

*“The only way to learn a new programming language is by writing programs in it.”*

Brian W. Kernighan and Dennis M. Ritchie

# C# Programming

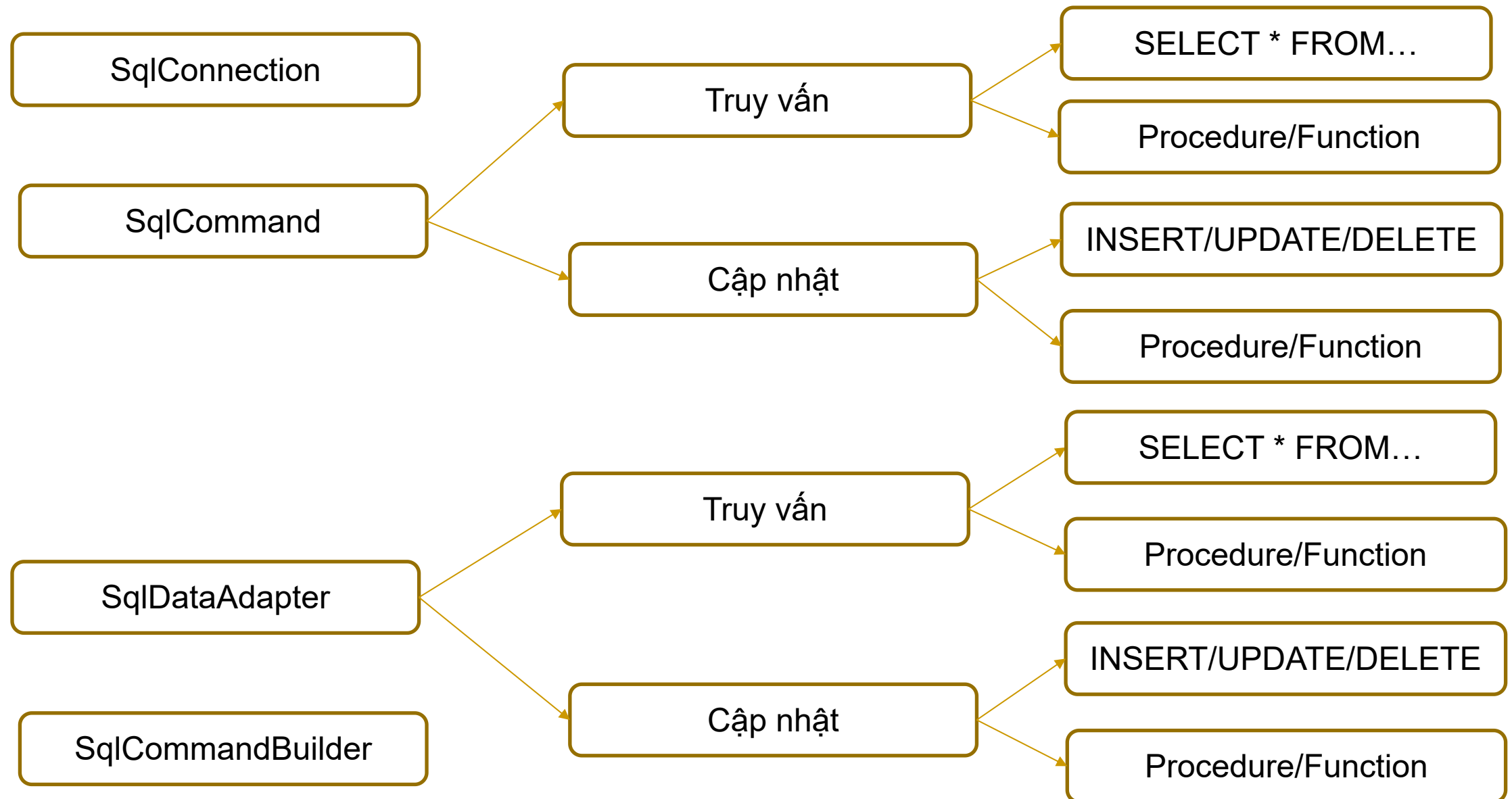
## Lập trình cơ sở dữ liệu với **ADO.NET**

# Nội dung

---

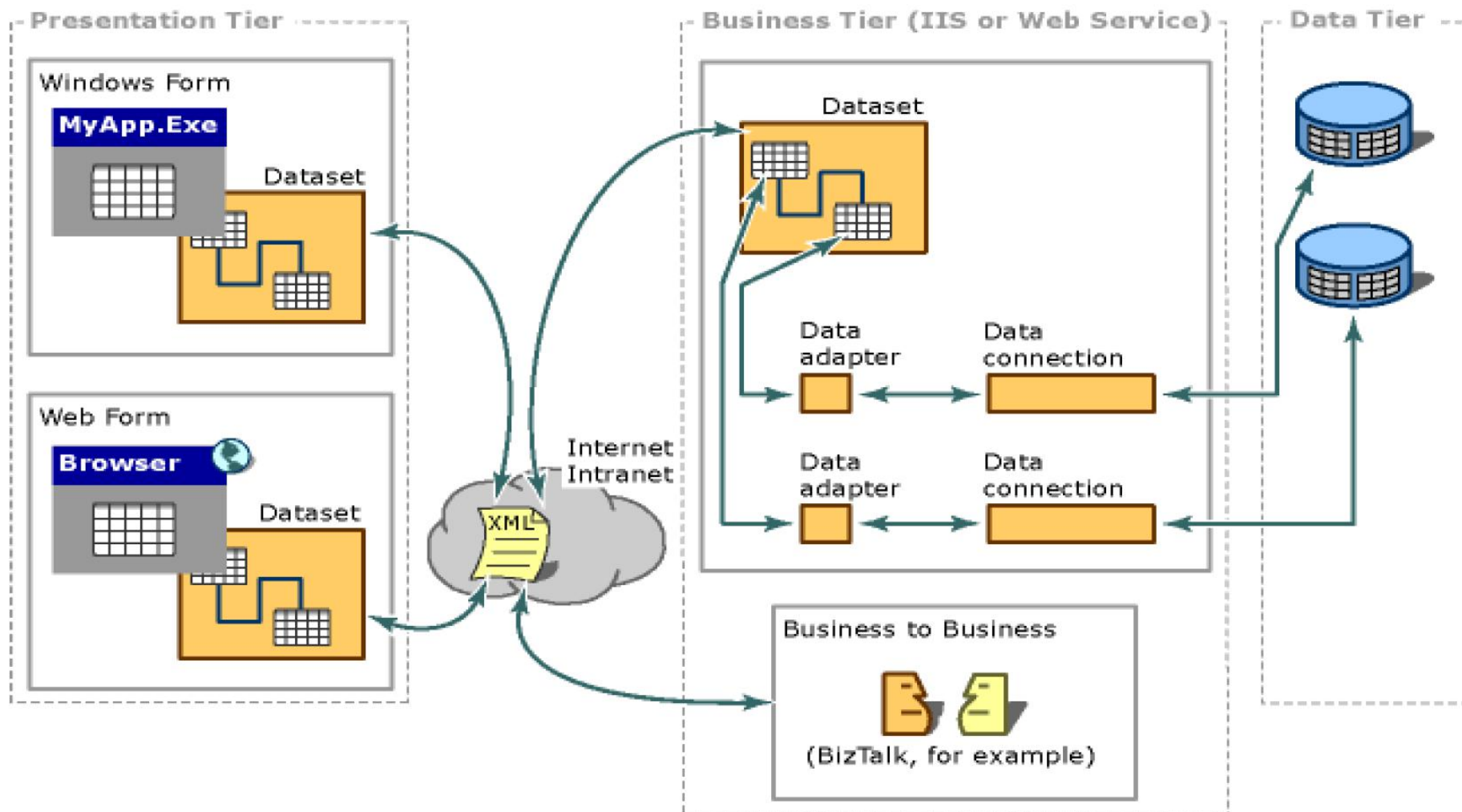
- Nội dung
  - Giới thiệu về ADO.NET
  - Kết nối với cơ sở dữ liệu dùng lớp `SqlConnection`
  - Truy vấn và cập nhật dữ liệu dùng lớp `SqlCommand`
  - Truy vấn và cập nhật dữ liệu dùng lớp `SqlDataAdapter`
  - Cập nhật dữ liệu dùng lớp `SqlCommandBuilder`

# Mục đích, chuẩn đầu ra và nội dung



# Giới thiệu ADO.NET

## ■ Microsoft 3-Tier Architecture



# Giới thiệu ADO.NET

---

- Là bước phát triển của ADO (ActiveX Data Objects).
- Làm việc với các cơ sở dữ liệu MS Access, SQL Server, XML, Oracle.
- Gồm một tập các lớp dùng để truy nhập và thao tác với các cơ sở dữ liệu:
  - Kiểu kết nối (connected): gồm các lớp cung cấp dịch vụ quản lý (managed provider) để kết nối trực tiếp với cơ sở dữ liệu và đồng bộ dữ liệu của ứng dụng với cơ sở dữ liệu.
  - Kiểu ngắt kết nối (disconnected): gồm các lớp dữ liệu chung (generic data) được thiết kế để lưu trữ bản sao dữ liệu trong ứng dụng được lấy từ cơ sở dữ liệu.
  - Các lớp dữ liệu chung không phụ thuộc vào cơ sở dữ liệu.

