

Seminar Hệ Quản trị CSDL – Triển khai ứng dụng trên .NET

# ADO.NET

Lương Vĩ Minh

Bộ môn Hệ thống thông tin ([www.is.fit.hcmuns.edu.vn](http://www.is.fit.hcmuns.edu.vn))

Khoa Công nghệ thông tin ([www.fit.hcmuns.edu.vn](http://www.fit.hcmuns.edu.vn))

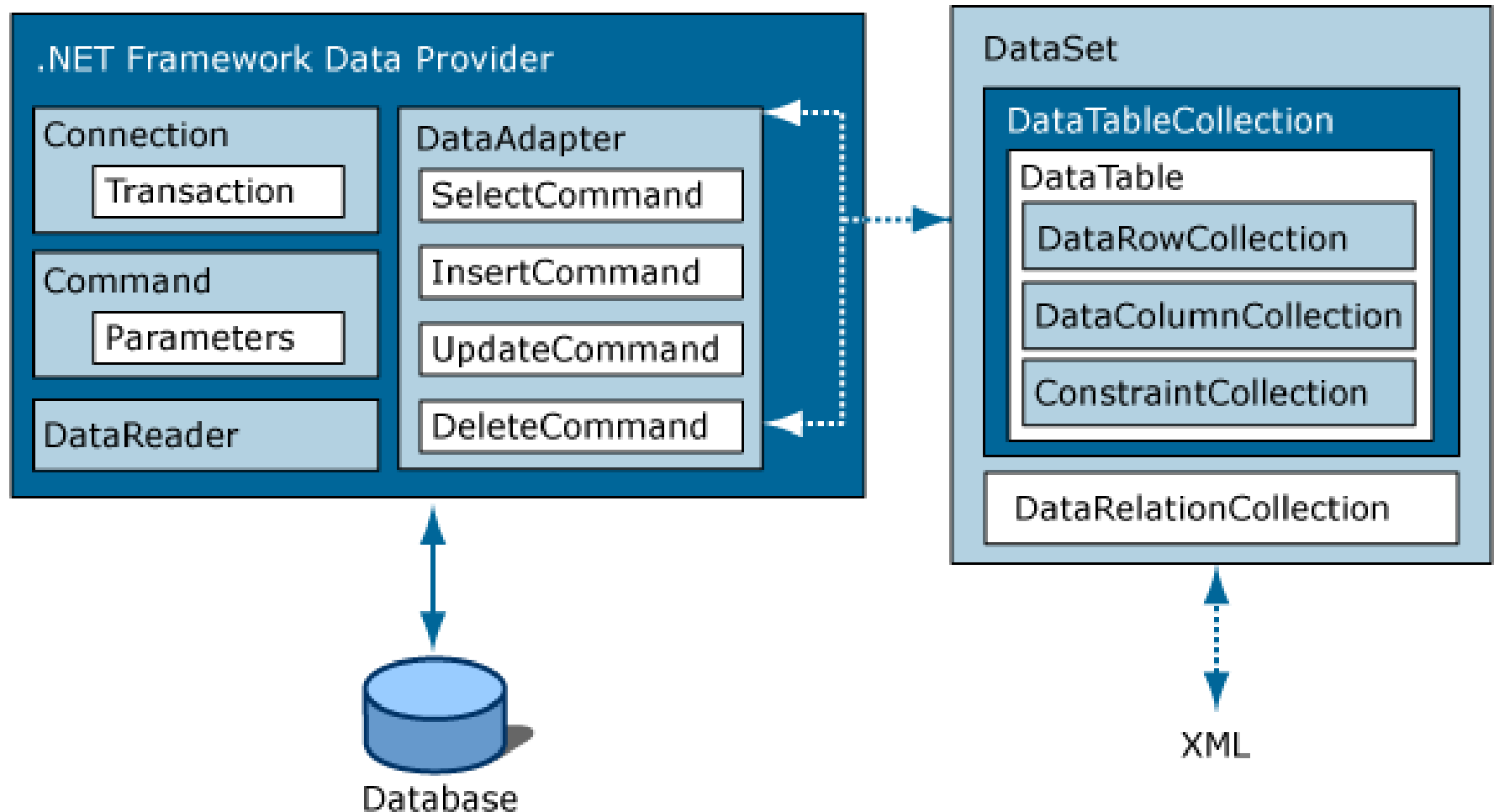
# Nội dung

- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction

# Nội dung

- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction

# Kiến trúc của ADO.NET



# ADO.NET Namespace

- System.Data  
System.Data.Common
- System.Data.SqlClient  
System.Data.OleDb
- System.Data.SqlTypes
- System.XML  
System.XML.Schema

# Sử dụng ADO.NET cho Ứng dụng

- Nếu ứng dụng .NET cần truy xuất dữ liệu → Phải khai báo namespace ADO.NET tương ứng với dữ liệu cho ứng dụng
- Đối với dữ liệu OLE
  - `using System.Data;`
  - `using System.Data.OleDb;`
- Đối với dữ liệu SQL Server
  - `using System.Data;`
  - `using System.Data.SqlClient;`

# SQL Namespace

**using** System.Data.SqlClient;

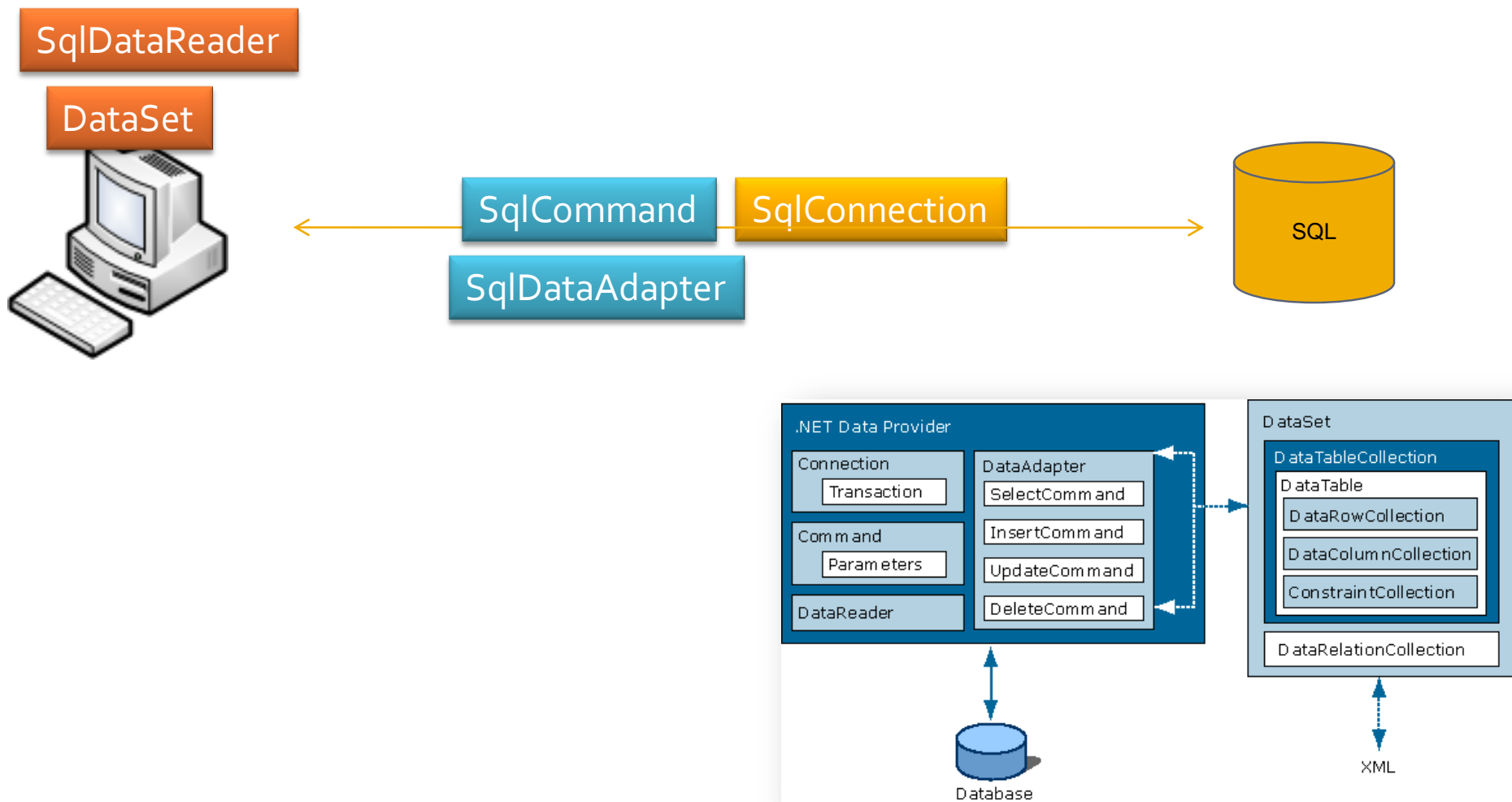
- SqlConnection
- SqlCommand
- SqlDataReader
- SqlDataAdapter
- SqlParameter
- SqlParameterCollection
- SqlError
- SqlErrorCollection
- SqlException
- SqlTransaction
- SqlDbType

