



# CHƯƠNG 8: LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ADO .NET

---





# Nội dung chính

---

1. Tổng quan
2. Tổng quan ADO .Net
3. .Net Data Provider
4. Demo: Lấy dữ liệu dùng ADO .Net
5. Đối tượng DataSet
6. Thiết kế và gắn kết dữ liệu
7. Tích hợp XML
8. Lab: Tạo ứng dụng ADO.Net



# Mục đích của chương

---

- ❑ Chương này cung cấp sinh viên kiến thức cần thiết để tạo các ứng dụng mức cao truy cập dữ liệu dùng VB.Net
- ❑ Sau bài này sinh viên có thể:
  - ❑ Liệt kê các lợi ích ADO .Net
  - ❑ Tạo các ứng dụng dùng ADO .Net
  - ❑ Liệt kê các thành phần chính của đối tượng ADO.Net và các chức năng của nó.
  - ❑ Dùng VS.Net để thiết kế và gắn kết dữ liệu
  - ❑ Giải thích cách tích hợp XML cùng ADO.Net



# 1. Tổng quan

---

- ❑ Bạn sẽ học cách sử dụng ADO .Net
- ❑ Cách sử dụng đối tượng **DataSet**
- ❑ Bạn sẽ học cách thiết kế dữ liệu trong VS.Net và cách gắn dữ liệu lên WinForm và WebForm.
- ❑ Cuối cùng bạn sẽ học cách tích hợp XML vào ADO .Net