# Giới thiệu ADO.NET

---

- Các lớp cung cấp dịch vụ quản lý:
  - Dùng để kết nối tới cơ sở dữ liệu, truy vấn và cập nhật dữ liệu vào cơ sở dữ liệu.
  - **SQL Server Managed Provider:**
    - Các lớp việc với cơ sở dữ liệu SQL Server.
    - Định nghĩa bằng tiền tố Sql, ví dụ SqlConnection.
  - **OLE DB Managed Provider:**
    - Các lớp dùng để làm việc với các cơ sở dữ liệu hỗ trợ OLEDB như Microsoft Access hoặc Oracle.
    - Định nghĩa bằng tiền tố OleDb, ví dụ OleDbConnection.
  - **ODBC Managed Provider:**
    - Các lớp dùng để làm việc với các cơ sở dữ liệu hỗ trợ ODBC.
    - Định nghĩa bằng tiền tố Odbc, ví dụ OdbcConnection.

# Giới thiệu ADO.NET

---

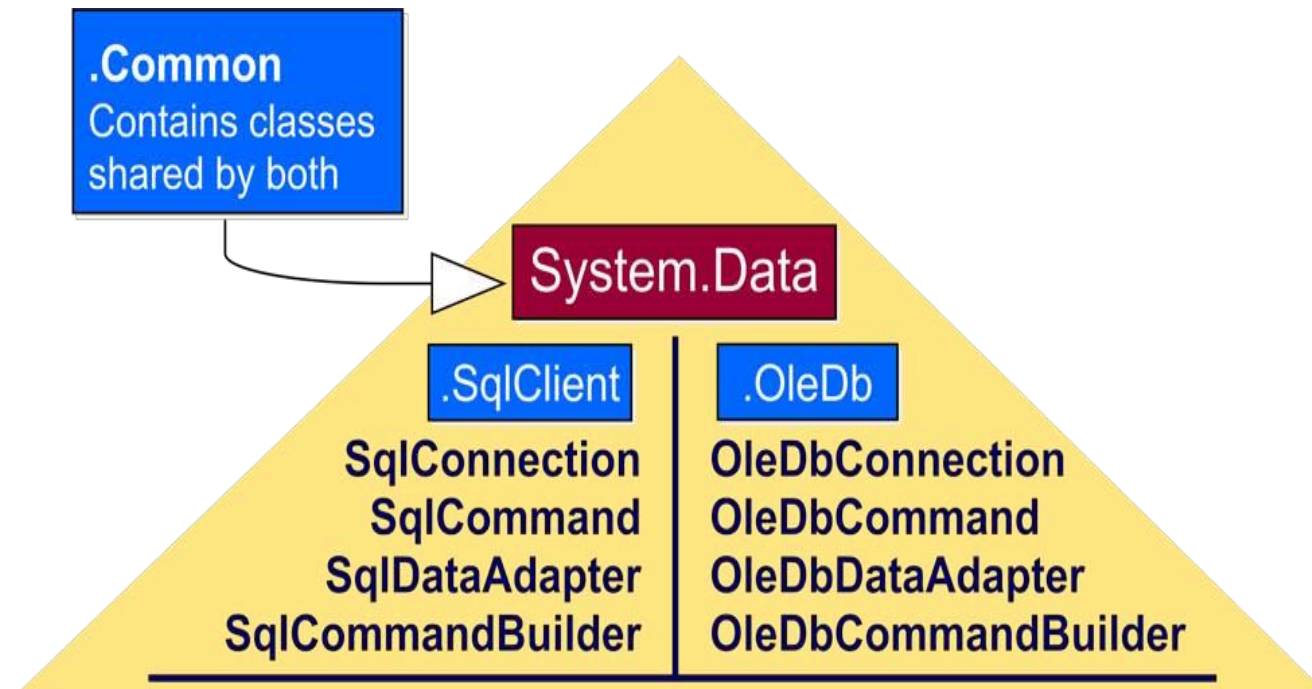
## ■ Các lớp cung cấp dịch vụ quản lý:

- ❑ **Connection**: gồm `SqlConnection` và `OleDbConnection` để kết nối tới sơ sở dữ liệu.
- ❑ **Command**: gồm `SqlCommand` và `OleDbCommand` để thực hiện một lệnh SQL hoặc dùng để gọi một thủ tục.
- ❑ **DataReader**: gồm `SqlDataReader` và `OleDbDataReader` để đọc dữ liệu từ cơ sở dữ liệu.
- ❑ **Parameter**: gồm `SqlParameter` và `OleDbParameter` để thiết lập các tham số lớp Command.
- ❑ **DataAdapter**: gồm `SqlDataAdapter` và `OleDbDataAdapter` để truy vấn và cập nhật dữ liệu.
- ❑ **CommandBuilder**: gồm `SqlCommandBuilder` và `OleDbCommandBuilder` để cập nhật dữ liệu.

# Giới thiệu ADO.NET

## ■ Các không gian tên:

- ❑ `System.Data.SqlClient`: Định nghĩa các lớp làm việc với cơ sở dữ liệu Microsoft SQL Server.
- ❑ `System.Data.OleDb`: Định nghĩa các lớp làm việc với cơ sở dữ liệu OLEDB như Microsoft Access.





