

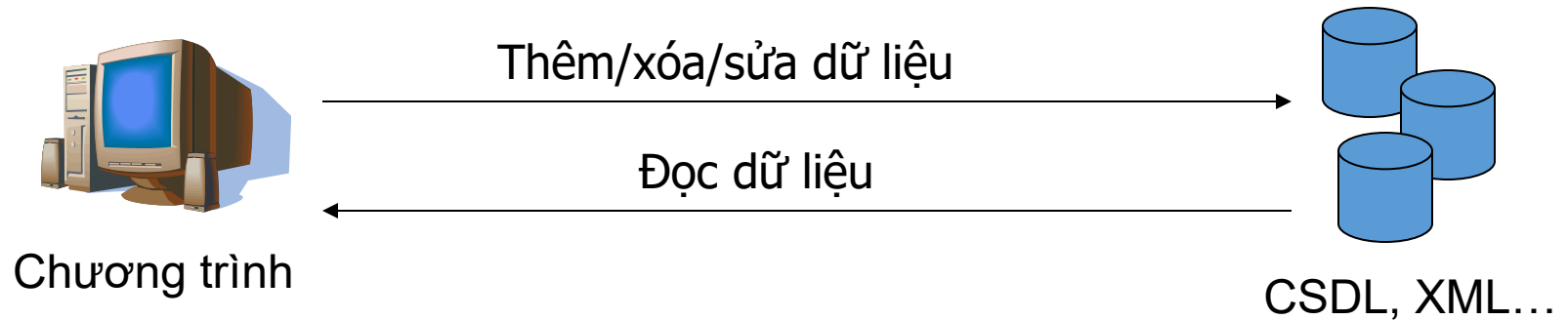
# LẬP TRÌNH KẾT NỐI CSDL ADO.NET

# 1. NỘI DUNG

- Giới thiệu ADO.NET
- Kết nối với cơ sở dữ liệu và đọc dữ liệu
- Đối tượng Command
- Đối tượng Dataset
- Xử lý mô hình ngắt kết nối
- Bài tập áp dụng

## 2. GIỚI THIỆU ADO.NET

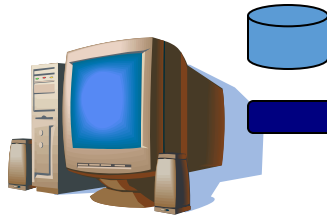
» ADO.NET là một tập các lớp thư viện được sử dụng để truy xuất dữ liệu



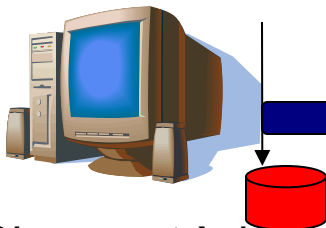
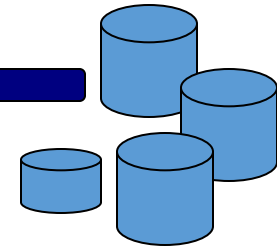
» Chiến lược truy xuất dữ liệu

