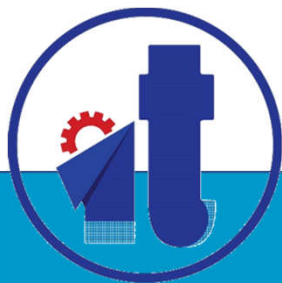




# LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS





## Chương 4:

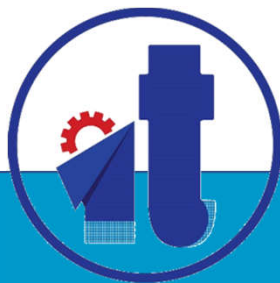
## ADO.NET





# Nội dung

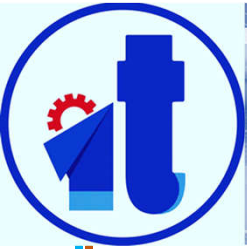
- ☐ Giới thiệu về ADO.NET
  - .NET Framework Data Provider
  - DataSet
- ☐ Mô hình tương tác CSDL trong ứng dụng .NET
  - Mô hình ngắt kết nối
  - Mô hình kết nối
- ☐ Combobox
- ☐ DataGridView



# Giới thiệu về ADO.NET







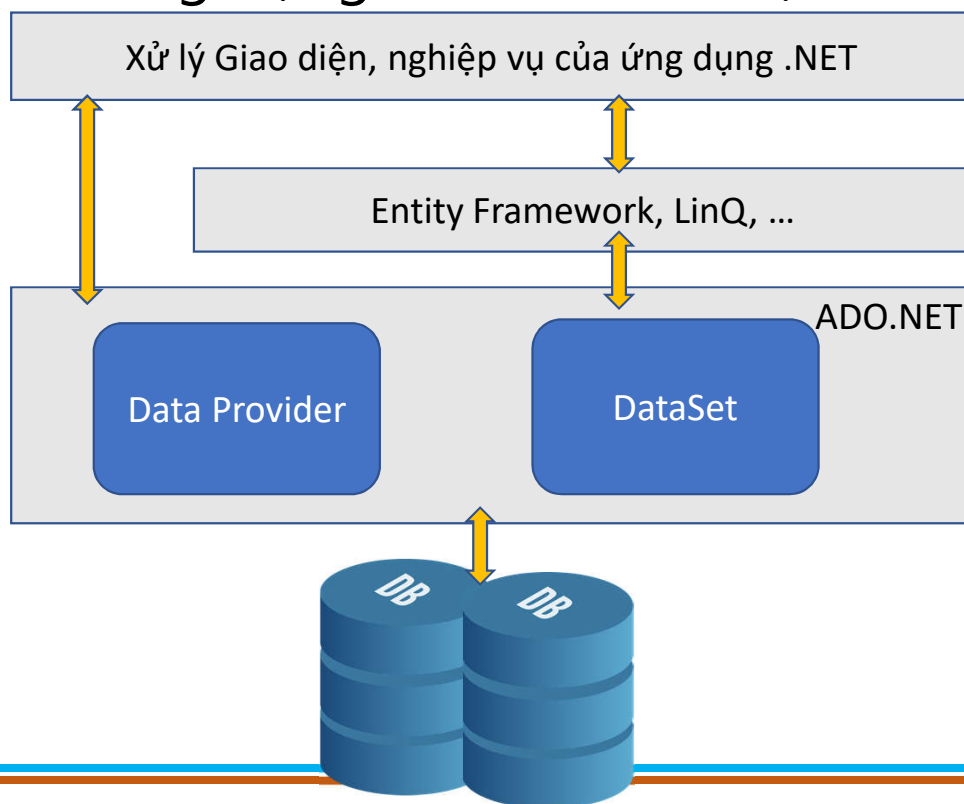
# ADO.NET

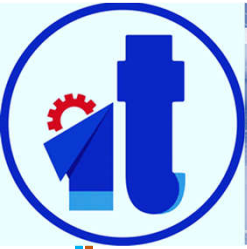
- ❑ Là một thành phần của .NET Framework
- ❑ Là một bộ các thư viện hướng đối tượng cho phép bạn tương tác với dữ liệu nguồn
- ❑ Dữ liệu nguồn:
  - Tập tin dạng text, xml, excel...
  - Cơ sở dữ liệu: MS SQL Server, MS Access, Oracle,...
- ❑ Hỗ trợ mô hình ngắt kết nối (*disconnected model*)
- ❑ Sử dụng XML để tương tác với cơ sở dữ liệu, nghĩa là dữ liệu trong CSDL được chuyển sang định dạng XML để thực hiện các thao tác truy vấn, cập nhật cơ sở dữ liệu.



# Tổ chức

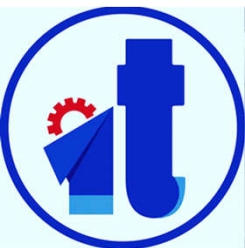
- ❑ Là cầu nối của ứng dụng với cơ sở dữ liệu



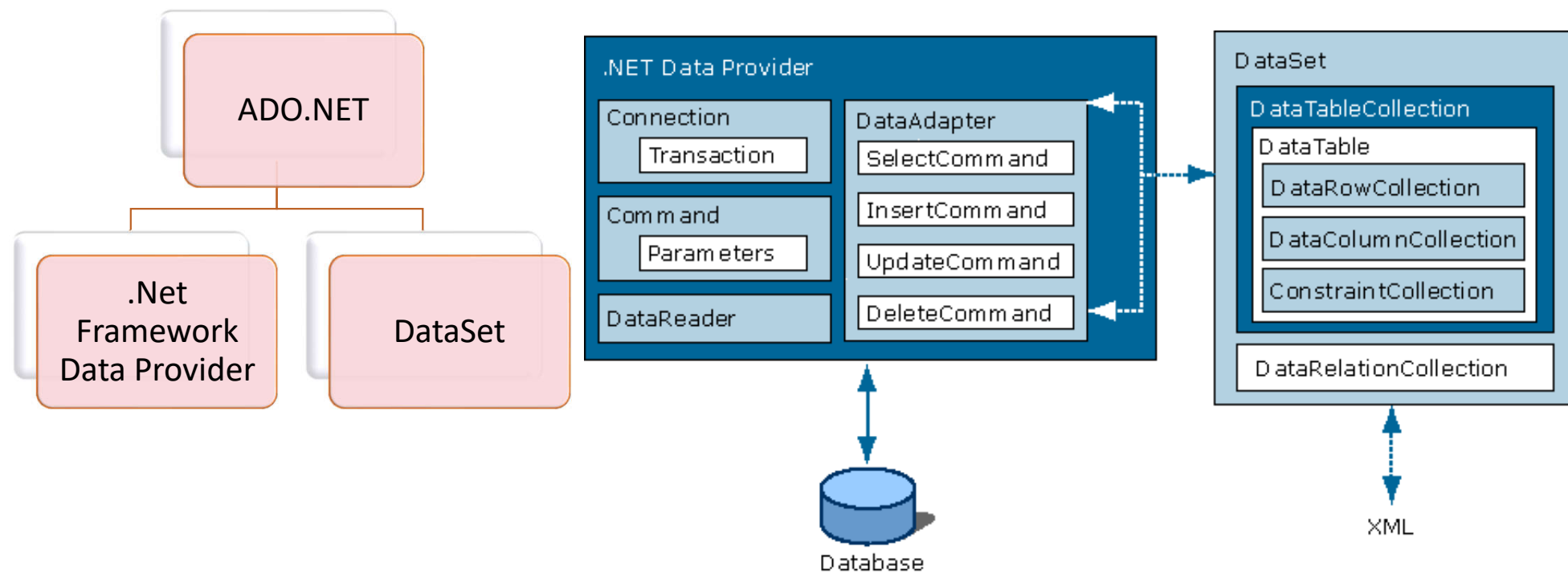


# ADO vs ADO.NET

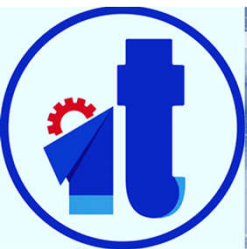
- ❑ ADO.NET không phải phát triển từ ADO
- ❑ ADO:
  - Dựa vào công nghệ COM
  - Sử dụng OLEDB data provider để truy xuất dữ liệu
  - Sử dụng mô hình kiến trúc kết nối dữ liệu (giữ kết nối với CSDL trong quá trình thao tác: xem, cập nhật,...)
- ❑ ADO.NET
  - Là thành phần của .Net Framework
  - Sử dụng cấu trúc XML để chuyển dữ liệu giữa CSDL với ứng dụng
  - Sử dụng kiến trúc ngắt kết nối