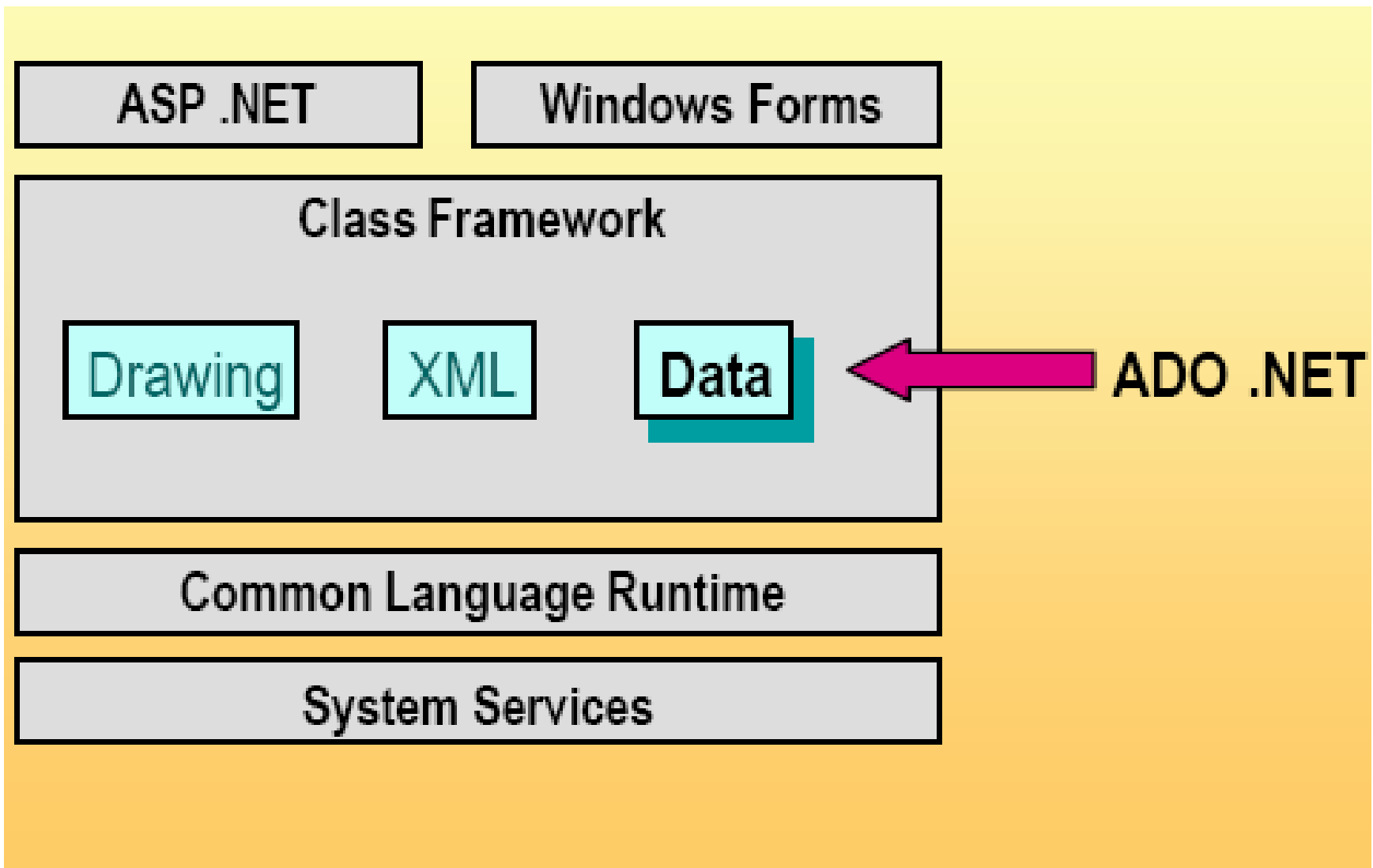
## 2. Tổng quan ADO .Net

---

- ❑ Giới thiệu ADO .Net
- ❑ Các lợi ích ADO .Net



# Giới thiệu ADO .Net





# Giới thiệu ADO.Net

---

- ❑ ADO.Net là một tập các lớp cho phép các ứng dụng .Net đọc và cập nhật thông tin DB và các nơi lưu trữ dữ liệu khác. Namespace: **System.Data**
- ❑ ADO.Net cung cấp đồng nhất cách truy cập các nguồn dữ liệu khác nhau như SQL Server, OLE DB, các nguồn dữ liệu không quan hệ như MS Exchange, và các tài liệu XML
- ❑ ADO.Net cải tiến khả năng disconnect tới dữ liệu.



# Giới thiệu ADO.Net

---

- ❑ ADO.Net cung cấp hai .Net Data Providers:
  - ❑ .Net Data Provider cho SQL Server
  - ❑ .Net Data Provider cho OLE DB
- ❑ ADO.Net cung cấp nhiều công cụ cho việc đọc, cập nhật, thêm mới và xóa dữ liệu. Nhiều đối tượng trong thư viện là tương tự nhau và được nhận diện qua tên tiền tố của chúng ví dụ như **SqlDataReader** và **OleDbDataReader** cả hai đều cung cấp đọc các bản ghi từ nguồn dữ liệu.





# Các lợi ích của ADO.Net

---

- ❑ Tương tự như ADO
- ❑ Được thiết kế cho dữ liệu không kết nối.
- ❑ Nằm trong nội tại .Net Framework nên dễ dàng trong việc sử dụng ngôn ngữ để phát triển.
- ❑ Hỗ trợ XML
  - ❑ ADO và XML có trước nhưng không tương thích
  - ❑ ADO dựa trên cơ sở dữ liệu quan hệ.
  - ❑ XML dựa trên cơ sở dữ liệu phân cấp.
  - ❑ ADO.Net kết hợp hai công nghệ này lại với nhau.



## 3. .Net Data Provider

---

- ❑ .Net Data Provider cho phép truy cập các nguồn dữ liệu xác định:
  - ❑ ***System.Data.SqlClient*** dùng truy cập SQL Server 7.0 trở lên
  - ❑ ***System.Data.OleDb*** dùng truy cập bất kỳ nguồn dữ liệu nào hỗ trợ OLE DB
- ❑ Dùng đối tượng ***Connection***
  - ❑ Kết nối tới một Database
- ❑ Dùng đối tượng ***Command***
  - ❑ Thực thi các câu lệnh
  - ❑ Dùng đối tượng ***Command*** với các Stored Procedure
- ❑ Dùng đối tượng ***DataReader***
  - ❑ Tạo một luồng dữ liệu chỉ đọc
- ❑ Dùng đối tượng ***DataAdapter***
  - ❑ để thay đổi dữ liệu nguồn và một ***DataSet***



# Dùng đối tượng *Connection*

---

## ❑ SqlConnection

*Dim conSQL As SqlClient.SqlConnection*

*conSQL = New SqlClient.SqlConnection()*

*conSQL.ConnectionString = "Server = localhost; DataSource = Northwind; Uid = sa;  
Pwd = sa;"*

*conSQL.Open()*

## ❑ OleDbConnection

*Dim conAccess As OleDb.OleDbConnection*

*conAccess = New OleDb.OleDbConnection()*

*conAccess.ConnectionString = "Provider=  
Microsoft.Jet.OLEDB.4.0;Data Source=c:\NWind.MDB"*

*conAccess.Open()*



# Dùng đối tượng *Command*

- ❑ Có hai cách để tạo đối tượng *Command*:
  - ❑ Sử dụng Constructor *Command*
  - ❑ Sử dụng phương thức *CreateCommand*
- ❑ Có ba cách để thực thi một *Command*:
  - ❑ *ExecuteReader*
  - ❑ *ExecuteScalar*
  - ❑ *ExecuteNonQuery*
  - ❑ *ExecuteXMLReader*

```
Dim commSQL As SqlClient.SqlCommand  
commSQL = New SqlClient.SqlCommand( )  
commSQL.Connection = conSQL  
commSQL.CommandText = "Select Count(*) from Authors"  
MessageBox.Show(commSQL.ExecuteScalar( ).ToString)
```