# Nội dung

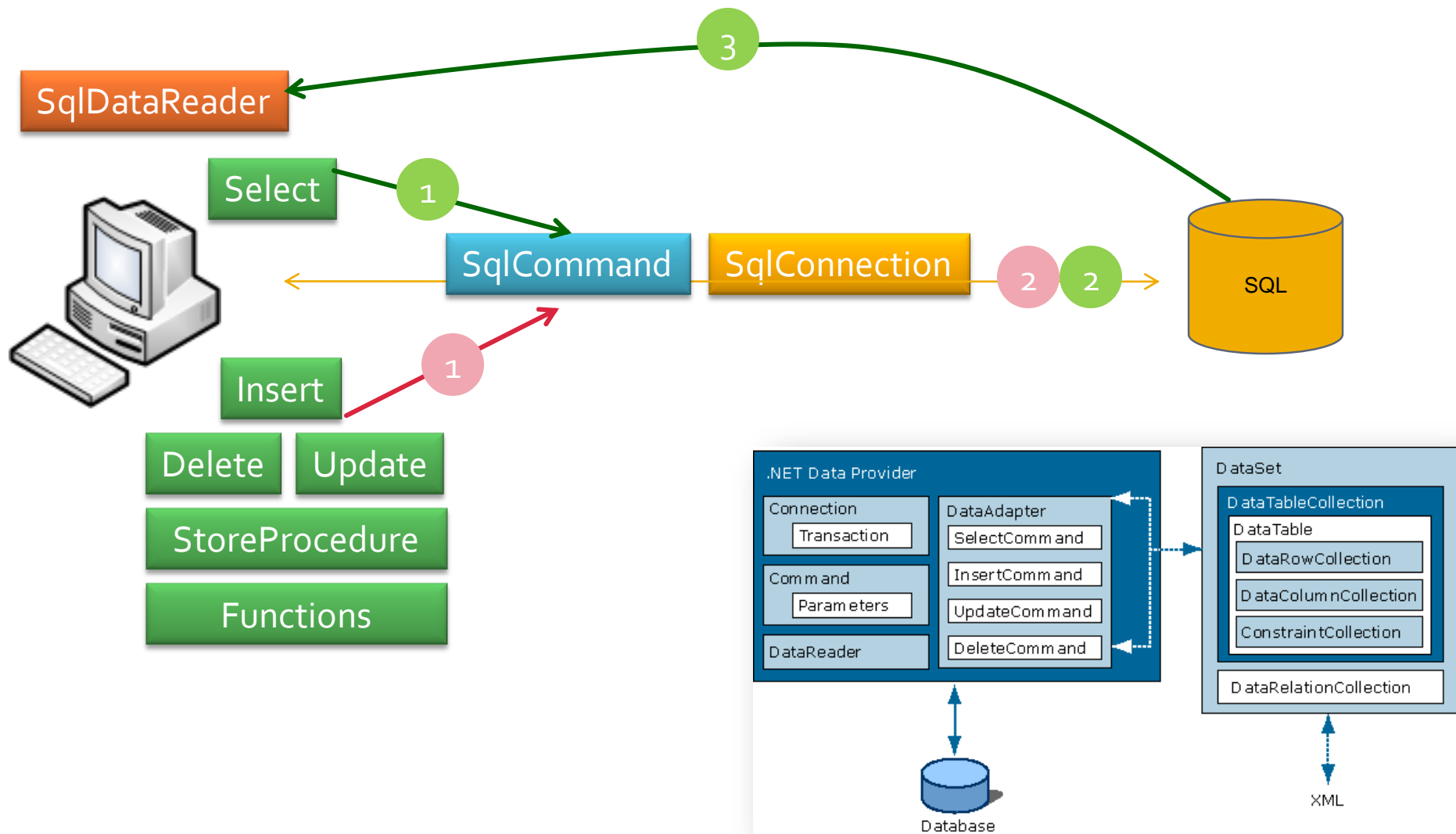
- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction



