

# ADO.NET

# Giới thiệu

- ❑ **ADO.NET (ActiveX Data Object)** là một bộ các thư viện hướng đối tượng (OOP) hỗ trợ tương tác với các loại dữ liệu
- ❑ Dữ liệu nguồn có thể là
  - Cơ sở dữ liệu (database)
  - Tập tin text, excel hoặc XML, ...
- ❑ Trong môn học này, chúng ta sẽ chỉ xem xét tới cách ADO.NET làm việc với database

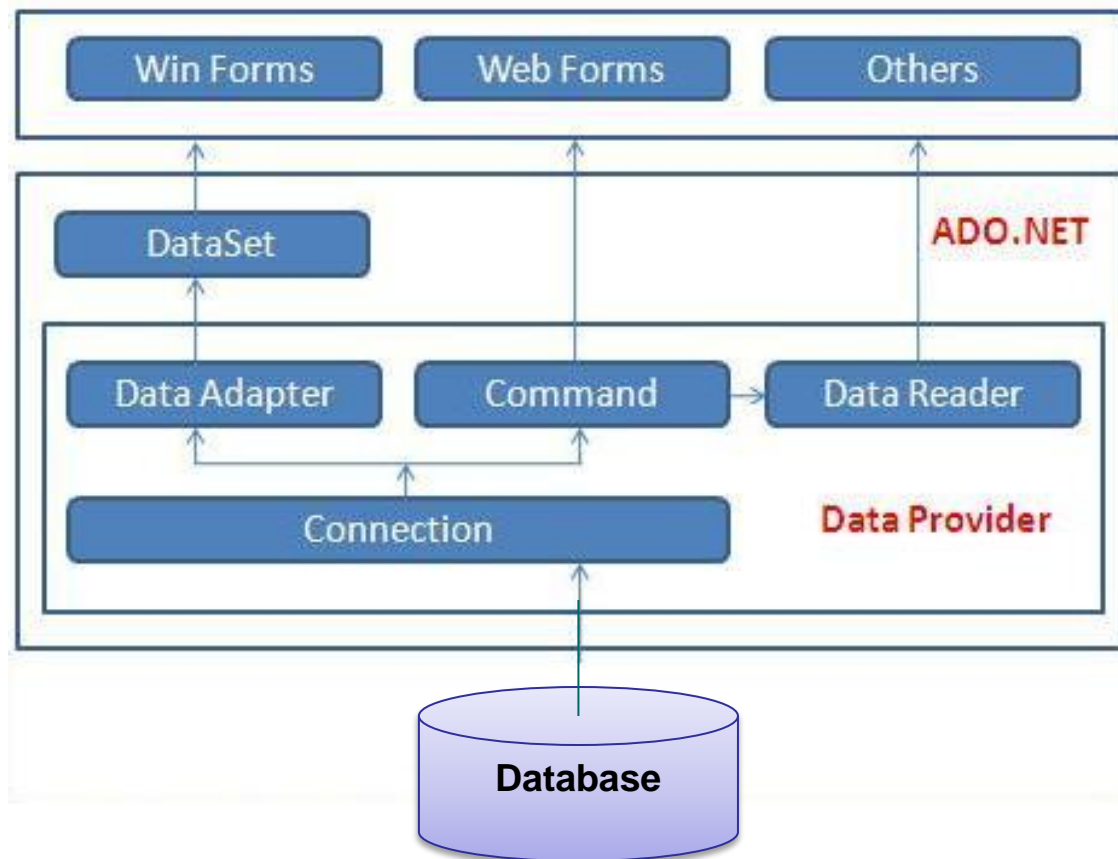
# Giới thiệu

## ❑ Tại sao dùng ADO.NET?

- Các loại dữ liệu cũ sử dụng giao thức ODBC.
- Các loại dữ liệu mới hơn sử dụng giao thức OleDb.

→ Vì vậy cần có một thư viện thống nhất để làm việc với chúng, đây chính là lý do mà ADO.NET được tạo ra.

# Giới thiệu



# Data provider

- ❑ ADO.NET cung cấp một cách thức chung để tương tác với nguồn dữ liệu, nhưng với mỗi loại dữ liệu phải sử dụng một thư viện khác nhau.
- ❑ Các thư viện này được gọi là **Data Provider** và thường được đặt tên theo giao thức hoặc loại dữ liệu mà chúng cho phép truy xuất.

