



# BÀI 13: TRUY XUẤT CƠ SỞ DỮ LIỆU (ADO.NET)

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

# Giới thiệu ADO.NET

- ActiveX Data Object .NET (ADO.NET)
  - Công nghệ của MS trên .NET Framework
  - Phát triển từ nền tảng ADO
  - Cung cấp các **lớp đối tượng** và **hàm thư viện** phục vụ cho việc kết nối và xử lý dữ liệu



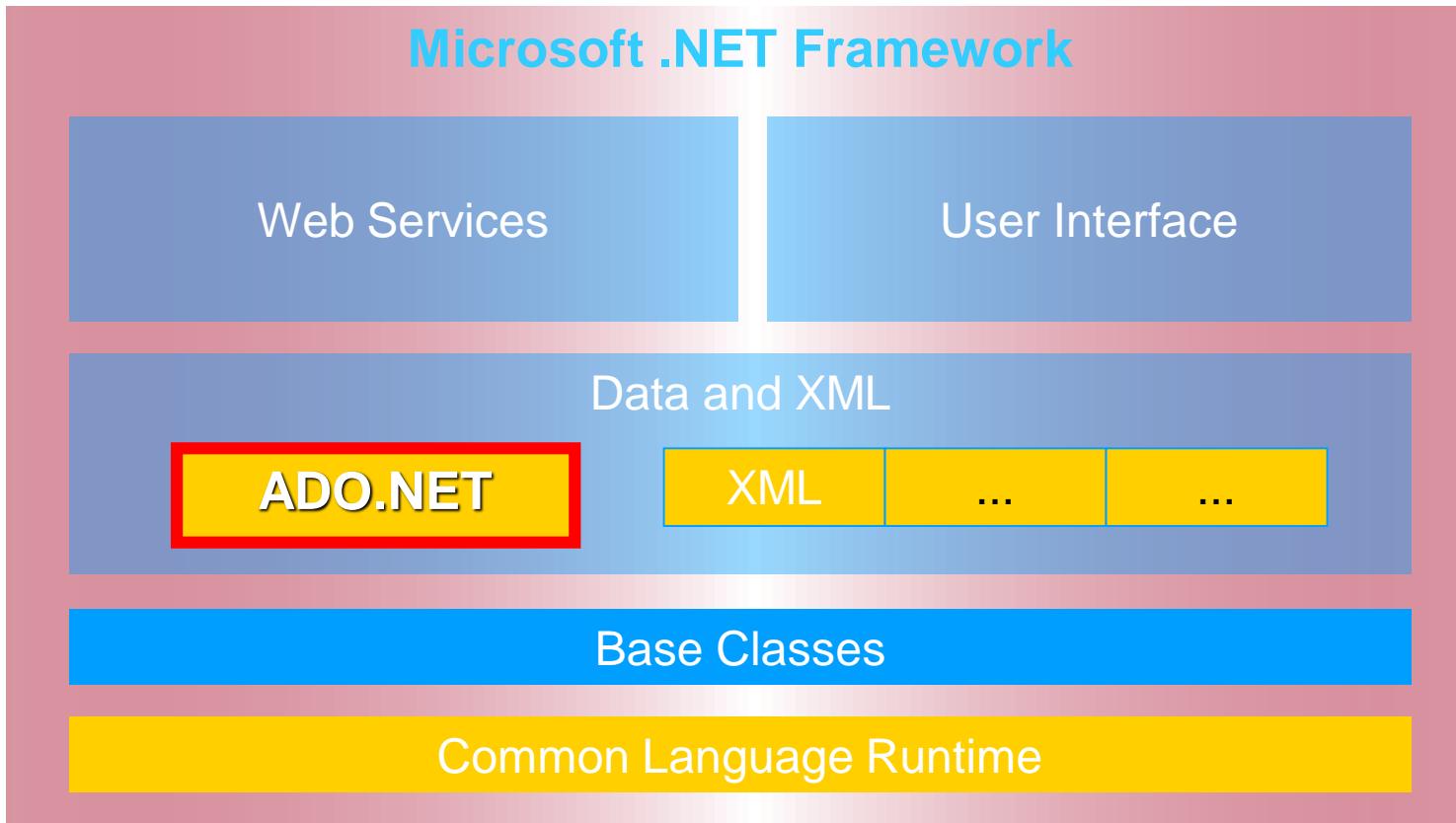
.NET Application

ADO.NET



# Giới thiệu ADO.NET

## ■ Mô hình .NET Framework





# Nội Dung

- ❑ ADO.NET
- ❑ **Đặc điểm ADO.NET**
- ❑ .NET Data Provider
- ❑ DataSet
- ❑ Data Source Configuration Wizard