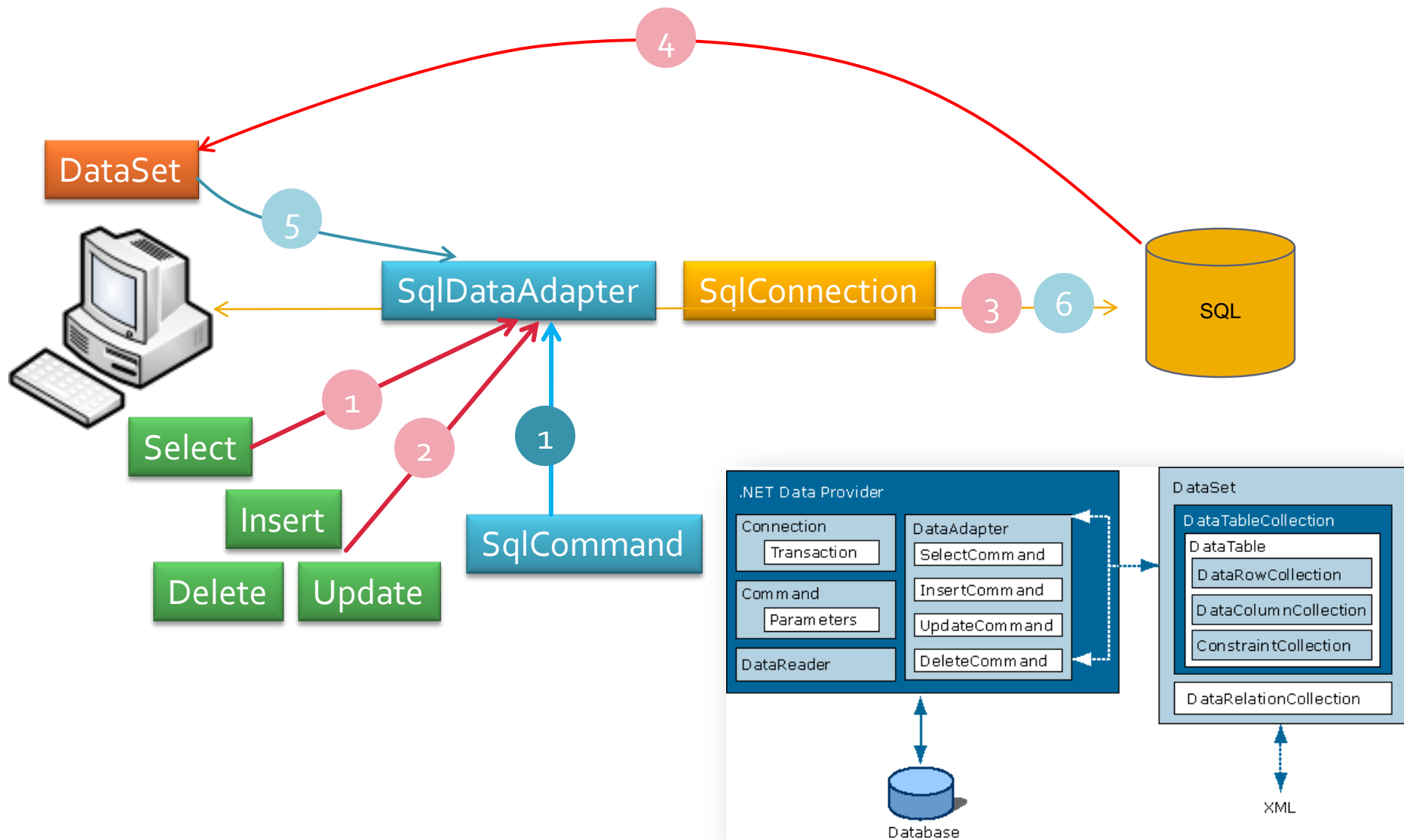
# Mô hình sử dụng ADO.NET với Phần mềm



# Mô hình sử dụng ADO.NET với Phần mềm



# Mô hình sử dụng ADO.NET với Phần mềm



# Nội dung

- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- **Đối tượng SqlConnection**
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction

# Đối tượng SqlConnection

- Dùng để tạo kết nối đến các CSDL Sql Server
- Thuộc tính ConnectionString : Lưu chuỗi kết nối đến hệ quản trị CSDL SQL Server
- Kết nối được mở thông qua 2 cách
  - Qua phương thức Open()
  - Qua việc thực thi đối tượng SqlDataAdapter
- Ngắt kết nối thông qua phương thức Close()
- Quản lý các giao tác của kết nối này

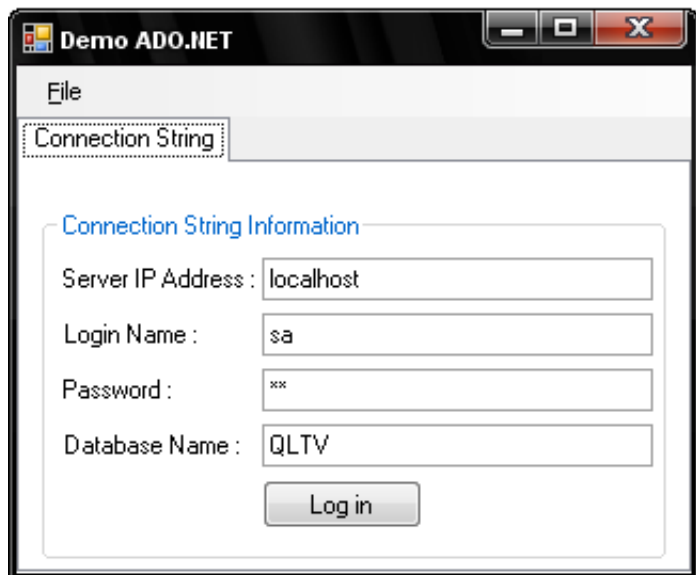
# Đối tượng SqlConnection (Connection String)

- `using System.Data.SqlClient;`

```
string sConnectionString =  
    "Initial Catalog=Northwind;  
    Data Source=localhost;  
    user=sa;  
    password=sa;";
```

- `SqlConnection cnn = new SqlConnection();`
- `cnn.ConnectionString = sConnectionString;`

# Đối tượng SqlConnection - Demo



Demo ADO.NET

File

Connection String

Connection String Information

Server IP Address : localhost

Login Name : sa

Password : xx

Database Name : QLTV

Log in

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DemoADO
{
    public partial class DemoADO : Form
    {
        private SqlConnection _cnn;
        public DemoADO()
        {
            InitializeComponent();
        }

        private void buttonLogin_Click(object sender, EventArgs e)
        {
            string sConnectionString;
            sConnectionString = "Initial Catalog=" + textBoxDBName.Text + "; ";
            sConnectionString += "Data Source=" + textBoxServerIP.Text + "; ";
            sConnectionString += "user=" + textBoxLoginName.Text + "; ";
            sConnectionString += "password=" + textBoxPassword.Text + "; ";
            // sConnectionString += "Security=SSPI;";

            _cnn = new SqlConnection(sConnectionString);
            _cnn.Open();
        }
    }
}
```

# Nội dung

- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction



# Đối tượng SqlCommand

- Truyền thông tin của một câu truy vấn cho đối tượng SqlConnection
- Hỗ trợ tham số vào, tham số ra, và giá trị trả về
- Kết quả trả về được lưu trữ dưới dạng luồng thông qua 2 đối tượng
  - DataReader
  - DataSet thông qua một đối tượng SqlDataAdapter

# Đối tượng SqlCommand

## Các hàm khởi tạo