# Kiến trúc





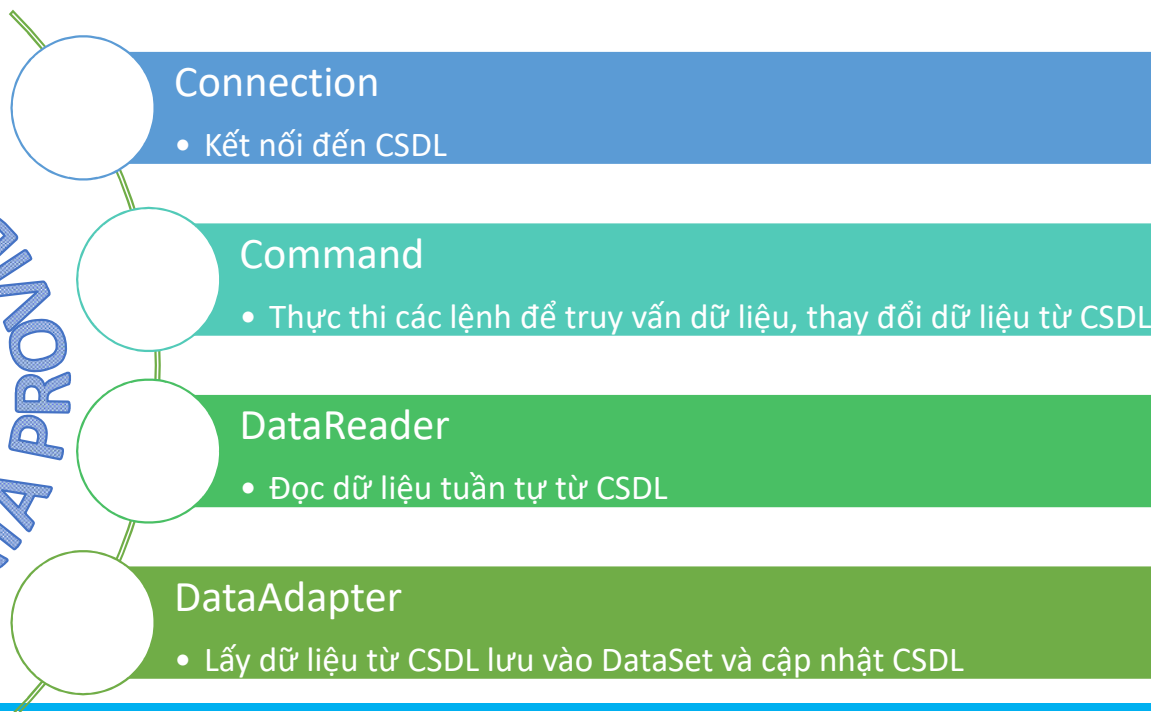


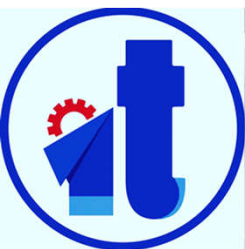
## .NET FRAMEWORK DATA PROVIDER

- ❑ Cung cấp các lớp dùng để kết nối, thao tác thực thi, lấy kết quả từ CSDL
- ❑ Các thành phần

.NET FRAMEWORK

DATA PROVIDER





## .NET FRAMEWORK DATA PROVIDER

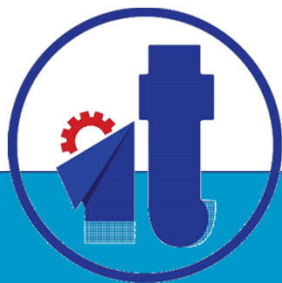
❑ Ứng với tùy loại CSDL có các .NET Framework Data Provider

| Loại CSDL                         | Tên Provider                                | Namespace                                    |
|-----------------------------------|---|--|
| ODBC                              | .Net Framework Data Provider for ODBC       | System.Data.Odbc<br>(prefix: Odbc)           |
| Access, Excel, MS SQL Server, ... | .Net Framework Data Provider for OLEDB      | System.Data.OleDb<br>(prefix: OleDb)         |
| MS SQL Server                     | .Net Framework Data Provider for SQL Server | System.Data.SqlClient<br>(prefix: Sql)       |
| Oracle                            | .Net Framework Data Provider for Oracle     | System.Data.OracleClient<br>(prefix: Oracle) |



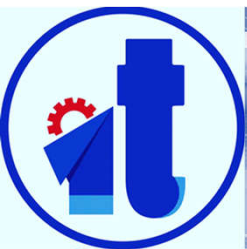
# DATASET

- ❑ Hỗ trợ mô hình truy cập ngắt lết nối
  - Kết nối CSDL chỉ được mở khi lấy dữ liệu về dataset, cập nhật dữ liệu xuống CSDL
- ❑ Là bản sao của CSDL hoặc một phần của CSDL
- ❑ Không phụ thuộc vào loại CSDL
- ❑ Hai thành phần:
  - DataTableCollection: tập hợp các DataTable
  - DataRelationCollection: tập hợp các mối quan hệ giữa các DataTable



# Mô hình tương tác CSDL trong ứng dụng .NET

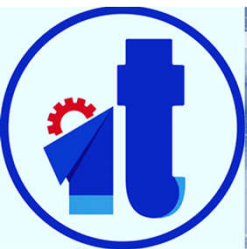




# Mô hình tương tác CSDL

- ❑ Mô hình kết nối (connected)
  - Tạo và kết nối đến CSDL
  - Thao tác với CSDL
  - Đóng kết nối
- ❑ Mô hình ngắt kết nối (disconnected)
  - Tạo bản sao CSDL vào bộ nhớ ở Client
  - Thao tác trên bản sao
  - Cập nhật bản sao về CSDL





# Mô hình tương tác CSDL

## Connected

Ứng dụng .NET

- 1 • Mở kết nối
- 2 • Thực thi lệnh
- 3 • Duyệt kết quả
- 4 • Đóng kết nối

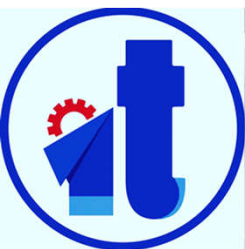


## Disconnected

Ứng dụng .NET

- 1 • Mở kết nối
- 2 • Lưu dữ liệu trên Client
- 3 • Đóng kết nối
- 4 • Thao tác dữ liệu trên Client
- 5 • Mở kết nối
- 6 • Cập nhật thay đổi
- 7 • Đóng kết nối





## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Connection

➤ Dùng để tạo một kết nối giữa ứng dụng với cơ sở dữ liệu

| Tên Provider                                       | Class Connection | <b>XYZConnection</b> |
|--|------------------|----------------------|
| .Net Framework Data Provider for <b>ODBC</b>       | OdbcConnection   |                      |
| .Net Framework Data Provider for <b>OleDB</b>      | OleDbConnection  |                      |
| .Net Framework Data Provider for <b>SQL Server</b> | SqlConnection    |                      |
| .Net Framework Data Provider for <b>Oracle</b>     | OracleConnection |                      |



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Connection

#### ➤ Khai báo

- ✓ Cách 1: `XYZConnection <tên_biến> = new XYZConnection();`
- ✓ Cách 2: `XYZConnection <tên_biến> = new XYZConnection(<chuỗi_kết_nối>);`

#### ➤ Chuỗi kết nối

- ✓ Mỗi HQT CSDL khác nhau sẽ có chuỗi kết nối khác nhau
- ✓ Access 2007

`Provider=Microsoft.ACE.OLEDB.12.0; Data Source = C:\myFolder\myAccessFile.accdb;`

- ✓ MS SQL Server

`Data Source=myServerAddress;Initial Catalog=myDataBase;Integrated Security=true;  
User ID=myUsername;Password=myPassword;`

<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-syntax>



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Connection

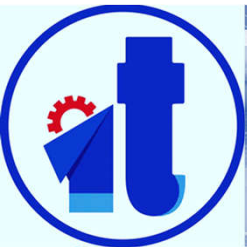
#### ➤ Một số thuộc tính

##### ✓ ConnectionString

- Kiểu dữ liệu: `string`
- Chuỗi kết nối đến cơ sở dữ liệu

##### ✓ State

- Kiểu dữ liệu: `ConnectionState`
- Trạng thái của đối tượng Connection (Broken, Closed, Connecting, Executing, Fetching, Open)



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Connection

#### ➤ Một số phương thức

##### ✓ Open()

- Cú pháp: `public void Open()`
- Mở kết nối đến cơ sở dữ liệu

##### ✓ Close()

- Cú pháp: `public void Close()`
- Đóng kết nối cơ sở dữ liệu





## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Connection

➤ Ví dụ: kết nối đến CSDL DemoDB

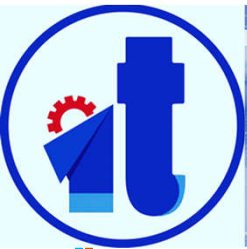
```
//Chuỗi kết nối
string strChuoiKetNoi = "Data Source=NguyenLe-PC; Initial CataLog=DemoDB;Integrated Security=true";

//Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection(strChuoiKetNoi);

//Mở kết nối
conn.Open();

//Thực hiện các thao tác dữ liệu: truy vấn, cập nhật
//...

//Đóng kết nối
conn.Close();
```



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Command

- Dùng để thực thi câu lệnh SQL như SELECT, INSERT, UPDATE, DELETE, hoặc một Store Procedure

| Tên Provider                                       | Class Command | XYZCommand |
|--|---------------|------------|
| .Net Framework Data Provider for <b>ODBC</b>       | OdbcCommand   |            |
| .Net Framework Data Provider for <b>OleDB</b>      | OleDbCommand  |            |
| .Net Framework Data Provider for <b>SQL Server</b> | SqlCommand    |            |
| .Net Framework Data Provider for <b>Oracle</b>     | OracleCommand |            |



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Command

#### ➤ Khai báo:

- ✓ Cách 1: `XYZCommand <tên_biến> = new XYZCommand ();`
- ✓ Cách 2: `XYZCommand <tên_biến> = new XYZCommand(<câu_lệnh_SQL>, <đối_tượng_Connection>);`

#### ➤ Một số thuộc tính

##### ✓ CommandText

- Kiểu dữ liệu: `string`
- Câu lệnh SQL hoặc Store Procedure để thực thi trên CSDL

##### ✓ Connection

- Kiểu dữ liệu: `XYZConnection`
- Đối tượng Connection được sử dụng bởi đối tượng Command



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Command

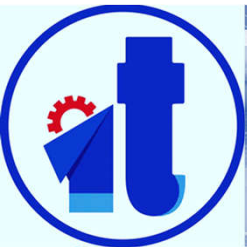
➤ Một số thuộc tính (tt)

#### ✓ CommandType

- Kiểu dữ liệu: `CommandType`
- Kiểu câu lệnh mà đối tượng Command sẽ thực thi (Text, StoreProcedure)

#### ✓ Parameters

- Kiểu dữ liệu: `XYZParameterCollection`
- Danh sách các tham số trong câu lệnh SQL. Các tham số trong câu lệnh SQL có thể được thêm (Add), xóa (Remove), ...



## Đối tượng kết nối, thực thi CSDL

### ❑ Đối tượng Command

#### ➤ Một số phương thức

##### ✓ ExecuteNonQuery()

- Cú pháp: `public int ExecuteNonQuery()`
- Thực thi các câu lệnh SQL INSERT, UPDATE, DELETE. Kết quả trả về là số dòng được thực thi

##### ✓ ExecuteScalar()

- Cú pháp: `public object ExecuteScalar()`
- Thực thi câu lệnh SQL SELECT. Kết quả trả về một giá trị duy nhất (giá trị của cột đầu tiên trong dòng đầu tiên của kết quả truy vấn) có kiểu object





## Đối tượng kết nối, thực thi CSDL

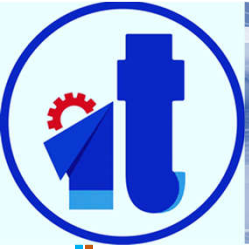
### ❑ Đối tượng Command

➤ Một số phương thức(tt)

✓ ExecuteReader()

○ Cú pháp: `public XYZDataReader ExecuteReader()`

○ Thực thi câu lệnh SQL SELECT. Kết quả trả về là đối tượng XYZDataReader chứa kết quả truy vấn SELECT



## Đối tượng kết nối, thực thi CSDL

## ❑ Đối tượng Command

- Ôn tập lại các câu lệnh SQL
  - ✓ INSERT INTO <bảng>(<danh\_sách\_cột>) VALUES (<danh\_sách\_giá\_trị>)
  - ✓ UPDATE <bảng> SET <cột\_1> = <giá\_trị\_1>, <cột\_2> = <giá\_trị\_2>, ... , <cột\_N> = <giá\_trị\_N> WHERE <điều\_kiện>
  - ✓ DELETE FROM <bảng> WHERE <điều\_kiện>
  - ✓ SELECT <danh\_sách\_cột> FROM <bảng> WHERE <điều\_kiện>
  - ✓ Truy vấn lồng, kết bảng
  - ✓ Các hàm thống kê: COUNT, MIN, MAX, SUM, AVERAGE...
  - ✓ Mệnh đề ORDER, HAVING
  - ✓ ...



