

# Chương 5

## LẬP TRÌNH CƠ SỞ DỮ LIỆU

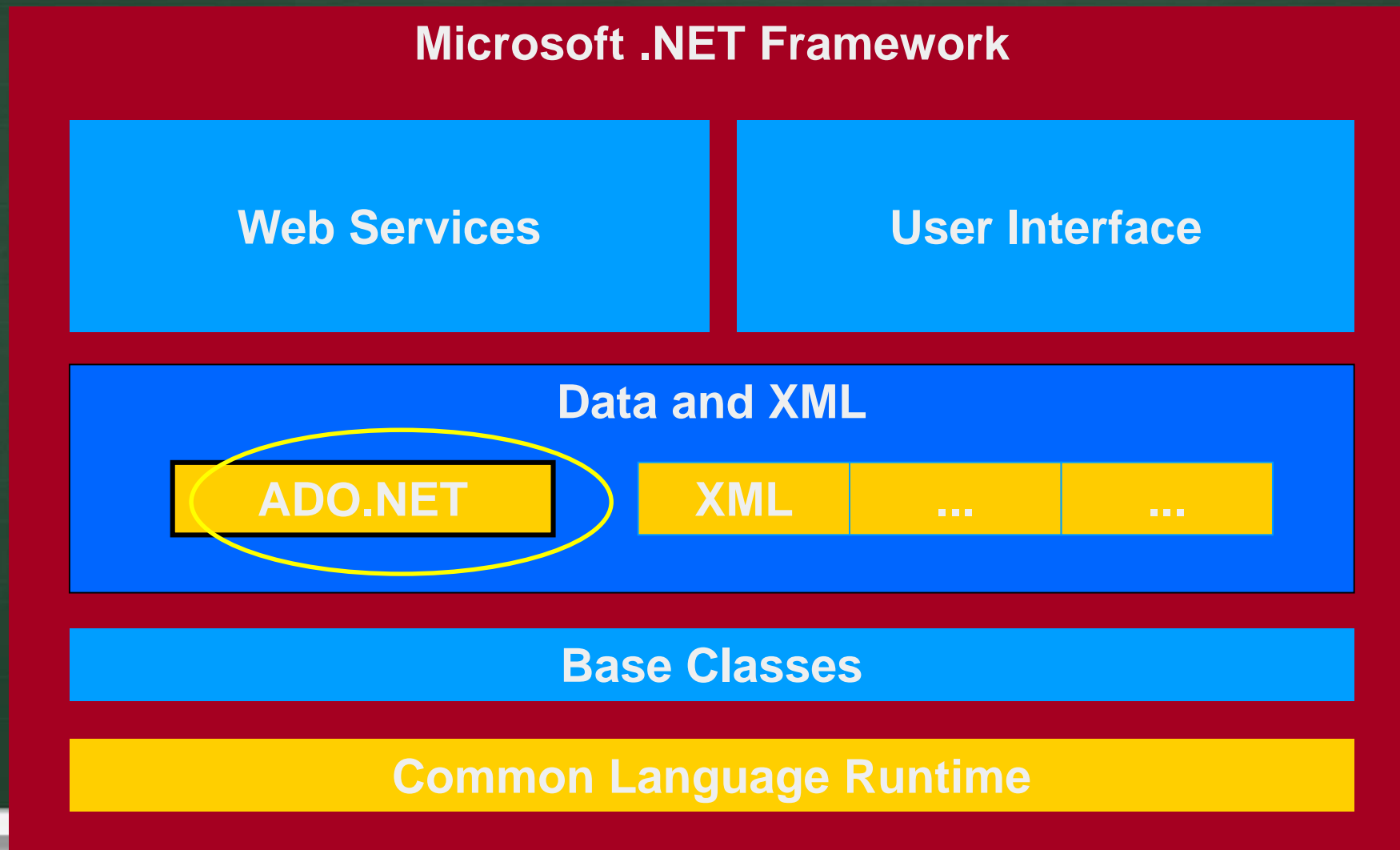
# Nội dung

- 1 Tổng quan về ADO.NET
- 2 Kết nối cơ sở dữ liệu bằng ADO.NET
- 3 Xây dựng ứng dụng minh họa