```
new SqlCommand()
```

```
new SqlCommand(cmdText)
```

```
new SqlCommand(cmdText, connection)
```

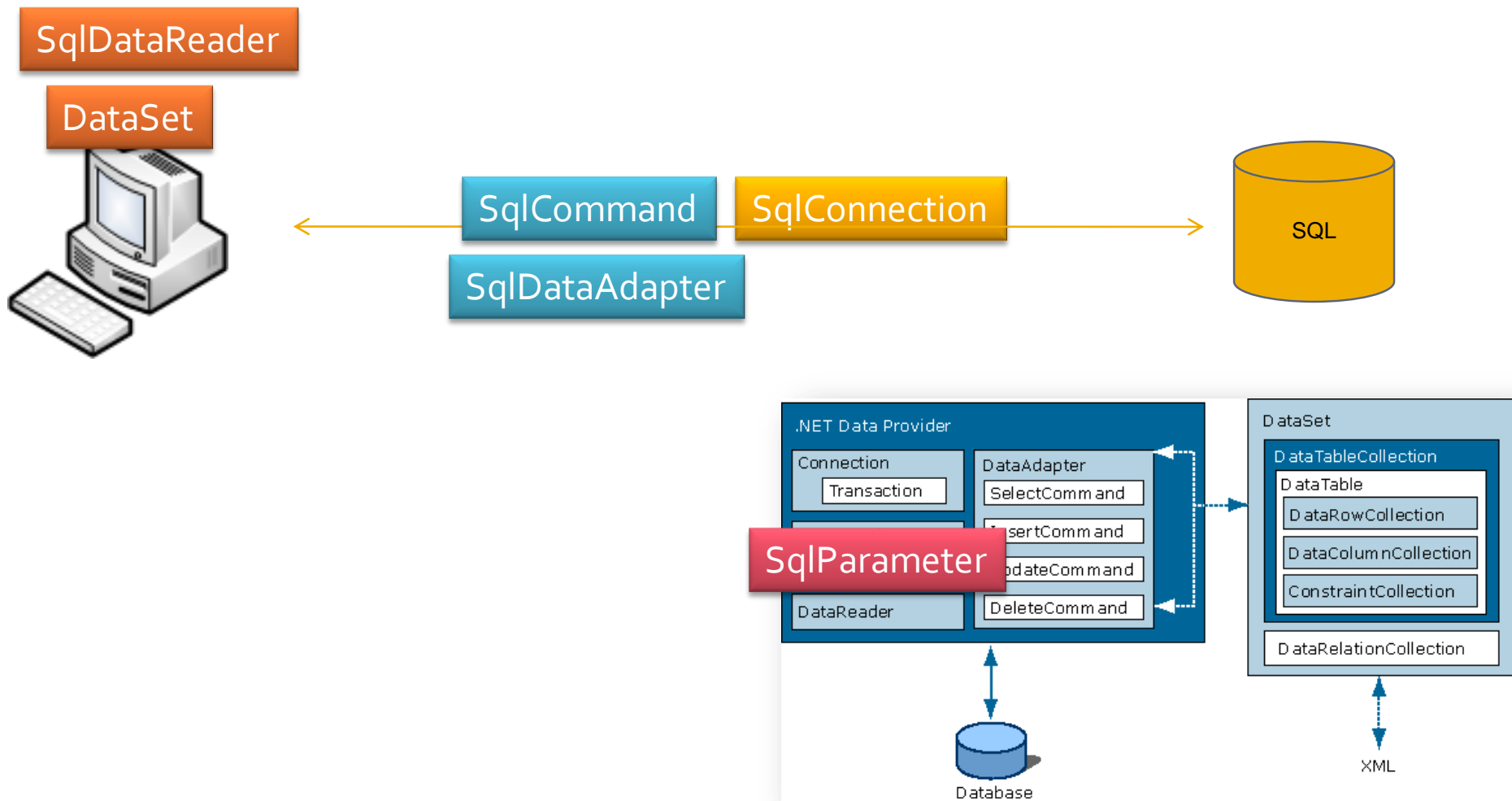
```
new SqlCommand(cmdText, connection, transaction)
```

Thuộc tính	Ý nghĩa
<b>.Connection</b>	Trỏ đến đối tượng kết nối
<b>.CommandType</b>	<code>CommandType.Text</code> (giá trị mặc định) <code>CommandType.StoredProcedure</code> <code>CommandType.TableDirect</code>
<b>.CommandText</b>	Câu truy vấn SQL hoặc tên Store, tên Bảng
<b>.CommandTimeout</b>	Thời gian chờ đợi thực thi 1 câu sql
<b>.Parameters</b>	Danh sách các tham số truyền vào

# Các phương thức - SqlCommand

Tên hàm	Ý nghĩa
<code>.ExecuteReader()</code>	Trả về một DataReader
<code>.ExecuteNonQuery()</code>	Trả về số lượng dòng bị ảnh hưởng trên CSDL
<code>.ExecuteScalar()</code>	Trả về 1 giá trị đầu tiên (VD: giá trị tính tổng)
<code>.ExecuteXMLReader()</code>	Trả về 1 XMLReader

# SqlCommand - SqlParameter

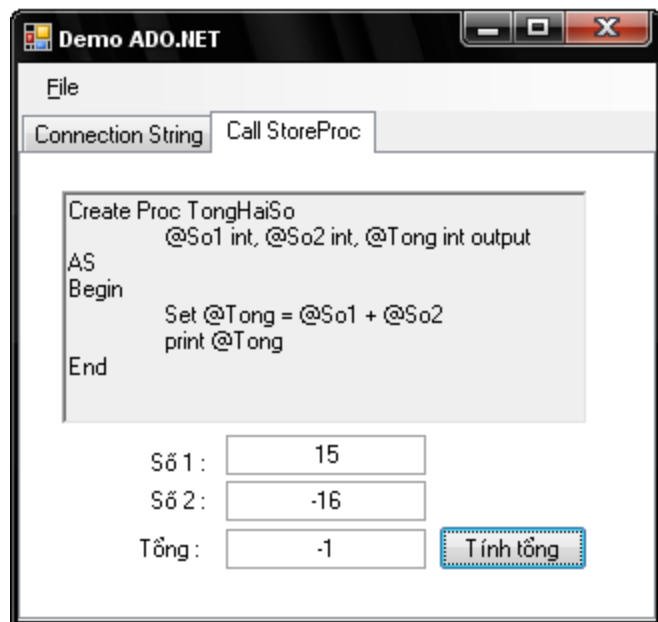


# SqlParameter

- Đối tượng tham số truyền vào cho đối tượng SqlCommand
- Có các thuộc tính sau :

Thuộc tính	Ý nghĩa
ParameterName	Tên tham số
SqlDbType	Kiểu dữ liệu của tham số tương ứng với kiểu dữ liệu của SqlServer
Direction	Input, Output, InputOutput, ReturnValue, ...
Size	Kích thước tối đa của dữ liệu
Value	Giá trị của tham số (input / Output)

# SqlCommand + SqlParameter : Demo – Gọi stored procedure



```
private void buttonTinhTong_Click(object sender, EventArgs e)
{
    _cnn.Open();

    // Tao doi tuong SqlCommand
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "TongHaiSo";
    cmd.Connection = _cnn;

    // Tao tham so cho SqlCommand
    SqlParameter para;
    para = new SqlParameter("So1", SqlDbType.Int, 4);
    para.Direction = ParameterDirection.Input;
    para.Value = int.Parse(textBoxSo1.Text);
    cmd.Parameters.Add(para);
    para = new SqlParameter("So2", SqlDbType.Int, 4);
    para.Direction = ParameterDirection.Input;
    para.Value = int.Parse(textBoxSo2.Text);
    cmd.Parameters.Add(para);
    para = new SqlParameter("Tong", SqlDbType.Int, 4);
    para.Direction = ParameterDirection.Output;
    cmd.Parameters.Add(para);

    // Thuc thi viec gọi Storeproc
    cmd.ExecuteNonQuery();

    // Xuat ket qua
    textBoxTong.Text = cmd.Parameters["Tong"].Value.ToString();