# Đối tượng kết nối, thực thi CSDL

## ❑ Đối tượng Command

➤ Ví dụ:

### SINH\_VIEN

|   | Column Name | Data Type    |
|---|-------------|--------------|
| 🔑 | MSSV        | nvarchar(10) |
|   | HoTen       | nvarchar(50) |
|   | NgaySinh    | date         |
|   | GioiTinh    | bit          |
|   | DiemTB      | float        |



# Đối tượng kết nối, thực thi CSDL

## ❑ Đối tượng Command

```
//B1: Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection(strChuoiKetNoi);

//B2: Mở kết nối
conn.Open();

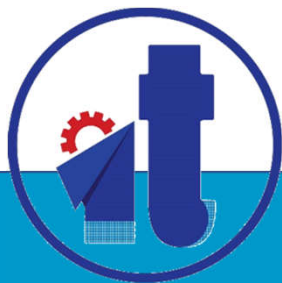
//B3: Thực hiện các thao tác dữ liệu: truy vấn, cập nhật
string strThemMoi = string.Format("Insert Into SINH_VIEN(MSSV,HoTen,NgaySinh,GioiTinh,DiemTB) Values ({0}', '{1}', '{2}', {3}, {4})",
    txtMSSV.Text, txtHoTen.Text, dtpNgaySinh.Value.ToString("yyyy-MM-dd"), (radNam.Checked ? 1 : 0), nudDiem.Value);

SqlCommand com = new SqlCommand(strThemMoi, conn);

int iNumOfRow = com.ExecuteNonQuery();

if(iNumOfRow > 0)
{
    MessageBox.Show("Thêm sinh viên thành công", "Thêm Mới");
}
else
{
    MessageBox.Show("Thêm không thành công", "Thêm Mới");
}

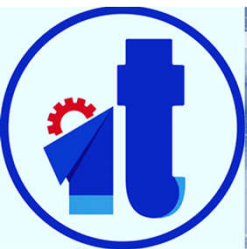
//B4: Đóng kết nối
conn.Close();
```



# Mô hình ngắt kết nối (Disconnected)

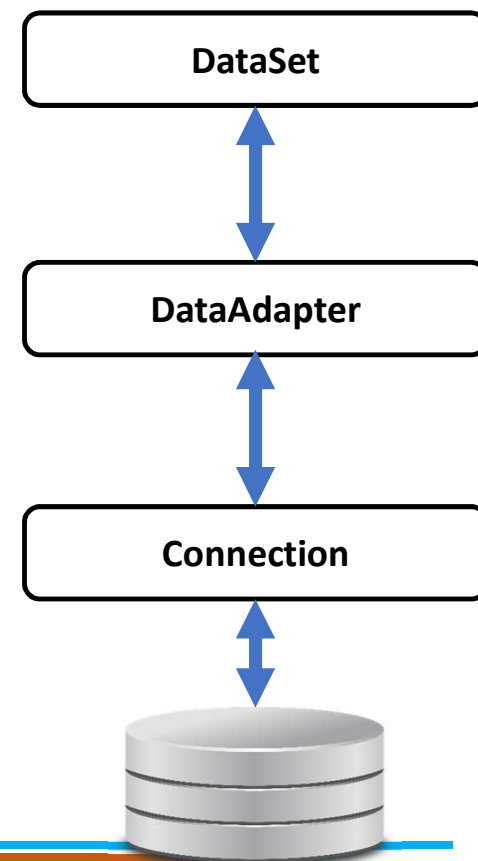


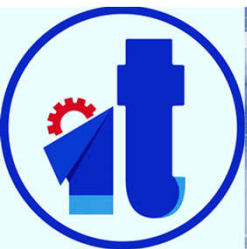




## Mô hình ngắt kết nối

- ☐ Mở kết nối
- ☐ Đổ dữ liệu vào dataset
- ☐ Đóng kết nối
- ☐ Thao tác với dữ liệu ở dataset
- ☐ Mở kết nối
- ☐ Cập nhật DL từ dataset vào DL nguồn
- ☐ Đóng kết nối





# Đối tượng DataAdapter

- ❑ Đọc dữ liệu từ cơ sở dữ liệu vào DataSet
- ❑ Cập nhật dữ từ DataSet ra cơ sở dữ liệu

| Tên Provider                                       | Class Command     | XYZDataAdapter |
|--|-------------------|----------------|
| .Net Framework Data Provider for <b>ODBC</b>       | OdbcDataAdapter   |                |
| .Net Framework Data Provider for <b>OleDB</b>      | OleDbDataAdapter  |                |
| .Net Framework Data Provider for <b>SQL Server</b> | SqlDataAdapter    |                |
| .Net Framework Data Provider for <b>Oracle</b>     | OracleDataAdapter |                |



# Đối tượng DataAdapter

## ❑ Khởi tạo đối tượng

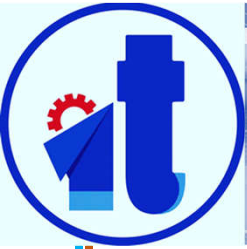
- Cách 1: `XyzDataAdapter <tên_biến> = new XyzDataAdapter();`
- Cách 2: `XyzDataAdapter <tên_biến> = new XyzDataAdapter  
(<câu_lệnh_SQL_SELECT>, <chuỗi_kết_nối>);`
- Cách 3: `XyzDataAdapter <tên_biến> = new XyzDataAdapter  
(<câu_lệnh_SQL_SELECT>, <đối_tượng_Connection>);`



# Đối tượng DataAdapter

## ❑ Các thuộc tính

- SelectCommand,
- InsertCommand,
- UpdateCommand,
- DeleteCommand
  - ✓ Kiểu dữ liệu: `XYZCommand`
  - ✓ Đối tượng thực thi câu lệnh SQL hoặc Store Procedure truy vấn, thêm, cập nhật, xóa dữ liệu



# Đối tượng DataAdapter

## ❑ Các Phương thức

### ➤ Fill()

- ✓ `public int Fill(DataSet dataSet)`
- ✓ `public int Fill(DataSet dataSet, string srcTable)`
- ✓ `public int Fill(DataSet dataSet, int startRecord, int maxRecords, string srcTable)`
- ✓ Đọc dữ liệu từ cơ sở dữ liệu và đổ vào một DataTable trong DataSet.
- ✓ Kết quả trả về số dòng dữ liệu được đổ vào DataSet