## Dùng đối tượng **Command** với các **Stored Procedure**

---

- ☐ Tạo một đối tượng **Command**
- ☐ Thiết lập **CommandType** là **StoredProcedure**
- ☐ Dùng phương thức **Add** để tạo và thiết lập các biến (Parameter)
- ☐ Dùng thuộc tính **ParameterDirection** để thiết lập kiểu biến
- ☐ Gọi phương thức **ExecuteReader**
- ☐ Dùng đối tượng **DataReader** để hiển thị hoặc duyệt qua các bản ghi và đóng khi kết thúc
- ☐ Truy cập đầu ra và trả về các biến



# Demo: Dùng đối tượng Command

## ❑ Tạo một Store trong SQL Server

```
CREATE PROCEDURE byroyalty @percentage int
AS
select au_id from titleauthor
where titleauthor.royaltyper = @percentage
```

## ❑ Code VB.Net

```
Imports System.Data.SqlClient
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
Dim conSQL As SqlConnection
conSQL = New SqlConnection( )
conSQL.ConnectionString = "Server = localhost;DataBase = Pubs;" & _
"UID = sa;PWD = sa"
conSQL.Open( )
Dim commSQL As SqlCommand = New SqlCommand( )
commSQL.Connection = conSQL
commSQL.CommandType = CommandType.StoredProcedure
commSQL.CommandText = "byroyalty"
```



## Demo: Dùng đối tượng Command

---

```
Dim paramSQL As New SqlClient.sqlParameter( _  
    "@percentage", SqlDbType.Int)  
paramSQL.Direction = ParameterDirection.Input  
paramSQL.Value = "30"  
commSQL.Parameters.Add(paramSQL)  
Dim datRead As SqlClient.SqlDataReader  
datRead = commSQL.ExecuteReader()  
Do While datRead.Read()  
    MessageBox.Show(datRead(0).ToString)  
Loop  
datRead.Close()  
End Sub
```



# Dùng đối tượng DataReader

---

## ❑ Đọc dữ liệu

```
Dim commSQL As SqlClient.SqlCommand = New _  
SqlClient.SqlCommand()  
commSQL.Connection = conSQL  
commSQL.CommandText = "Select au_lname, au_fname from authors"  
Dim datRead As SqlClient.SqlDataReader  
datRead = commSQL.ExecuteReader()  
Do Until datRead.Read = False  
MessageBox.Show(datRead.GetString(1) & " "  
& datRead.GetString(0))  
Loop  
datRead.Close()
```

## ❑ Lấy thông tin

## ❑ Trả về nhiều tập hợp kết quả





# Dùng đối tượng **DataAdapter**

---

- ❑ Được sử dụng như một liên kết giữa DataSource và Các Table lưu trữ trong Cache

```
Dim adaptSQL As New SqlClient.SqlDataAdapter( _  
    "Select * from authors", conSQL)  
Dim datPubs As DataSet = New DataSet()  
adaptSQL.Fill(datPubs, "NewTable")  
' Manipulate the data locally  
adaptSQL.Update (datPubs, "NewTable")
```



## 4. Demo: Lấy dữ liệu dùng ADO .Net

---

- ❑ Bạn sẽ học cách lấy dữ liệu từ CSDL SQL Server bằng việc dùng đối tượng **SQLDataAdapter** trong VB.Net.