    // Dong ket noi
    _cnn.Close();
}
```

# Nội dung

- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- **Đối tượng SqlDataReader**
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction

# Đối tượng SqlDataReader

- Có tốc độ xử lý dữ liệu nhanh
- Chỉ lưu lại 1 record kết quả trong bộ nhớ với mỗi lần truy xuất (buffer mạng)
- Không thực hiện các thao tác phức tạp trên DataReader (Sắp xếp, bỏ qua record, ..)
- Chỉ đọc, không thay đổi dữ liệu



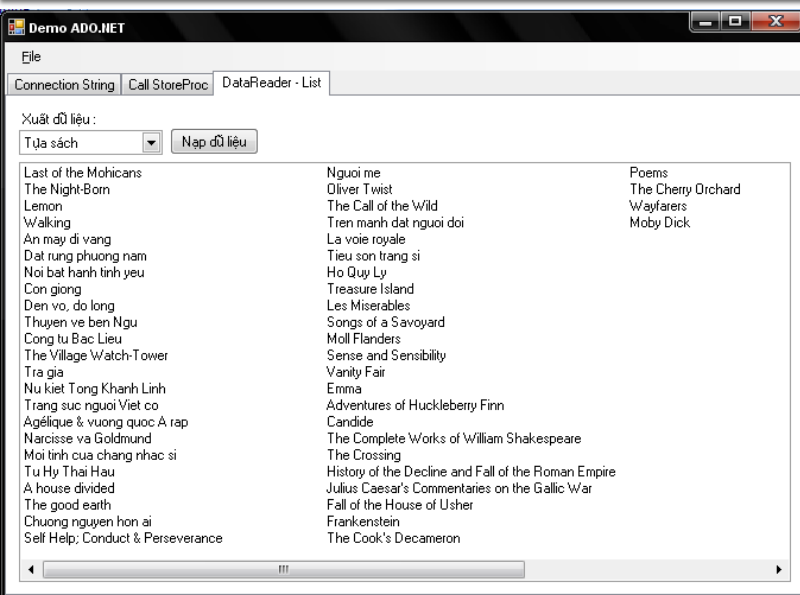
# Đối tượng SqlDataReader

- Một số thuộc tính & hàm :

Thuộc tính	Ý nghĩa
<b>HasRows</b>	Trả về xem DataReader có đọc được dữ liệu nào không.
<b>FieldCount</b>	Trả về số lượng thuộc tính trong dòng hiện tại (đang đọc)
<b>Item(int/string)</b>	Trả về giá trị của thuộc tính đang yêu cầu

Hàm	Ý nghĩa
<b>Read()</b>	Đọc record dữ liệu kế tiếp
<b>IsDBNull(i)</b>	Kiểm tra xem giá trị cột i có bị null không
<b>NextResult()</b>	Chuyển qua KQ kế tiếp cho các câu truy vấn cụm

# SqlDataReader + SqlCommand : Demo – Load dữ liệu vào list



```
private void buttonNapDuLieu_Click(object sender, EventArgs e)
{
    listViewDS.Items.Clear();
    string sql;
    if (comboBoxBang.SelectedIndex == 0) // Doc gia
        sql = "Select ho+' '+ten+' '+tenlot as Hoten From Docgia";
    else // Tua sach
        sql = "Select Tuasach From Tuasach";

    _cnn.Open();

    SqlCommand cmd = new SqlCommand(sql, _cnn);
    SqlDataReader reader = cmd.ExecuteReader();

    if (reader.HasRows == true)
    {
        // Doc du lieu
        if (comboBoxBang.SelectedIndex == 0) // Doc gia
        {
            while (reader.Read())
                listViewDS.Items.Add(reader["Hoten"].ToString());
        }
        else // Tua sach
        {
            while (reader.Read())
                listViewDS.Items.Add(reader["Tuasach"].ToString());
        }
    }