# 1. Tổng quan về ADO.NET

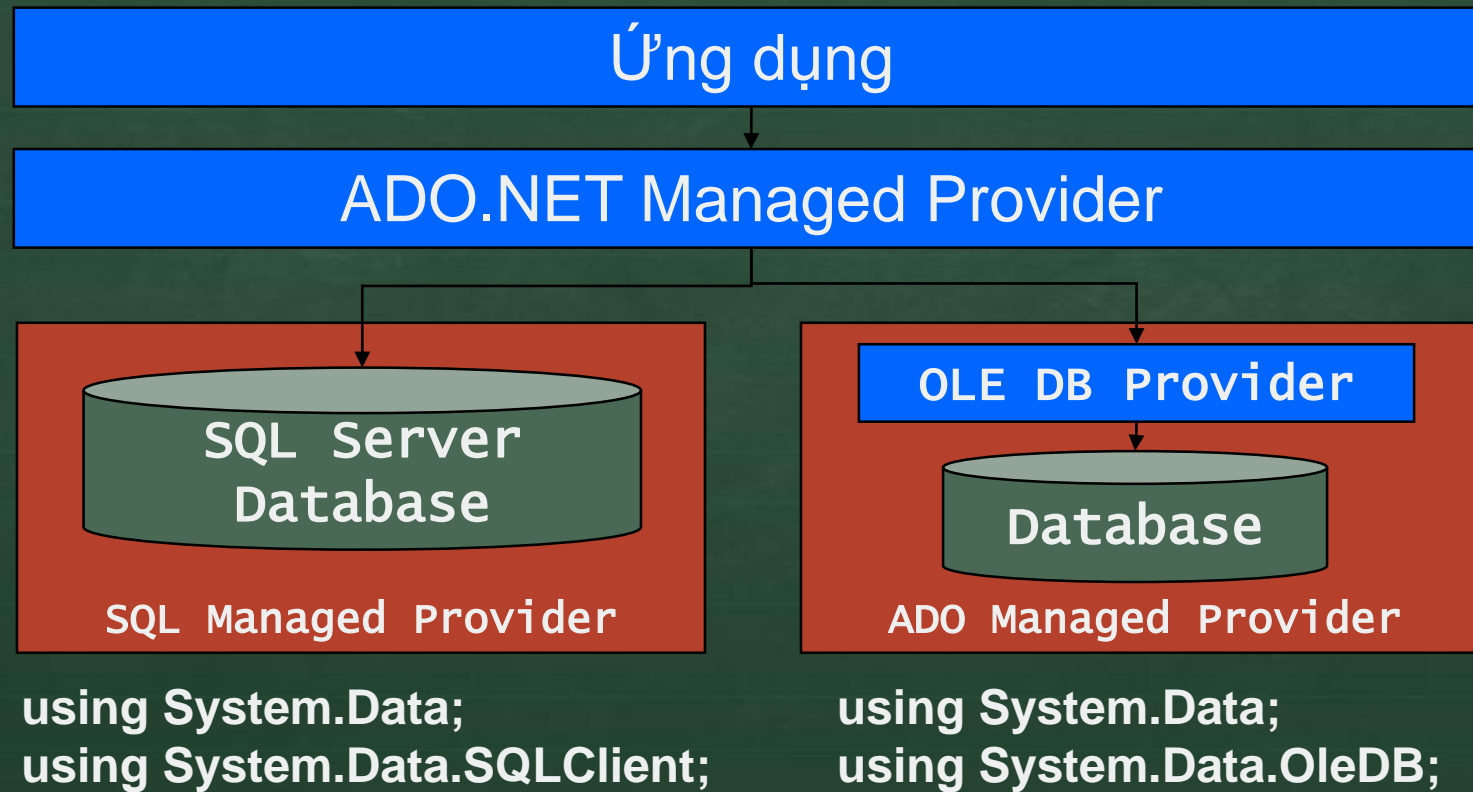
- ☐ ADO.NET và .NET framework
- ☐ ADO.NET là gì?
- ☐ Kiến trúc ADO.NET
- ☐ Các thành phần trong ADO.NET