- Connected Model
- Disconnected Model

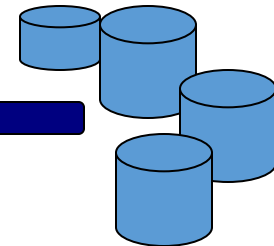
## 2. GIỚI THIỆU ADO.NET



Chương trình

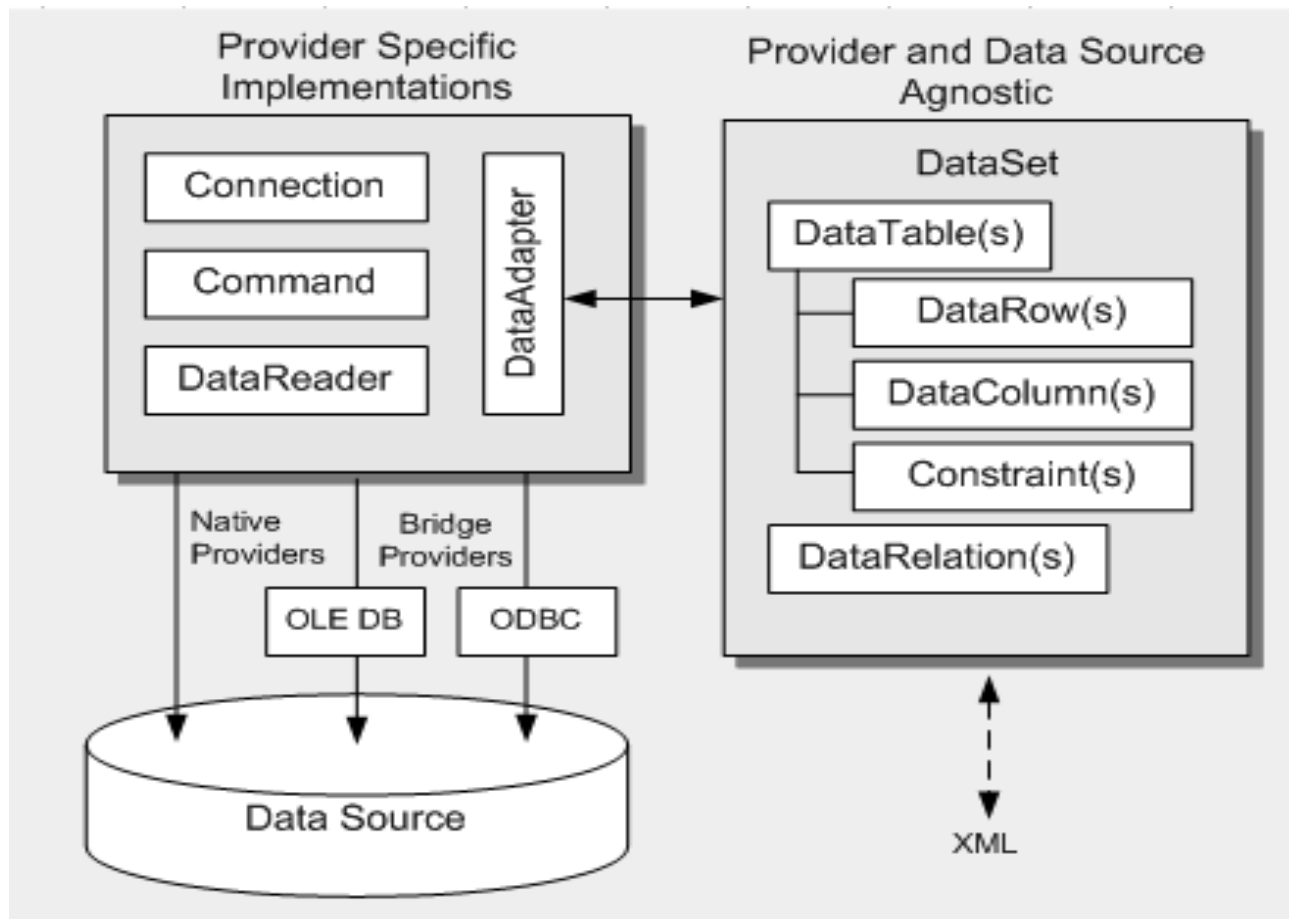


Chương trình



## 2. GIỚI THIỆU ADO.NET

- Kiến trúc của ADO.NET



## 2. GIỚI THIỆU ADO.NET

- .NET Data Provider
  - Dùng để kết nối đến CSDL, thực hiện các câu lệnh và nhận kết quả trả về.
  - Hiện có 4 loại Data Provider:
    - **SQL .NET Framework Data Provider**
    - **OLE DB .NET Framework Data Provider**
    - ODBC .NET Framework Data Provider
    - Oracle .NET Framework Data Provider

SQL .NET Framework Data Provider	OLE DB .NET Framework Data Provider
System.Data	System.Data
System.Data.SqlClient	System.Data.OleDb

# Kết nối với cơ sở dữ liệu và đọc dữ liệu

- **Một đối tượng Connection thể hiện một kết nối đến CSDL**
  - ***Các lớp phụ trách kết nối***
    - ODBCConnection
    - OleDbConnection
    - SqlConnection
    - OracleConnection
  - ***Thuộc tính quan trọng***
    - ConnectionString: xác định nguồn dữ liệu cần kết nối
  - ***Phương thức quan trọng***
    - Open(): mở kết nối
    - Close(): đóng kết nối
    - CreateCommand(): Tạo một đối tượng command nhờ một câu truy vấn

# Kết nối với cơ sở dữ liệu và đọc dữ liệu

- Sử dụng phương thức xây dựng có tham số của lớp SqlConnection.
    - Tham số này là chuỗi kết nối (ConnectionString):
      - Tên Server lưu cơ sở dữ liệu (Server)
      - Tên cơ sở dữ liệu (Database)
      - Tài khoản, Mật khẩu (uid, pwd)
      - ...
- Mỗi thuộc tính cách nhau dấu ;

```
SqlConnection sqlcon = new SqlConnection("Server=SQLDB;  
uid=mylogin;pwd=mylogin;Database=diemthi_hk");
```



# Kết nối với cơ sở dữ liệu và đọc dữ liệu

- Sử dụng phương thức xây dựng có tham số của lớp OleDbConnection.
    - Tham số này là chuỗi kết nối (ConnectionString):
      - Tên trình cung cấp (Provider)
      - Tên Server lưu cơ sở dữ liệu (Server, Data Source)
      - Tên cơ sở dữ liệu (Database)
      - Tài khoản, Mật khẩu (uid, pwd)
      - ...
- Mỗi thuộc tính cách nhau dấu ;

# Kết nối với cơ sở dữ liệu và đọc dữ liệu

```
OleDbConnection con=new  
    OleDbConnection("Provider=SQLOLEDB;Server=SQLDB;  
    Database=diemthi_hk;uid=sa;pwd=password");
```

```
OleDbConnection con=new  
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;  
    Data Source=D:\\diemthi_hk.mdb");
```

- Một số Provider:
  - SQLOLEDB: Provider của SQL Server
  - Microsoft.Jet.OLEDB.4.0: Provider của Access

# Đối tượng Command

- Một đối tượng Command cho phép truy xuất hoặc thao tác dữ liệu trong cơ sở dữ liệu (thông qua câu truy vấn dạng chuỗi).
- Đối tượng Command được khởi tạo sau khi 1 đối tượng Connection được thiết lập.
- Gồm:
  - SqlCommand: cho phép thực thi câu truy vấn với SQL .Net Framework Data Provider.
  - OleDbCommand: cho phép thực thi câu truy vấn với OleDb .Net Framework Data Provider.