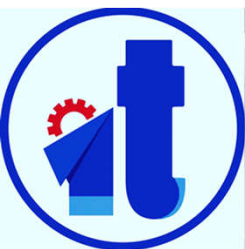


# Đối tượng DataAdapter

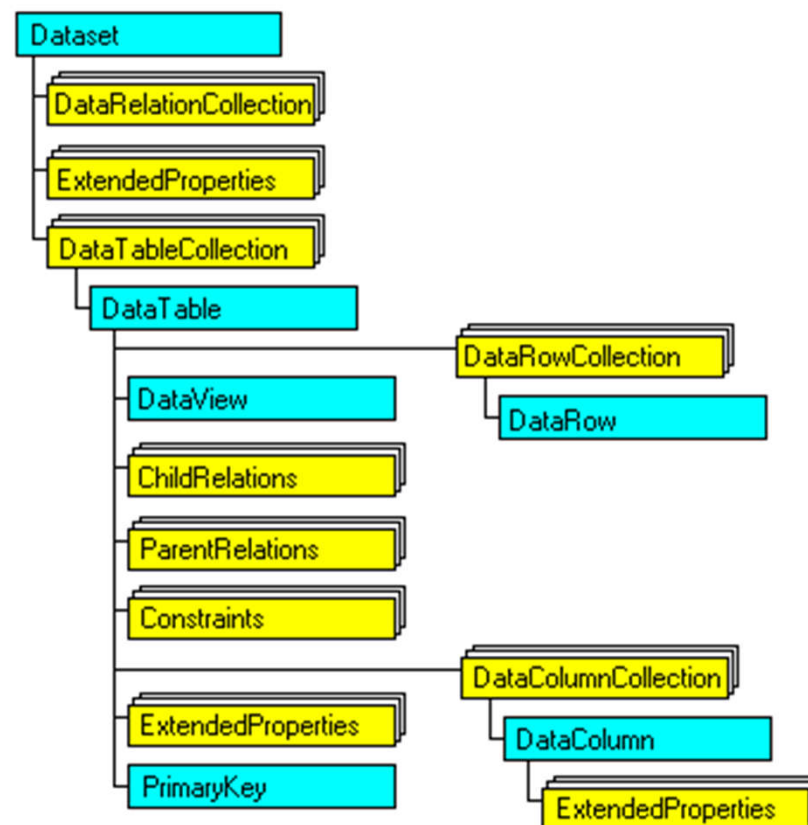
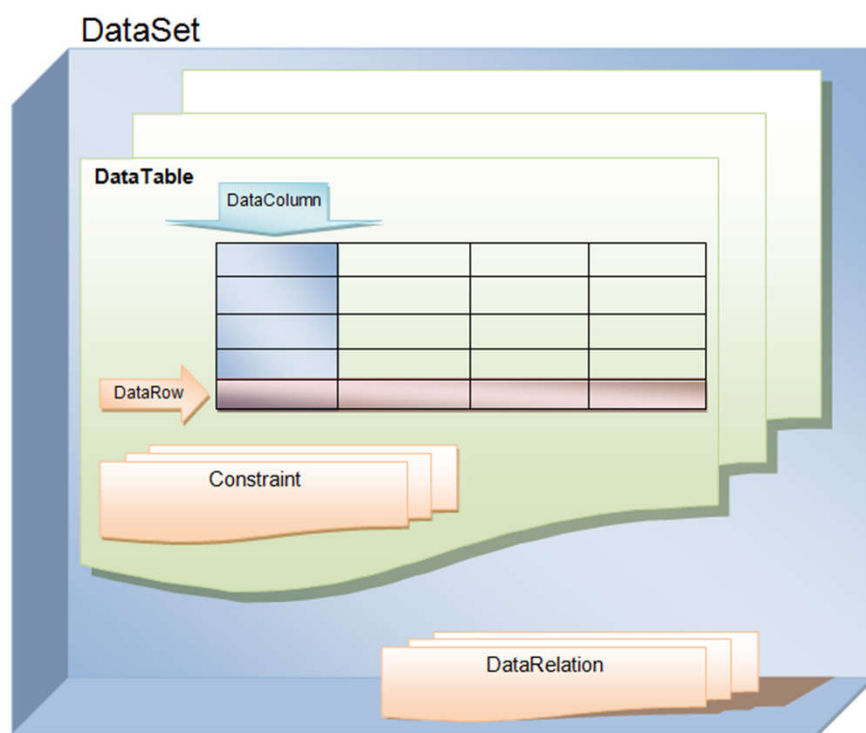
## ❑ Các Phương thức

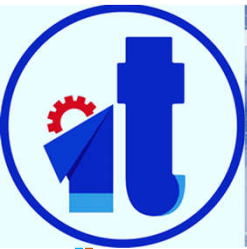
### ➤ Update()

- ✓ `public int Update(DataSet dataSet)`
- ✓ `public int Update(DataTable dataTable)`
- ✓ `public int Update(DataSet dataSet, string srcTable)`
- ✓ Cập nhật các thay đổi của dữ liệu trong DataSet (DataTable) ra cơ sở dữ liệu



# Đối tượng DataSet





## Đối tượng DataSet

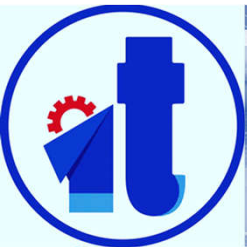
- ☐ Thuộc namespace System.Data
- ☐ Dữ liệu lưu trong **DataSet** được tổ chức dưới dạng một tập các bảng (**DataTable**)
- ☐ Mỗi dòng được biểu diễn bởi **DataRow**
- ☐ Mỗi cột được biểu diễn bởi **DataColumn**.
- ☐ Đối tượng **DataRelation** để biểu diễn mối quan hệ giữa các **DataTable**.
- ☐ Đối tượng **Constraint** để thiết lập ràng buộc trên **DataTable** để đảm bảo tính toàn vẹn dữ liệu.



# Đối tượng DataSet

## ❑ Khởi tạo

- `DataSet <tên_biến> = new DataSet();`
- `DataSet <tên_biến> = new DataSet(<tên_DataSet>);`
- Lưu ý: với cách 1 thì `DataSet` được tạo sẽ có tên là `NewDataSet`



# Đối tượng DataSet

## ❑ Các thuộc tính

### ➤ DataSetName

- ✓ Kiểu dữ liệu: `string`
- ✓ Thiết lập/lấy tên của DataSet

### ➤ Tables

- ✓ Kiểu dữ liệu: `DataTableCollection`
- ✓ Danh sách các DataTable của DataSet

### ➤ Relations

- ✓ Kiểu dữ liệu: `DataRelationCollection`
- ✓ Tập hợp các mối liên kết giữa các DataTable trong DataSet





# Đối tượng DataSet

## ❑ Phương thức

### ➤ AcceptChanges()

- ✓ Cú pháp: `public void AcceptChanges()`
- ✓ Thực hiện các thay đổi dữ liệu trên DataSet

### ➤ RejectChanges()

- ✓ Cú pháp: `public void RejectChanges()`
- ✓ Hủy bỏ các thay đổi dữ liệu trên DataSet

### ➤ Clear()

- ✓ Cú pháp: `public void Clear()`
- ✓ Xóa tất cả các dòng dữ liệu trong các DataTable trong DataSet



# Đối tượng DataSet

## ❑ Phương thức (tt)

### ➤ ReadXML()

- ✓ Cú pháp: `public XMLReadMode ReadXML (string filename)`
- ✓ Đọc dữ liệu từ tập tin XML vào DataSet

### ➤ WriteXML()

- ✓ Cú pháp: `public void WriteXML(string filename)`
- ✓ Ghi dữ liệu từ DataSet ra tập tin XML



# Đối tượng DataTable

- ❑ Biểu diễn một bảng trong DataSet
- ❑ Lưu dữ liệu từ dữ liệu nguồn
- ❑ Nằm trong namespace System.Data
- ❑ **Khởi tạo**
  - Cách 1: `DataTable <tên_biến> = new DataTable();`
  - Cách 2: `DataTable <tên_biến> = new DataTable(<tên_DataTable>);`



# Đối tượng DataTable

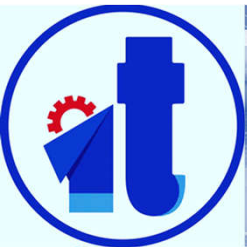
## ❑ Thuộc tính

### ➤ Columns

- ✓ Kiểu dữ liệu: `DataColumnCollection`
- ✓ Danh sách các cột trong DataTable

### ➤ DataSet

- ✓ Kiểu dữ liệu: `DataSet`
- ✓ Đối tượng DataSet chứa DataTable hiện hành



# Đối tượng DataTable

## ❑ Thuộc tính (tt)

### ➤ DefaultValue

- ✓ Kiểu dữ liệu: `DataRow`
- ✓ Đối tượng `DataRow` của `DataTable` hiện hành

### ➤ PrimaryKey

- ✓ Kiểu dữ liệu: `DataRow[]`
- ✓ Danh sách các cột làm khóa chính của `DataTable`





# Đối tượng DataTable

## ❑ Thuộc tính (tt)

### ➤ Rows

- ✓ Kiểu dữ liệu: `DataRowCollection`
- ✓ Danh sách các dòng dữ liệu trong DataTable

### ➤ TableName

- ✓ Kiểu dữ liệu: `string`
- ✓ Tên của DataTable



# Đối tượng DataTable

## ❑ Phương thức

### ➤ AcceptChanges()

- ✓ Cú pháp: `public void AcceptChanges()`
- ✓ Thực hiện các thay đổi dữ liệu trên DataTable

### ➤ RejectChanges()

- ✓ Cú pháp: `public void RejectChanges()`
- ✓ Hủy bỏ các thay đổi dữ liệu trên DataTable