# Giới thiệu ADO.NET

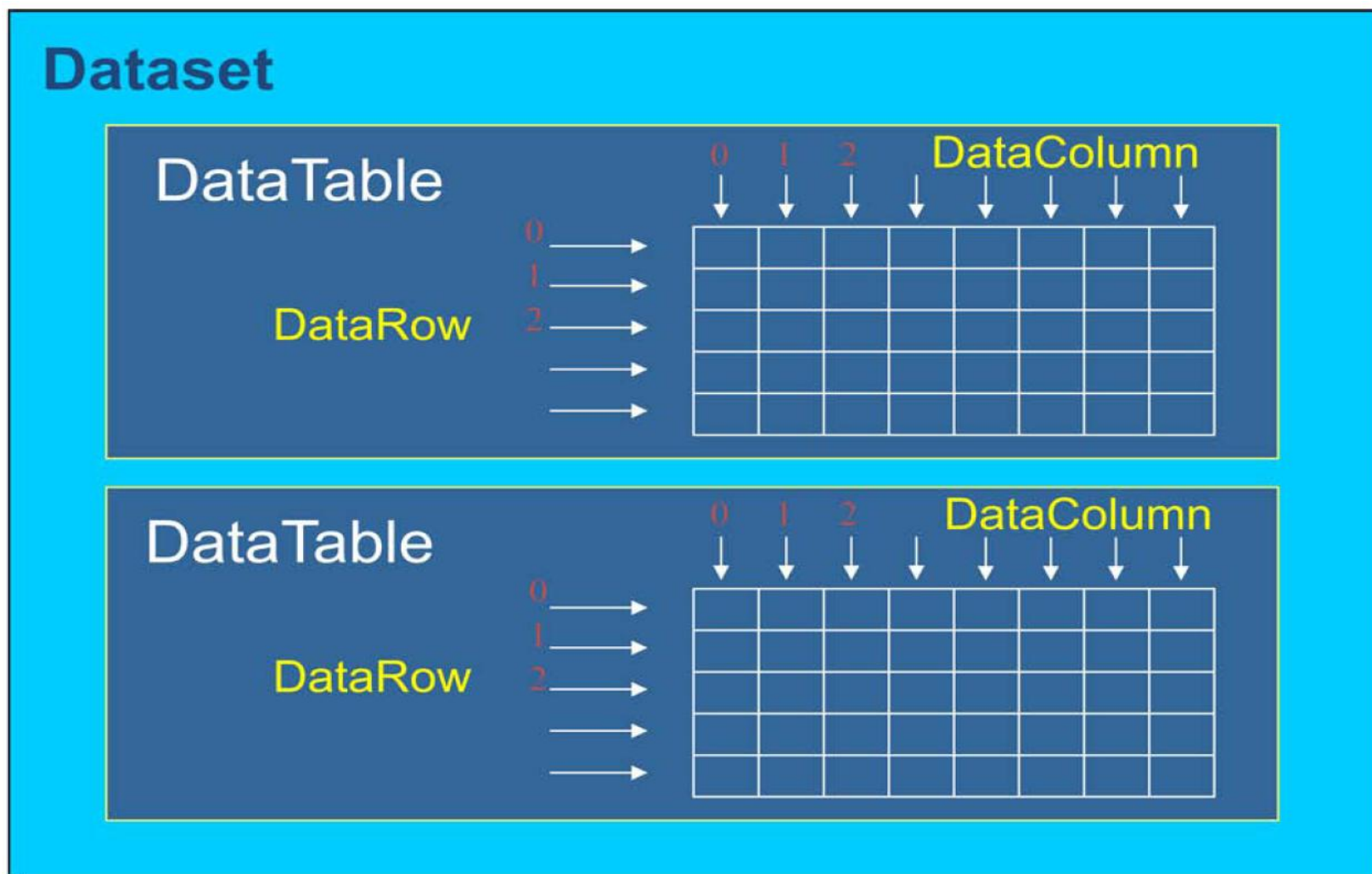
---

## ■ Các lớp dữ liệu chung:

- ❑ Các lớp dữ liệu chung (generic data) làm việc không phụ thuộc vào cơ sở dữ liệu.
- ❑ **DataSet**: Dùng để lưu trữ một bản sao của một cơ sở dữ liệu trong ứng dụng.
- ❑ **DataTable**: Dùng để lưu một bảng dữ liệu bao gồm các hàng và các cột
- ❑ **DataRow**: Dùng để lưu một hàng của một bảng dữ liệu.
- ❑ **DataColumn**: Dùng để lưu một cột của một bảng dữ liệu.
- ❑ *Không gian tên cho các lớp dữ liệu chung: System.Data.*

# Giới thiệu ADO.NET

- Các lớp dữ liệu chung:



# Kết nối cơ sở dữ liệu

---

- Các bước kết nối cơ sở dữ liệu:
  - Định nghĩa chuỗi kết nối tới cơ sở dữ liệu SQL Server:
  - Tạo đối tượng kết nối lớp SqlConnection:
  - Tạo kết nối tới cơ sở dữ liệu:

```
string conStr = "server = <computer>;  
                database = <dataname>;  
                uid = <user>;  
                pwd = <password>;  
                connect timeout = <time>";  
  
SqlConnection mySqlConnection = new SqlConnection(conStr);  
mySqlConnection.Open();
```

# Kết nối cơ sở dữ liệu

---

## ■ Ví dụ kết nối cơ sở dữ liệu Books

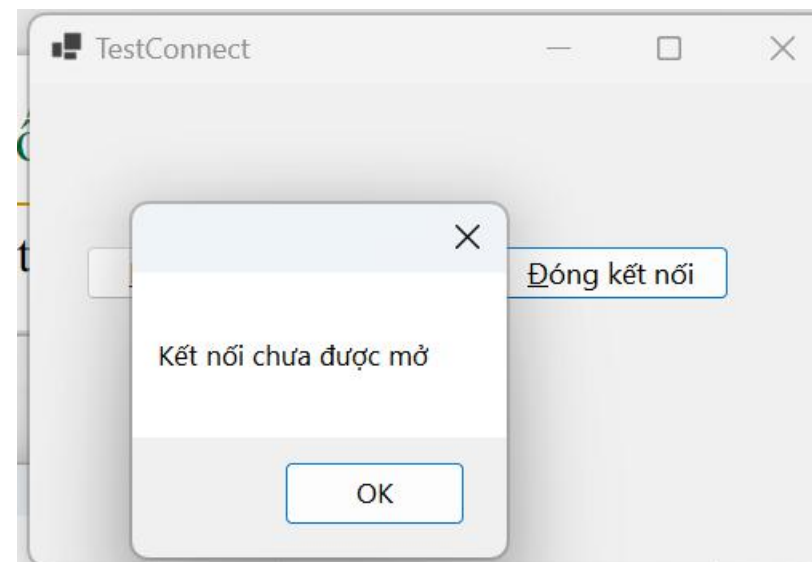
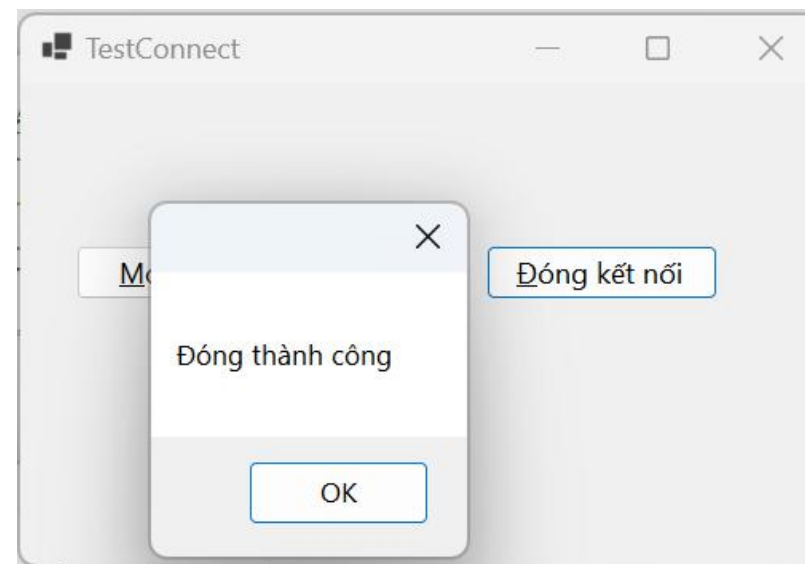
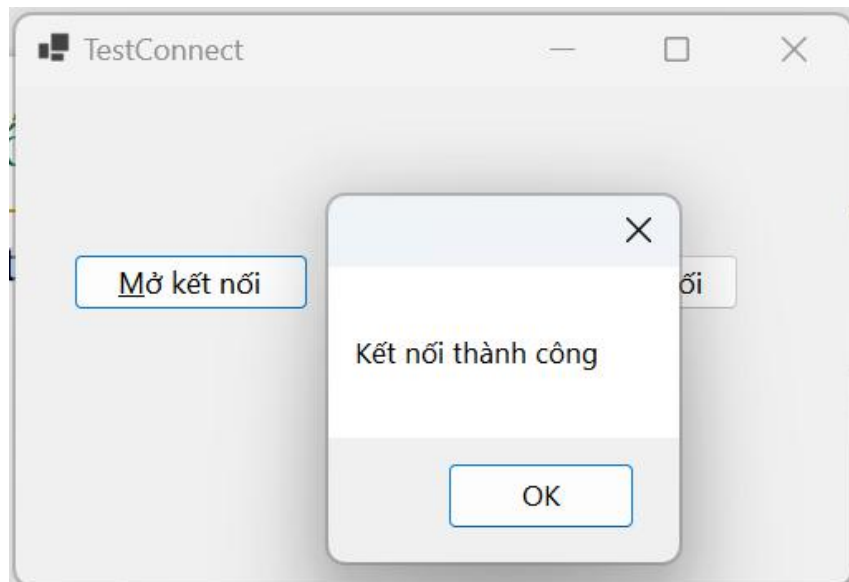
```
string conStr = "server = localhost;database = Books; uid = sa; pwd = sa;  
                connect timeout =150";  
SqlConnection mySqlConnection = new SqlConnection(conStr);  
mySqlConnection.Open();
```

## ■ Phát hiện các lỗi khi kết nối tới cơ sở dữ liệu:

```
try  
{  
    mySqlConnection = new SqlConnection(conStr);  
    mySqlConnection.Open();  
}  
catch (SqlException mySqlException)  
{  
    MessageBox.Show(mySqlException.Message, "Error");  
    return;  
}
```

# Ví dụ - Kết nối CSDL

## ■ Tạo form test kết nối



## Ví dụ:

```
SqlConnection sqlcon;  
String sqlstr = @"Data Source=DESKTOP-  
NUT8NPF\HONGANH;Initial  
Catalog=QLSINHVIEN;Integrated Security=True";  
  
private void TestConnect_Load(object sender,  
EventArgs e)  
{  
    sqlcon = new SqlConnection(sqlstr);  
}
```

```
private void button1_Click(object sender, EventArgs e)  
{  
    if (sqlcon.State == ConnectionState.Closed)  
    {  
        sqlcon.Open();  
        MessageBox.Show("Kết nối thành công");  
    }  
}  
  
private void button2_Click(object sender, EventArgs e)  
{  
    if (sqlcon.State == ConnectionState.Open)  
    {  
        sqlcon.Close();  
        MessageBox.Show("Đóng thành công");  
    }  
    else  
        MessageBox.Show("Kết nối chưa được mở");  
}
```

# Lớp SqlCommand

---

- Đối tượng lớp SqlCommand dùng để truy vấn và cập nhật (nhập, sửa và xóa) dữ liệu trong cơ sở dữ liệu.
- Đối tượng lớp SqlCommand dùng để thực hiện một câu lệnh SQL thông qua một đối tượng lớp SqlConnection:
  - SELECT: Truy vấn dữ liệu.
  - INSERT: Nhập dữ liệu.
  - UPDATE: Sửa dữ liệu
  - DELETE: Xóa dữ liệu.
- Đối tượng lớp SqlCommand có thể dùng để gọi một thủ tục (stored procedure) của cơ sở dữ liệu SQL Server.