# Đối tượng Command

- **Thuộc tính**
  - **CommandText**: là chuỗi thể hiện câu truy vấn hoặc tên của 1 stored procedure hay tên 1 bảng.
  - **CommandType**: loại của đối tượng Command (**StoredProcedure, TableDirect, Text**).
  - **Connection**: đối tượng Connection đến 1 cơ sở dữ liệu.
- **Phương thức**
  - **ExecuteNonQuery()**: thực thi câu truy vấn hành động (Insert, Update, Delete)
  - **ExecuteReader()**: thực thi câu truy vấn dạng Select; kết quả trả về là 1 đối tượng DataReader.
  - **ExecuteScalar()**: thực thi câu truy vấn dạng Select với kết quả của câu truy vấn là 1 giá trị đơn.

# Ví dụ

```
SqlConnection conn = null;
try
{
    //Tạo đối tượng Connection
    conn = new SqlConnection("Server=HOANG-PC; Database = diemthi_hk; uid=tvhoang; pwd=12345");
    //Mở kết nối
    conn.Open();
    String querystring = "Insert Into Sinh_Vien(MaSo_SV, HoTen_SV, Phai_SV, NgaySinh, NoiSinh_SV, MaSo_LH) "+
        "Values('1234567', N'Nguyễn Thanh Tú', N'Nam', '1990/12/13', N'Cà Mau', 'TH2602')";
    SqlCommand comm = new SqlCommand(querystring, conn);
    //Thực hiện câu lệnh
    comm.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    if (conn != null)
    {
        conn.Close();
    }
}
```

# Ví dụ

```
OleDbConnection conn = null;
try
{
    //Tạo đối tượng Connection
    conn = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=D:\\diemthi_hk.mdb");
    //Mở kết nối
    conn.Open();
    String querystring = "Insert Into Sinh_Vien(MaSo_SV, HoTen_SV, Phai_SV, NgaySinh, NoiSinh_SV, MaSo_LH)"+
        "Values('1234567', 'Nguyễn Thanh Tú', 'Nam', '1990/12/13', 'Cà Mau', 'TH2602')";
    OleDbCommand comm = new OleDbCommand(querystring, conn);
    //Thực hiện câu lệnh
    comm.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    if (conn != null)
    {
        conn.Close();
    }
}
```

# Ví dụ

```
SqlConnection conn = null;
try
{
    //Tạo đối tượng Connection
    conn = new SqlConnection("Server=HOANG-PC; Database=diemthi_hk;" +
                             "uid=tvhoang; pwd=12345");

    //Mở kết nối
    conn.Open();
    //Tìm năm sinh lớn nhất
    String querystring = "Select Max(Year(NgaySinh)) From Sinh_Vien";
    SqlCommand comm = new SqlCommand(querystring, conn);
    //Thực hiện câu lệnh
    int year = Convert.ToInt32(comm.ExecuteScalar());
    MessageBox.Show(year.ToString());
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    if (conn != null)
    {
        conn.Close();
    }
}
```

# Ví dụ

```
OleDbConnection conn = null;
try
{
    //Tạo đối tượng Connection
    conn = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; "+
                                "Data Source=D:\\diemthi_hk.mdb");

    //Mở kết nối
    conn.Open();
    //Tìm năm sinh lớn nhất
    String querystring = "Select Max(Year(NgaySinh)) From Sinh_Vien";
    OleDbCommand comm = new OleDbCommand(querystring, conn);
    //Thực hiện câu lệnh
    int year = Convert.ToInt32(comm.ExecuteScalar());
    MessageBox.Show(year.ToString());
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    if (conn != null)
    {
        conn.Close();
    }
}
```



# Parameters

- **Vấn đề:** Câu truy vấn được thực hiện dựa trên những giá trị được nhập từ bàn phím.
- **Giải pháp:**
  - Viết câu truy vấn dạng chuỗi trực tiếp
  - Lớp Parameter

```
String querystring = "Delete From Mon_Hoc "+  
                    " Where MaSo_MH='" + txtMaMonHoc.Text + "'";  
SqlCommand comm = new SqlCommand(querystring, conn);  
//Thực hiện câu lệnh  
comm.ExecuteNonQuery();
```

# Parameters

- Viết câu truy vấn dạng chuỗi trực tiếp
  - Ví dụ:

```
String querystring = "Delete From Mon_Hoc "+  
                    " Where MaSo_MH='" + txtMaMonHoc.Text + "'";  
SqlCommand comm = new SqlCommand(querystring, conn);  
//Thực hiện câu lệnh  
comm.ExecuteNonQuery();
```

# Parameters

- **Lớp Parameter**
  - 1 đối tượng Parameter là 1 tham số được truyền vào khi thực hiện 1 câu truy vấn hoặc 1 stored procedure thông qua 1 đối tượng Command.
- **Gồm:**
  - Lớp SqlParameter: tham số với SQL .Net Framework Data Provider.
  - Lớp OleDbParameter: tham số với OleDb .Net Framework Data Provider

