int rowsToDelete = dvwCustomer.Find("Tomy");
```

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

- DataGridView: Hiển thị dữ liệu dưới dạng bảng (grid).
- Hỗ trợ nhiều thuộc tính và phương thức thao tác dữ liệu.



	OrderID	CustomerID	EmployeeID	OrderDate
▶	10248	WILMK	5	7/4/1996
	10249	TRADH	6	7/5/1996
	10250	HANAR	4	7/8/1996
	10251	VICTE	3	7/8/1996
	10252	SUPRD	4	7/9/1996
	10253	HANAR	3	7/10/1996
	10254	CHOPS	5	7/11/1996

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

- Các thuộc tính:

Thuộc tính	Ý nghĩa
AutoGenerateColumns	Tự sinh ra các cột tương ứng với bảng dữ liệu sau khi thiết lập DataSource cho DataGridView.
Columns	Tập các cột của DataGridView.
DataSource	Xác định nguồn dữ liệu cho DataGridView.
DataMember	Xác định bảng dữ liệu sẽ hiển thị trên DataGridView.
Rows	Tập các hàng của DataGridView.
SelectedRows	Tập các hàng được chọn trong DataGridView.

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

Ví dụ: Đọc dữ liệu từ bảng Customer trong CSDL AdventureWorks và đưa vào DataGridView

Bước 1: Kéo thả trên giao diện đối tượng DataGridView và đặt tên là gvwCustomer

Bước 2: Khai báo đối tượng SqlConnection, SqlDataAdapter, DataSet

```
SqlConnection conn;
```

```
SqlDataAdapter da;
```

```
DataSet ds;
```

Bước 3: Viết thủ tục OpenConnection

```
private void OpenConnection()  
{  
    string connectionString = "Data Source=Server; Initial  
Catalog=AdventureWorks; UID=sa; PassWord='';  
    conn = new SqlConnection(connectionString);  
    conn.Open();  
}
```

7.3. LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO.NET TRONG C# (tiếp theo)

Bước 4: Đọc dữ liệu vào DataSet hiển thị trên DataGridView

```
string strSQL = "Select * From Sales.Customer";  
//Gọi OpenConnection để mở kết nối với CSDL  
OpenConnection();  
//Đọc dữ liệu vào DataAdapter  
SqlDataAdapter adapter = new SqlDataAdapter(strSQL, conn);  
//Đổ dữ liệu vào DataSet  
DataSet ds = new DataSet();  
adapter.Fill(ds);  
//Hiển thị dữ liệu từ DataSet ra DataGridView  
gvwCustomer.DataSource = ds.Tables[0];  
conn.Close();
```

Trong bài này, chúng ta đã nghiên cứu các nội dung chính sau:

- Tổng quan về ADO.NET;
- Kiến trúc và các thành phần cơ bản của ADO.NET;
- Cách lập trình CSDL với ADO.NET trong C#.