# Lớp SqlCommand

---

## ■ Tạo đối tượng SqlCommand:

```
SqlCommand mySqlCommand = new SqlCommand(string sSql, SqlConnection mySqlConnection);
```

Trong đó:

- sSql: Định nghĩa lệnh SQL hoặc lệnh gọi thủ tục.
- mySqlConnection: Đối tượng đã kết nối tới cơ sở dữ liệu.

## ■ Các phương thức:

- `ExecuteNonQuery()`: Dùng để thực hiện một lệnh INSERT, UPDATE, DELETE hoặc một thủ tục không trả về kết quả.
- `ExecuteReader()`: Dùng để thực hiện lệnh SELECT hoặc các thủ tục với kết quả trả về là một đối tượng lớp DataReader.
- `ExecuteScalar()`: Dùng để thực hiện lệnh SELECT với kết quả trả về là một giá trị vô hướng như hàm count(), sum().



# Lớp SqlCommand

---

## ■ Truy vấn dữ liệu dùng lớp SqlCommand:

1. Tạo đối tượng lớp SqlConnection và kết nối tới cơ sở dữ liệu:

```
SqlConnection mySqlConnection = new SqlConnection(conStr);  
mySqlConnection.Open();
```

2. Tạo đối tượng lớp SqlCommand:

```
SqlCommand mySqlCommand = new SqlCommand(sSql, mySqlConnection);
```

hoặc

```
SqlCommand mySqlCommand = mySqlConnection.CreateCommand();  
mySqlCommand.CommandText = sSql;
```

3. Thực hiện phương thức ExecuteReader():

```
SqlDataReader mySqlDataReader = mySqlCommand.ExecuteReader();
```

# Lớp SqlCommand

---

- Các thuộc tính thường dùng của SqlDataReader:
  - `FieldCount`: Số cột của dòng hiện thời.
  - `IsClosed`: Có/không đối tượng đã đóng.
- Các phương thức thường dùng của SqlDataReader:
  - `Read()`: Di chuyển con trỏ để đọc dòng tiếp theo.
  - `Close()`: Đóng đối tượng.
  - `HasRows()`: cho biết có dòng dữ liệu nào hay không
- Phương thức `Load()` của đối tượng `DataTable`:
  - `Load()`: Chuyển dữ liệu đối tượng `SqlDataReader` vào đối tượng `DataTable`.

## ■ Lấy dữ liệu

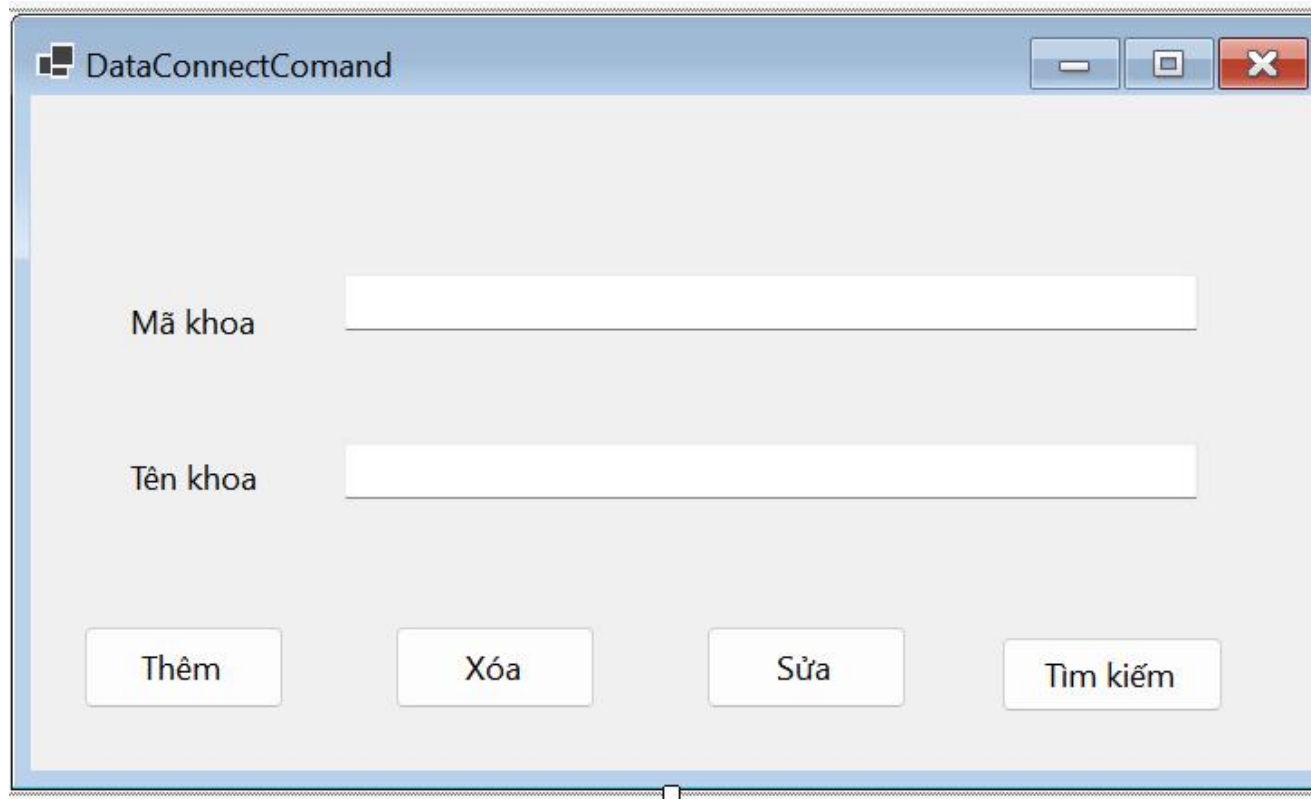
- ❑ `get<type>(index)`: lấy dữ liệu kiểu `type` của cột thứ `index` trong database. Lưu ý: các cột đánh số từ 0.
- ❑ Ví dụ: `String ten = rd.GetString(1);`

```
private void btnTimKiem_Click(object sender, EventArgs e)
{
    try
    {
        if (sqlcon.State == ConnectionState.Closed)
        {
            sqlcon.Open();
        }
        string sqlstr = "select * from Khoa where makhoa='" + txtMaKhoa.Text + "'";
        SqlCommand cmd = new SqlCommand(sqlstr, sqlcon);
        SqlDataReader rd = cmd.ExecuteReader();
        if (rd.Read())
        {
            String ten = rd.GetString(1);
            MessageBox.Show("Khoa cua ma can tim la " + ten);
        }
        else
        {
            MessageBox.Show("Không tìm thấy mã cần tìm");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally { sqlcon.Close(); }
}
```

## Ví dụ

---

- Cho bảng dữ liệu Khoa (MaKhoa, TenKhoa)
- Viết lệnh thực hiện thêm, sửa, xóa. Yêu cầu kiểm tra mã trùng khi thực hiện



The image shows a screenshot of a Windows application window titled "DataConnectComand". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains two text input fields. The first field is labeled "Mã khoa" (Department Code) and the second is labeled "Tên khoa" (Department Name). Below these fields, there are four buttons arranged horizontally: "Thêm" (Add), "Xóa" (Delete), "Sửa" (Edit), and "Tìm kiếm" (Search). The buttons are white with black text and a slight shadow effect.

# Kiểm tra mã trùng

---

```
public bool kt_KhoaChinh(string ma)
{
    try
    {
        if (sqlcon.State == ConnectionState.Closed)
        {
            sqlcon.Open();
        }
        string sqlstr = "select * from Khoa where maKhoa='" + ma + "'";
        SqlCommand cmd = new SqlCommand(sqlstr, sqlcon);
        var rd = cmd.ExecuteReader();
        if(rd.HasRows)
        {
            sqlcon.Close();
            return true;
        }
        else
        {
            sqlcon.Close();
            return false;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        return false;
    }
}
```

# Lớp SqlCommand

---

## ■ Cập nhật dữ liệu dùng lớp SqlCommand:

1. Tạo đối tượng lớp SqlConnection và kết nối tới cơ sở dữ liệu:

```
SqlConnection mySqlConnection = new SqlConnection(conStr);  
mySqlConnection.Open();
```

2. Tạo đối tượng lớp SqlCommand:

```
SqlCommand mySqlCommand = new SqlCommand(sSql,mySqlConnection);
```

hoặc

```
SqlCommand mySqlCommand = mySqlConnection.CreateCommand();  
mySqlCommand.CommandText = sSql;
```

3. Thực hiện phương thức ExecuteNonQuery():

```
mySqlCommand.ExecuteNonQuery();
```

## Ví dụ

---

- Cho bảng dữ liệu Khoa (MaKhoa, TenKhoa)
- Viết lệnh thực hiện thêm, sửa, xóa. Yêu cầu kiểm tra mã trùng khi thực hiện

The screenshot shows a Windows application window titled "DataConnectComand". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light gray and contains two text input fields. The first field is labeled "Mã khoa" (Department Code) and the second is labeled "Tên khoa" (Department Name). Below these fields are four buttons: "Thêm" (Add), "Xóa" (Delete), "Sửa" (Edit), and "Tìm kiếm" (Search). The buttons are arranged horizontally and have a light gray background with a thin border.