- ADO.NET là một phần của .NET Framework
  - Thư viện lớp có chức năng **thao tác dữ liệu** trong ngôn ngữ MS.NET
- ADO.NET là dạng “***Disconnected***”
  - Cho phép lấy cả một cấu trúc phức tạp của DL từ CSDL, sau đó ngắt kết nối rồi mới thực hiện thao tác xử lý!
  - ADO (***Connected***) luôn phải duy trì kết nối trong suốt quá trình làm việc.



# Môi trường “connected”

- Mỗi user có một kết nối cố định tới data source
- **Ưu điểm**
  - Môi trường được bảo vệ tốt
  - Kiểm soát được sự đồng bộ
  - Dữ liệu luôn được mới
- **Nhược**
  - Phải có một kết nối mạng cố định



# Môi trường “disconnected”

- Một tập con của dữ liệu trung tâm được sao chép và bổ sung độc lập, sau đó sẽ được merge lại vào dữ liệu trung tâm.
- **Ưu điểm**
  - Có thể làm việc bất cứ lúc nào, cũng như có thể kết nối bất kỳ vào Data Source
  - Cho phép user khác có thể kết nối
  - Nâng cao hiệu suất thực hiện của ứng dụng
- **Khuyết**
  - Dữ liệu không được cập nhật một cách nhanh nhất
  - Sự tranh chấp có thể xuất hiện và phải giải quyết



- **ADO.NET mạnh mẽ**
  - Kế thừa các ưu điểm của ADO
  - Kết hợp với ý tưởng thiết kế hoàn toàn mới
- **Đặc điểm nổi bật**
  - Thiết kế hoàn toàn dựa vào XML
    - Chuẩn giao tiếp dữ liệu phổ biến nhất trên môi trường Internet hiện nay
  - Thiết kế hoàn toàn hướng đối tượng
    - Đặc trưng của thư viện .NET Framework



# ADO.NET vs. ADO

Đặc Điểm	ADO	ADO.NET
Data xử lý được đưa vào bộ nhớ dưới dạng	Recordset : tương đương 1 bảng dữ liệu trong database	Dataset : tương đương 1 database
Duyệt dữ liệu	Recordset chỉ cho phép duyệt tuần tự, từng dòng một.	Dataset : duyệt “tự do, ngẫu nhiên”, truy cập thẳng tới bảng, dòng, cột mong muốn.
Dữ liệu ngắt kết nối	Recordset thiên về hướng kết nối, nên việc hỗ trợ ngắt kết nối không mạnh	Dataset hỗ trợ hoàn toàn ngắt kết nối
Trao đổi dữ liệu qua Internet	Khả năng trao đổi dữ liệu ADO qua Internet thường có nhiều hạn chế. Do dùng chuẩn COM	ADO.NET trao đổi dữ liệu qua Internet rất dễ dàng vì ADO.NET được thiết kế theo chuẩn XML, là chuẩn dữ liệu chính được sử dụng để trao đổi trên Internet.



- **Kiến trúc của ADO.NET gồm 2 phần chính**
- **Phần kết nối:** sử dụng khi kết nối CSDL và thao tác dữ liệu, phải thực hiện kết nối khi thao tác
  - **Connection:** quản lý việc đóng mở DB
    - ???Connection: `SqlConnection`, `OleDbConnection`
  - **Command:** lệnh truy vấn, tương tác dữ liệu khi đang lập kết nối
    - ???Command: `SqlCommand`, `OleDbCommand`
  - **DataReader:** đọc dữ liệu, chỉ xử lý 1 dòng dữ liệu tại một thời điểm
    - ???DataReader: `SqlDataReader`, `OleDbDataReader`
  - **DataAdapter:** cầu nối giữa DB và DataSet

