

# CHƯƠNG 5: ADO.NET

1

*NGUYỄN THỊ LƯƠNG*

KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC ĐÀ LẠT

3/26/2010

## Mục tiêu

2

- Giúp sinh viên nắm được:
  - Kiến thức cơ sở về các lớp của ADO.Net
  - Cách viết một chương trình đơn giản có kết nối và đọc các dòng từ cơ sở dữ liệu.
  - Cách xây dựng ứng dụng Windows Form

## Nội dung

3

- Giới thiệu ADO.Net
- Phân loại các lớp của ADO.Net
  - Các lớp trong nhóm kết nối
    - ✦ Connection, Command
    - ✦ Parameter, ParameterCollection
    - ✦ DataReader, DataAdapter
    - ✦ ...
- Các lớp trong nhóm không kết nối
  - ✦ DataSet, DataTable, DataView
  - ✦ DataColumn, DataRow
  - ✦ DataRelation, Constraint

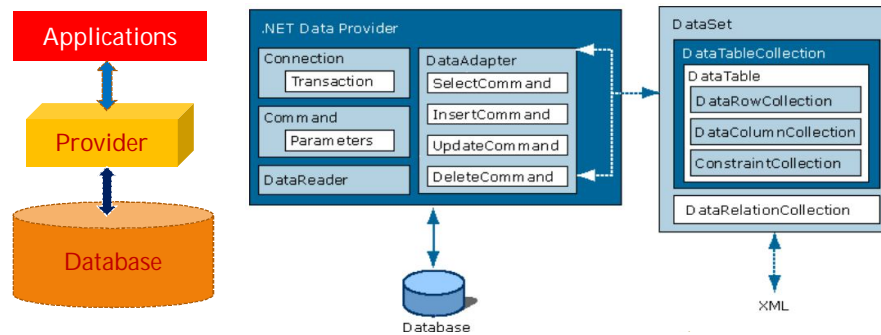
## Giới thiệu ADO.Net

4

- **ADO** (ActiveX Data Objects) là một thư viện của các thành phần COM, bao gồm các thành phần:
  - Connection, Command, Recordset, và Field.
- **ADO.NET** chứa 2 không gian tên cơ sở dữ liệu:
  - Một cho SQL Server.
  - Một cho các cơ sở dữ liệu được trình bày thông qua một giao diện OLE DB.

## Giới thiệu ADO.Net (tt)

5



- Data Provider (hay Driver – trình điều khiển) là một thuật ngữ ám chỉ một thư viện nhị phân - được viết dưới dạng các hàm API (giao diện lập trình ứng dụng).

## Phân loại các lớp của ADO.Net

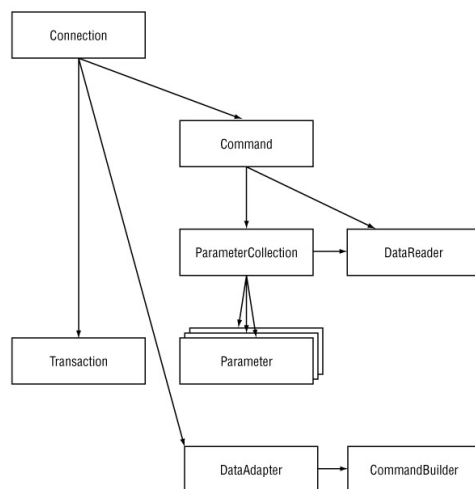
6

- Các lớp kết nối (**managed provider**)
  - Tạo kết nối trực tiếp
  - Đồng bộ hóa dữ liệu được lưu trên máy cục bộ với csdl.
  - Đọc các dòng dữ liệu từ cơ sở dữ liệu theo một hướng.
  - Sử dụng tùy thuộc vào loại cơ sở dữ liệu muốn kết nối tới.
- Các lớp không kết nối (**generic data**)
  - Lưu trữ bản sao thông tin lấy được từ cơ sở dữ liệu.
  - Không phụ thuộc vào loại cơ sở dữ liệu mà bạn dùng, nghĩa là các lớp này được dùng chung cho dữ liệu được lấy từ cơ sở dữ liệu SQL Server, Access hay Oracle...
  - Các lớp này biểu diễn thông tin lấy từ cơ sở dữ liệu ở định dạng XML.