---

```
private void btnThem_Click(object sender, EventArgs e)
{
    try
    {
        if (!kt_KhoaChinh(txtMaKhoa.Text))
        {
            if (sqlcon.State == ConnectionState.Closed)
            {
                sqlcon.Open();
            }
            string sqlstr = "insert into Khoa values('" + txtMaKhoa.Text + "','" + txtTenKhoa.Text + "')";
            SqlCommand cmd = new SqlCommand(sqlstr, sqlcon);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Thêm thành công");
        }
        else
        {
            MessageBox.Show("Trùng mã");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally { sqlcon.Close(); }
}
```





//Chuoi ket noi ung voi yeu cau sua du lieu

```
string updateString = "update Khoa set TenKhoa='" +  
txtTenKhoa.Text + "' where MaKhoa='" + txtMaKhoa.Text + "'";
```

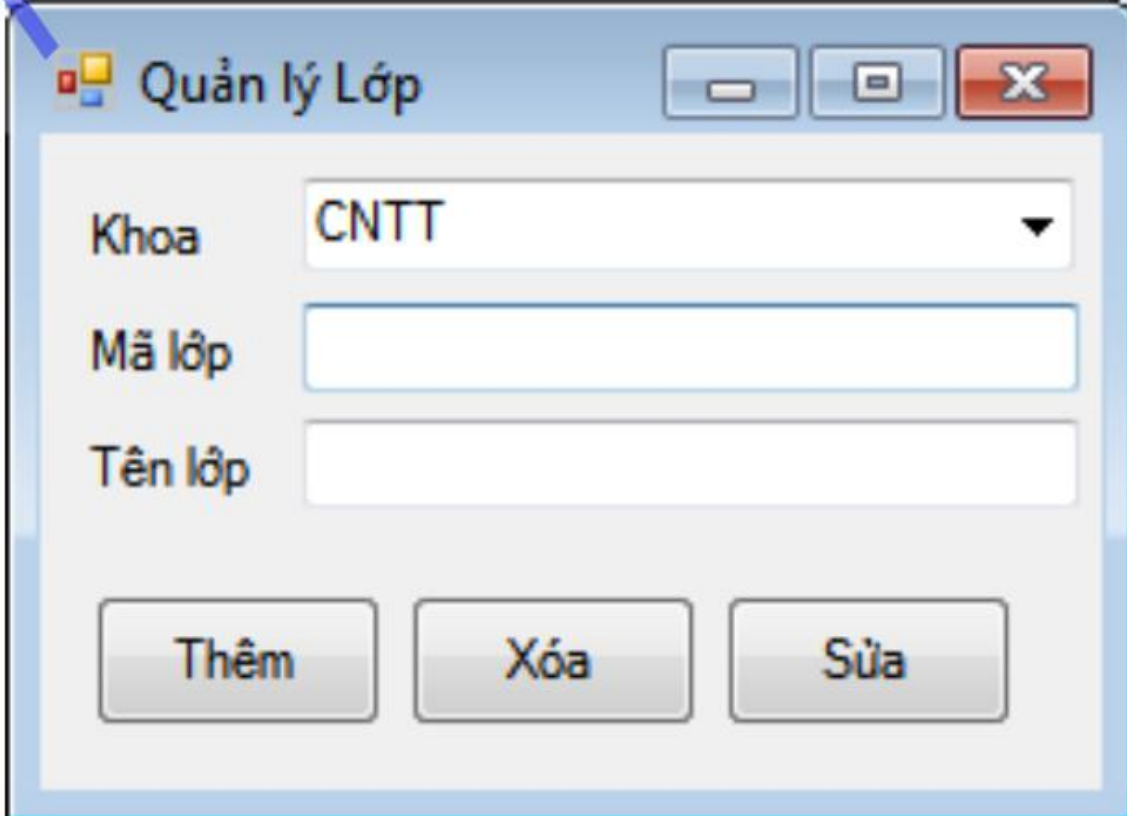
//Chuoi ket noi ung voi yeu cau xoa du lieu

```
string deleteString = "delete Khoa where MaKhoa='" +  
txtMaKhoa.Text + "'";
```

# Bài tập

---

- Cho bảng:
  - ❑ Lop (MaLop, TenLop, MaKhoa)
  - ❑ Khoa(MaKhoa, Ten Khoa)
  - ❑ Với comboBox khoa hiển thị các mã khoa từ bảng khoa



Quản lý Lớp

Khoa CNTT

Mã lớp

Tên lớp

Thêm Xóa Sửa

frmSinhvien\_Combo

Lớp 04CDNTH

Mã sv

Họ tên

Ngày sinh \_\_/\_\_/\_\_

Thêm Xóa Sửa

■ Cho bảng:

- ❑ Lop (MaLop, TenLop, MaKhoa)
- ❑ Khoa(MaKhoa, Ten Khoa)

Với comboBox khoa hiển thị các mã khoa từ bảng khoa

- ❑ SinhVien (MaSV, HoTen, NgaySinh, MaLop)

- Với comboBox mã lớp
- Ngày sinh: đặt masked TextBox

# Lớp SqlCommand

- **Bài tập:** xây dựng form để thêm, sửa, xóa, tìm kiếm và hiển thị dữ liệu bảng Suppliers lên DataGridView:

The application window is titled "Sửa, xóa nhà cung cấp". It features a form on the left for editing a supplier and a DataGridView on the right for displaying a list of suppliers.

**Form Fields:**

- Tên nhà cung cấp (\*): Công ty Cổ phần Sách Thái Hà
- Họ tên người liên lạc (\*): Hoàng Thái Hà
- Địa chỉ công ty (\*): P.Phúc Diễn, Q.Bắc Từ Liêm, Hà Nội
- Điện thoại (\*): 02437920995
- Email: sachthaiha@thaihabooks.com
- Mô tả thêm: <http://nhasachphuongnam.com/>

**Search Section:**

Tim kiếm nhà cung cấp ☐ Tim kiếm theo tên nhà cung cấp ☐ Tim kiếm họ tên người liên lạc

**DataGridView:**

	SupplierID	CompanyName	ContactName	Address	Ph
▶	3	Công ty Cổ phần Sách Thái Hà	Hoàng Thái Hà	P.Phúc Diễn, Q....	024
	4	Công ty TNHH Một Thành Viên S...	Nguyễn Phương Nam	Lầu 1, số 940 Đ...	190
	6	Công ty cổ phần Sách Alpha	Trần Mạnh Tường	11a ngõ 282 Ngu...	093
	8	Công ty Văn hóa Sáng tạo Trí Việt	Võ Trí Việt	11H Nguyễn Thị ...	028

**Buttons:** Ghi, Xóa, Kết thúc

# Lớp SqlCommand

---

- Tạo đối tượng lớp DataGridView, nháy chuột phải và chọn Add, mỗi cột khai báo:
  - **Name**: Tên cột dùng trong mã lệnh.
  - **Header text**: Tiêu đề hiển thị của cột.
  - **DataPropertyName**: Tên cột dữ liệu của DataTable.
- Các thuộc tính và sự kiện thường dùng:
  - **DataSource**: Tên DataTable cần hiển thị lên lưới.
  - **AutoGenerateColumns**: Có/không tự động lấy các cột.
  - **AllowUserToAddRows**: Có/không cho phép thêm dòng.
  - **AllowUserToDeleteRows**: Có/không cho phép xóa dòng.
  - **RowEnter()**: Sự kiện xảy ra khi con trỏ đưa vào một dòng.