# ADO.NET và .NET framework



# ADO.NET là gì?

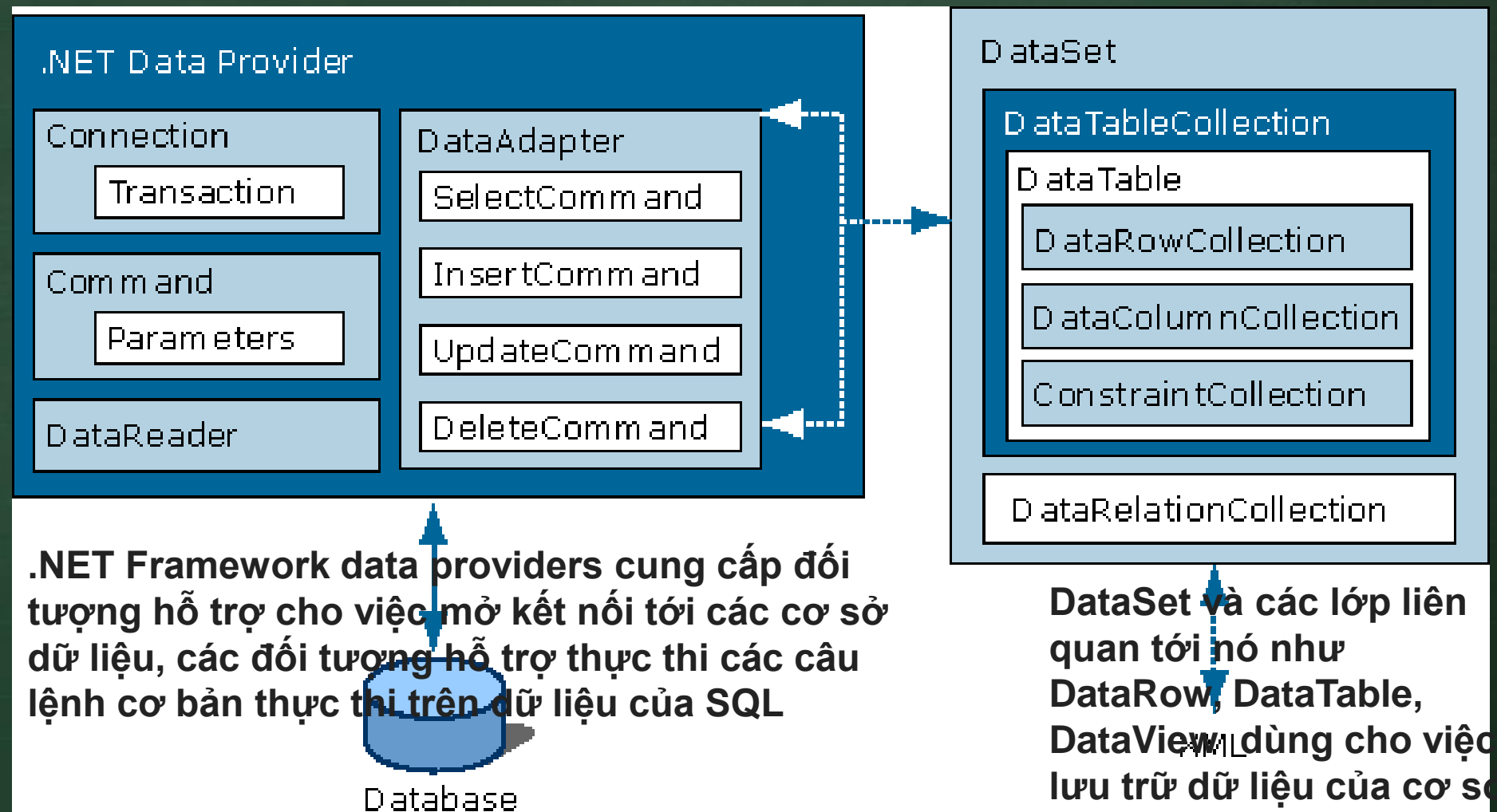
- ADO.NET là tập hợp các lớp, interface, cấu trúc, kiểu dữ liệu định sẵn để quản lý việc truy xuất với dữ liệu



01/08/2023

# Kiến trúc ADO.NET

1. Connection
2. Command
3. DataReader
4. DataAdapter
5. Dataset



## 2. Kết nối CSDL bằng ADO.NET

- Connection
- Command
- DataReader
- DataAdapter
- Dataset

## 2.1. Connection

- Thiết lập và quản lý kết nối với CSDL
- Các loại Connection:
  - SqlConnection
  - OleDbConnection
  - ...
- Thuộc tính quan trọng:
  - ConnectionString
- Phương thức quan trọng:
  - Open()
  - Close()
- Chú ý:
  - Luôn đóng Connection sau khi sử dụng



