



# BÀI 14: TRUY XUẤT CƠ SỞ DỮ LIỆU – DATASET

## Thời lượng: 180 phút

Giảng viên: PHẠM PHÚ KHƯƠNG

Email: [phamphukhuong@dtu.edu.vn](mailto:phamphukhuong@dtu.edu.vn)

Điện thoại: 0905635421

# Nội Dung

## □ MỤC TIÊU

*Sau khi hoàn thành chương này sinh viên sẽ nắm được kiến thức về:*

## □ Về lý thuyết

- Nắm vững kiến thức về khả năng hỗ trợ các hệ quản trị cơ sở dữ liệu khác nhau của một ứng dụng web.
- Tập trung khai thác khả năng xử lý dữ liệu với hệ quản trị cơ sở dữ liệu SQL server.
- Củng cố kiến thức đã học về các câu lệnh truy vấn và xử lý dữ liệu đã học ở các môn học trước đây.
- Tìm hiểu nguyên lý xử lý dữ liệu bao gồm các bước:
  - Kết nối bao gồm các khai báo kết nối xử lý kết nối và đóng kết nối khi kết thúc tiến trình khai thác dữ liệu
  - Xử lý dữ liệu, bao gồm chèn, hiệu chỉnh, xóa dữ liệu trong cơ sở dữ liệu theo yêu cầu lưu trữ.
  - Đọc và hiển thị thông tin dữ liệu
- Ngoài ra sinh viên còn được học các kiến thức nâng cao như khả năng khai thác các thủ tục của SQL Server, xây dựng các lớp chung để các nhóm lập trình có thể làm việc với nhau dễ dàng hơn.

## □ Về thực hành

- Người học có khả năng xây dựng một ứng dụng web động dựa trên việc khai thác cơ sở dữ liệu