# Lớp SqlCommand

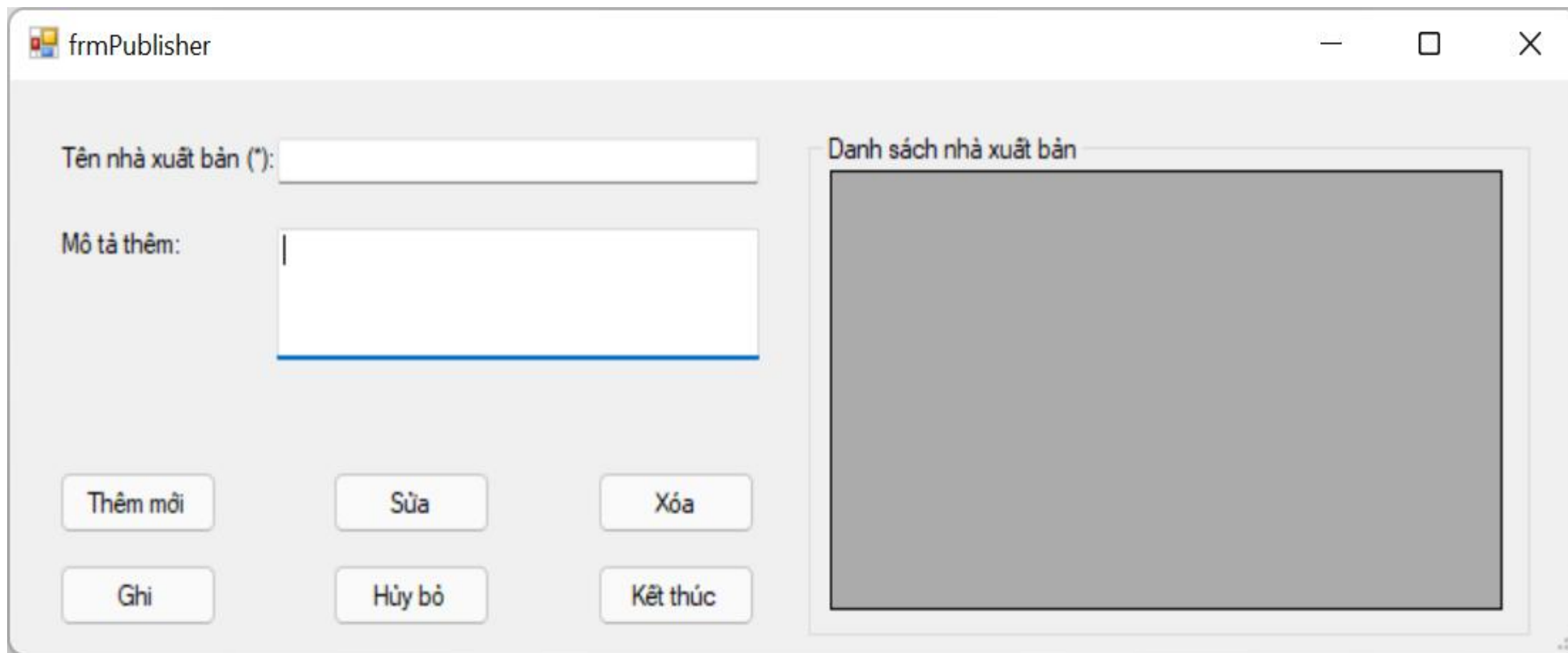
- Ví dụ xây dựng Form gồm nhập, sửa, xóa loại sách

The screenshot shows a Windows application window with the title bar text "Nhập, sửa. xóa loại sách". The window contains a form with the following elements:

- A label "Tên loại sách (\*):" followed by a single-line text input field.
- A label "Mô tả thêm:" followed by a multi-line text area.
- A panel on the right titled "Danh sách loại sách" containing a large, empty list box.
- A set of six buttons arranged in two rows:
  - Top row: "Thêm mới", "Sửa", "Xóa"
  - Bottom row: "Ghi", "Hủy bỏ", "Kết thúc"

# Lớp SqlCommand

- Sử dụng tham số trong lớp SqlCommand - ví dụ về truy vấn, nhập, sửa, xóa trong bảng Publishers.



The screenshot shows a Windows Forms application window titled "frmPublisher". The window contains the following elements:

- Form Fields:**
  - "Tên nhà xuất bản (\*):" followed by a text input box.
  - "Mô tả thêm:" followed by a larger text input box.
- Buttons:**
  - A row of three buttons: "Thêm mới", "Sửa", and "Xóa".
  - A second row of three buttons: "Ghi", "Hủy bỏ", and "Kết thúc".
- Table:**
  - A table titled "Danh sách nhà xuất bản" (Publisher List) is located on the right side of the form. The table area is currently empty and shaded gray.

# Lớp SqlCommand

---

## ■ Sử dụng tham số trong lớp SqlCommand:

### 1. Tạo đối tượng lớp SqlCommand:

```
string sSql = "INSERT INTO Publishers (PublisherName, Description) " +  
              "VALUES (@PublisherName, @Description)";  
SqlCommand mySqlCommand = new SqlCommand(sSql, mySqlConnection);
```

### 2. Định nghĩa và gán giá trị tham số cho đối tượng lớp SqlCommand:

```
mySqlCommand.Parameters.Add("@PublisherName", SqlDbType.NVarChar, 50).Value = "x";  
mySqlCommand.Parameters.Add("@Description", SqlDbType.NVarChar, 250).Value = "y";
```

### 3. Thực hiện phương thức ExecuteNonQuery():

```
mySqlCommand.ExecuteNonQuery();
```



---

# Lớp SqlCommand

---

- Thực hiện thủ tục của cơ sở dữ liệu:
  - ❑ Thủ tục là phương pháp thực hiện hiệu quả để cập nhật và sửa đổi dữ liệu của cơ sở dữ liệu.
  - ❑ `ExecuteNonQuery()`: thực hiện thủ tục không trả về kết quả.
  - ❑ `ExecuteReader()`: thực hiện thủ tục trả về kết quả.
- Thực hiện thủ tục không trả về kết quả:
  - ❑ Định nghĩa câu lệnh SQL gọi thủ tục/hàm với các tham số của thủ tục/hàm
  - ❑ Tạo một đối tượng lớp `SqlCommand`
  - ❑ Định nghĩa các tham số cho thủ tục/hàm.
  - ❑ Gán giá trị cho các tham số truyền vào cho thủ tục/hàm.
  - ❑ Dùng phương thức `ExecuteNonQuery()`.
  - ❑ Đọc giá trị của các tham số ra của thủ tục nếu có.

# Lớp SqlCommand

- Ví dụ tạo các thủ tục nhập/sửa/xóa người dùng và xây dựng form:

The screenshot shows a Windows application window titled "Nhập, sửa, xóa người dùng" (Add, edit, delete user). The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is divided into two sections. On the left, there is a form with several input fields and buttons. On the right, there is a list box titled "Danh sách người dùng" (User list).

The form fields and buttons are as follows:

- Họ và tên (\*):
- Tên tài khoản (\*):
- Mật khẩu (\*):
- Điện thoại (\*):
- Email:
- Trạng thái (\*):
- Mô tả thêm:
- Buttons: Thêm mới (Add new), Sửa (Edit), Xóa (Delete), Ghi (Save), Hủy bỏ (Cancel), Kết thúc (End)

The list box on the right, titled "Danh sách người dùng", is currently empty.

# Lớp SqlCommand

---

- Thủ tục truy vấn người dùng:

```
CREATE PROCEDURE DisplayUser
AS
BEGIN
    SELECT * FROM Users
END
```

- Thủ tục thêm một người dùng:

```
CREATE PROCEDURE AddUser
    @FullName nvarchar(35), @UserName nvarchar(20), @Password nvarchar(20),
    @Phone nvarchar(11), @Email nvarchar(30), @Status int, @Description nvarchar(250)
AS
BEGIN
    INSERT INTO Users (FullName, UserName, Password, Phone, Email, Status, Description)
    VALUES (@FullName, @UserName, @Password, @Phone, @Email, @Status, @Description)
END
```

# Lớp SqlCommand

---

- Thủ tục cập nhật một người dùng trong SQL Server:

```
CREATE PROCEDURE UpdateUser
```

```
    @UserID int, @FullName nvarchar(35), @UserName nvarchar(20), @Password nvarchar(20),  
    @Phone nvarchar(11), @Email nvarchar(30), @Status int, @Description nvarchar(250)
```

```
AS
```

```
BEGIN
```

```
    UPDATE Users
```

```
    SET FullName = @FullName, UserName = @UserName, Password = @Password,
```

```
    Phone = @Phone, Email = @Email, Status = @Status, Description = @Description
```

```
    WHERE UserID = @UserID
```

```
END
```

- Thủ tục xóa một người dùng:

```
CREATE PROCEDURE DeleteUser
```

```
    @UserID int
```

```
AS
```

```
BEGIN
```

```
    DELETE FROM Users WHERE UserID = @UserID
```

```
END
```

# Lớp SqlCommand

---

## 1. Truy vấn Users

```
string sSql = "EXECUTE DisplayUser";  
mySqlCommand = new SqlCommand(sSql,mySqlConnection);  
SqlDataReader mySqlDataReader = mySqlCommand.ExecuteReader();
```