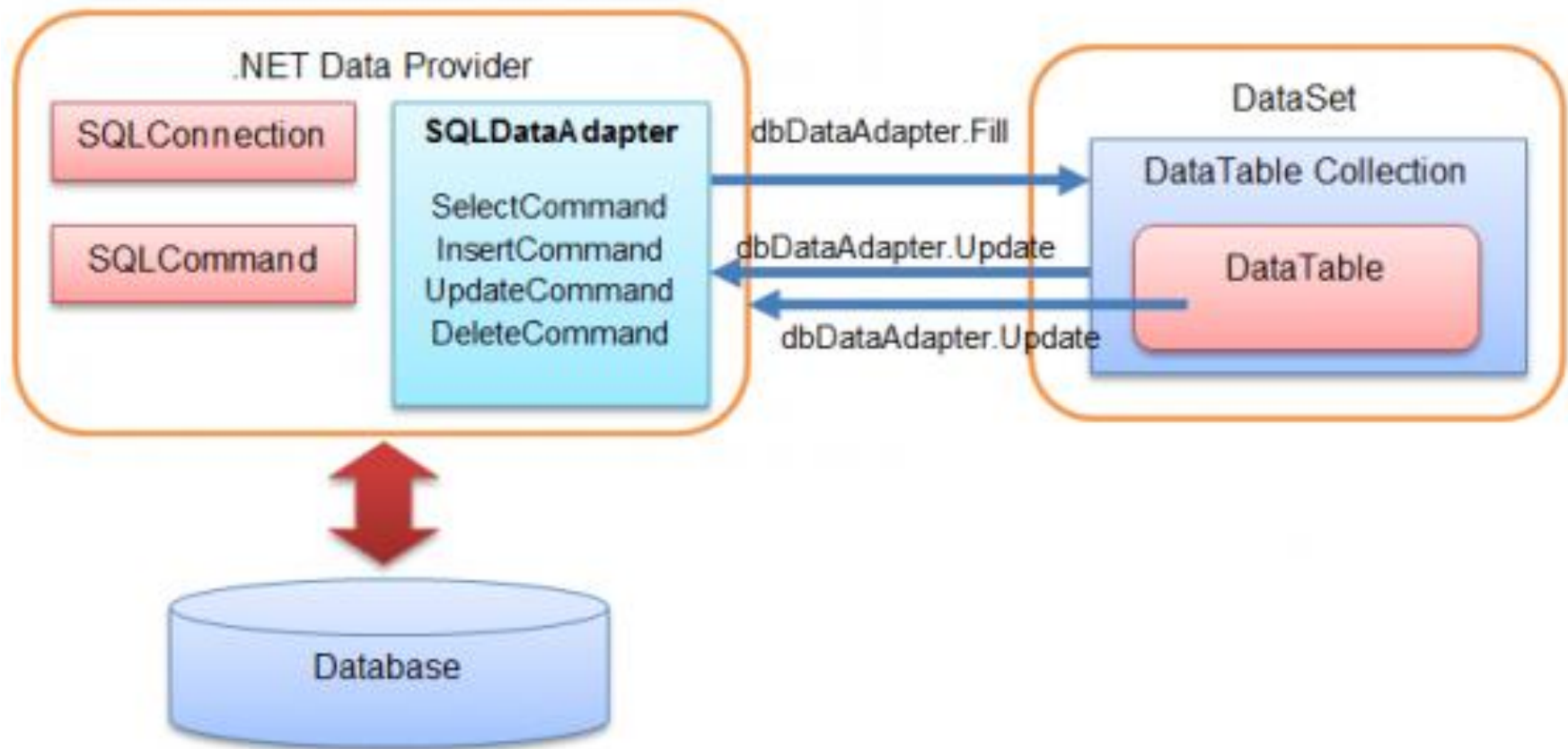
# Data provider

## ❑ Tiền tố của các **Data provider** thông dụng

Provider Name	API prefix	Data Source Description
ODBC Data Provider	Odbc	Các loại cơ sở dữ liệu cũ
OleDb Data Provider	OleDb	Access (hoặc Excel)
Oracle Data Provider	Oracle	Oracle
SQL Data Provider	Sql	SQL Server
Borland Data Provider	Bdp	Interbase, SQL Server, IBM DB2, Oracle.