## 2.1.Connection (tiếp)

- Ví dụ:

```
SqlConnection con; //Đối tượng để kết nối  
con = new SqlConnection(); //Khởi tạo  
con.ConnectionString = @"Data Source=  
    (LocalDB)\MSSQLLocalDB; AttachDbFilename=“  
    +Application.StartupPath+ @"\QLSV.mdf;  
    Integrated Security=True;Connect  
    Timeout=30;User Instance=True”;  
con.Open(); //Mở kết nối
```



***Tạo ConnectString chính xác và nhanh nhất?***

## 2.2. Command

- Thực hiện các thao tác với CSDL
  - DDL, DML, gọi thủ tục, ....
- Các loại Command:
  - SqlCommand
  - OleDbCommand
  - ....
- Thuộc tính quan trọng:
  - Connection
  - CommandText
- Phương thức quan trọng:
  - ExecuteNonQuery()  
*(thực hiện lệnh: INSERT, UPDATE, DELETE)*
  - ExecuteScalar()  
*(thực hiện lệnh SELECT trả về 1 giá trị)*
  - ExecuteReader()  
*(thực hiện lệnh SELECT trả về một hay nhiều bản ghi)*

## 2.2.Command (tiếp)

▪ Ví dụ:

```
SqlCommand cmd = new SqlCommand();  
cmd.Connection = con;  
cmd.CommandText = "UPDATE NhanVien set Luong = Luong + 100000  
WHERE MaNV = 01";  
cmd.ExecuteNonQuery();  
...  
con.Close();
```