# Parameters

- Lớp Parameter
  - **Thuộc tính**
    - ParameterName: tên tham số.
    - DbType: Kiểu tham số
    - Value: Giá trị của tham số
  - **Phương thức xây dựng**
    - SqlParameter(string, SqlDbType)
    - OleDbParameter(string, OleDbType)
    - Định nghĩa 1 tham số với tên và kiểu thích hợp
- Một tham số sau khi định nghĩa sẽ được thêm vào tập hợp Parameters của 1 đối tượng Command

# Parameters

- Lớp Parameter

- Ví dụ:

```
//Câu truy vấn có chứa tham số @MS
String querystring = "Delete From Mon_Hoc "+
    " Where MaSo_MH = @MS";

//Định nghĩa một tham số có tên @MS có kiểu string
SqlParameter para = new SqlParameter("@MS", SqlDbType.NVarChar);
//Giá trị của tham số @MS lấy từ textbox txtMaMonHoc
para.Value = txtMaMonHoc.Text;

SqlCommand comm = new SqlCommand(querystring, conn);
//Thêm vào tập hợp các Parameters
comm.Parameters.Add(para);
//Thực hiện câu lệnh
comm.ExecuteNonQuery();
```

# Parameters

- Thuộc tính CommandText của đối tượng Command là tên của **stored procedure**.
- Thuộc tính CommandType là **StoredProcedure**
- Dùng lớp Parameter để định nghĩa các tham số.
- Thêm các tham số vào tập hợp Parameters của đối tượng Command.
- Thực thi câu truy vấn.

# Ví dụ sử dụng procedure

- Giả sử ta có 1 stored procedure sau:
  - CREATE PROCEDURE **SPPUBLISHER**
  - AS
  - SELECT \* FROM tblEbooks
  - GO

```
string connetionString = null;
    SqlConnection connection ;
    SqlDataAdapter adapter ;
    SqlCommand command = new SqlCommand();
    DataSet ds = new DataSet();

    int i = 0;
    connetionString = "Data Source=servername;Initial
Catalog=PUBS;User ID=sa;Password=yourpassword";
    connection = new SqlConnection(connetionString);

    connection.Open();
    command.Connection = connection;
    command.CommandType = CommandType.StoredProcedure;
    command.CommandText = "SPPUBLISHER";
    adapter = new SqlDataAdapter(command);
    adapter.Fill(ds);

    for (i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)
    {
        MessageBox.Show
(ds.Tables[0].Rows[i][0].ToString ());
    }

    connection.Close();
```

# Ví dụ sử dụng procedure

- Giả sử ta có 1 stored procedure có tham số như sau:

```
CREATE PROCEDURE SPBooks
```

```
    @bookTitle VARCHAR(20)
```

```
AS
```

```
    SELECT publisherName FROM tblEbooks
```

```
WHERE bookTitle = @bookTitle
```

```
GO
```



# Ví dụ sử dụng procedure

```
SqlConnection connection ;
SqlDataAdapter adapter ;
SqlCommand command = new SqlCommand();
SqlParameter param ;
DataSet ds = new DataSet();
int i = 0;

    connection = new SqlConnection(connt);
    connection.Open();
    command.Connection = connection;
    command.CommandType = CommandType.StoredProcedure;
    command.CommandText = "SPBooks";
    param = new SqlParameter("@bookTitle", "SQL 2008");
    param.Direction = ParameterDirection.Input;
    param.DbType = DbType.String;
    command.Parameters.Add(param);

    adapter = new SqlDataAdapter(command);
    adapter.Fill(ds);

    for (i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)
    {
        MessageBox.Show (ds.Tables[0].Rows[i][0].ToString ());
    }

    connection.Close();
}
```

# Đối tượng Dataset

- Đối tượng thuộc lớp DataReader chứa luồng dữ liệu chỉ đọc là kết quả của việc thực thi câu truy vấn dạng Select.
  - Luồng dữ liệu này chỉ cho phép truy xuất 1 chiều (từ đầu đến cuối - forward only).
- 1 đối tượng DataReader được tạo ra khi phương thức ExecuteReader() của 1 đối tượng Command được gọi thực hiện.
- Gồm:
  - Lớp SqlDataReader: đọc dữ liệu với SQL .Net Framework Provider
  - Lớp OleDbDataReader: đọc dữ liệu với OleDb .Net Framework Provider

# Đối tượng Dataset

- **Thuộc tính**

- HasRows: xác định đối tượng DataReader còn trả về kết quả nữa (true) hay không (false)?