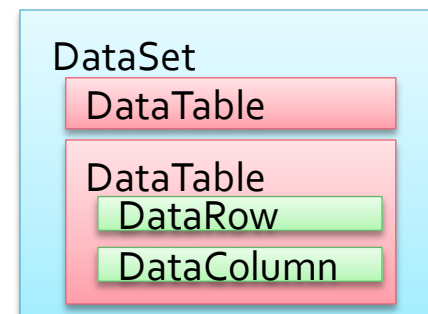
    _cnn.Close();
}
```

# Nội dung

- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction

# Đối tượng DataSet

- Lưu trữ dữ liệu của DB trong bộ nhớ
- Mọi thao tác thay đổi dữ liệu được thực hiện trên DataSet, không làm ảnh hưởng đến DB
- Có thể lấy dữ liệu từ SqlDataAdapter và XML
- Theo vết các thay đổi trên dữ liệu và có thể cập nhật dữ liệu ngược vào DB
- Gồm các đối tượng : DataTable, DataRelationship, Constraint



# Đối tượng DataSet

- Một số thuộc tính và phương thức

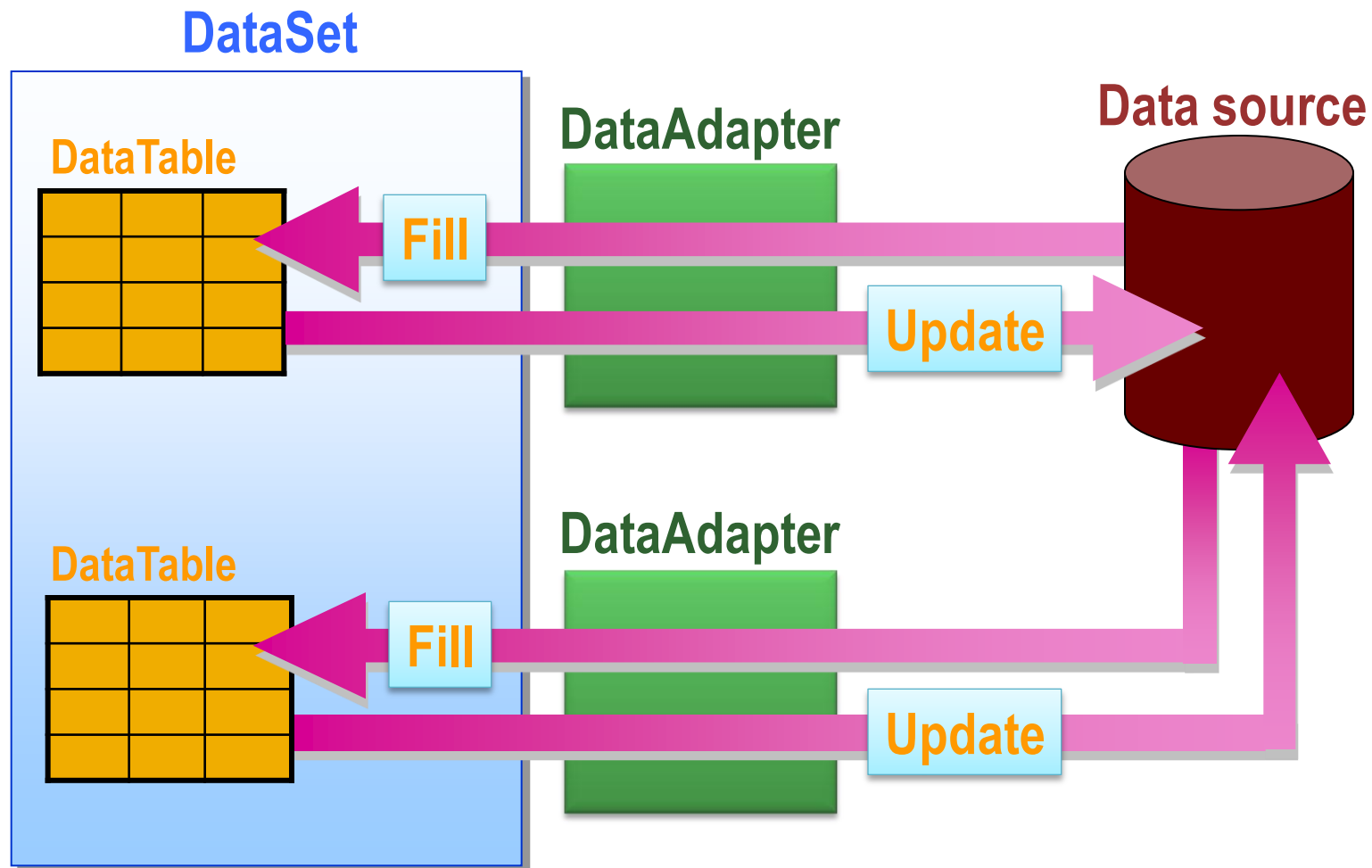
Thuộc tính	Ý nghĩa
<code>DataSetName</code>	
<code>Relations</code>	
<code>Tables</code>	Danh sách các table có trong Dataset

Hàm	Ý nghĩa
<code>GetChange()</code>	Trả về DataSet chứa các thay đổi trên DataSet đang xét
<code>RejectChanges()</code>	
<code>AcceptChanges()</code>	
<code>GetXML()</code> , <code>ReadXML()</code> , <code>WriteXML()</code>	

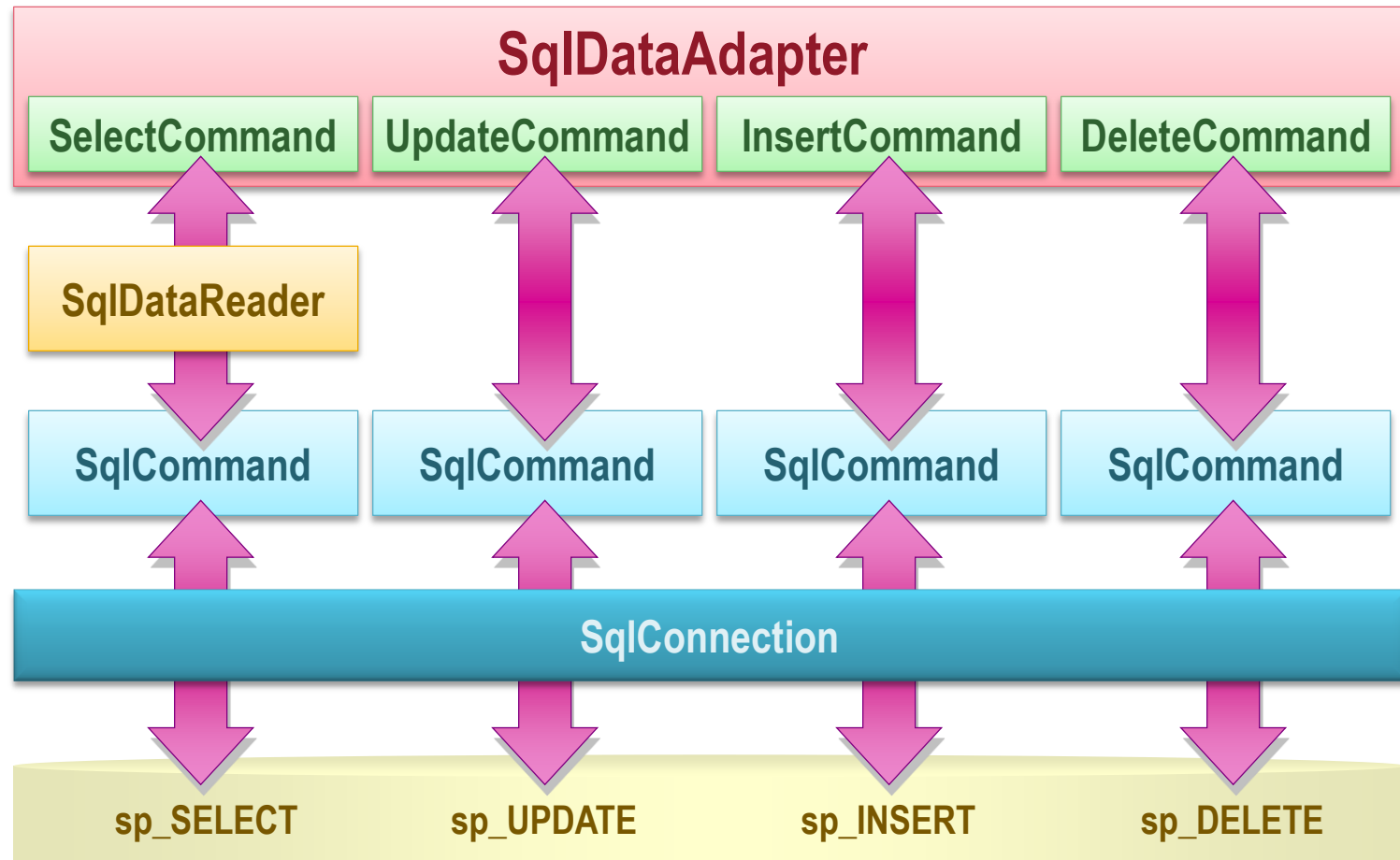
# Đối tượng SqlDataAdapter

- Cung cấp các phương thức và thuộc tính để lấy và lưu dữ liệu giữa DataSet và CSDL
- Sử dụng DataSet để lưu trữ dữ liệu, đồng thời, cho cập nhật dữ liệu ngược lại vào Database

# Đối tượng SqlDataAdapter



# Mô hình đối tượng SqlDataAdapter





# Đối tượng SqlDataAdapter

- Một số thuộc tính và phương thức

Thuộc tính	Ý nghĩa
SelectCommand	
UpdateCommand	
InsertCommand	
DeleteCommand	

Hàm	Ý nghĩa
Fill (Dataset)	Lấy dữ liệu từ DB và trả vào Dataset
FillSchema ()	
Update (...)	Tiến hành cập nhật dữ liệu trên DataSet với DB

# Đối tượng SqlDataAdapter

```
using System.Data.SqlClient;

namespace DemoADO
{
    public partial class DemoADO : Form
    {
        private SqlConnection _cnn;
        private DataSet _ds;
        private SqlDataAdapter _da;

        public DemoADO()
        {
            InitializeComponent();
        }
    }
}
```

```
private void buttonNapDuLieu2_Click(object sender, EventArgs e)
{
    dataGridViewData.DataSource = null;
    string sql;
    if (comboBoxChonBang.SelectedIndex == 0) // Doc gia
        sql = "Select * From Docgia";
    else // Tua sach
        sql = "Select * From Tuasach";

    _cnn.Open();
    _da = new SqlDataAdapter(sql, _cnn);
    SqlCommandBuilder sqlcb = new SqlCommandBuilder(_da);

    _ds = new DataSet();
    _da.Fill(_ds);

    dataGridViewData.DataSource = _ds.Tables[0];
    _cnn.Close();
}

private void buttonUpdateData_Click(object sender, EventArgs e)
{
    if (_ds.HasChanges() == false)
    {
        MessageBox.Show("Nothing Change");
        return;
    }