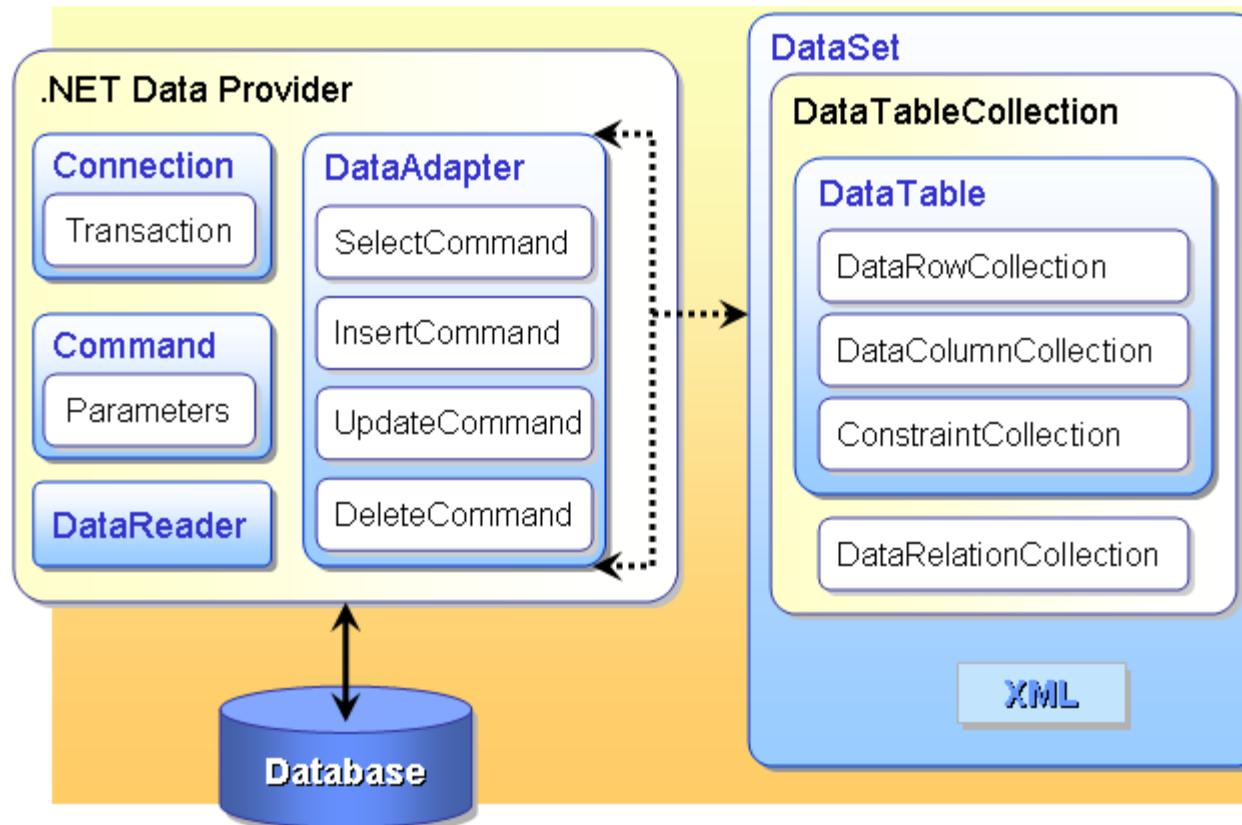


## ■ **Phần ngắt kết nối:** là **DataSet**

- **DataSet không quan tâm đến DB thuộc kiểu gì, và lấy dữ liệu từ DataAdapter để xử lý**
- **DataSet xem như một DB trong bộ nhớ: bảng, quan hệ...**
- **DataSet có các thành phần con như**
  - **DataTable**
  - **DataRow**
  - **DataColumn**
  - **DataRelation**
  - **Các đối tượng nhóm: DataTableCollection, DataRowCollection, DataColumnCollection**



# Mô hình đối tượng ADO.NET





# Các lớp thư viện ADO.NET

- **System.Data.OleDb:** Access, SQL Server, Oracle
- **System.Data.SqlClient:** SQL Server
- **System.Data.OracleClient:** Oracle
- **Đặc điểm:**
  - Cả ba thư viện trên về giao tiếp lập trình là giống nhau
  - Dùng thư viện SqlCommand truy xuất SQL Server nhanh hơn OleDb
  - Tương tự cho OracleClient