- **Phương thức**

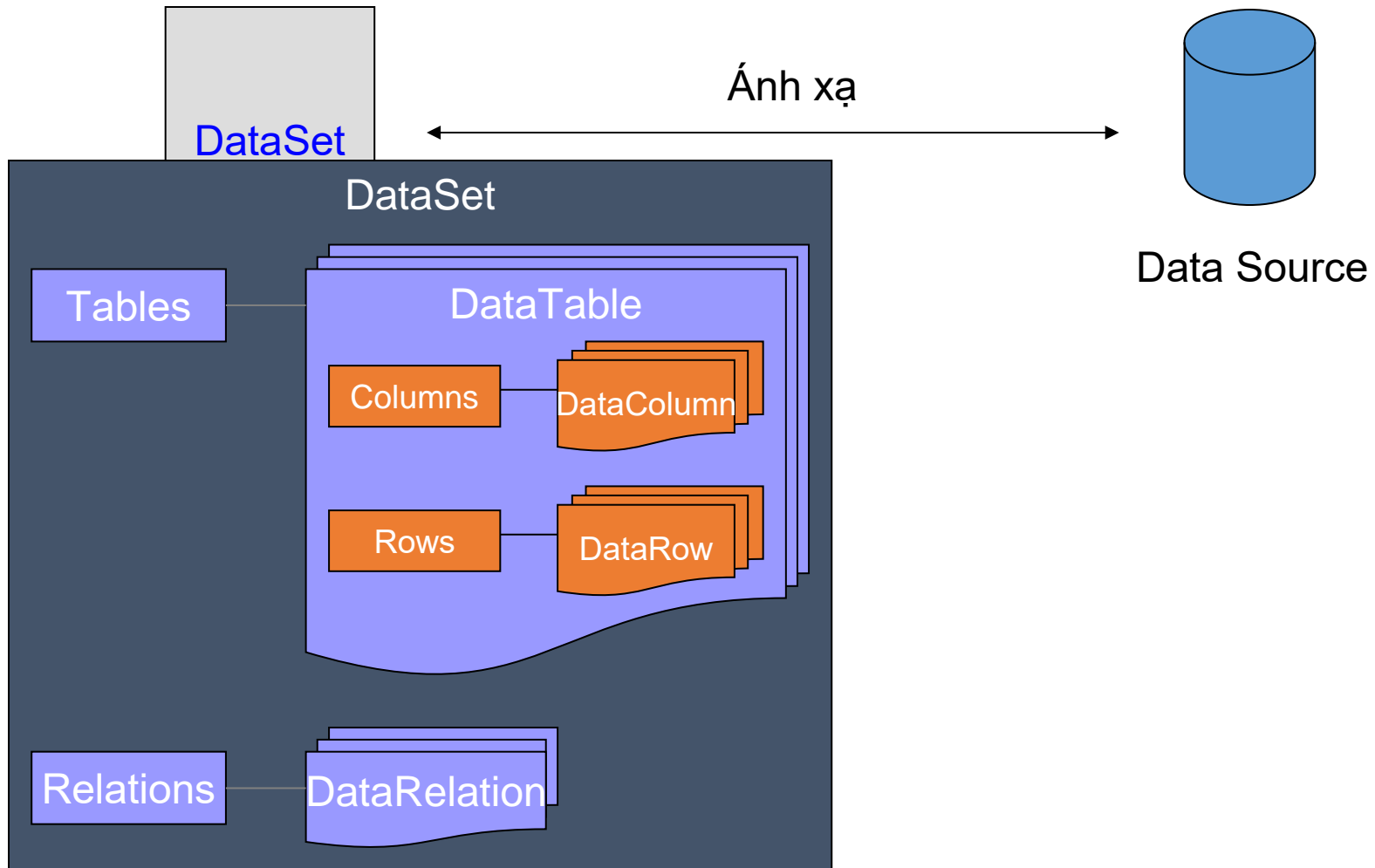
- Close: đóng lại đối tượng DataReader
- Read: di chuyển đến mẫu tin kế tiếp, kết quả trả về là true nếu di chuyển thành công, ngược lại false.
- GetBoolean: trả về giá trị của cột chỉ định như là 1 giá trị kiểu bool.
- GetDateTime: trả về giá trị của cột chỉ định như là 1 giá trị kiểu DateTime
- GetInt32: trả về giá trị của cột chỉ định như là 1 giá trị kiểu int
- GetString: trả về giá trị của cột chỉ định như là 1 giá trị kiểu string
- GetValue: trả về giá trị của cột chỉ định

# Đối tượng Dataset

- Ví dụ: Hiển thị ngày sinh của những sinh viên có phái là nam

```
SqlCommand comm = new SqlCommand("Select * From Sinh_Vien Where Phai_SV='Nam'", conn);
SqlDataReader dr;
dr = comm.ExecuteReader();
while (dr.Read())
{
    if (!dr.IsDBNull(3))
    {
        MessageBox.Show(dr.GetDateTime(3).ToString());
    }
}
```

# Đối tượng Dataset



# Đối tượng Dataset

- DataTable
  - **TableName**: tên bảng.
  - **Columns**: danh sách các cột (DataColumn).
  - **Rows**: danh sách các mẫu tin (DataRow).
  - **PrimaryKey**: danh sách các cột làm khóa chính (DataColumn).
  - **NewRow()**: tạo một mẫu tin mới.
- DataColumn
  - **ColumnName**: tên cột.
  - **DataType**: kiểu dữ liệu cột.
- DataRow
  - **RowState**: trạng thái của mẫu tin (Added, Modified, Deleted, Unchanged...).
  - Toán tử **[ i ]**: truy xuất đến cột i của mẫu tin.
  - **Delete()**: đánh dấu xóa mẫu tin.