    DataSet ds = _ds.GetChanges();
    _da.Update(ds);
}
```

Demo ADO.NET

File

Connection String Call StoreProc DataReader - List DataAdapter

Xuất dữ liệu:

Tua sách

	ma_tuasach	TuaSach	tacgia	tomtat
▶	1	123	James Fenimore ...	
	2	The Night-Born	Jack London	
	3	Lemon	Motojirou	
	4	Walking	Henry David Tho...	
	5	An may di vang	Chu Lai	
	6	Dat rung phuong ...	Doan Gioi	
	7	Noi bat hanh tinh...	Hoang Lai Giang	
	8	Con giong	Le Van Thao	
	9	Den vo, do long	Mai Thanh Hai	

# Nội dung

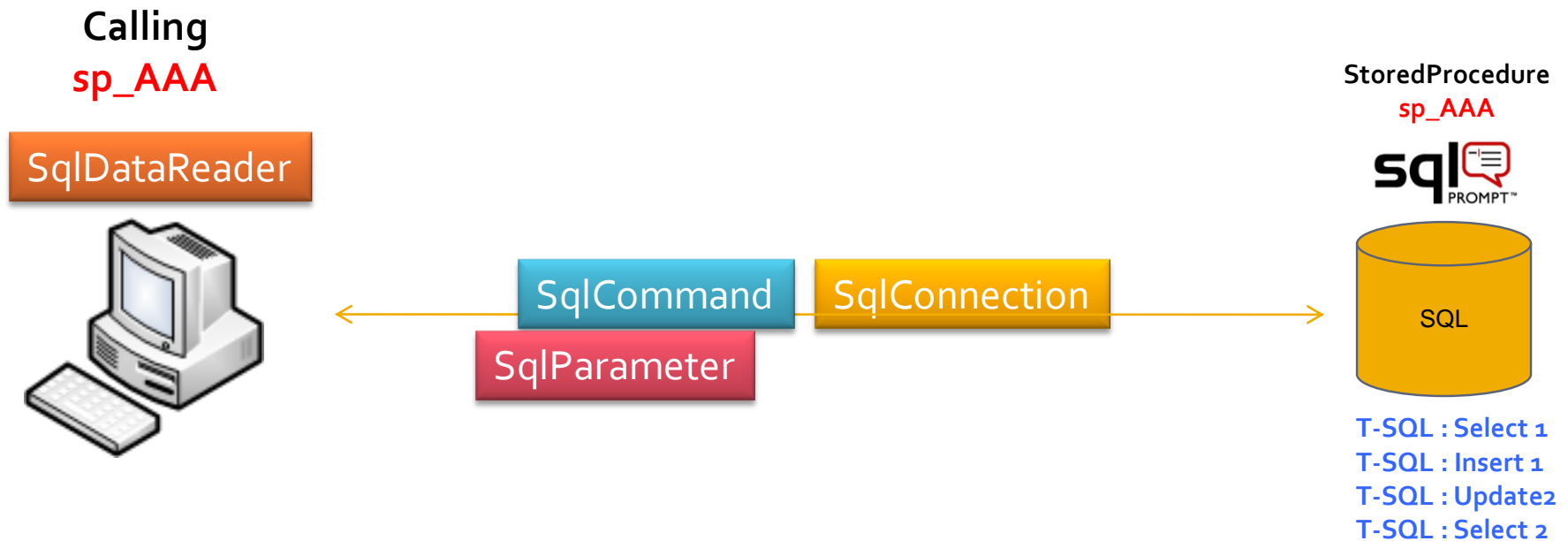
- ADO.NET
- Mô hình sử dụng ADO.NET với Phần mềm
- Đối tượng SqlConnection
- Đối tượng SqlCommand & SqlParameter
- Đối tượng SqlDataReader
- Đối tượng DataSet & SqlDataAdapter
- Thực hiện giao tác với ứng dụng CSDL & Đối tượng SqlTransaction

# Thực hiện giao tác với ứng dụng CSDL

- Có 2 cách để thực hiện 1 giao tác (nhiều lệnh T-SQL) từ phía ứng dụng.
  - **Cách 1** : Viết storeproc ở HQTSQL + Gọi thực hiện store thông qua đối tượng SqlCommand + SqlParameterer
  - **Cách 2** : Quản lý Giao tác ở phía ứng dụng, chỉ gọi thực hiện từng câu truy vấn T-SQL

# Thực hiện giao tác với ứng dụng CSDL

## ■ Cách 1 :



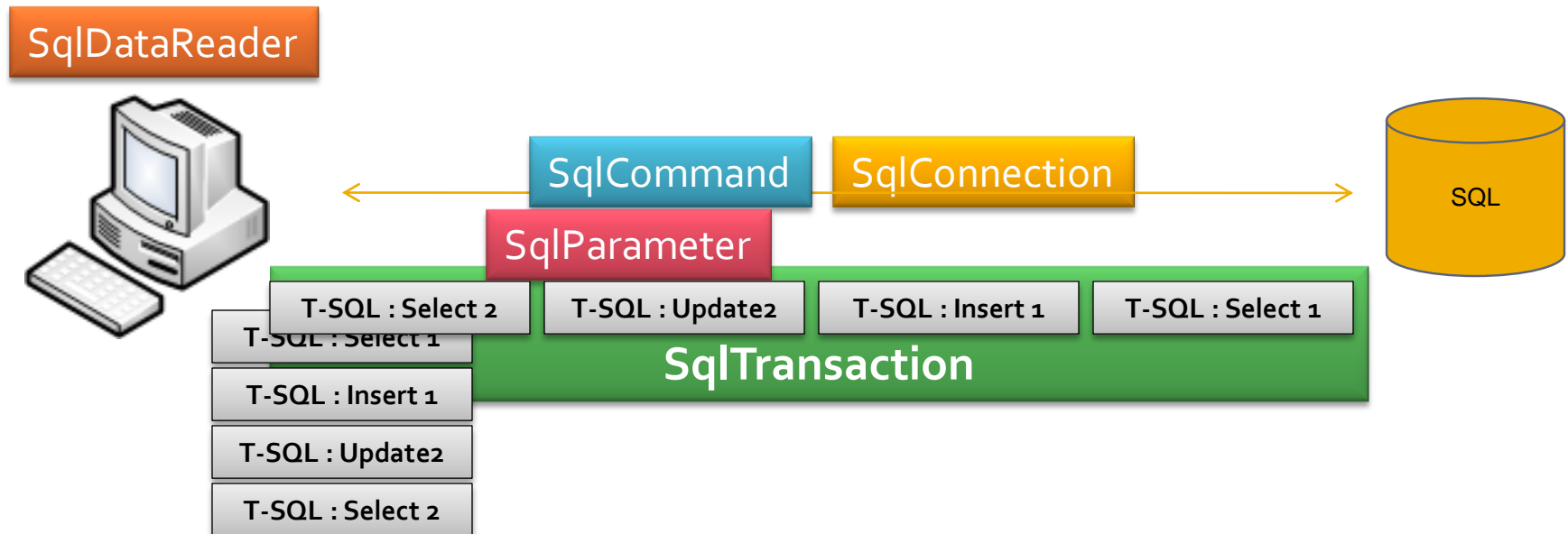
# Thực hiện giao tác với ứng dụng CSDL

## ■ Cách 2 :



# Thực hiện giao tác với ứng dụng CSDL

## ■ Cách 2 :



# Đối tượng SqlDataAdapter

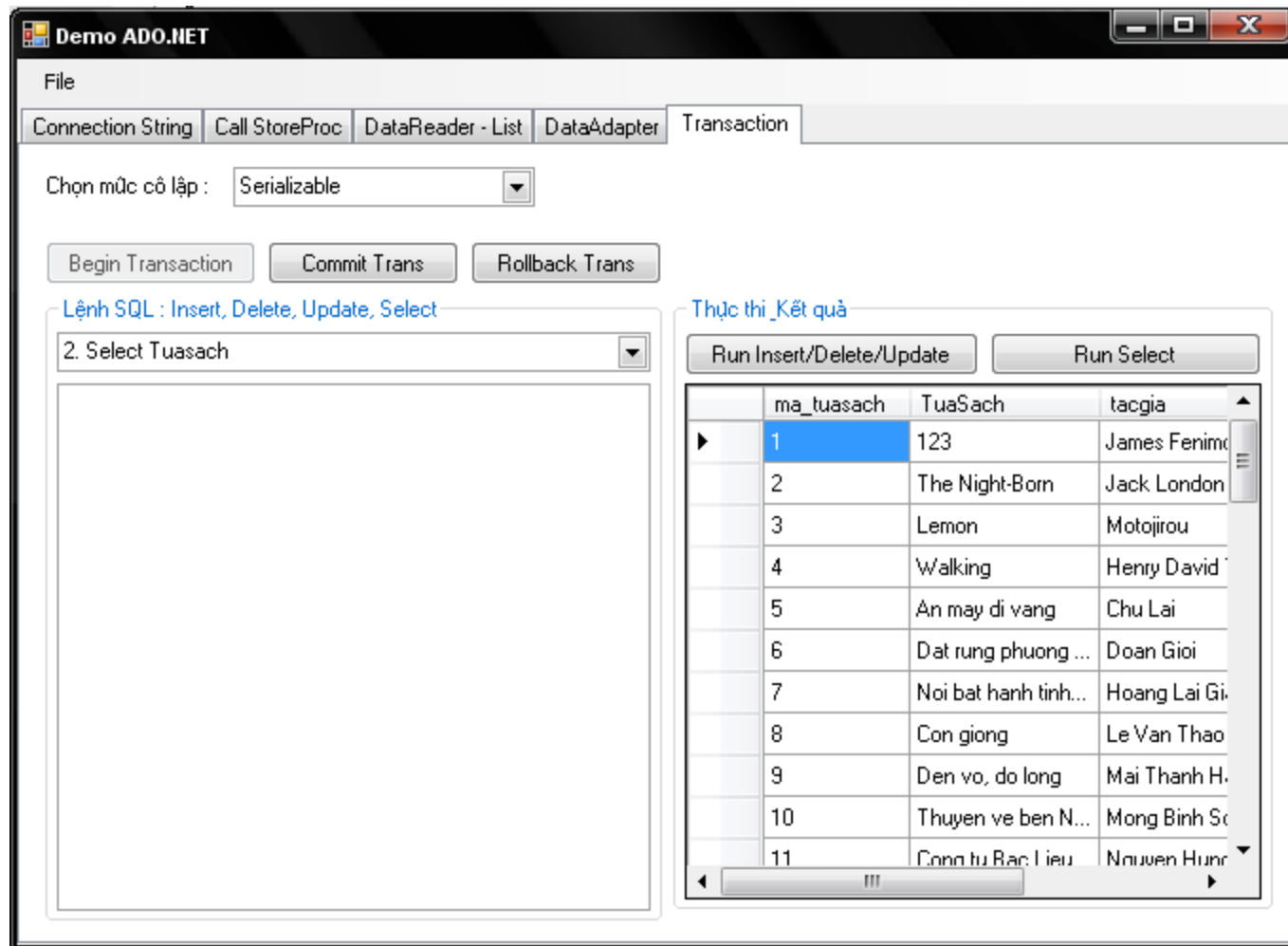
- Một số thuộc tính và phương thức

Thuộc tính	Ý nghĩa
<code>Connection</code>	Trỏ đến đối tượng <code>SqlConnection</code> cần tạo transaction
<code>IsolationLevel</code>	Enum Thiết lập mức cô lập cho Transaction

Hàm	Ý nghĩa
<code>ConObj.BeginTransaction()</code>	Khởi tạo Đối tượng <code>SqlTransaction</code> bằng Đối tượng <code>SqlConnection</code>
<code>Commit(...)</code>	
<code>RollBack(...)</code>	
<code>Save(SavePointName)</code>	Tạo save point để có thể Rollback



# Đối tượng SqlTransaction



# Đối tượng SqlTransaction