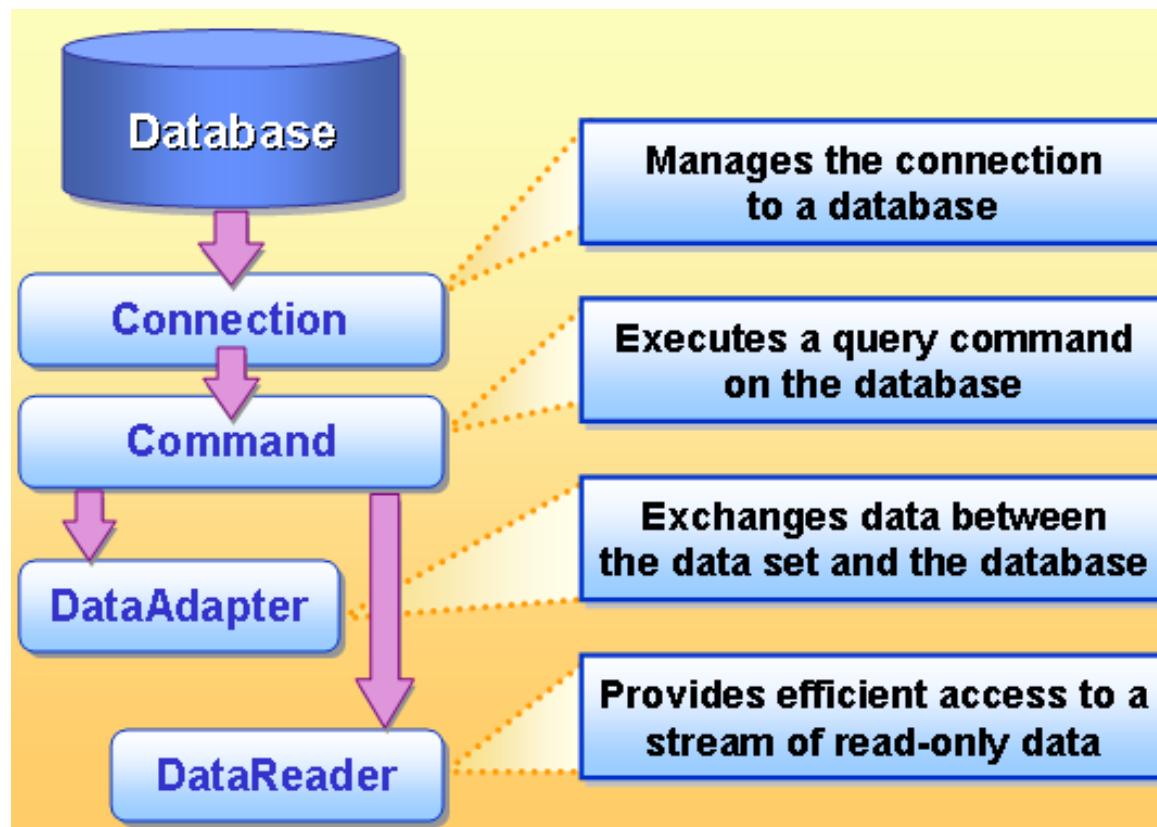
# Nội Dung

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



# NET Data Provider

## ■ Mô hình .NET Data Provider

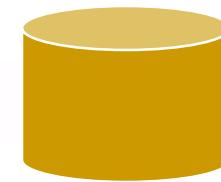


# NET Data Provider - Connection



Application

Connection



Database

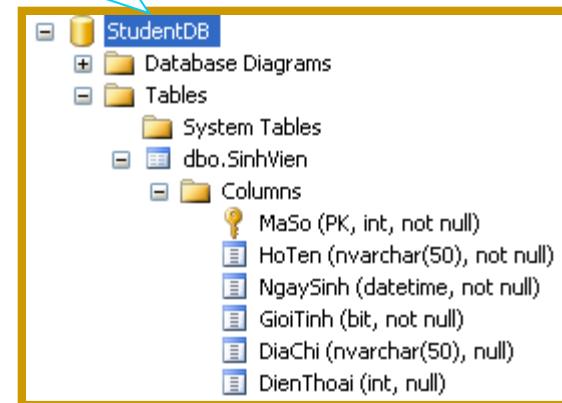
## Thuộc tính & Phương thức

- **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

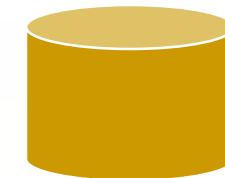


Application

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