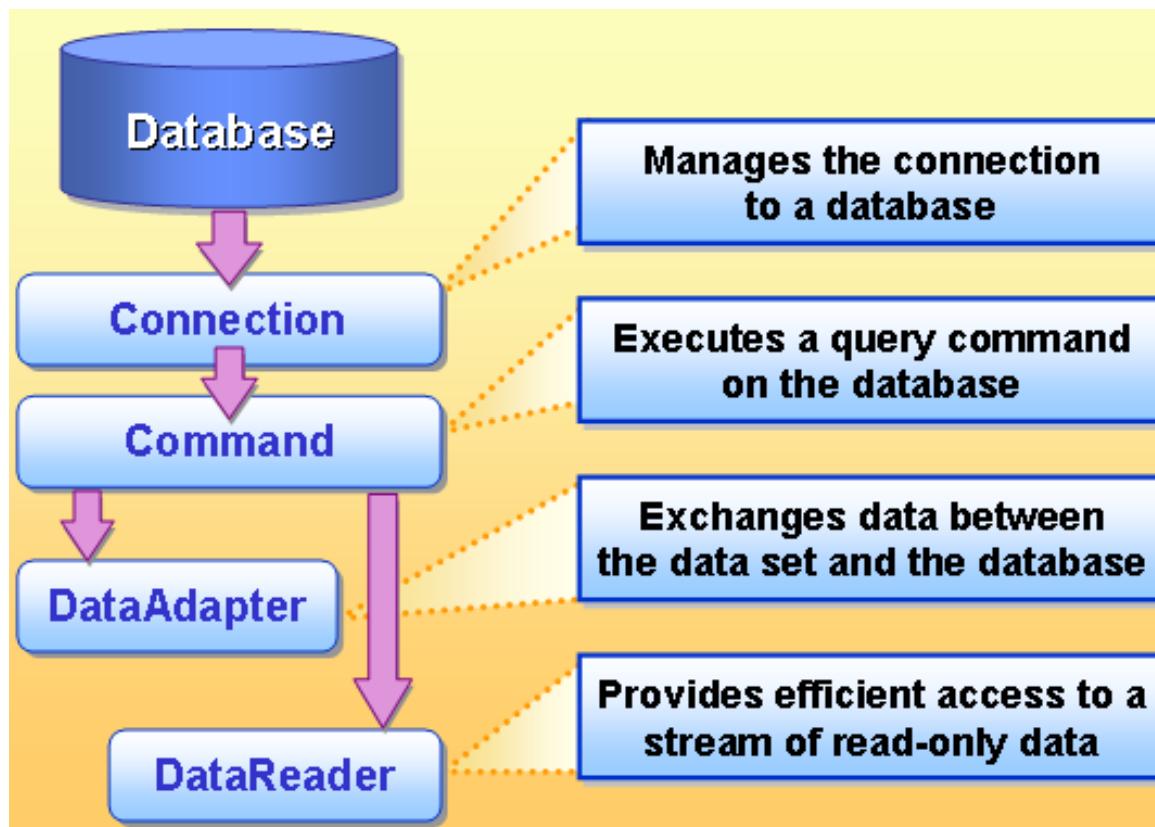
# Nội Dung

- ❑ ADO.NET
- ❑ Quá trình phát triển
- ❑ Đặc điểm ADO.NET
- ❑ **.NET Data Provider**
- ❑ DataSet



# NET Data Provider

## ■ Mô hình .NET Data Provider

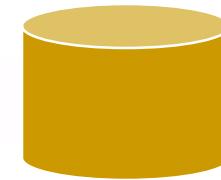


# NET Data Provider - Connection



Application

Connection



Database

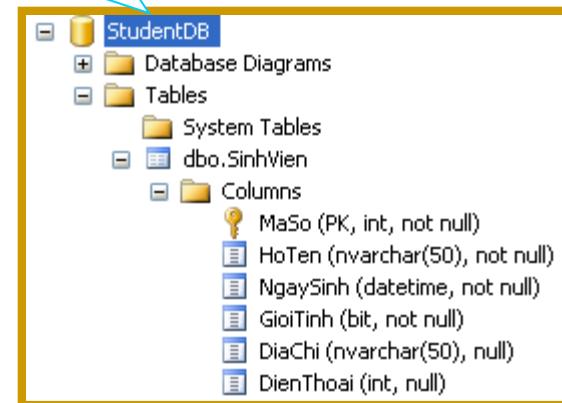
## Thuộc tính & Phương thức

- **ConnectionString:** chuỗi kết nối DataSource
- **Open():** thiết lập kết nối đến DS
- **Close():** đóng kết nối với DS



## ■ Mô tả CSDL sử dụng minh họa

Đặc điểm	Thông tin
DBMS	MS SQL Server Express 2005
DB 1	CSDL StudentDB
DB 2	CSDL Northwind





# NET Data Provider - Connection

## SQL Connection

```
SqlConnection cnn = new SqlConnection();  
cnn.ConnectionString = "server=.\SQLEXPRESS;  
database=Northwind; Trusted_Connection=true";
```

```
cnn.Open();
```

```
//xử lý trong quá trình kết nối
```

```
...
```

```
cnn.Close();
```

Thay bằng “localhost”  
trong SQL Server 2005



## Database

## ODBC/OLE DB Connection (\*)

MS Access

**Driver = {Microsoft Access Driver (\*.mdb)};**  
**DBQ = <đường dẫn file access>**

**Provider=Microsoft.Jet.OLEDB.4.0; Data Source = <đường dẫn file access>**

SQL Server

**Driver = {SQLServer}; Server = ServerName;**  
**Database= DatabaseName; Uid=Username;**  
**Pwd=Password;**

**Provider= SQLOLEDB; Data Source=ServerName;**  
**Initial Catalog=DatabaseName; UserId=Username;**  
**Password=Password**

(\*): Xem thêm chuỗi kết nối trong: <http://www.connectionstrings.com/?carrier=sqlserver2005>



## OLEDB Connection

```
OleDbConnection cnn = new OleDbSqlConnection();  
cnn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;  
Data Source =Sinhvien.mdb";
```

```
cnn.Open();
```

```
//xử lý trong quá trình kết nối
```

```
...
```

```
cnn.Close();
```