# Đối tượng DataTable

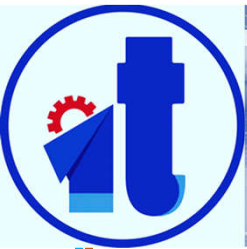
## ❑ Phương thức (tt)

### ➤ Clear()

- ✓ Cú pháp: `public void Clear()`
- ✓ Xóa tất cả các dòng dữ liệu trong DataTable

### ➤ Copy()

- ✓ Cú pháp: `public DataTable Copy()`
- ✓ Sao chép cấu trúc và dữ liệu của DataTable hiện hành



# Đối tượng DataTable

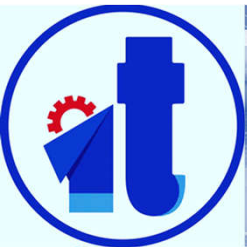
## ❑ Phương thức (tt)

### ➤ GetChanges()

- ✓ Cú pháp: `public DataTable GetChanges()`
- ✓ Một DataTable mới gồm những thay đổi từ DataTable hiện hành sẽ được tạo ra. Nếu trong DataTable hiện hành không có sự thay đổi nào thì kết quả trả về là null

### ➤ WriteXML()

- ✓ Cú pháp: `public void WriteXML(string filename)`
- ✓ Ghi dữ liệu từ DataTable ra tập tin XML



# Đối tượng DataTable

## ❑ Phương thức (tt)

### ➤ ImportRow()

- ✓ Cú pháp: `public void ImportRow(DataRow row)`
- ✓ Thêm một DataRow vào DataTable hiện hành

### ➤ Merge()

- ✓ Cú pháp: `public void Merge(DataTable dt)`
- ✓ Gộp DataTable dt vào DataTable hiện hành





# Đối tượng DataTable

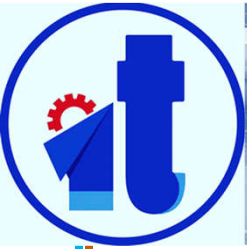
## ❑ Phương thức (tt)

### ➤ NewRow()

- ✓ Cú pháp: `public DataRow NewRow()`
- ✓ Tạo ra một dòng mới có cấu trúc của DataTable hiện hành

### ➤ ReadXML()

- ✓ Cú pháp: `public XMLReadMode ReadXML(string filename)`
- ✓ Đọc dữ liệu từ tập tin XML vào DataTable



# Đối tượng DataView

- ❑ Thuộc namespace System.Data
- ❑ Dùng cho việc sắp xếp, lọc, tìm kiếm và thay đổi các dòng dữ liệu của một bảng
- ❑ **Khởi tạo**
  - Cách 1: `DataView <tên_biến> = new DataView (<đối_tượng_DataTable>);`
  - Cách 2: `DataView <tên_biến> = <đối_tượng_DataTable>.DefaultView;`



# Đối tượng DataView

- ❑ Thuộc namespace System.Data
- ❑ Dùng cho việc sắp xếp, lọc, tìm kiếm và thay đổi các dòng dữ liệu của một bảng
- ❑ **Khởi tạo**
  - Cách 1: `DataView <tên_biến> = new DataView(<đối_tượng_DataTable>);`
  - Cách 2: `DataView <tên_biến> = <đối_tượng_DataTable>.DefaultView;`



# Đối tượng DataView

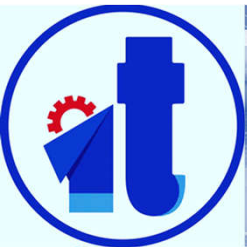
## □ Thuộc tính

### ➤ Count

- ✓ Kiểu dữ liệu: `int`
- ✓ Số lượng dòng dữ liệu có trong DataView

### ➤ RowFilter

- ✓ Kiểu dữ liệu: `string`
- ✓ Biểu thức dùng để lọc dữ liệu trên DataView



# Đối tượng DataView

## ❑ Thuộc tính (tt)

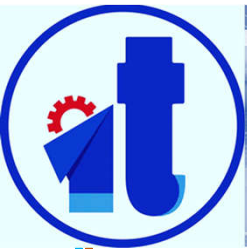
### ➤ Sort

- ✓ Kiểu dữ liệu: `string`
- ✓ Tên các cột được sắp xếp và thứ tự sắp xếp

### ➤ Table

- ✓ Kiểu dữ liệu: `DataTable`
- ✓ Đối tượng `DataTable` tạo `DataView`





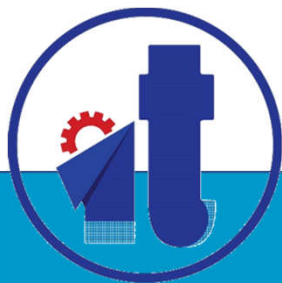
# Đối tượng DataView

## ❑ Phương thức

### ➤ Delete()

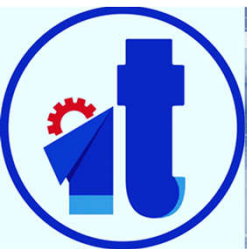
- ✓ Cú pháp: `public void Delete(int position)`
- ✓ Xóa một dòng tại vị trí position

**Các thay đổi trên DataView cũng thay đổi trên DataTable tạo ra DataView**



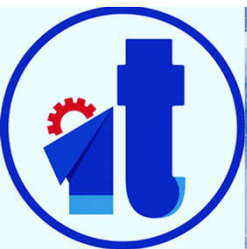
# DataGridView - ComboBox





# DataGridView

- ❑ Điều khiển chuyên dùng cho việc hiển thị dữ liệu dưới dạng bảng với các chức năng:
  - Có thể biểu diễn dữ liệu trên bảng trong **Textbox**, **CheckBox** hoặc **ComboBox**,...
  - Có khả năng sử dụng DataSource để kết nối với bất kì cơ sở dữ liệu nào, với bất kì bảng nào
  - Dữ liệu trên điều khiển được lưu sẵn trong bộ nhớ cache nên tăng tốc độ xử lý mà vẫn đảm bảo khả năng xử lý hàng trăm bản ghi một lúc



# DataGridView

## ☐ Thuộc tính

### ➤ AllowUserToResizeRows

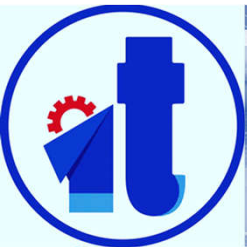
- ✓ Kiểu dữ liệu: `bool`
- ✓ Cho phép (`true`) hoặc không cho phép (`false`) người dùng thay đổi kích thước các dòng trên DataGridView

### ➤ ColumnCount

- ✓ Kiểu dữ liệu: `int`
- ✓ Số lượng cột hiển thị trên DataGridView

### ➤ Columns

- ✓ Kiểu dữ liệu: `DataGridViewColumnCollection`
- ✓ Danh sách cột trên DataGridView



# DataGridView

## ☐ Thuộc tính (tt)

### ➤ CurrentCell

- ✓ Kiểu dữ liệu: [DataGridViewCell](#)
- ✓ Ô đang được trỏ đến trên DataGridView

### ➤ CurrentRow

- ✓ Kiểu dữ liệu: [DataGridViewRow](#)
- ✓ Dòng chứa ô đang được trỏ đến trên DataGridView

### ➤ DataSource

- ✓ Kiểu dữ liệu: [Object](#)
- ✓ Nguồn dữ liệu của DataGridView





# DataGridView

## ☐ Thuộc tính (tt)

### ➤ MultiSelect

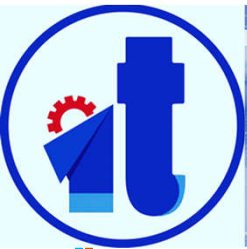
- ✓ Kiểu dữ liệu: **bool**
- ✓ True: cho phép người dùng chọn nhiều dòng trên DataGridView; False: ngược lại