# Data provider

## ❑ Cách thức truy cập CSDL của ADO.NET



# Các đối tượng của ADO.NET

- ❑ \_Connection
- ❑ \_Command
- ❑ \_DataReader
- ❑ DataSet
- ❑ \_DataAdapter



# Các đối tượng của ADO.NET

**\_Connection**

**\_Command**

**\_DataReader**

**DataSet**

**\_DataAdapter**

# Các đối tượng của ADO.NET

**\_Connection**

**\_Command**

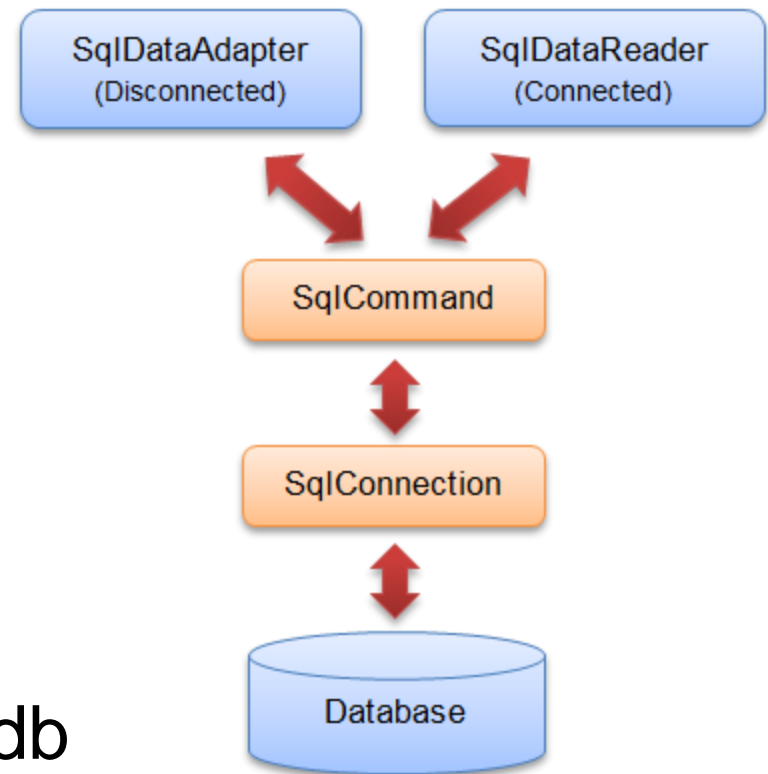
**\_DataReader**

**DataSet**

**\_DataAdapter**

# Đối tượng Connection

- ❑ Để tương tác với database, phải có một kết nối tới nó
- ❑ Kết nối giúp xác định:
  - Database server
  - Database name
  - User name
  - Password
  - Và các tham số cần thiết để kết nối tới db



# Đối tượng Connection

❑ Ví dụ SqlConnection:

```
SqlConnection conn = new  
SqlConnection("Data Source=(local);  
Initial Catalog=QLNV;  
Integrated Security=True");
```

```
SqlConnection conn=new SqlConnection  
("Data Source=MT01demo\\SQL2008R2;  
Initial Catalog=QLNV;  
User Id=sa;Password=sql2008;");
```

# Đối tượng Connection

## ❑ Ví dụ SqlConnection:

Connection String Parameter Name	Description
Data Source	Identifies the server. Could be local machine, machine domain name, or IP Address.
Initial Catalog	Database name.
Integrated Security	Set to <u>SSPI</u> to make connection with user's Windows login
User ID	Name of user configured in SQL Server.
Password	Password matching SQL Server User ID.

# Đối tượng Connection

❑ Ví dụ OleDbConnection:

```
OleDbConnection conn = new  
OleDbConnection("Provider=Microsoft.ACE.0  
LEDB.12.0;Data Source=qInv.accdb");
```