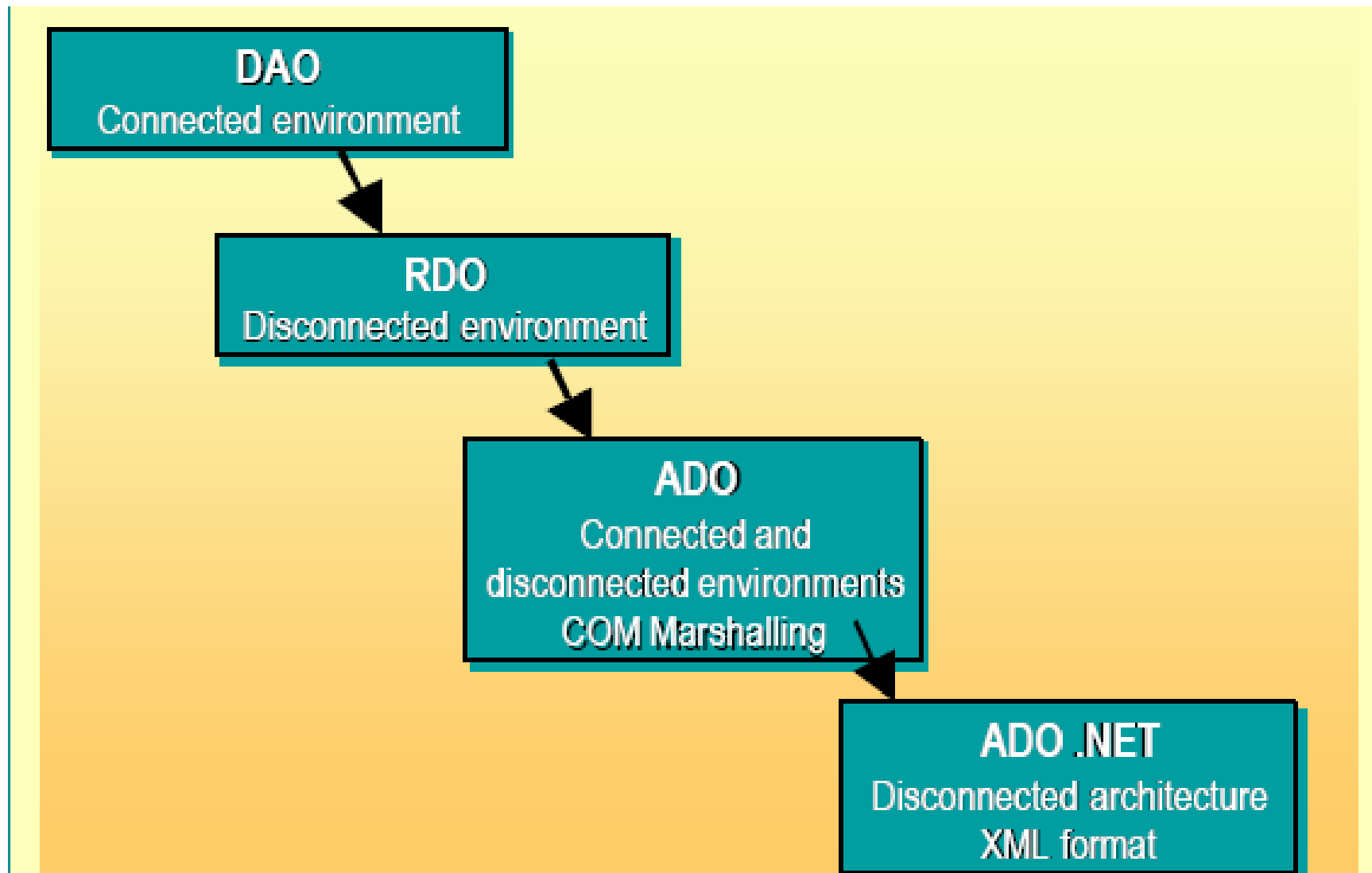
## 5. Đối tượng DataSet

---

- ❑ Tổng quan về dữ liệu không kết nối
- ❑ Đối tượng DataSet
- ❑ Cư trú dữ liệu trong DataSet
- ❑ Sử dụng Relationship trong DataSet
- ❑ Sử dụng các Constraint
- ❑ Cập nhật dữ liệu trong DataSet
- ❑ Cập nhật dữ liệu tại nguồn