### ➤ ReadOnly

- ✓ Kiểu dữ liệu: **bool**
- ✓ True: không thay đổi được dữ liệu trên DataGridView; False: ngược lại

### ➤ RowCount

- ✓ Kiểu dữ liệu: **int**
- ✓ Số lượng dòng hiển thị trên DataGridView



# DataGridView

## ❑ Thuộc tính (tt)

### ➤ Rows

- ✓ Kiểu dữ liệu: `DataGridViewRowCollection`
- ✓ Danh sách dòng trên DataGridView

### ➤ SelectedCells

- ✓ Kiểu dữ liệu: `DataGridViewSelectedCellCollection`
- ✓ Danh sách các ô được chọn trên DataGridView

### ➤ SelectedRows

- ✓ Kiểu dữ liệu: `DataGridViewSelectedRowCollection`
- ✓ Danh sách các dòng được chọn trên DataGridView



# DataGridView

## ❑ Phương thức

### ➤ ClearSelection()

- ✓ Cú pháp: `public void ClearSelection()`
- ✓ Bỏ chọn các ô trên DataGridView

### ➤ SelectAll()

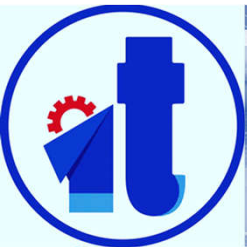
- ✓ Cú pháp: `public void SelectAll ()`
- ✓ Chọn tất cả các ô trên DataGridView



# DataGridView

## ☐ Sự kiện

- **Click**: Xảy ra khi nhấn chuột vào DataGridView
- **CellClick**: Xảy ra khi nhấn chuột vào ô bất kỳ trên DataGridView
- **CellContentClick**: Xảy ra khi nhấn chuột vào ô có chứa nội dung trên DataGridView
- **CellValueChanged**: Xảy ra khi nội dung của ô trên DataGridView bị thay đổi
- **UserAddedRow**: Xảy ra sau khi người dùng thêm dòng mới vào DataGridView
- **UserDeletedRow**: Xảy ra sau khi người dùng xóa dòng trên DataGridView
- **CellFormating**: Định dạng hiển thị giá trị/ nội dung của ô trên DataGridView



# ComboBox

❑ Namespace System.Windows.Forms

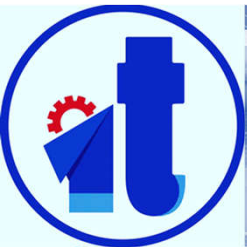
❑ **Thuộc tính nâng cao**

➤ DataSource

- ✓ Kiểu dữ liệu: **Object**
- ✓ Nguồn dữ liệu của Combobox

➤ DisplayMember

- ✓ Kiểu dữ liệu: **string**
- ✓ Tên trường/thuộc tính của đối tượng được dùng để hiển thị giá trị lên Combobox



# ComboBox

## ❑ Thuộc tính nâng cao (tt)

### ➤ ValueMember

- ✓ Kiểu dữ liệu: **string**
- ✓ Tên trường/thuộc tính của đối tượng được dùng để làm giá trị thực cho phần tử trong Combobox

### ➤ SelectedItem

- ✓ Kiểu dữ liệu: **Object**
- ✓ Phần tử được chọn trong Combobox

### ➤ SelectedValue

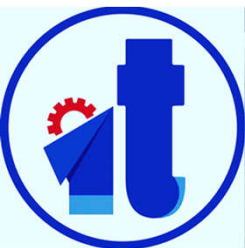
- ✓ Kiểu dữ liệu: **Object**
- ✓ Giá trị thực của phần tử được chọn trong Combobox





# ComboBox

- ❑ Gán nguồn dữ liệu là một danh sách các đối tượng vào combobox
  - <điều\_khiển\_combobox>.DataSource = <danh\_sách\_đối\_tượng>;
  - <điều\_khiển\_combobox>.DisplayMember = "<tên\_trường\_hiển\_thị>";
  - <điều\_khiển\_combobox>.ValueMember = "<tên\_trường\_giá\_trị>";
- ❑ Lấy phần tử được chọn và ép về kiểu dữ liệu ban đầu của phần tử
  - <kdl\_phần\_tử> <tên\_biến> = (<kdl\_phần\_tử>)<điều\_khiển\_combobox>.SelectedItem;



# Bài tập

Quản lý sinh viên

## QUẢN LÝ SINH VIÊN

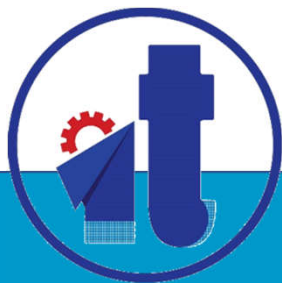
**Tìm Kiếm**

☒ Tất Cả ☐ Họ tên:

☐ MSSV:   ☐ Lớp:

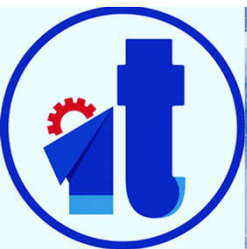
| ID    | MSSV       | Họ          | Tên    | Ngày sinh  | Lớp học    | Địa chỉ |
|-------|------------|-------------|--------|------------|------------|---------|
| 133   | 0306161449 | Nguyễn Văn  | Nghĩa  | 07/04/1997 | CĐ TH16PMB | TP. HCM |
| 134   | 0306161451 | Lê Thị Minh | Ngọc   | 14/12/1998 | CĐ TH16PMB | TP. HCM |
| 135   | 0306161452 | Nguyễn Hoài | Nhân   | 16/08/1998 | CĐ TH16PMB | TP. HCM |
| 136   | 0306161460 | Lê Trung    | Phú    | 07/10/1998 | CĐ TH16PMB | TP. HCM |
| 137   | 0306161466 | Văn Nam     | Phương | 16/11/1998 | CĐ TH16PMB | TP. HCM |
| 138   | 0306161469 | Lâm Thế     | Quang  | 16/01/1998 | CĐ TH16PMB | TP. HCM |
| 139   | 0306161470 | Vũ Hoàng    | Quân   | 09/09/1998 | CĐ TH16PMB | TP. HCM |
| ▶ 141 | 0306151220 | Võ Thành    | An     | 10/06/1996 | CĐ TH16PMC | TP. HCM |
| 142   | 0306151281 | Bùi Minh    | Quý    | 11/12/1997 | CĐ TH16PMC | TP. HCM |
| 143   | 0306151309 | Hứa Vũ Minh | Trí    | 26/12/1996 | CĐ TH16PMC | TP. HCM |
| 144   | 0306151313 | Nguyễn Ngọc | Tuấn   | 02/06/1997 | CĐ TH16PMC | TP. HCM |