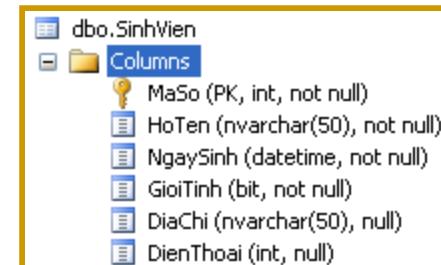
```
SqlConnection conn = new SqlConnection();
conn.ConnectionString = "Data source=.\SQLEXPRESS;"+
    "Initial Catalog=StudentDB; Integrated security=yes;";

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();

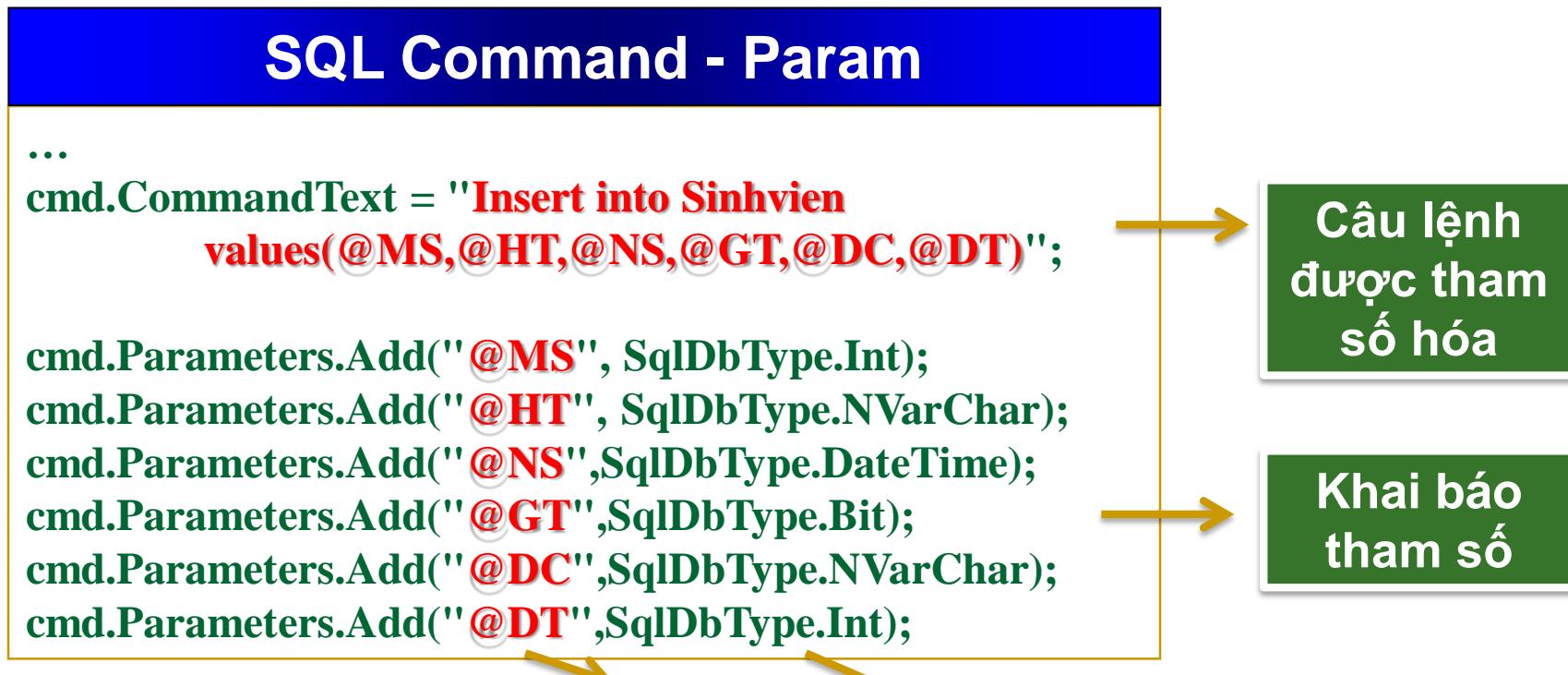
conn.Close();
```





# 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();
```

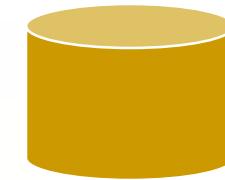
Truyền giá  
trị cho các  
tham số  
tương ứng



# 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();
```