# Tổng quan về dữ liệu không kết nối



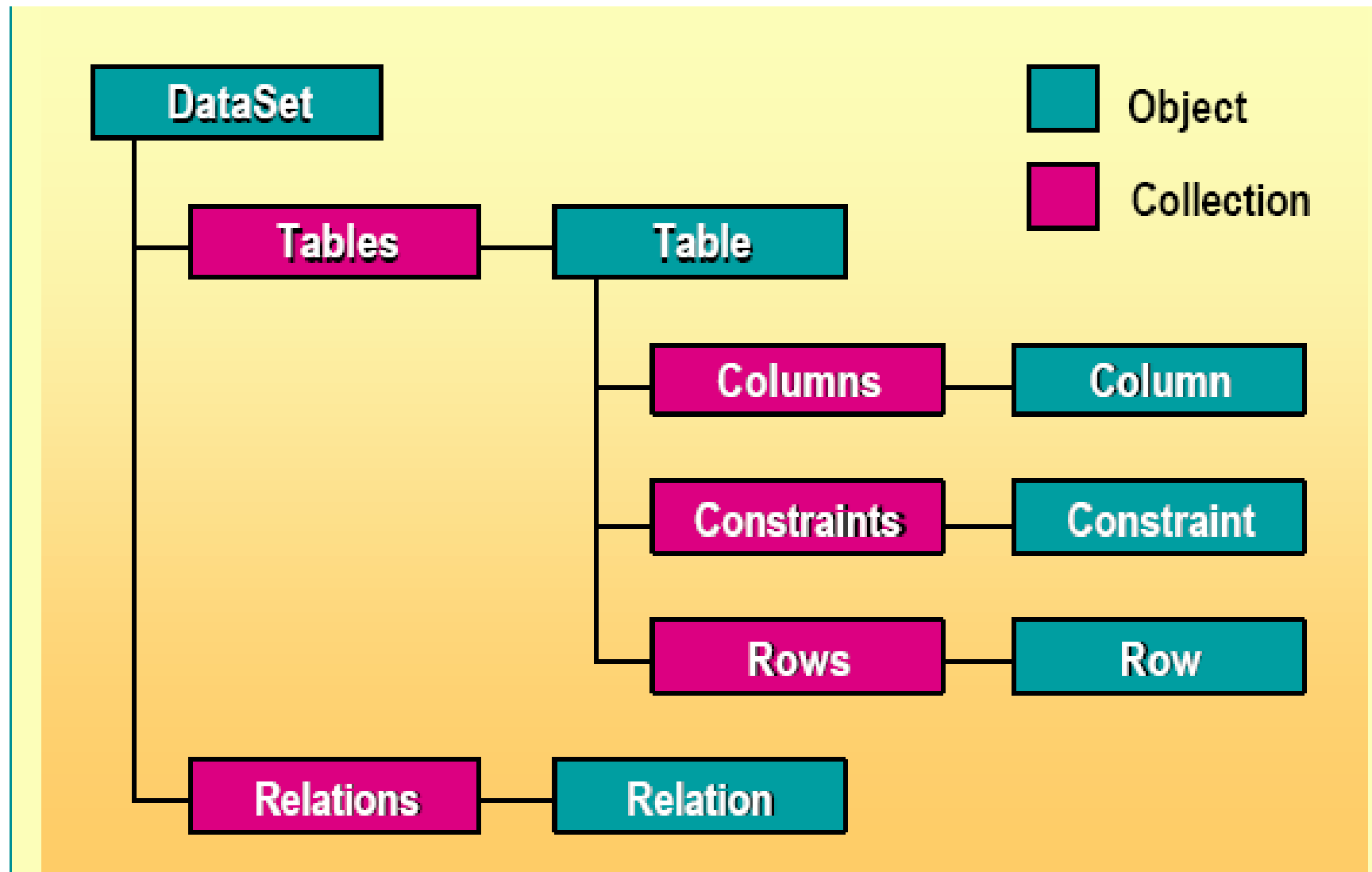


# Tổng quan về dữ liệu không kết nối

---

- ❑ Mỗi công nghệ truy cập dữ liệu đều đã được cải thiện khái niệm không kết nối, nhưng đến ADO.Net mới cung cấp giải pháp cách đầy đủ.
- ❑ ADO.Net được thiết kế dùng cho Internet
- ❑ ADO.Net sử dụng XML như là các định dạng truyền tải.
- ❑ ADO.Net cung cấp một đối tượng mới cho việc caching dữ liệu trên máy client. Đối tượng này là **DataSet**

# Đối tượng DataSet





# Cư trú dữ liệu trong DataSet

---

- ❑ Cư trú dữ liệu trong DataSet từ một RDBMS

```
Dim adaptSQL As SqlConnection.SqlDataAdapter
```

```
adaptSQL = New SqlConnection.SqlDataAdapter(  
"Select * from authors", conSQL)
```

```
Dim datPubs As DataSet = New DataSet()
```

```
adaptSQL.Fill(datPubs, "NewTable")
```

- ❑ Lập trình tạo Dataset

```
Dim datPubs As DataSet = New DataSet()
```

```
Dim tblAuthors As DataTable = New DataTable("authors")
```

```
tblAuthors.Columns.Add("AuthorID", System.Type.GetType(  
"System.Int32"))
```



# Sử dụng Relationship trong DataSet

---

## ❑ Tạo Relationship

```
Dim relPubsTitle As DataRelation = New DataRelation(  
    "PubsTitles",  
    datPubs.Tables("Publishers").Columns("pub_id"),  
    datPubs.Tables("Titles").Columns("pub_id"))  
  
datPubs.Relations.Add(relPubsTitle)
```

## ❑ Truy cập dữ liệu quan hệ

```
Dim PubRow As DataRow, TitleRows( ) As DataRow  
PubRow = datPubs.Tables("Publishers").Rows(0)  
TitleRows = PubRow.GetChildRows("PubsTitles")
```





# Sử dụng các Constraint

---

## ❑ Tạo New Constraint

- ❑ **ForeignKeyConstraints:** đây là ràng buộc xảy ra khi một hàng con cũng được cập nhật khi hàng cha được update hoặc delete
- ❑ **UniqueConstraints:** đây là ràng buộc đảm bảo các giá trị trong một cột hoặc các cột là duy nhất.

## ❑ Sử dụng Constraint tồn tại

```
adaptSQL = New SqlClient.SqlDataAdapter("Select title_id,  
title, type, price from titles", conSQL)  
adaptSQL.FillSchema(datPubs, schematype.Source, "Titles")  
adaptSQL.Fill(datPubs, "Titles")  
'Edit some data  
adaptSQL.Fill(datPubs, "Titles")
```



# Cập nhật dữ liệu trong DataSet

---

## ❑ Thêm Rows

*Dim drNewRow As DataRow = datPubs.Tables("Titles").NewRow*

*'Populate columns*

*datPubs.Tables("Titles").Rows.Add(drNewRow)*

## ❑ Thay đổi Rows

*drChangeRow.BeginEdit()*

*drChangeRow("Title") = drChangeRow("Title").ToString & " 1"*

*drChangeRow.EndEdit()*

## ❑ Xóa dữ liệu

*datPubs.Tables("Titles").Rows.Remove(drDelRow))*



# Cập nhật dữ liệu tại nguồn

---

## ❑ Chỉ rõ ràng các cập nhật

```
Dim cmd As cmd.CommandText = "Insert into titles(" & _  
"title_id, title, type) values(@t_id,@title,@type)"  
cmd.Parameters.Add("@t_id", SqlDbType.VarChar,6,"title_id")  
cmd.Parameters.Add("@title", SqlDbType.VarChar,80,"title")  
cmd.Parameters.Add("@type", SqlDbType.Char,12,"type")  
adaptSQL.InsertCommand = cmd  
adaptSQL.Update(datPubs, "titles")
```