# NET Data Provider - Command

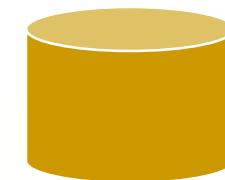


Application

Connection



Command



Database

## Thuộc tính & Phương thức

- Connection: kết nối để thực hiện lệnh
- CommandText: câu lệnh cần thực hiện
- CommandType: loại câu lệnh (Text,TableDirect, StoredProcedure)
- ExecuteScalar(): thực hiện câu lệnh và trả về giá trị đơn
- ExecuteNonQuery(): gọi các lệnh SQL, store, trả về số row bị tác động (Insert, Update, Delete...)
- ExecuteReader(): thực hiện lệnh và trả về DataReader



# NET Data Provider - Command

## SQL Command

```
SqlConnection cnn = new SqlConnection();  
cnn.ConnectionString = "server=.\SQLEXPRESS;  
database=Northwind; Trusted_connection=true";  
  
SqlCommand cmd = new SqlCommand("",cnn);  
cmd.CommandText = "Select COUNT(*) From Customers";  
  
cnn.Open();  
  
int count = (int) cmd.ExecuteScalar();  
cnn.Close();
```



# NET Data Provider - Command

## SQL Command

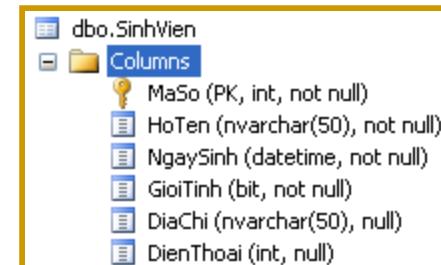
```
SqlConnection conn = new SqlConnection();
conn.ConnectionString = "Data source=.\SQLEXPRESS;"+
    "Initial Catalog=StudentDB; Integrated security=yes;";

SqlCommand cmd = new SqlCommand();
cmd.CommandText =
    "Insert into Sinhvien values(007,'Le Nam','1/1/1980',1,null,null)";
cmd.Connection = conn;

conn.Open();

cmd.ExecuteNonQuery();

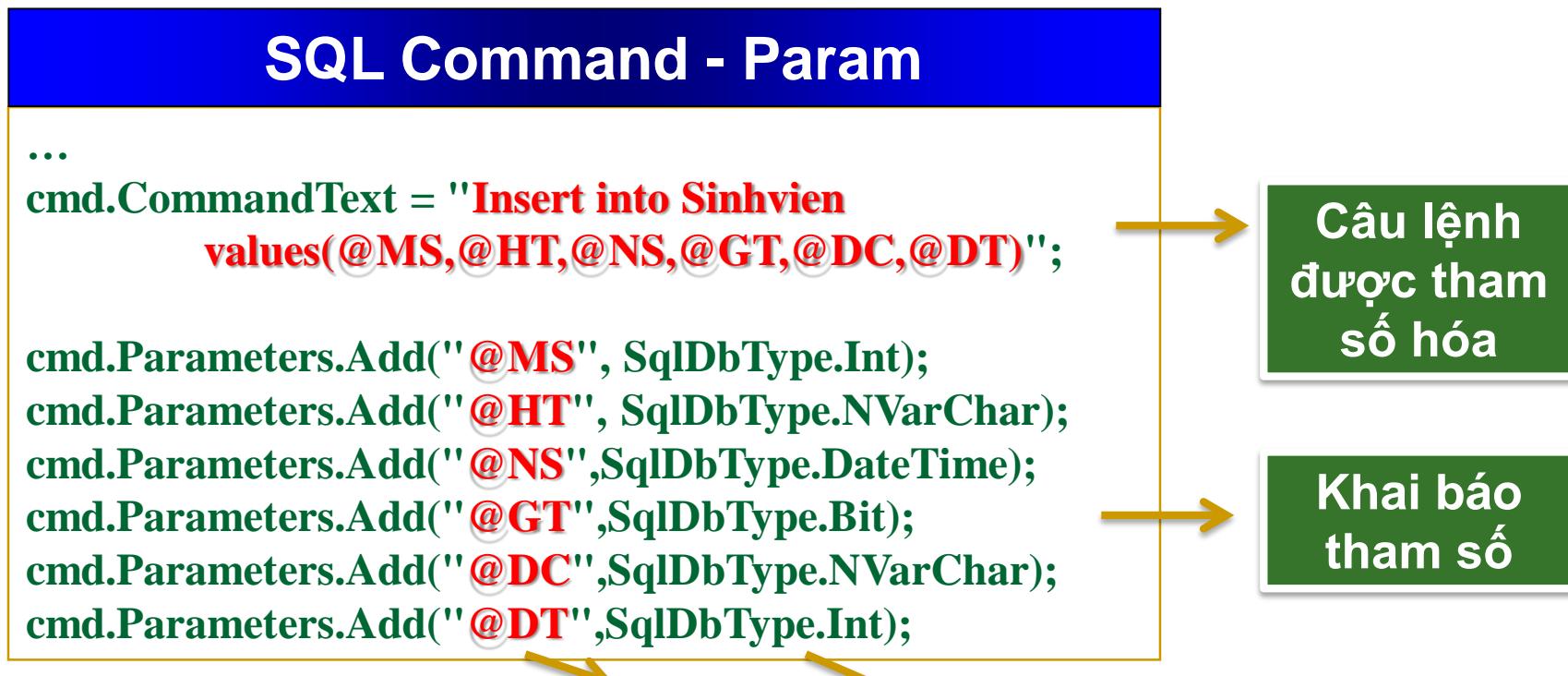
conn.Close();
```