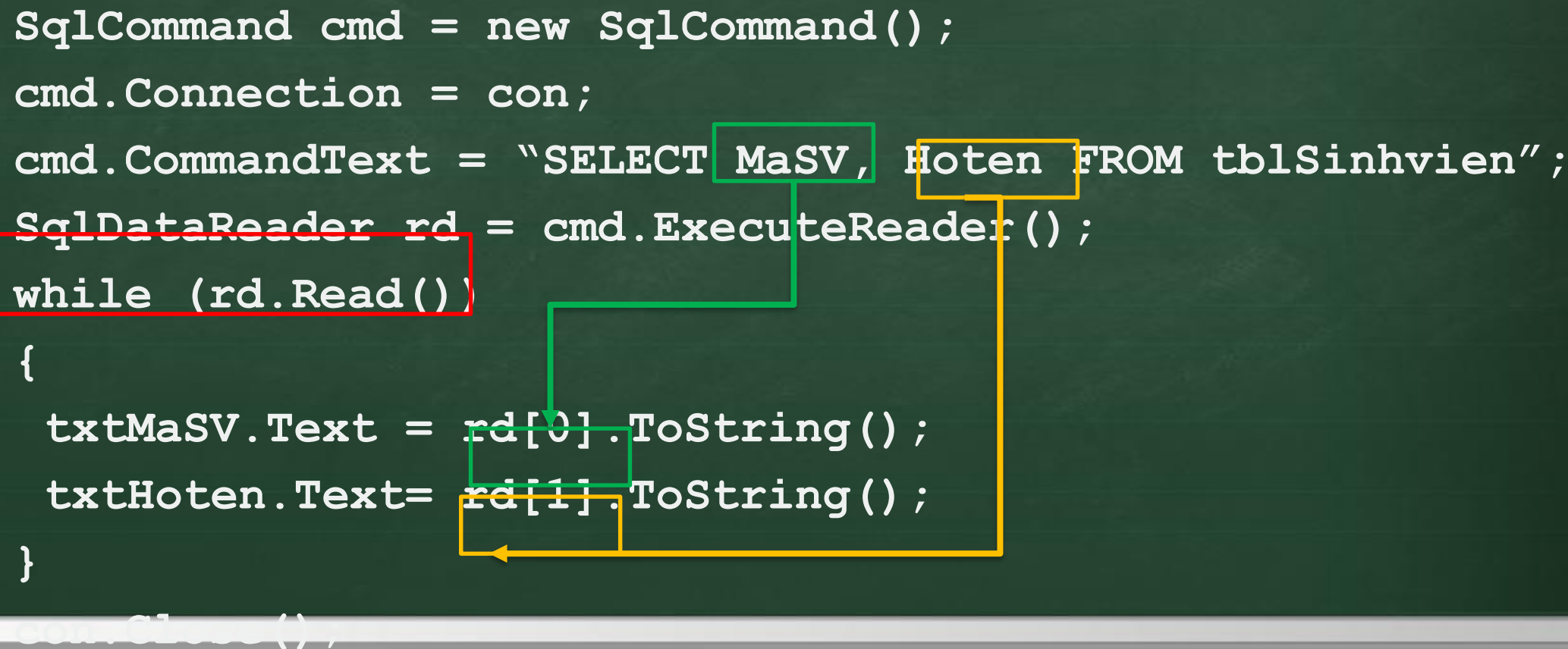
## 2.3. DataReader

- Có dạng con trỏ, dùng để handle dữ liệu trả về từ CSDL
- Đặc điểm:
  - Con trỏ không thể lùi
  - Thường handle dữ liệu trả về từ phương thức ExecuteReader() của Command
- Có 2 loại:
  - SqlDataReader
  - OleDbReader
- Khuyến cáo:
  - Đối với các form chỉ SELECT dữ liệu, **NÊN DÙNG** DataReader để có tốc độ xử lý nhanh hơn

## 2.3. DataReader (tiếp)

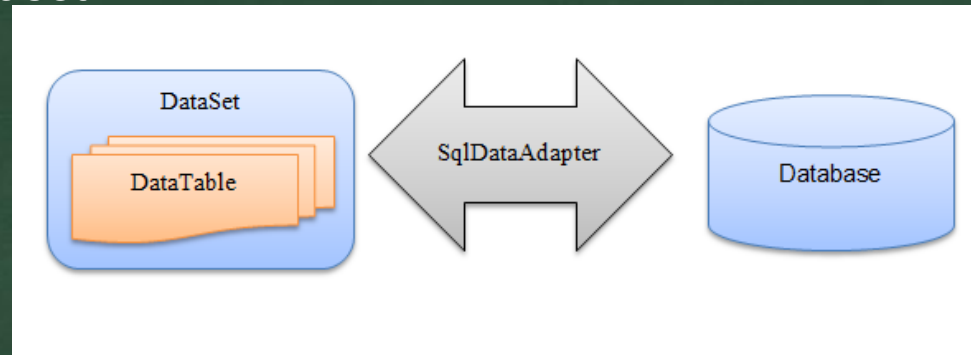
- Ví dụ:

```
SqlCommand cmd = new SqlCommand();  
cmd.Connection = con;  
cmd.CommandText = "SELECT MaSV, Hoten FROM tblSinhvien";  
SqlDataReader rd = cmd.ExecuteReader();  
while (rd.Read())  
{  
    txtMaSV.Text = rd[0].ToString();  
    txtHoten.Text = rd[1].ToString();  
}  
con.Close();
```



## 2.4. DataAdapter

- Là cầu nối giữa CSDL và Dataset



- Các thuộc tính quan trọng:

- SelectCommand
- InsertCommand

- Các phương thức quan trọng:

- Fill()
- Update()

- Có cơ chế tự động đóng Connection

■ UpdateCommand

■ DeleteCommand

## 2.4. DataAdapter (tiếp)

- Ví dụ - Hiển thị dữ liệu:

```
string sql;  
sql = "SELECT * from tblSinhVien";  
//Đối tượng DataAdapter  
SqlDataAdapter MyData = new  
SqlDataAdapter(sql,con);  
tblSinhvien = new DataTable(); //Khởi tạo bảng  
//Đổ dữ liệu từ DataAdapter vào bảng  
MyData.Fill(tblSinhvien);  
dataGridView.DataSource = tblSinhvien;
```