## 2. Thêm một user

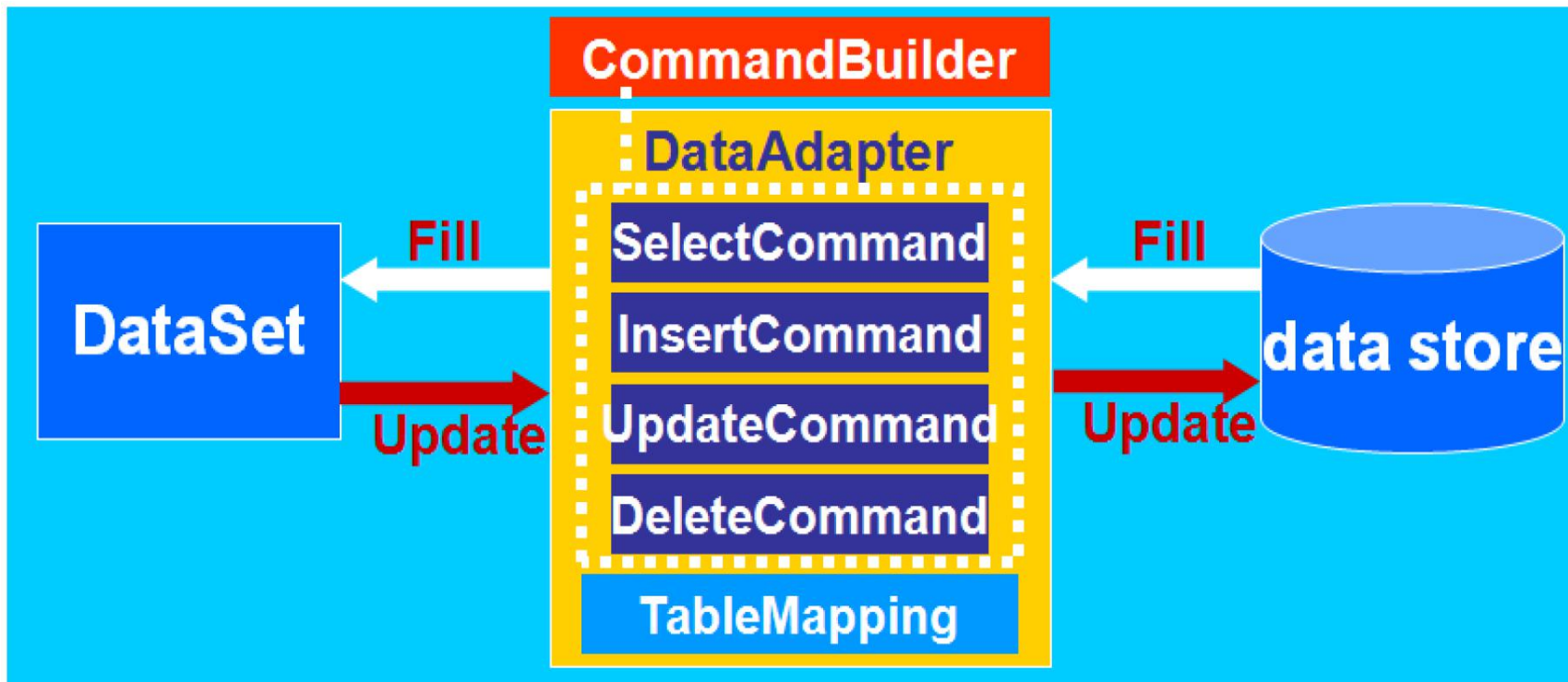
```
string sSql = "EXECUTE AddUser @FullName,@UserName,@Password,@Phone,@Email,@Status,@Description";  
mySqlCommand = new SqlCommand(sSql,mySqlConnection);  
//định nghĩa các tham số @FullName,@UserName,@Password, ...  
mySqlCommand.ExecuteNonQuery();
```

## 3. Xóa một User

```
string sSql = "EXECUTE DeleteUser @UserID";  
//định nghĩa tham số @UserID  
mySqlCommand = new SqlCommand(sSql,mySqlConnection);  
mySqlCommand.ExecuteNonQuery();
```

# Lớp SqlDataAdapter

- Một đối tượng lớp SqlDataAdapter là một đường chuyển dữ liệu giữa ứng dụng và cơ sở dữ liệu:
  - Truy vấn dữ liệu: **Fill**
  - Cập nhật dữ liệu: **Update**



# Lớp SqlDataAdapter

---

- Tạo đối tượng lớp SqlDataAdapter:

- `SqlDataAdapter(SqlCommand mySqlCommand);`

- `SqlDataAdapter(string sSql, SqlConnection mySqlConnection);`

- Các thuộc tính:

- ❑ DeleteCommand: thực hiện lệnh DELETE hoặc thủ tục.

- ❑ InsertCommand: thực hiện lệnh INSERT hoặc thủ tục.

- ❑ UpdateCommand: thực hiện lệnh UPDATE hoặc thủ tục.

- ❑ SelectCommand: thực hiện lệnh SELECT hoặc thủ tục.

- Các phương thức:

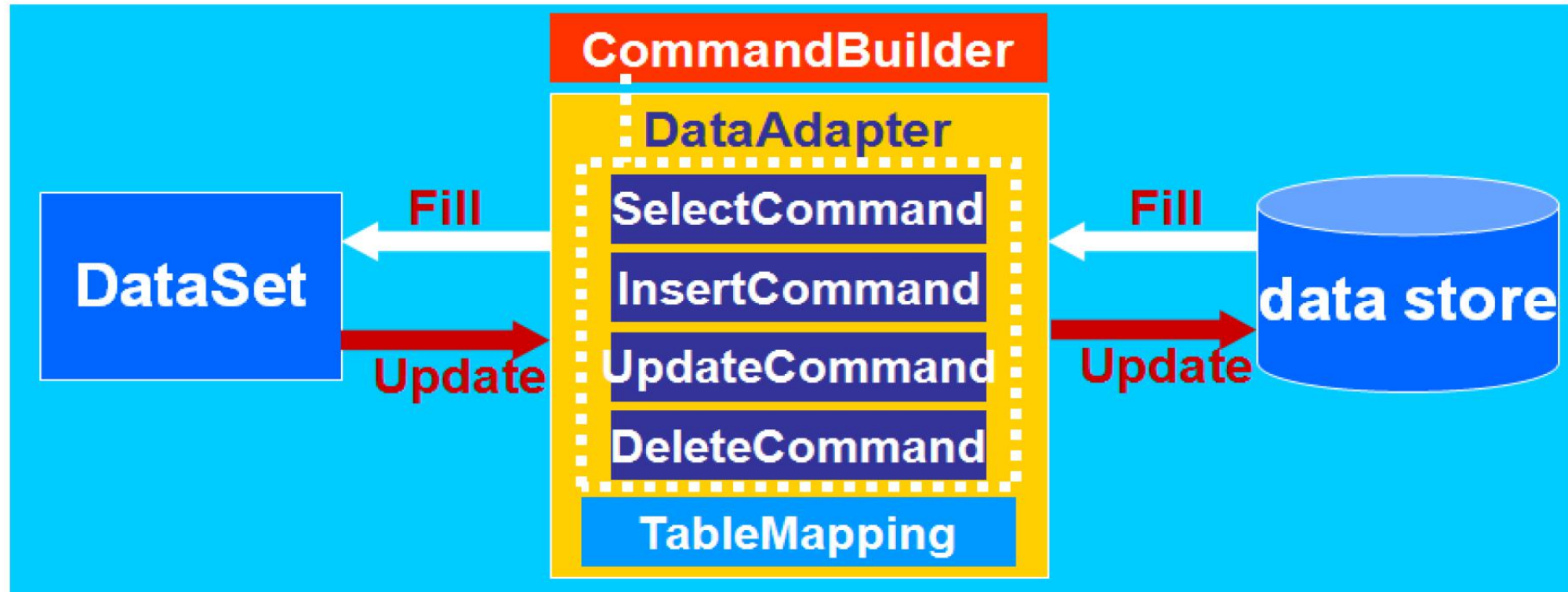
- ❑ Fill(): truy vấn dữ liệu vào DataSet hoặc DataTable.

- ❑ Update(): cập nhật dữ liệu từ DataSet hoặc DataTable.



# Lớp SqlCommandBuilder

- Sử dụng lớp `SqlCommandBuilder` để giảm mã lệnh viết cho các thuộc tính `InsertCommand`, `UpdateCommand` và `DeleteCommand` của `SqlDataAdapter`.
- Khi thay đổi dữ liệu trong `DataSet`/`DataTable` và gọi phương thức `Update()` thì dữ liệu trong `DataSet`/`DataTable` sẽ đồng bộ với cơ sở dữ liệu.