# Các đối tượng của ADO.NET

**\_Connection**

**\_Command**

**\_DataReader**

**DataSet**

**\_DataAdapter**

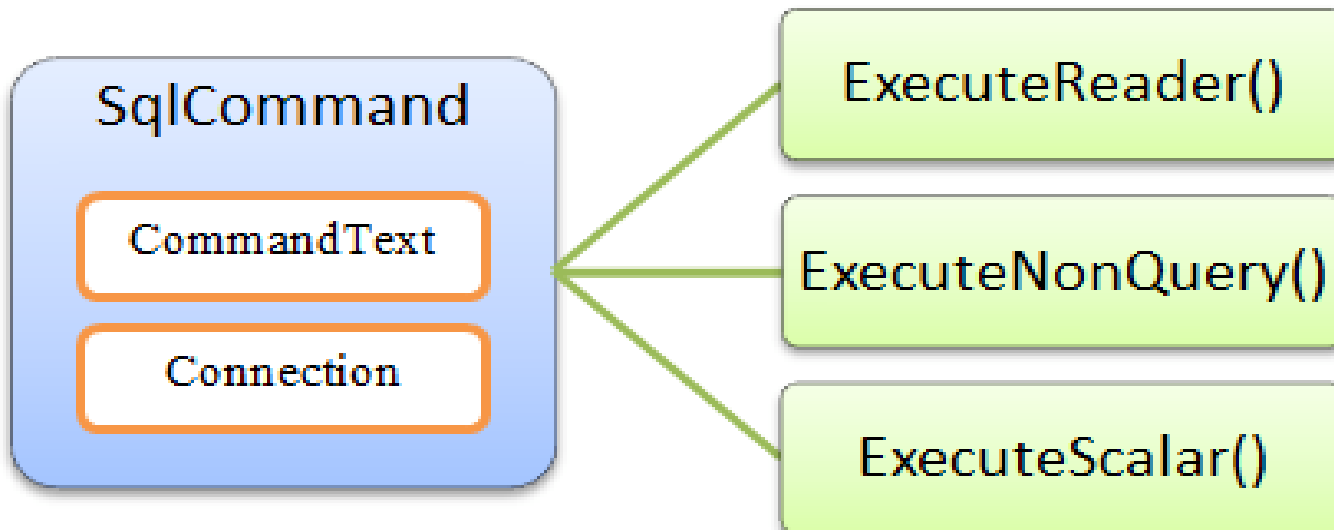
# Đối tượng Command

- ❑ Đối tượng Command thể hiện hành động cụ thể trong quá trình tương tác với database.
- ❑ Đối tượng command
  - gửi một câu lệnh SQL tới database
  - dùng một đối tượng connection để xác định database nào sẽ được truy xuất



# Đối tượng Command

## ❑ Kiến trúc SqlCommand



# Đối tượng Command

- ❑ Khai báo SqlCommand

```
SqlCommand cmd = new SqlCommand  
("select * from nhanvien", conn);
```

# Các đối tượng của ADO.NET

**\_Connection**

**\_Command**

**\_DataReader**

**DataSet**

**\_DataAdapter**

# Đối tượng DataReader

- ❑ Đối tượng data reader cho phép bạn lấy được kết quả của một câu lệnh SELECT từ một đối tượng command.
- ❑ Dữ liệu trả về từ một data reader là một luồng
  - tốc độ nhanh khi truy xuất theo một thứ tự nhất định

# Đối tượng SqlDataReader

## □ Ví dụ SqlDataReader

```
SqlCommand cmd = new SqlCommand ("select  
* from nhanvien", conn);  
SqlDataReader rdr = cmd.ExecuteReader();
```