## Nhóm lớp kết nối

(7)

Managed Provider Objects



## Nhóm lớp kết nối

(8)

- Kết nối tới cơ sở dữ liệu (**SqlConnection**)
- Sử dụng **SqlCommand**

## Kết nối tới cơ sở dữ liệu

9

### • Namespace sử dụng

#### ○ System.Data

- ✦ Các lớp truy xuất dữ liệu chung

```
using System.Data;
```

#### ○ System.Data.OleDb

- ✦ Các lớp của OLE DB provider

```
using System.Data.OleDb;
```

#### ○ System.Data.SqlClient

- ✦ Các lớp của SQL Server provider

```
using System.Data.SqlClient;
```

## Kết nối tới cơ sở dữ liệu (tt)

10

### • Ví dụ mô tả, cách tạo kết nối đến cơ sở dữ liệu NorthWind

```
//tao chuỗi kết nối tới NorthWind
string connectionString =
    "server=localhost; database=NorthWind; uid=sa; pwd=sa";

//Tạo đối tượng kết nối
SqlConnection mySqlConnection = new SqlConnection(connectionString);

//Mở kết nối
mySqlConnection.Open();

//SQL truy vấn, thủ tục,...

//Đóng kết nối
mySqlConnection.Close();
```

Nhóm câu lệnh  
xử lý

## Sử dụng SqlCommand

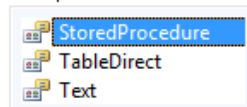
11

- Đối tượng lớp `SqlCommand` dùng chuỗi lệnh SQL để truy xuất dữ liệu hay sử dụng Store Procedure.
- Ví dụ tạo 1 đối tượng thuộc lớp `SqlCommand`

```
SqlConnection mySqlConnection = new SqlConnection(
    "server=localhost;database=Northwind; integrated security=true");

string sql = "SELECT CustomerID, CompanyName, Country "
    + "FROM Customers";
SqlCommand mySqlCommand = new SqlCommand(sql, mySqlConnection);
```

- **CommandType**: thuộc tính chỉ loại của Command  
`CommandType.`



`CommandType.CommandType.StoredProcedure`  
The name of a stored procedure.

## Sử dụng SqlCommand (tt)

12

- Sử dụng CommandType: **Text** (default)

```
//Tao doi tuong thi hanh cau lenh sql
string strsql = "select CategoryID,CategoryName,Description
    from Categories";
```

```
SqlCommand mySqlCommand = mySqlConnection.CreateCommand();
mySqlCommand.CommandText = strsql;
```

```
SqlCommand mySqlCommand = new SqlCommand(strsql, mySqlConnection);
```

## Sử dụng SqlCommand (tt)

13

- Sử dụng CommandType: **StoredProcedure**

- Ví dụ tạo thủ tục thêm dữ liệu vào bảng Categories

```
CREATE PROCEDURE MyInSertCategories
    ( @CategoryName nvarchar(15),
      @Description ntext
    )
AS
Begin
    Insert into Categories(CategoryName, Description)
    values( @CategoryName, @Description)
end
```

## Sử dụng SqlCommand (tt)

14

- Sử dụng CommandType: **StoredProcedure**

```
SqlCommand mySqlCommand =
    new SqlCommand("MyInsertCategories", mySqlConnection);
mySqlCommand.CommandType = CommandType.StoredProcedure;

//Thêm các tham số cho thủ tục MyInsertCategories

SqlParameter paramCategoryName =
    new SqlParameter("@CategoryName", SqlDbType.NVarChar, 15);
paramCategoryName.Value = "Car";
mySqlCommand.Parameters.Add(paramCategoryName);

SqlParameter paramDescription =
    new SqlParameter("@Description", SqlDbType.NText);
paramDescription.Value = "Car for ....";
mySqlCommand.Parameters.Add(paramDescription);
```

## Sử dụng SqlCommand (tt)

15

- Sử dụng CommandType: **TableDirect**

```
string connectionString = "Provider=SQLOLEDB.1;User ID=sa;
    Password=sa;Initial Catalog=NorthWind;Data Source=(local)";
OleDbConnection myOleDbConnection=new OleDbConnection(connectionString);

OleDbCommand myOleDbCommand =
    new OleDbCommand("Categories", myOleDbConnection);
myOleDbCommand.CommandType = CommandType.TableDirect;
```

## Sử dụng SqlCommand (tt)

METHOD	RETURN TYPE	DESCRIPTION
ExecuteReader()	SqlDataReader	-Được dùng để thực thi các lệnh SELECT, TableDirect hay lời gọi thủ tục có trả về một tập dữ liệu. -Kết quả trả về này được lưu trong một đối tượng DataReader.
ExecuteScalar()	object	-Dùng để thực thi các lệnh SELECT chỉ trả về một giá trị (những giá trị khác sẽ bị bỏ qua). -Kết quả trả về được lưu trong một object.
ExecuteNonQuery()	int	-Được dùng để thực thi các lệnh SQL không trả về tập kết quả như INSERT, UPDATE, DELETE, các lệnh DDL, lời gọi thủ tục không trả về dữ liệu. -Giá trị trả về của hàm này chính là số dòng bị ảnh hưởng khi thực thi truy vấn.