# NET Data Provider - Command

- Tham số hóa câu lệnh
  - Một câu lệnh được sử dụng nhiều lần!





## SQL Command - Param

```
...  
cmd.Parameters["@MS"].Value = 999;  
cmd.Parameters["@HT"].Value = "Nguyen Ha Giang";  
cmd.Parameters["@NS"].Value = new DateTime(1978,12,4);  
cmd.Parameters["@GT"].Value = 1;  
cmd.Parameters["@DC"].Value = "Tan Binh";  
cmd.Parameters["@DT"].Value = 5120791;
```

```
conn.Open();  
int count = (int) cmd.ExecuteNonQuery();
```

```
conn.Close();
```

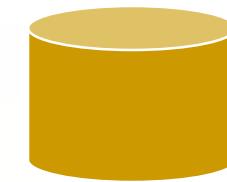
Truyền giá  
trị cho các  
tham số  
tương ứng



# NET Data Provider - DataReader



Application



Database

## Thuộc tính & Phương thức

- **HasRow**: cho biết câu truy vấn có trả về dữ liệu
- **Read()**: đọc một mẩu tin
- **[i]**: truy xuất đến cột i của mẩu tin được đọc
- **Close()**: đóng

Truy xuất tuần tự

Chỉ đọc dữ liệu

Cơ chế kết nối



# NET Data Provider - DataReader

## DataReader

...

```
SqlCommand cmd = new SqlCommand("Select *\nFrom Sinhvien", conn);
```

```
SqlDataReader reader;
```

```
conn.Open();
```

```
reader = cmd.ExecuteReader();
```

```
while (reader.Read())
```

```
    listBox1.Items.Add(reader["Hoten"]);
```

```
reader.Close();
```

```
conn.Close();
```