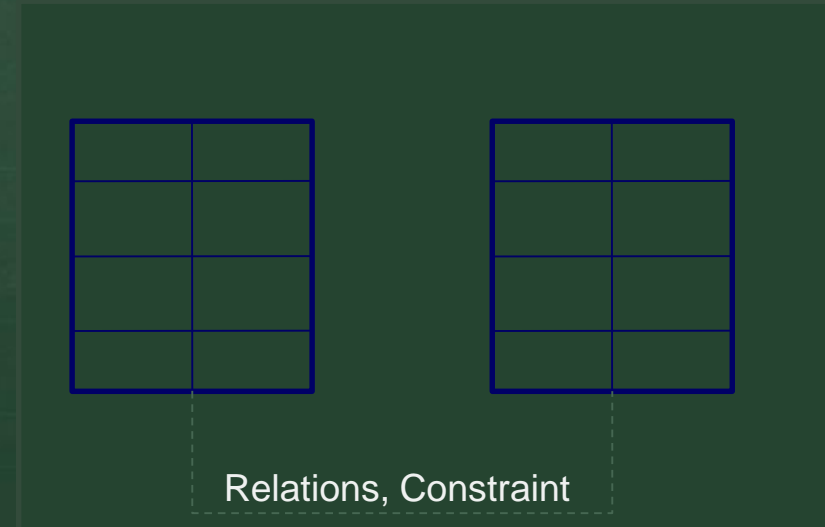
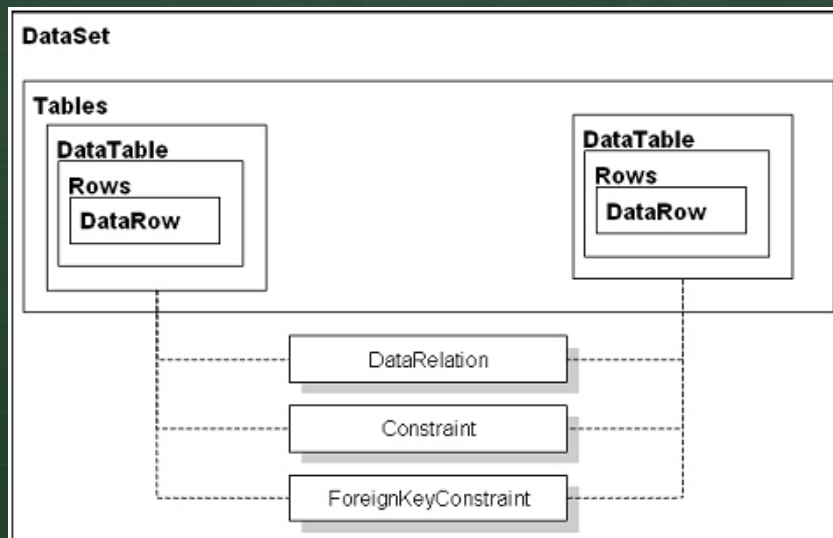
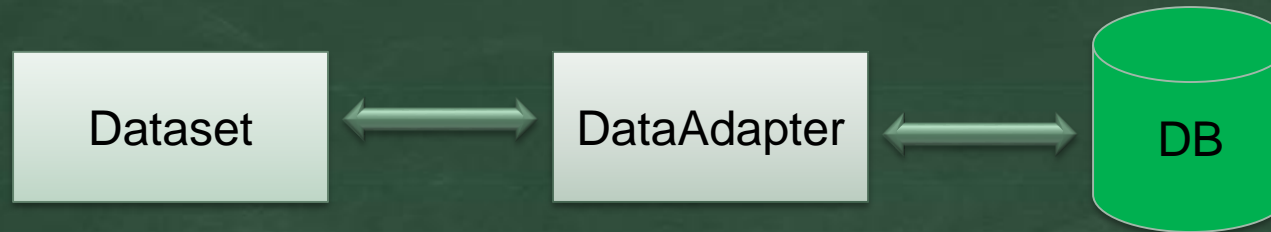
# Các bước làm việc với CSDL

- Bước 1: Tạo kết nối (thiết lập chuỗi kết nối, khai báo và tạo Connection)
- Bước 2: Mở kết nối dữ liệu
- Bước 3: Tạo lệnh SQL
- Bước 4: Thực thi lệnh SQL
- Bước 5: Đóng kết nối và hủy các đối tượng (nếu cần)



## 2.5. Dataset

- Là đối tượng lưu dữ liệu trả về từ CSDL



**Dataset**

**Database**

### 3. Xây dựng ứng dụng minh họa

- Bài toán

Xây dựng ứng dụng quản lý sinh viên đơn giản

- Cho phép đọc dữ liệu từ csdl sinh viên
- Thực hiện các thao tác: Thêm, sửa, xoá
- Cơ sở dữ liệu: gồm bảng SINHVIEN(MaSV, Hoten, Ngaysinh, Khoa, Lop, Diachi)

- Các bước thực hiện

- Bước 1: Tạo cơ sở dữ liệu
- Bước 2: Thiết kế giao diện
- Bước 3: Thực hiện kết nối cơ sở dữ liệu
- Bước 4: Xử lý các sự kiện