Lưu



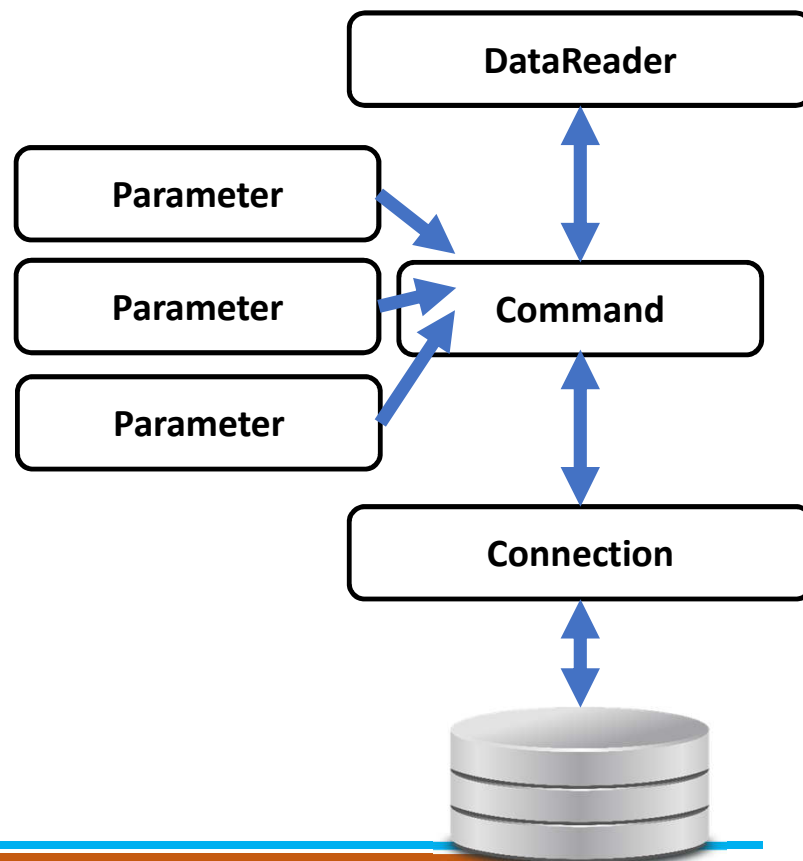
# Mô hình kết nối (Connected)

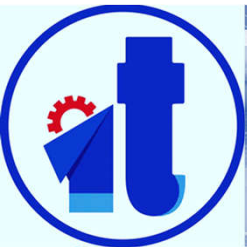




# Mô hình kết nối

- ☐ Tạo và mở kết nối với CSDL
- ☐ Tạo đối tượng truy vấn, thao tác với CSDL
- ☐ Truy vấn, thao tác CSDL và xử lý kết quả trả về nếu có
- ☐ Đóng kết nối CSDL





## Đối tượng DataReader

- ❑ Dùng để đọc dữ liệu từ cơ sở dữ liệu
- ❑ Chỉ đọc dữ liệu tuần tự từ đầu đến cuối (không theo chiều ngược lại) (không có chức năng cập nhật dữ liệu vào CSDL)
- ❑ Tạo đối tượng
  - `XYZDataReader <tên_biến> = <đối_tượng_Command>.ExecuteReader();`





# Đối tượng DataReader

## ❑ Thuộc tính

- <đối\_tượng\_DataReader> [<chỉ số cột>]
- <đối\_tượng\_DataReader> ["tên\_cột"]
  - ✓ Kiểu dữ liệu: **object**
  - ✓ Giá trị của ô có chỉ số cột hoặc tên cột tương ứng
- **FieldCount**
  - ✓ Kiểu dữ liệu: **int**
  - ✓ Số cột trong dòng mà đối tượng DataReader đang đọc





# Đối tượng DataReader

## ❑ Thuộc tính (tt)

### ➤ HasRow

- ✓ Kiểu dữ liệu: **bool**
- ✓ Có giá trị true nếu đối tượng DataReader có chứa ít nhất một dòng dữ liệu, ngược lại có giá trị false

### ➤ IsClosed

- ✓ Kiểu dữ liệu: **bool**
- ✓ Có giá trị true nếu đối tượng DataReader đã bị đóng lại, ngược lại có giá trị false



# Đối tượng DataReader

## ❑ Phương thức

### ➤ Read()

- ✓ Cú pháp: `public bool Read()`
- ✓ Có giá trị `true` nếu đối tượng DataReader đọc được một dòng dữ liệu, ngược lại có giá trị `false`

### ➤ Close()

- ✓ Cú pháp: `public void Close()`
- ✓ Đóng đối tượng DataRead



# Đối tượng DataReader

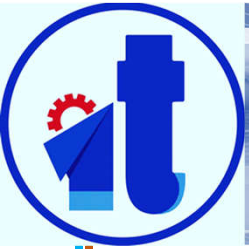
## ❑ Phương thức

### ➤ IsDBNull(index)

- ✓ Cú pháp: `public bool IsDBNull (int index)`
- ✓ Tham số truyền vào là chỉ số cột
- ✓ Có giá trị true nếu giá trị của cột trong dòng hiện hành là DBNull, ngược lại có giá trị false

### ➤ GetOrdinal (name)

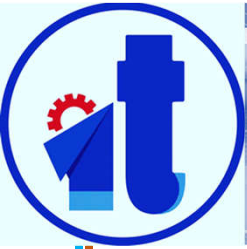
- ✓ Cú pháp: `public int GetOrdinal (string name)`
- ✓ Tham số truyền vào là tên cột
- ✓ Có giá trị là chỉ số cột theo tên hiện hành



# Đối tượng DataReader

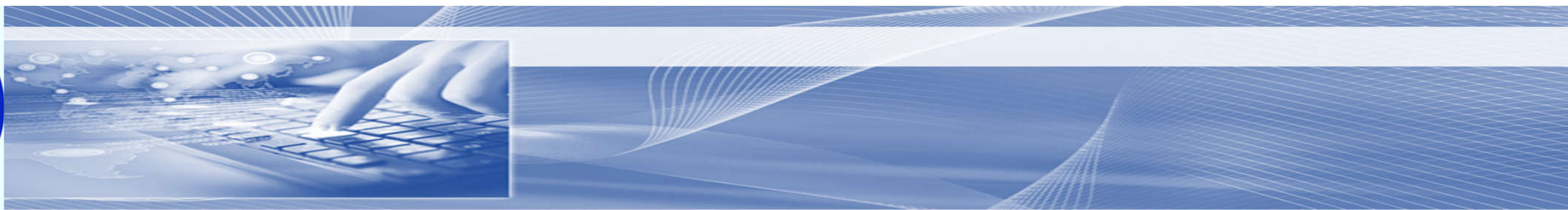
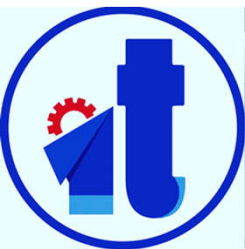
❑ Một số phương thức lấy các giá trị với các kiểu dữ liệu tương ứng với tham số được truyền vào là chỉ số của cột

- `public bool GetBoolean (int index)`
- `public byte GetByte (int index)`
- `public char GetChar (int index)`
- `public DateTime GetDateTime (int index)`
- `public decimal GetDecimal (int index)`
- `public double GetDouble (int index)`
- `public float GetFloat (int index)`
- `public short GetInt16 (int index)`
- `public int GetInt32 (int index)`
- `public long GetInt64 (int index)`
- `public string GetString (int index)`
- `public object GetValue (int index)`



# Ví dụ

```
SqlConnection conn = new SqlConnection(@"Data Source=NguyenLe-PC; Initial Catalog=DemoDB;
Integrated Security=true"); // Tạo đối tượng kết nối
SqlDataReader rdr = null;
try
{
    conn.Open(); // Mở kết nối
    SqlCommand cmd = new SqlCommand(@"SELECT * FROM SINH_VIEN", conn);
    // Truyền đối tượng kết nối cho đối tượng command
    rdr = cmd.ExecuteReader(); // nhận kết quả từ truy vấn
    // Xử lý kết quả nhận được
    while (rdr.Read())
        { /*code xử lý*/ }
}
finally
{
    if (rdr != null)
        rdr.Close(); // close the reader
    if (conn != null)
        conn.Close(); // Close the connection
}
```



Q&A