## Sử dụng SqlCommand (tt)

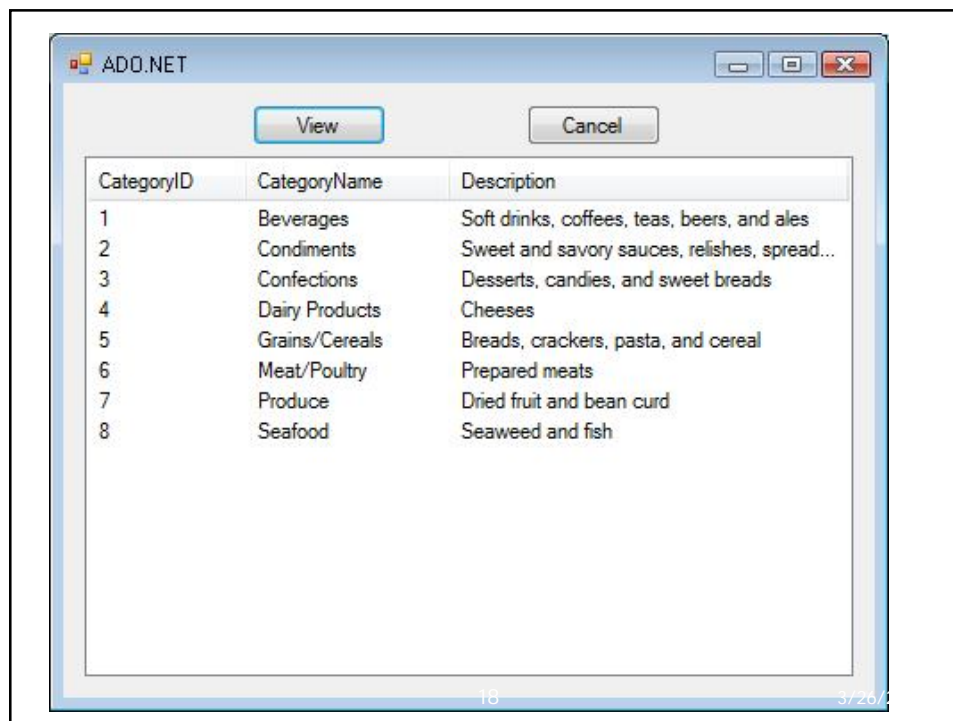
17

- Phương thức **ExecuteReader()**

- **SqlDataReader**: cách đơn giản nhất và nhanh nhất để chọn một vài dữ liệu từ một nguồn cơ sở dữ liệu, nhưng có ít tính năng.

```
//Tạo đối tượng thi hành câu lệnh sql
mySqlConnection.Open();
SqlDataReader mySqlDataReader = mySqlCommand.ExecuteReader();
ListViewItem lvitem;
this.lvCategories.Items.Clear();
while (mySqlDataReader.Read())
{
    lvitem = new ListViewItem(mySqlDataReader[0].ToString());
    lvitem.SubItems.Add(mySqlDataReader[1].ToString());
    lvitem.SubItems.Add(mySqlDataReader[2].ToString());
    lvCategories.Items.Add(lvitem);
}
mySqlDataReader.Close();
mySqlConnection.Close();
```

lvCategories: ListView



### Sử dụng SqlCommand (tt)

- Phương thức **ExecuteNonQuery()**: (gọi thủ tục)

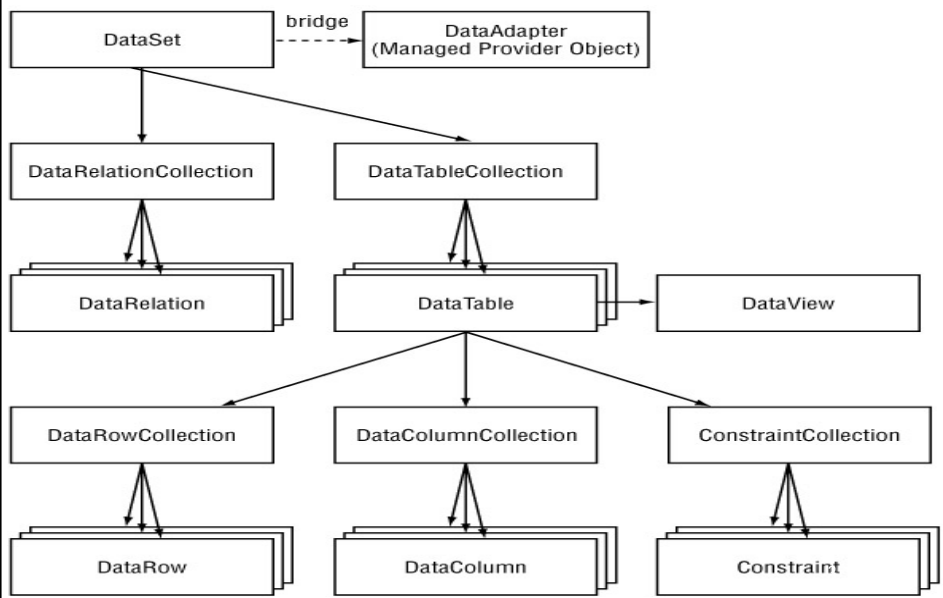
```
CREATE PROCEDURE MyInsertCategories
    ( @CategoryName nvarchar(15),
      @Description ntext
    )
AS
Begin
    Insert into Categories(CategoryName, Description)
    values( @CategoryName, @Description)
end

SqlCommand mySqlCommand =
    new SqlCommand("MyInsertCategories", mySqlConnection);
mySqlCommand.CommandType = CommandType.StoredProcedure;

SqlParameter paramCategoryName =
    new SqlParameter("@CategoryName", SqlDbType.NVarChar, 15);
mySqlConnection.Open();
int numRoweffect=mySqlCommand.ExecuteNonQuery();
if (numRoweffect == 0)
    MessageBox.Show("Them khong thanh cong");
else
    MessageBox.Show("Them thanh cong", "Thong bao");
mySqlConnection.Close();
```