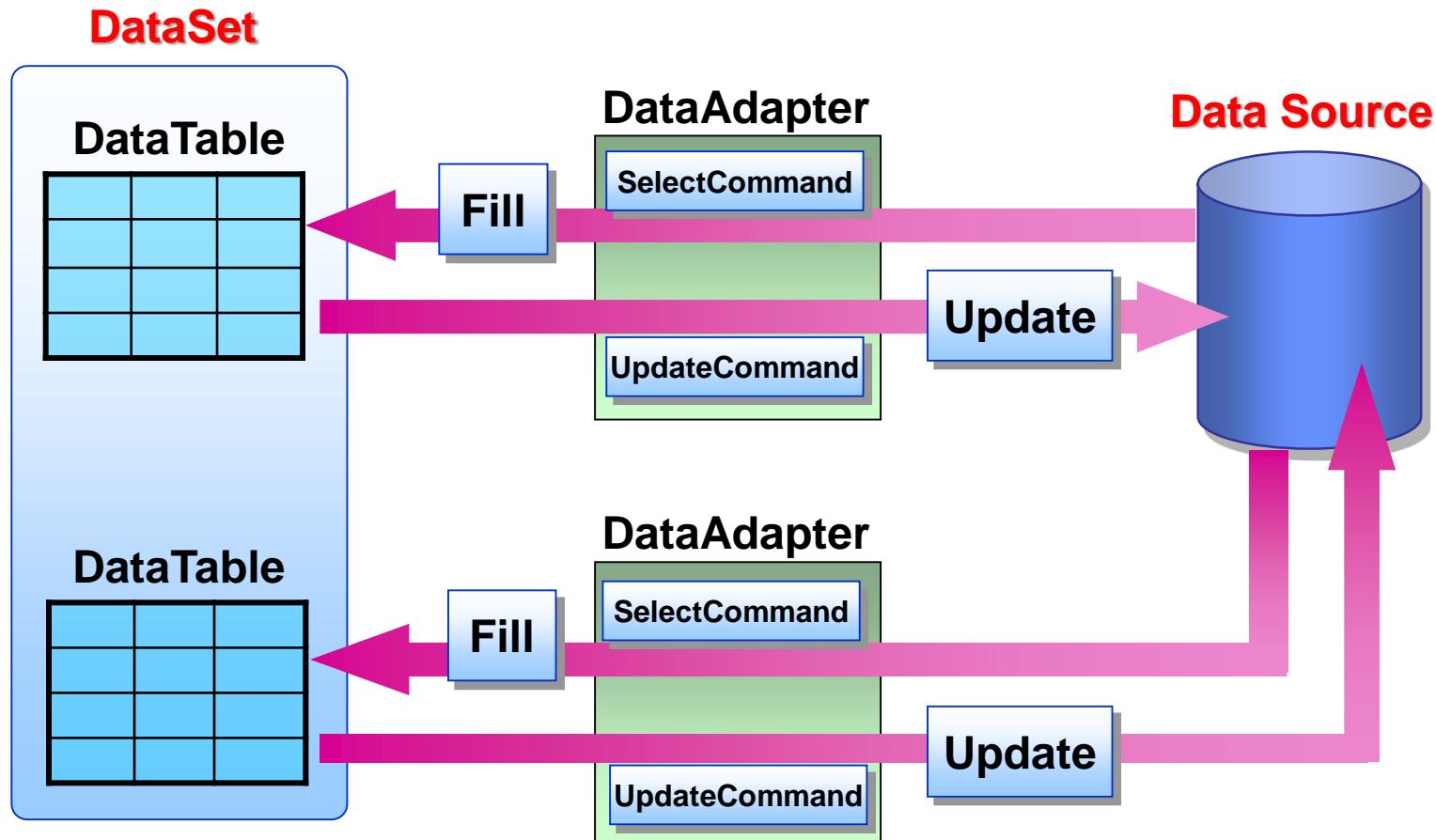
Thực thi lệnh truy vấn

Đọc một dòng

Lấy cột Hoten của  
dòng vừa đọc



# NET Data Provider - DataAdapter





# NET Data Provider - DataAdapter

## Thuộc tính & Phương thức

- **Fill(DataSet)**: sử dụng SelectCommand lấy dữ liệu từ Data Source đổ vào Data Set
- **Update(DataSet)**: InsertCommand, UpdateCommand, DeleteCommand cập nhật dữ liệu trong DataSet vào DataSource





# NET Data Provider - DataAdapter

## DataAdapter

```
string strConn="Server=.\SQLEXPRESS; Database=StudentDB;  
Trusted_connection=true";  
  
SqlDataAdapter adapter = new  
    SqlDataAdapter("Select * From Sinhvien", strConn);  
  
DataSet ds = new DataSet();  
  
adapter.Fill(ds);  
// thao tác trên dataset  
...  
adapter.Update(ds);
```



# Nội Dung

- ❑ ADO.NET
- ❑ Đặc điểm ADO.NET
- ❑ .NET Data Provider
- ❑ **DataSet**



# DataSet

- Là phần cơ sở dữ liệu được lưu trữ trong bộ nhớ (in-memory database)
- Cơ chế không kết nối
- Nhờ đối tượng DataAdapter làm trung gian
- Hỗ trợ đầy đủ đặc tính XML
- Thao tác được với tất cả mô hình lưu trữ hiện tại:
  - Relational database.