# Đối tượng Dataset

- Một DataTable là một bảng trong 1 DataSet (1 bảng trong bộ nhớ của cơ sở dữ liệu quan hệ).
- Thuộc tính
  - **TableName**: tên bảng.
  - **Columns**: danh sách các cột (DataColumn).
  - **Rows**: danh sách các mẫu tin (DataRow).
  - **PrimaryKey**: danh sách các cột làm khóa chính (DataColumn).
  - **Constraints**: các constraints của 1 DataTable.
- Phương thức
  - **NewRow()**: tạo một mẫu tin mới.

# Đối tượng Dataset

- Một DataColumn là 1 cột của 1 DataTable trong 1 DataSet.
- **Thuộc tính**
  - **ColumnName:** tên của cột
  - **DataType:** kiểu dữ liệu



# Đối tượng Dataset

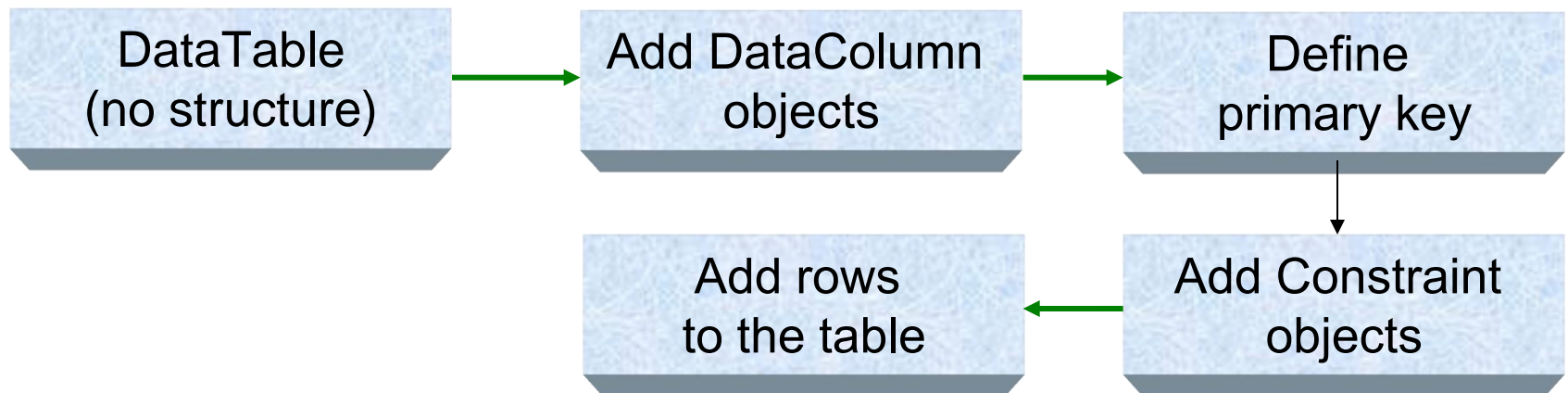
- Một DataRow là 1 dòng của 1 DataTable trong 1 DataSet.
- Thuộc tính
  - **RowState**: trạng thái của mẫu tin (Added, Modified, Deleted, Unchanged...).
  - **Toán tử [ i ]**: truy xuất đến cột i của mẫu tin.
- Phương thức
  - **Delete()**: đánh dấu xóa mẫu tin.

# Đối tượng Dataset

- Dùng phương thức xây dựng của lớp DataColumn.
- Sử dụng tập hợp Columns của đối tượng DataTable để thêm cột mới vào.

```
DataTable dt = new DataTable();  
dt.TableName = "Sinh_Vien";  
//hoặc DataTable dt = new DataTable("Sinh_Vien");  
  
//Dùng phương thức xây dựng của lớp DataColumn  
DataColumn ms = new DataColumn();  
ms.ColumnName = "MSSV";  
ms.DataType = typeof(Int32);  
ms.AllowDBNull = false;  
ms.Unique = true;  
dt.Columns.Add(ms);  
  
dt.Columns.Add("HoTen", typeof(String));  
dt.Columns.Add("NgaySinh", typeof(DateTime));
```

# Đối tượng Dataset



# Đối tượng Dataset

**Khóa chính là 1 cột**

```
dt.PrimaryKey=new DataColumn[]  
{dt.Columns["MSSV"]};
```

hay

```
DataColumn[]col_arr=new DataColumn[1];  
col_arr[0]=dt.Columns["MSSV"];  
dt.PrimaryKey = col_arr;
```

**Khóa chính là khóa tổ hợp**

```
dt.PrimaryKey=new DataColumn[]  
{dt.Columns["MSSV"],  
dt.Columns["HoTen"]};
```

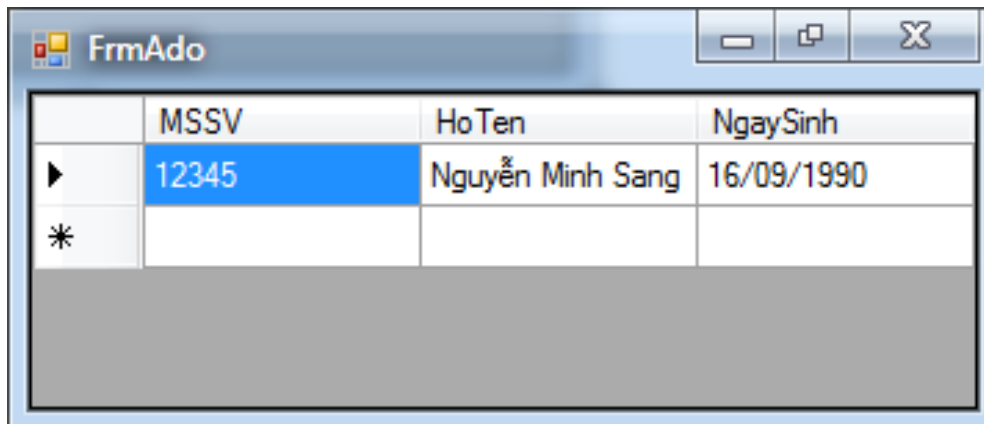
hay

```
DataColumn[]pkey=new DataColumn[2];  
pkey[0] = dt.Columns["MSSV"];  
pkey[1] = dt.Columns["HoTen"];  
dt.PrimaryKey = pKey;
```