### Các lớp không kết nối

Generic Data Set Objects



## DataSet

21

- DataSet
  - DataTable
    - ✦ DataRow
    - ✦ DataColumn
    - ✦ UniqueConstraint
    - ✦ ForeignKeyConstraint
  - DataRelation
  - DefaultView

## Sử dụng DataSet

22

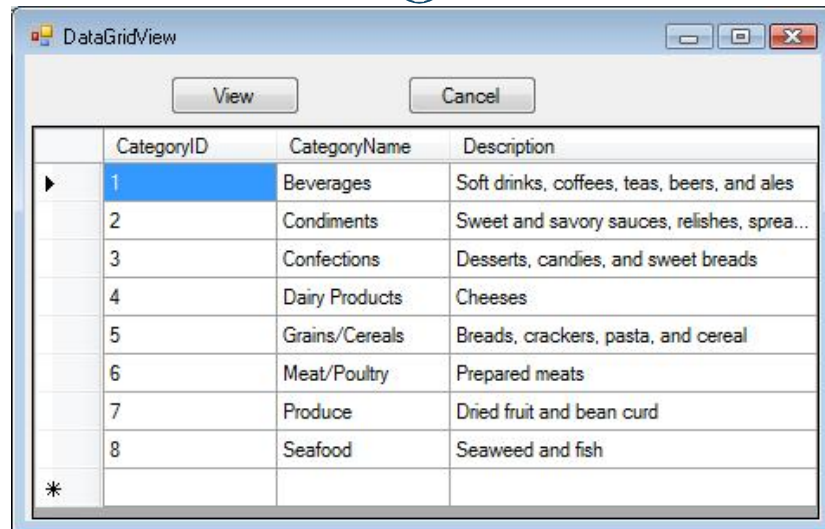
```
//Tao doi tuong thi hanh cau lenh sql
string strSql = "select CategoryID,CategoryName,Description
                from Categories";
SqlCommand mySqlCommand =new SqlCommand(strSql,mySqlConnection);

SqlDataAdapter mySqlDataAdapter = new
    SqlDataAdapter(strSql,mySqlConnection);
DataSet dtset = new DataSet();
mySqlConnection.Open();
mySqlDataAdapter.Fill(dtset, "tblCategories");
mySqlConnection.Close();

DataTable dttable = dtset.Tables["tblCategories"];
foreach (DataRow dr in dttable.Rows)
{
    this.lvCategories.Items.Add(new ListViewItem(new string[]
    {dr[0].ToString(),dr[1].ToString(), dr[2].ToString()}));
}
lvCategories: ListView
```

## DataGridView

23



## DataGridView

24

```
private void btnView_Click(object sender, EventArgs e)
{
    string strcon = "Integrated security=true; server=(local); "
        + " database=NorthWind";
    SqlConnection mysqlconnection = new SqlConnection(strcon1);
    try
    {
        mysqlconnection.Open();
        string strcommand = "select CategoryID,CategoryName,Description "
            + "from Categories";
        SqlDataAdapter mydtAdapter =
            new SqlDataAdapter(strcommand, mysqlconnection);
        DataSet ds = new DataSet();
        mydtAdapter.Fill(ds);
        DataTable dt = ds.Tables[0];
        dataGridView1.DataSource = dt;
        mysqlconnection.Close();
    }
    catch
    {
        MessageBox.Show("Loi ket noi");
    }
}
```

## KẾT THÚC CHƯƠNG

25

# Q&A

3/26/2010