```
namespace DemoADO
{
    public partial class DemoADO : Form
    {
        private SqlConnection _cnn;
        private DataSet _ds;
        private SqlDataAdapter _da;
        private SqlTransaction _trans;
```

```
private void buttonBeginTrans_Click(object sender, EventArgs e)
{
    IsolationLevel iso;
    switch (comboBoxMucCoLap.SelectedIndex)
    {
        case 0:
            iso = IsolationLevel.ReadUncommitted;
            break;
        case 1:
            iso = IsolationLevel.ReadCommitted;
            break;
        case 2:
            iso = IsolationLevel.RepeatableRead;
            break;
        case 3 :
            iso = IsolationLevel.Serializable;
            break;
        default :
            iso = IsolationLevel.ReadCommitted;
            break;
    }
    _cnn.Open();
    _trans = _cnn.BeginTransaction(iso);
}
```

# Đối tượng SqlTransaction

```
private void buttonCommitTrans_Click(object sender, EventArgs e)
{
    try
    {
        _trans.Commit();
    }
    catch
    {
    }

    _cnn.Close();
}
```

```
private void buttonRollbackTrans_Click(object sender, EventArgs e)
{
    try
    {
        _trans.Rollback();
    }
    catch
    {
    }
    _cnn.Close();
}
```

# Đối tượng SqlTransaction

```
private void buttonRunAlter_Click(object sender, EventArgs e)
{
    int n = 0;
    if (_cnn.State != ConnectionState.Open)
    {
        n = 1;
        _cnn.Open();
    }

    string sql = textBoxSQL.Text;

    // -----
    SqlCommand cmd = new SqlCommand(sql, _cnn);
    cmd.ExecuteNonQuery();
    // -----

    if (n == 0)
    {
        _cnn.Close();
    }
}
```

# Đối tượng SqlTransaction

```
private void buttonRunSelect_Click(object sender, EventArgs e)
{
    int n = 0;
    if (_cnn.State != ConnectionState.Open)
    {
        n = 1;
        _cnn.Open();
    }

    string sql = textBoxSQL.Text;

    // -----
    SqlCommand cmd = new SqlCommand(sql, _cnn, _trans);
    _da = new SqlDataAdapter(cmd);
    _ds = new DataSet();
    _da.Fill(_ds);

    if (_ds.Tables.Count > 0 && _ds.Tables[0] != null)
    {
        dataGridViewData2.DataSource = _ds.Tables[0];
    }

    // -----
    if (n == 0)
    {
        _cnn.Close();
    }
}
```