## ❑ Tự động phát sinh update

```
Dim sqlCommBuild As New SqlCommandBuilder(adaptSQL)  
adaptSQL.Update(datPubs, "titles")  
MsgBox(sqlCommBuild.GetInsertCommand.ToString)
```

# Demo: Sử dụng DataSet





## 6. Thiết kế và gắn kết dữ liệu

---

- ❑ Thiết kế DataSet
- ❑ Tùy chỉnh form dữ liệu
- ❑ Gắn dữ liệu trong WinForms
- ❑ Gắn dữ liệu trong WebForms



# Thiết kế DataSet

---

- ❑ Tùy chỉnh cấu hình DataAdapter
  - ❑ Phát sinh đối tượng **DataAdapter** trong thủ tục **InitializeComponent** dùng trong code của bạn.
- ❑ Công cụ phát sinh DataSet
  - ❑ Phát sinh một **DataSet** trên cơ sở dữ liệu đã tồn tại một **DataAdapter**.



# Tùy chỉnh cấu hình DataAdapter

---

- ☐ Tên kết nối
- ☐ Kiểu truy vấn
  - ☐ Câu lệnh SQL
  - ☐ Hoặc Stored Procedure mới
  - ☐ Hoặc Stored Procedure đã tồn tại
- ☐ Các chi tiết lựa chọn câu truy vấn



# Công cụ phát sinh DataSet

- ❑ Đây là công cụ phát sinh **DataSet** tự động từ **DataAdapter**

**Generate Dataset**

Generate a dataset that includes the specified tables.

Choose a dataset:

☒ Existing: Test.DataSet1

☐ New: DataSet2

Choose which table(s) to add to the dataset:

☒ authors (SqlDataAdapter1)

☒ Add this dataset to the designer.