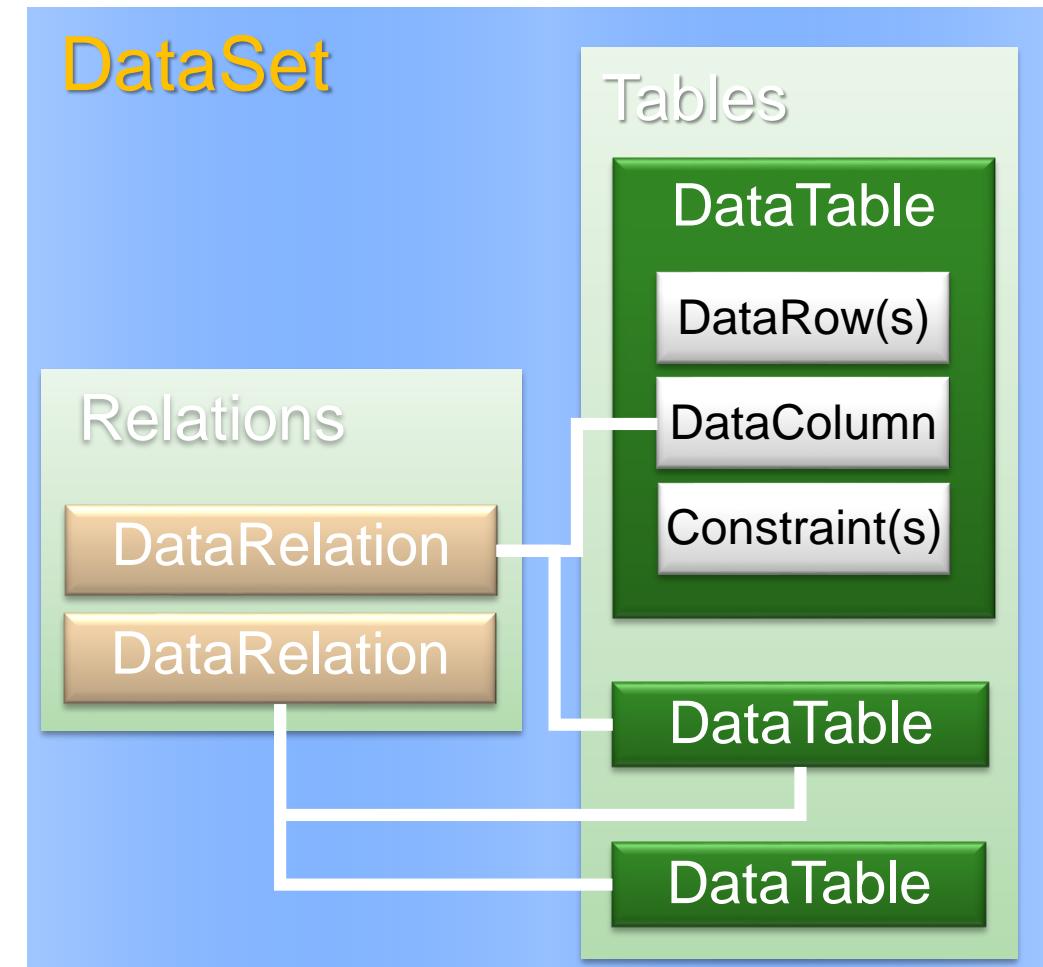
# DataSet vs. DataReader

DataSet	DataReader
Read/write access to data	Read-only
Includes multiple tables from different databases	Based on one SQL statement from one database
Disconnected	Connected
Bind to multiple controls	Bind to one control only
Forward and backward scanning of data	Forward-only
Slower access	Faster access
Supported by Visual Studio .NET tools	Manually coded



# DataSet

- Các thành phần
  - Tables
  - Relations





# DataSet

## DataSet

.Tables[...]