# Các đối tượng của ADO.NET

**\_Connection**

**\_Command**

**\_DataReader**

**DataSet**

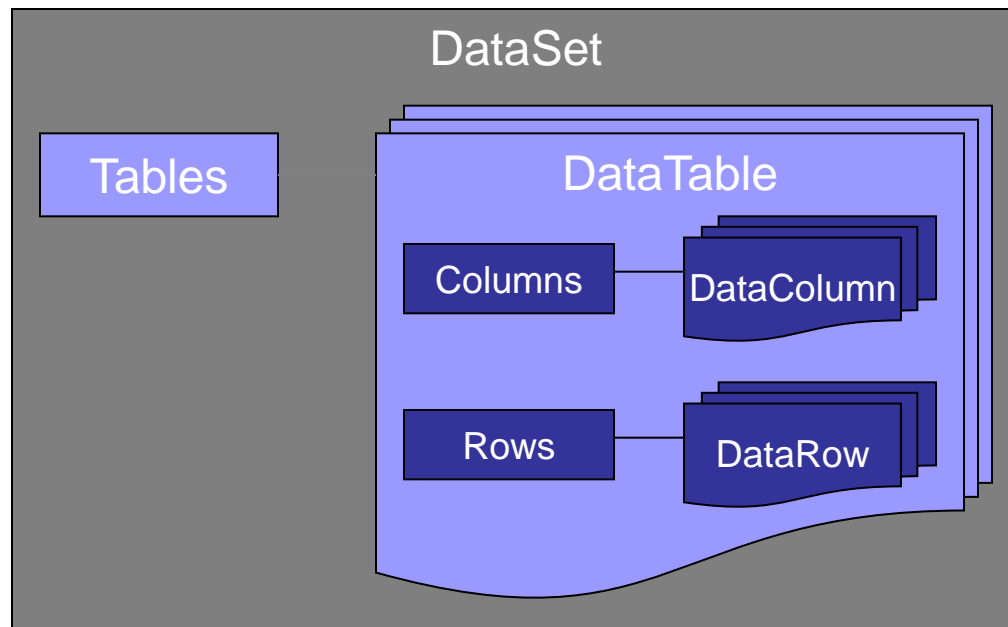
**\_DataAdapter**

# Đối tượng DataSet

- ❑ DataSet là một đối tượng được dùng bởi tất cả Data Provider.
- ❑ Đối tượng DataSet là một thể hiện của dữ liệu trong bộ nhớ.
- ❑ DataSet được thiết kế đặc biệt để giúp quản lý dữ liệu trong bộ nhớ và để hỗ trợ các thao tác không cần kết nối (disconnected) trên dữ liệu.

# Đối tượng Dataset

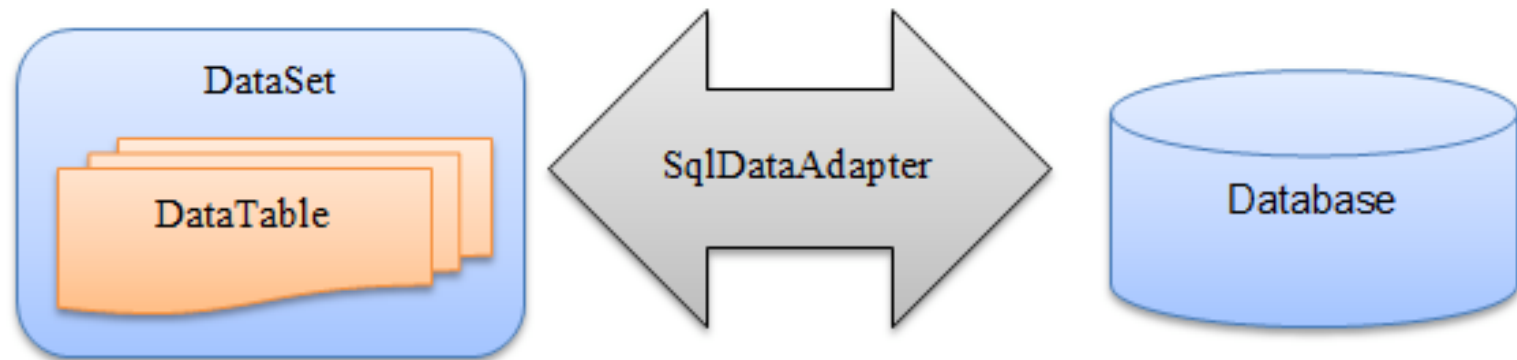
- ❑ Dataset chứa nhiều đối tượng DataTable, bên trong DataTable lại có nhiều column và row, giống như các database table thông thường.





# Đối tượng Dataset

## ❑ Cách kết nối và làm việc



# Đối tượng Dataset

## □ Ví dụ

```
//Khoi tao SqlCommand
cmd = new SqlCommand("Select * From Album",conn);
//Khoi tao SqlDataAdapter
daQLD = new SqlDataAdapter(cmd);
//Khoi tao SqlCommandBuilder
cbQLD = new SqlCommandBuilder(daQLD);
//Du du lieu vao DataSet
daQLD.Fill(dsQLD, "Album");
//Hien thi du lieu tu DataSet len DataGridView
dgAlbum.DataSource = dsQLD;
dgAlbum.DataMember = "Album";
}

private void btnUpdate_Click(object sender, EventArgs e)
{
    //Cap nhat Du lieu tu DataSet len Server
    daQLD.Update(dsQLD, "Album");
}
```

# Các đối tượng của ADO.NET

\_Connection

\_Command

\_DataReader

DataSet

\_DataAdapter

# Đối tượng DataAdapter

- ❑ Data adapter chứa một tham chiếu đến đối tượng connection và mở/đóng kết nối tự động khi đọc và ghi dữ liệu vào database.
- ❑ Data adapter chứa đối tượng command cho những thao tác SELECT, INSERT, UPDATE và DELETE trên dữ liệu.