OK Cancel Help





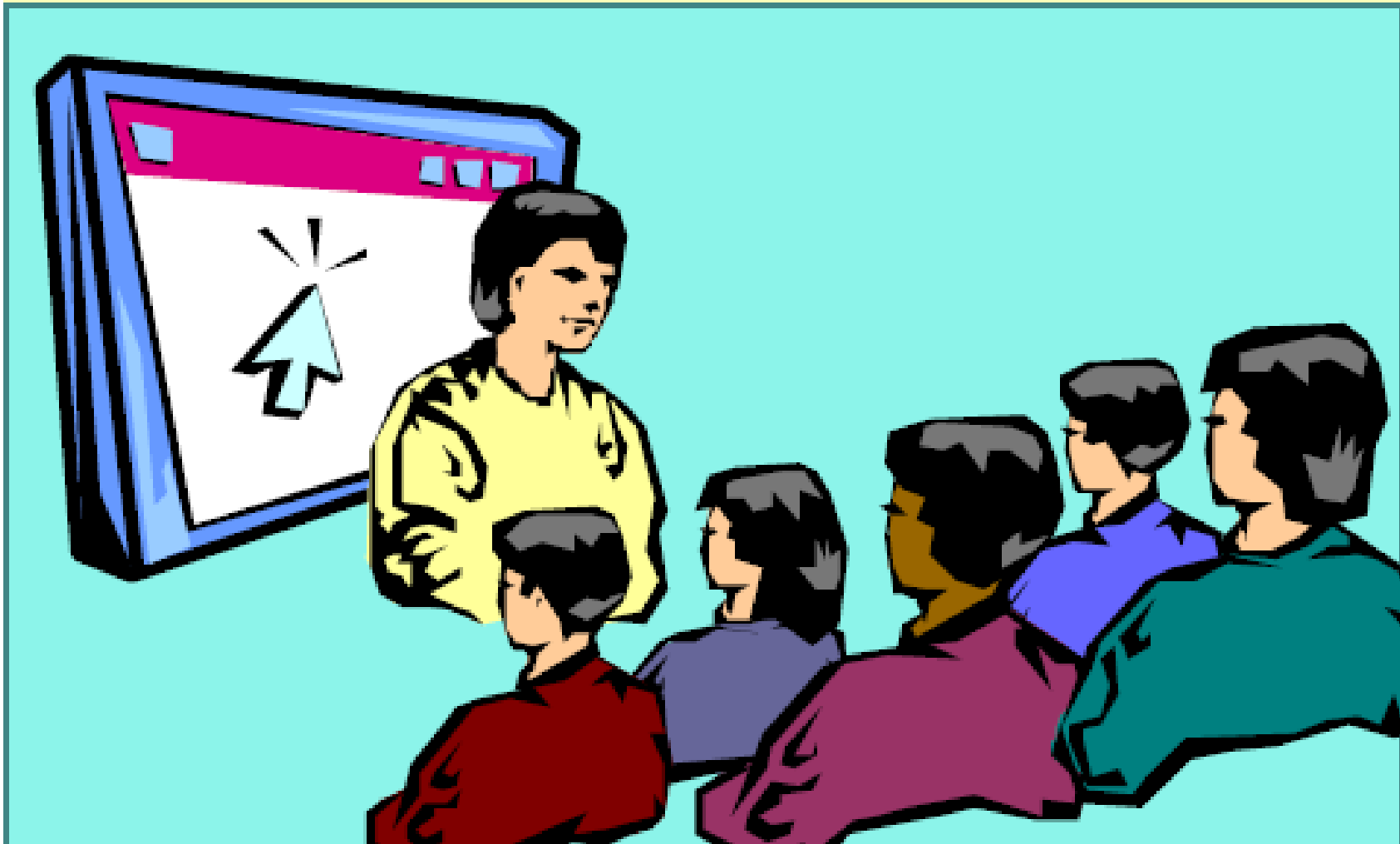
# Tùy chỉnh form dữ liệu

---

- ☐ Các thông tin được yêu cầu:
  - ☐ Tên của **DataSet**
  - ☐ Connection được sử dụng
  - ☐ Các Table hoặc các View nào, và các cột của chúng.
  - ☐ Cách hiển thị dữ liệu
  - ☐ Những button nào được tạo

# Demo: Sử dụng Form tùy chỉnh dữ liệu

---





# Gắn dữ liệu trong WinForms

---

## ❑ Cách gắn đơn giản

```
da = New SqlClient.SqlDataAdapter("Select au_lname, " & _  
"au_fname from authors", sqlconn)  
da.Fill(ds, "authors")  
TextBox1.DataBindings.Add("Text", _  
ds.Tables("authors"), "au_fname")
```

## ❑ Cách gắn phức tạp

```
da = New SqlClient.SqlDataAdapter("Select au_lname, " & _  
"au_fname from authors", sqlconn)  
da.Fill(ds, "authors")  
DataGrid1.DataSource = ds.Tables("authors")
```



# Gắn dữ liệu trong WebForms

---

## ❑ Gắn dữ liệu chỉ đọc

```
Dim sqlComm As New SqlClient.SqlCommand("Select" & _  
" * from authors", sqlconn)  
Dim sqlReader As SqlClient.SqlDataReader  
sqlReader = sqlComm.ExecuteReader  
DataGrid1.DataSource( ) = sqlReader  
DataGrid1.DataBind( )
```



## 7. Tích hợp XML

---

- ☐ Tại sao phải dùng schemas (giản đồ)?
- ☐ Miêu tả cấu trúc XML
- ☐ Tạo Schemas
- ☐ Sử dụng XML và Schemas trong ADO.Net
- ☐ DataSets và XmlDocument

# Tại sao phải dùng Schemas

---

- ❑ Định nghĩa định dạng của dữ liệu
- ❑ Sử dụng để kiểm tra tính hợp lệ
- ❑ Có lợi thế hơn qua việc định nghĩa các kiểu tài liệu (document type definitions - DTSs )
- ❑ Cú pháp XML
- ❑ Sử dụng lại các kiểu
- ❑ Nhóm một nhóm các phần tử ...

# Miêu tả cấu trúc XML

---

- ❑ Schemas có thể được miêu tả gồm:
  - ❑ Các phần tử trong tài liệu (E)
  - ❑ Các thuộc tính trong tài liệu (A)
  - ❑ Các quan hệ giữa phần tử và thuộc tính (R)
  - ❑ Các kiểu dữ liệu
  - ❑ Thứ tự của các phần tử
  - ❑ Các phần tử nào là tùy chọn



# Ví dụ một tài liệu XML

---

```
<?xml version="1.0" ?>
<pubs>
  <Publishers>
    <pub_id>0736</pub_id>
    <pub_name>Lucerne Publishing</pub_name>
    <city>Boston</city>
    <state>MA</state>
    <country>USA</country>
  </Publishers>
  <Publishers>
    <pub_id>0877</pub_id>
    <pub_name>Litware, Inc.</pub_name>
    <city>Washington</city>
    <state>DC</state>
    <country>USA</country>
  </Publishers>
</pubs>
```





## Ví dụ Tài liệu này liên kết một schema để miêu tả cấu trúc

```
<xsd:schema id="pubs" targetNamespace="http://tempuri.org/Publishers.xsd"
xmlns="http://tempuri.org/Publishers.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" attributeFormDefault="qualified"
elementFormDefault="qualified">
<xsd:element name="pubs" msdata:IsDataSet="true" msdata:EnforceConstraints="False">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="Publishers">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="pub_id" type="xsd:string" minOccurs="0" />
            <xsd:element name="pub_name" type="xsd:string" minOccurs="0" />
            <xsd:element name="city" type="xsd:string" minOccurs="0" />
            <xsd:element name="state" type="xsd:string" minOccurs="0" />
            <xsd:element name="country" type="xsd:string" minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```



# Tạo Schemas

---

- ☐ Tạo Schemas từ các tài liệu XML đã tồn tại.
- ☐ Tạo Schemas từ Databases
- ☐ Làm việc với Schemas
- ☐ Kiểm tra các tài liệu XML dựa vào Schema



# Dùng dữ liệu XML và schemas trong ADO.Net

---

## ☐ Nạp dữ liệu XML vào một DataSet