DataTable

schema

.Columns[...]

DataColumn

DataColumn

.Rows[...]

DataRow

DataRow

data

.DefaultView

DataView

.Relations[...]

DataRelation

...



# DataSet - DataTable

## ■ **DataTable: thể hiện một bảng trong CSDL**

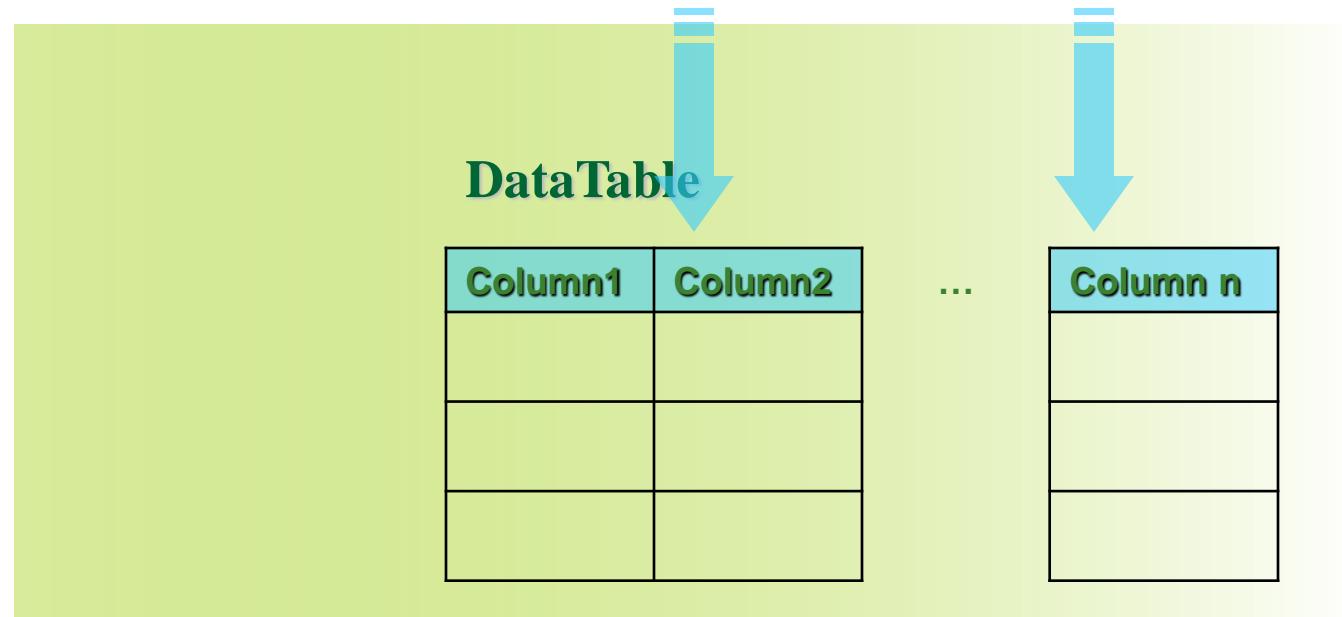
### Thuộc tính & Phương thức

- **TableName:** tên của bảng dữ liệu
- **Columns:** danh sách các cột
- **Rows:** danh sách các mẫu tin
- **PrimaryKey:** danh sách các cột là khóa chính
- **NewRow():** tạo một mẫu tin mới