# Đối tượng Dataset

- Sử dụng phương thức NewRow của lớp DataTable.

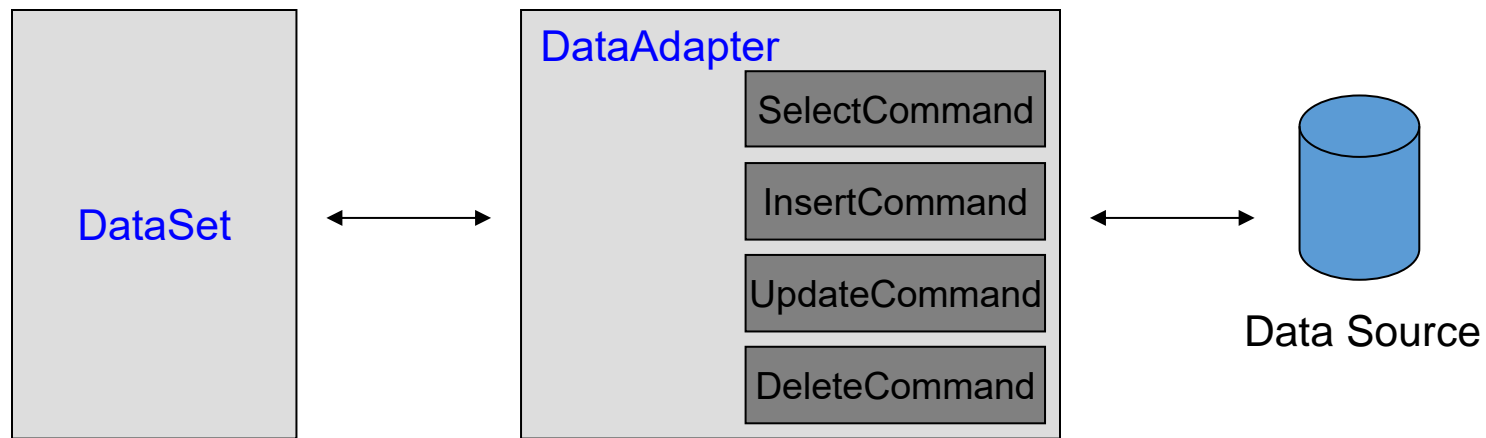
```
DataRow dr = dt.NewRow();  
dr["MSSV"] = 12345; // hoặc dr[0] = 12345;  
dr["HoTen"] = "Nguyễn Minh Sang"; // hoặc dr[1] = "Nguyễn Minh Sang";  
dr["NgàySinh"] = "1990/09/16"; // hoặc dr[2] = "1990/09/16"  
dt.Rows.Add(dr);  
//Hiển thị DataTable bằng điều khiển DataGridView  
dataGridViewSinhVien.DataSource = dt;
```



	MSSV	HoTen	NgàySinh
▶	12345	Nguyễn Minh Sang	16/09/1990
*			

# Đối tượng Dataset

- Được sử dụng để lấy dữ liệu từ CSDL vào đưa vào DataSet, và ngược lại cho phép cập nhật dữ liệu được thay đổi từ DataSet vào CSDL.



Disconnected Model

- Gồm:
  - Lớp `SqlDataAdapter`: SQL .Net Framework Data Provider
  - Lớp `OleDbDataAdapter`: OleDb .Net Framework Data Provider

# Đối tượng Dataset

- Thuộc tính
  - SelectCommand: Lấy dữ liệu từ CSDL
  - InsertCommand:
  - UpdateCommand:
  - DeleteCommand:
- Phương thức
  - Fill: dùng SelectCommand để lấy dữ liệu từ CSDL đổ vào DataSet.
  - Update: dùng InsertCommand, UpdateCommand, DeleteCommand để cập nhật dữ liệu trong DataSet vào CSDL.