```
Dim datXML As DataSet = New DataSet()  
datXML.ReadXml("c:\publishers.xml")  
MessageBox.Show(datXML.Tables(0).Rows(0)(0).ToString)
```

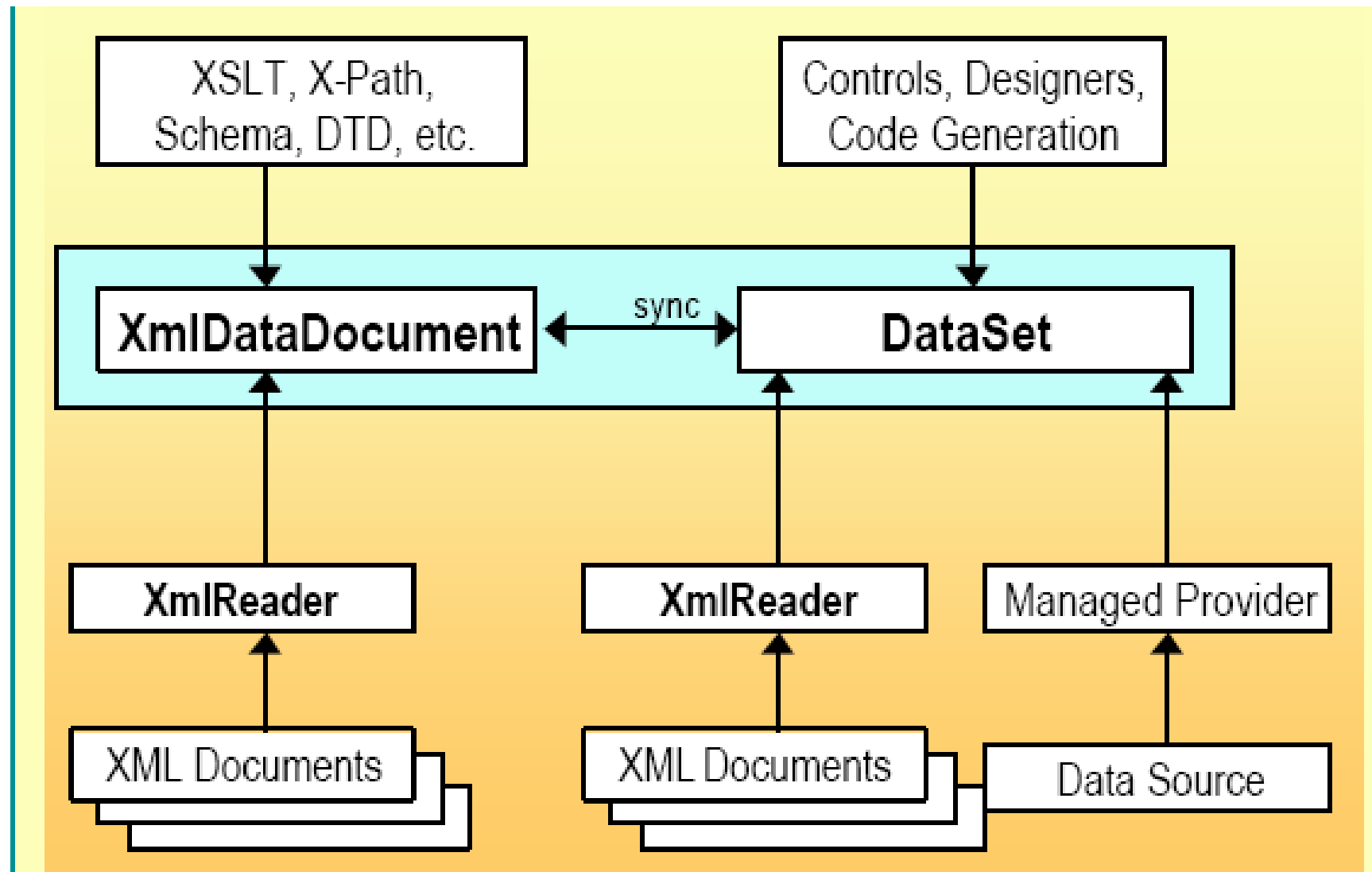
## ☐ Sử dụng kiểu DataSet được định nghĩa

### ☐ Tăng hiệu năng

### ☐ Đơn giản trong việc coding

```
MessageBox.Show(pubs.Publishers(0).pub_id)
```

# DataSets và XmlDocument Documents



# Demo: Sử dụng XML Schema