# DataSet - DataColumn

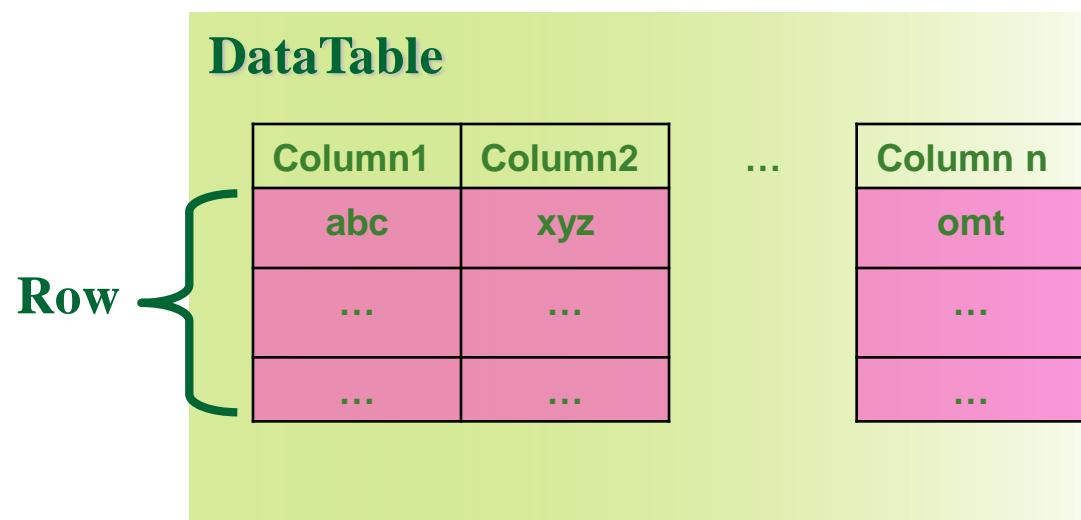
- **DataColumn: đại diện cho một cột trong bảng**
  - **ColumnName: tên cột**
  - **DataType: kiểu dữ liệu**





# DataSet - DataRow

- **DataRow: đại diện cho mẫu tin trong bảng**
  - RowState: trạng thái Added, Modified, Deleted,...
  - [i]: truy xuất đến cột i
  - Delete(): đánh dấu xóa mẫu tin





## DataSet – Update Row

```
string strConn = "data source=.\\"SQLEXPRESS; Initial  
Catalog=StudentDB; integrated security=yes;";  
string strCmd = "Select * From Sinhvien";  
SqlDataAdapter da = new SqlDataAdapter(strCmd, strConn);  
  
SqlCommandBuilder builder = new SqlCommandBuilder(da);  
DataSet ds = new DataSet();  
  
da.Fill(ds);  
  
foreach (DataRow dr in ds.Tables[0].Rows)  
    dr["Ngaysinh"] = DateTime.Now; ←————  
  
da.Update(ds);
```

VD: Cập nhật tất cả các  
cột với ngày giờ hiện  
hành



## DataSet – Delete Row

...

```
SqlDataAdapter da = new SqlDataAdapter(strCmd, strConn);  
SqlCommandBuilder builder = new SqlCommandBuilder(da);
```

```
DataSet ds = new DataSet();  
da.Fill(ds);
```

```
DataTable table = ds.Tables[0];
```

```
DataRow[] rows = table.Select("Ngaysinh<'1/1/1980'");
```

```
foreach (DataRow r in rows)  
    r.Delete(); ←  
da.Update(ds);
```

Lấy các dòng có  
ngày sinh trước  
1/1/1980

Đánh dấu xóa các  
dòng này

# CÂU HỎI NHANH

- Phương thức nào của Data Set dùng để cập nhật dữ liệu ?
- Có phải Dataset là bản sao của cơ sở dữ liệu hay không?
- Cấu trúc lưu trữ của Dataset được định dạng XML?
- Phương thức nào sau đây của Data Set dùng để cập nhật dữ liệu ?



# LINK YOUTUBE

- <https://www.youtube.com/watch?v=aoFDyt8oG0k&list=PL6n9fhu94yhX5dzHunAl2t4kE0kOuv4D7>
- <https://www.youtube.com/watch?v=RriFg1AFoT0>
- [https://www.youtube.com/watch?v= IiEd9xCLpo](https://www.youtube.com/watch?v=IiEd9xCLpo)
- <https://www.youtube.com/watch?v=2QMozMqBibk&list=PLUU6Q8aGTpKX-qilx1zvzXxE7R9xedrPs>



# Thank you!