# Lớp SqlCommandBuilder

---

- Các hạn chế của lớp SqlCommandBuilder - trong trường hợp có cập nhật lại dữ liệu :
  - ❑ Câu lệnh SELECT của đối tượng lớp SqlDataAdapter phải được truy vấn dữ liệu từ một bảng.
  - ❑ Bảng truy vấn trong cơ sở dữ liệu phải chứa một khóa chính.
  - ❑ Khóa chính của bảng phải có trong câu lệnh SELECT truy vấn dữ liệu.

# Lớp SqlCommandBuilder

---

## ■ Truy vấn dữ liệu vào đối tượng lớp DataTable:

- ❑ Tạo đối tượng lớp SqlConnection và kết nối cơ sở dữ liệu:

```
SqlConnection mySqlConnection = new SqlConnection(conStr);  
mySqlConnection.Open();
```

- ❑ Tạo đối tượng lớp SqlDataAdapter:

```
SqlDataAdapter mySqlDataAdapter = new SqlDataAdapter(sSql, mySqlConnection);
```

- ❑ Tạo đối tượng lớp SqlCommandBuilder (khi có cập nhật sau đó):

```
SqlCommandBuilder mySqlCommandBuilder = new SqlCommandBuilder(mySqlDataAdapter);
```

- ❑ Tạo đối tượng lớp DataTable và truy vấn dữ liệu:

```
DataTable myDataTable = new DataTable();  
mySqlDataAdapter.Fill(myDataTable);
```

# Lớp SqlCommandBuilder

---

- Cập nhật dữ liệu vào đối tượng lớp DataTable:

- Nhập dữ liệu vào đối tượng lớp DataTable:

```
DataRow myDataRow = myDataTable.NewRow();  
myDataRow["FirstName"] = "Peter";  
myDataRow["LastName"] = "Aitken";  
myDataTable.Rows.Add(myDataRow);  
mySqlDataAdapter.Update(myDataTable);
```

- Xóa dữ liệu trong đối tượng lớp DataTable:

```
myDataTable.Rows[pos].Delete();  
myDataAdapter.Update(myDataTable);
```

Chú ý: pos là dòng cần xóa.

# Lớp SqlCommandBuilder

---

- Sửa dữ liệu trong đối tượng lớp DataTable, pos là dòng cần sửa:

```
DataRow editRow =myDataTable.Rows[pos];  
editRow["FirstName"] = "abc";  
editRow["LastName"] = "def";  
myDataAdapter.Update(myDataTable);
```

# Lớp SqlCommandBuilder

## ■ Ví dụ xây dựng form

frmCustomer (SqlDataAdapter + SqlCommandBuilder)

Họ và tên (\*):

Số CCCD:

Địa chỉ:

Điện thoại:

Email:

Mô tả thêm:

Danh sách khách hàng:

CustomerID	FullName	Identification	Address	Phone
------------	----------	----------------	---------	-------

# Lớp ListBox và ComboBox

---

- **ListBox**: Cho phép người dùng xem và chọn các dòng dữ liệu từ danh sách.
- **ComboBox**: Sự kết hợp của TextBox và ListBox.
- Các thuộc tính thường dùng:
  - **DataSource**: Nguồn dữ liệu, là một DataTable.
  - **DisplayMember**: Cột hiển thị trong ListBox.
  - **ValueMember**: Cột giá trị trả về khi chọn ListBox.
  - **SelectedIndex**: Dòng hiện thời được chọn.
  - **SelectedValue**: Giá trị được chọn trên ListBox.

# Xử lý lỗi ngoại lệ dùng lớp SQLException

---

## ■ Khối lệnh xử lý lỗi:

```
try
{
    <khối lệnh có thể xảy ra lỗi>
} catch (SQLException mySQLException)
{
    <xử lý lỗi sinh ra do cơ sở dữ liệu>
}
```

## ■ Các thuộc tính cơ bản của lớp SQLException:

- ❑ Number: Số hiệu lỗi.
- ❑ Message: Xâu chứa thông báo lỗi.
- ❑ StackTrace: Vị trí sinh ra lỗi gồm dòng lệnh và phương thức sinh ra lỗi.



# Xử lý lỗi ngoại lệ dùng lớp SQLException

---

- Một số mã lỗi của đối tượng lớp SQLException:
  - ❑ 53: Tên máy chủ cơ sở dữ liệu sai.
  - ❑ 4060: Không đúng tên cơ sở dữ liệu.
  - ❑ 18456: Không đúng tên và/hoặc mật khẩu truy nhập.
  - ❑ 547: Dữ liệu nhập/sửa vi phạm ràng buộc khóa ngoài.
  - ❑ 2601|2627: Trùng giá trị dữ liệu trường khóa.
  - ❑ 8152: Dữ liệu lưu trữ vào trường của bảng quá dài.

# Xây dựng lớp dịch vụ dữ liệu

---

- Mọi form không phụ thuộc vào cơ sở dữ liệu.
- Chỉ kết nối dữ liệu một lần khi chạy ứng dụng.
- Mỗi form sử dụng các đối tượng lớp.
  - Truy vấn dữ liệu vào DataTable
  - Cập nhật dữ liệu từ DataTable vào cơ sở dữ liệu.
  - Thực hiện các câu lệnh SQL thao tác với dữ liệu.
- Khai báo:

```
private static SqlConnection mySqlConnection;  
private SqlDataAdapter myDataAdapter;  
public DataServices()  
{  
}
```

# Xây dựng lớp dịch vụ dữ liệu

---

```
public bool OpenDB(string myComputer, string myDB, string uid, string psw)
{
    string conStr = "server = '" + myComputer + "'; database = '" + myDB + "';
                    uid = '" + uid + "'; pwd = '" + psw + "'";

    try
    {
        MySqlConnection = new SqlConnection(conStr);
        MySqlConnection.Open();
    }
    catch (SqlException ex)
    {
        MessageBox.Show(ex.Message, "Error " + ex.Number.ToString());
        MySqlConnection = null;
        return false;
    }
    return true;
}
```

# Xây dựng lớp dịch vụ dữ liệu

---

```
public DataTable RunQuery(string sSql)
{
    DataTable myDataTable = new DataTable();
    try
    {
        myDataAdapter = new SqlDataAdapter(sSql, mySqlConnection);
        SqlCommandBuilder mySqlCommandBuilder = new SqlCommandBuilder(myDataAdapter);
        myDataAdapter.Fill(myDataTable);
    } catch (SqlException ex)
    {
        MessageBox.Show(ex.Message, "Error " + ex.Number.ToString());
        return null;
    }
    return myDataTable;
}
```

# Xây dựng lớp dịch vụ dữ liệu

---

```
public void Update(DataTable myDataTable)
{
    try
    {
        myDataAdapter.Update(myDataTable);
    } catch (SQLException ex)
    {
        MessageBox.Show(ex.Message, "Error " + ex.Number.ToString());
    }
}

public void ExecuteNonQuery(string sSql)
{
    SqlCommand mySqlCommand = new SqlCommand(sSql, mySqlConnection);
    try
    {
        mySqlCommand.ExecuteNonQuery();
    } catch (SQLException ex)
    {
        MessageBox.Show(ex.Message, "Error " + ex.Number.ToString());
    }
}
```

# Xây dựng lớp dịch vụ dữ liệu

- Sử dụng lớp dịch vụ dữ liệu xây dựng form:

The screenshot shows a Windows application window titled "frmCustomer". The window is divided into two main sections. On the left is a form for entering customer information, and on the right is a table titled "Danh sách khách hàng" (Customer List).

The form contains the following fields and buttons:

- Họ và tên (\*):
- Số CMND:
- Địa chỉ:
- Điện thoại (\*):
- Email:
- Mô tả thêm:
- Buttons: Thêm mới, Sửa, Xóa, Ghi, Hủy bỏ, Kết thúc

The table "Danh sách khách hàng" has the following columns:

CustomerID	FullName	Identification	Address
------------	----------	----------------	---------

# Xây dựng lớp dịch vụ dữ liệu

- Sử dụng lớp dịch vụ dữ liệu xây dựng form:

frmAddBook

Loại sách: Ngôn ngữ lập trình máy tính

Tên sách: Ngôn ngữ lập trình C

Các tác giả: (Một tác giả 1 dòng)  
Trần Xuân Hào  
Nguyễn Quang Ninh  
Lê Văn Minh

Nhà xuất bản: Đại học Vinh

Phiên bản: 1

Năm xuất bản: 2019

Ngôn ngữ: Tiếng Việt

Giá bìa: 25000

Mô tả thêm: Giáo trình dùng cho SV

Thêm mới Ghi Hủy ghi Kết thúc

Danh sách các quyển sách

	BookID	CategoryID	Book Name	PublisherID
▶	16	1040	Ngôn ngữ lập trìn...	14
	17	41	Tự học Excel	4

# Kết thúc

---



**HỎI & TRẢ LỜI**