# Xử lý mô hình ngắt kết nối

- Đưa dữ liệu từ CSDL vào DataSet
  - Nối kết đến CSDL (Connection)
  - Tạo đối tượng DataAdapter
  - Chỉ ra câu truy vấn trong thuộc tính SelectCommand của DataAdapter.
  - Điền dữ liệu từ DataAdapter vào Dataset nhờ phương thức Fill của lớp DataAdapter

```
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinh_Vien", conn);  
DataSet ds = new DataSet();  
da.Fill(ds, "T_SinhVien");  
da = new SqlDataAdapter("Select * From Mon_Hoc", conn);  
da.Fill(ds, "T_MonHoc");  
dataGridViewSinhVien.DataSource = ds.Tables["T_SinhVien"];
```



# Xử lý mô hình ngắt kết nối

- Xử lý dữ liệu

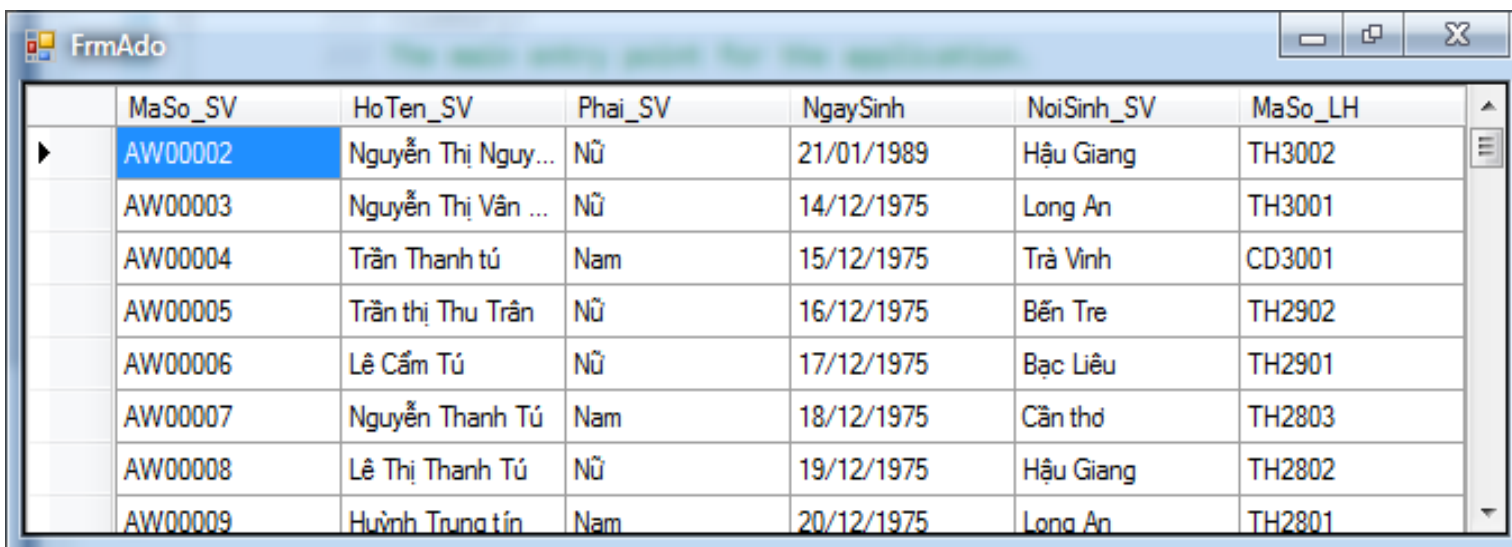
```
/******Thêm mới dữ liệu*****/  
DataRow dr1 = ds.Tables["T_SinhVien"].NewRow();  
dr1["MaSo_SV"] = "1111111";  
dr1["HoTen_SV"] = "Nguyễn Cường";  
dr1["Phai_SV"] = "Nam";  
dr1["NgàySinh"] = "1992/07/16";  
ds.Tables["T_SinhVien"].Rows.Add(dr1);  
/******Xóa dữ liệu*****/  
ds.Tables["T_SinhVien"].Rows[0].Delete();  
/******Thay đổi dữ liệu*****/  
DataRow dr2 = ds.Tables["T_SinhVien"].Rows[1];  
dr2.BeginEdit();  
dr2["NgàySinh"] = "1989/01/21";  
dr2.EndEdit();
```

- Thay đổi dữ liệu

```
SqlCommandBuilder builder = new SqlCommandBuilder(da);  
da.Update(ds, "T_SinhVien");
```

# Bài tập áp dụng

```
conn = new SqlConnection("Server=HOANG-PC; Database=diemthi_hk;" +  
    "uid=tvhoang; pwd=12345");  
conn.Open();  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinh_Vien", conn);  
DataSet ds = new DataSet();  
da.Fill(ds, "T_SinhVien");  
dataGridViewSinhVien.DataSource = ds.Tables["T_SinhVien"];
```



	MaSo_SV	HoTen_SV	Phai_SV	NgaySinh	NoiSinh_SV	MaSo_LH
▶	AW00002	Nguyễn Thị Nguy...	Nữ	21/01/1989	Hậu Giang	TH3002
	AW00003	Nguyễn Thị Vân ...	Nữ	14/12/1975	Long An	TH3001
	AW00004	Trần Thanh tú	Nam	15/12/1975	Trà Vinh	CD3001
	AW00005	Trần thị Thu Trần	Nữ	16/12/1975	Bến Tre	TH2902
	AW00006	Lê Cẩm Tú	Nữ	17/12/1975	Bạc Liêu	TH2901
	AW00007	Nguyễn Thanh Tú	Nam	18/12/1975	Cần thơ	TH2803
	AW00008	Lê Thị Thanh Tú	Nữ	19/12/1975	Hậu Giang	TH2802
	AW00009	Huỳnh Trung tín	Nam	20/12/1975	Long An	TH2801