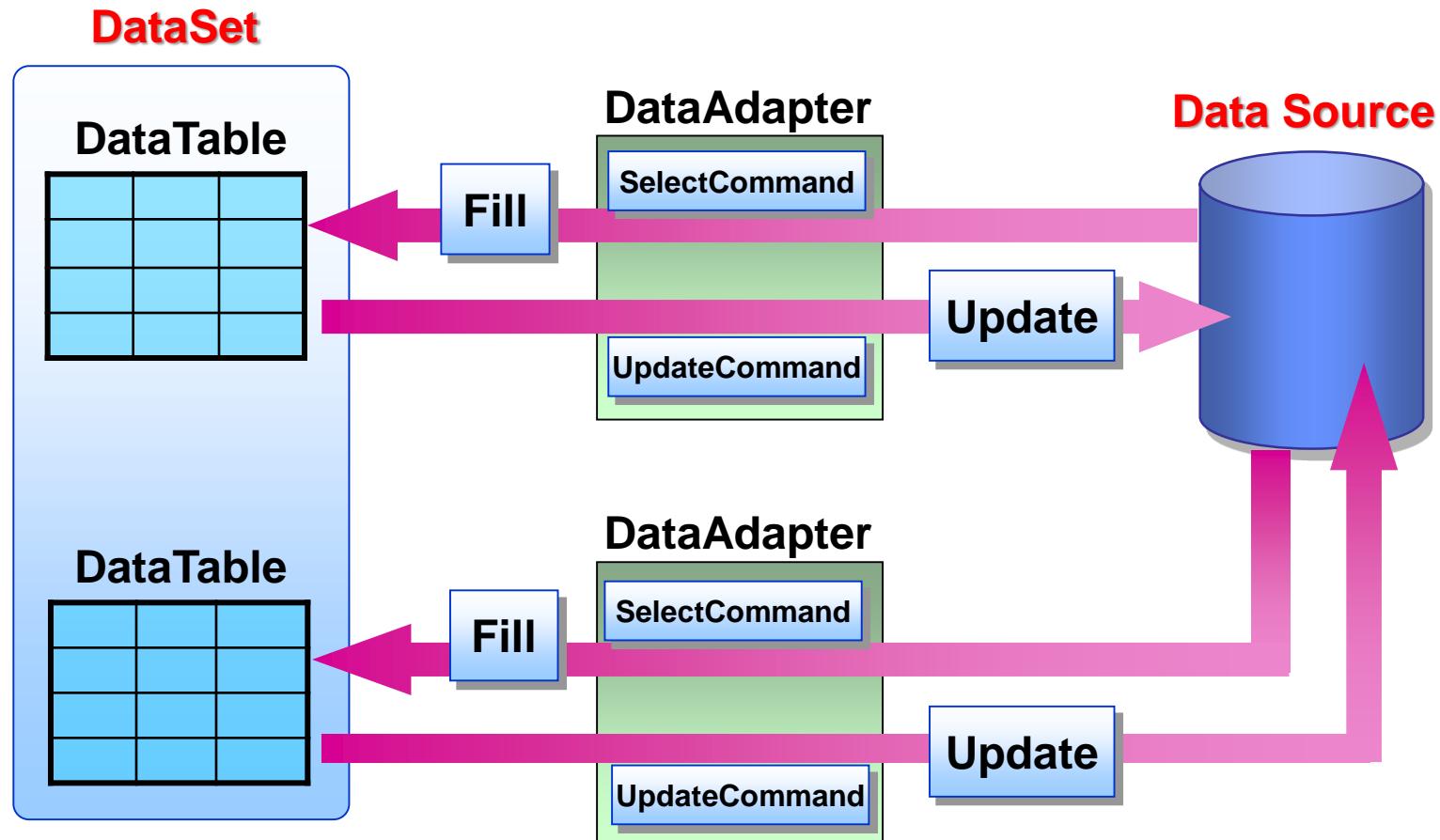
Thực thi lệnh truy vấn

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);
```

# CÂU HỎI NHANH

- Cho biết các thành phần nào không phải là thuộc tính của đối tượng Data Adapter ?
- Một điều khiển DataAdapter dùng phương thức nào để nạp một DataSet ?
- Trong một ứng dụng ASP.Net, khi dùng đối tượng SqlCommand để gọi stored procedure tạo một bảng CSDL trong MS SQL Server 2005 hoặc cao hơn, chúng ta gọi phương thức nào?
- Cho biết các thuộc tính của đối tượng Data Adapter?



# 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!