### 3. Xây dựng ứng dụng minh họa

- Bước 1: Tạo cơ sở dữ liệu
  - Tạo ứng dụng mới
  - Tạo cơ sở dữ liệu
    - Ở khung Solution Explorer, nháy phải chuột lên tên ứng dụng, chọn Add → New Item...
    - Chọn Data → Service-based Database (hoặc *SQL Database* trong *Visual Studio Net 2019*)
    - Tạo bảng tblSinhvien

dbo.tblSinhVien:...\IN\DEBUG\QLSV.MDF			
Program.cs   frmMain.cs			
	Column Name	Data Type	Allow Nulls
PK	MaSV	nvarchar(10)	<input type="checkbox"/>
	Hoten	nvarchar(50)	<input type="checkbox"/>
	Ngaysinh	datetime	<input type="checkbox"/>
	Khoa	nvarchar(20)	<input type="checkbox"/>
	Lop	nvarchar(20)	<input type="checkbox"/>
	Diachi	nvarchar(100)	<input checked="" type="checkbox"/>

### 3. Xây dựng ứng dụng minh họa

- Bước 2: Thiết kế giao diện

- Các đối tượng:

- Textbox:

- txtMaSV

- txtHoten

- txtKhoa

- txtLop

- txtDiachi

- Maskedtextbox

- txtNgaysinh

- DataGridView

- DataGridView

- Buttons

- btnMoi, btnSua, btnXoa, btnLuu, btnHuy, btnThoat

### 3. Xây dựng ứng dụng minh họa

- Bước 3: Thực hiện kết nối cơ sở dữ liệu

SqlConnection con; //Đối tượng để kết nối

DataTable tblSinhvien; //Đối tượng lưu bảng sinh viên

**public void Connect() //Kết nối**

{ con = new SqlConnection(); //Khởi tạo đối tượng

con.ConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB;

AttachDbFilename = E:\Luye\Hướng dẫn thực hành

\ConnectSQL\QLBanHang.mdf;Integrated Security=True";

con.Open(); //Mở kết nối

}

**public void Disconnect() //Ngắt kết nối**

{ if (con.State == ConnectionState.Open) //nếu đang mở

{ con.Close(); //đóng

con.Dispose(); //huỷ

}

### 3. Xây dựng ứng dụng minh họa

- Bước 4: Xử lý các sự kiện
  - Hiển thị thông tin trong DataGridView

```
public void LoadDataGridView()  
{  
    string sql;  
    sql = "SELECT * from tblSinhVien";  
    SqlDataAdapter MyData = new SqlDataAdapter(sql, con); //Đối tượng  
DataAdapter  
    tblSinhvien = new DataTable(); //Khởi tạo bảng  
    MyData.Fill(tblSinhvien); //Đổ dữ liệu từ DataAdapter vào bảng  
    dataGridView.DataSource = tblSinhvien;  
}
```

### 3. Xây dựng ứng dụng minh họa

- Bước 4: Xử lý các sự kiện
  - Thực hiện lệnh SQL

```
public void RunSQL(string sql) //Thực hiện một câu lệnh SQL
{
    SqlCommand cmd = new SqlCommand(); //Đối tượng để thực hiện lệnh
    cmd.CommandText = sql;
    cmd.Connection = con;
    try
    { cmd.ExecuteNonQuery(); //Thực hiện câu lệnh
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
```



### 3. Xây dựng ứng dụng minh họa

- Bước 4: Xử lý các sự kiện
  - Nhấn nút Sửa

```
private void btnSua_Click(object sender, EventArgs e)
{
    string sql;
    sql = "UPDATE tblSinhVien SET Hoten='" + txtHoten.Text +
        "',Ngaysinh='" + txtNgaysinh.Text +
        "',Khoa='" + txtKhoa.Text + "',Lop='" + txtLop.Text +
        "',Diachi='" + txtDiachi.Text + "' WHERE MaSV='" + txtMaSV.Text + "'";
    RunSQL(sql); //thực hiện lệnh sql
    LoadDataGridView(); //hiển thị lại thông tin lên DataGridView
}
```



### 3. Xây dựng ứng dụng minh họa

- Bước 4: Xử lý các sự kiện

- Nhấn nút Lưu

```
private void btnLuu_Click(object sender, EventArgs e)
{
    string sql;
    sql = "SELECT MaSV FROM tblSinhVien WHERE MaSV=N'" + txtMaSV.Text + "'";
    SqlDataAdapter MyData = new SqlDataAdapter(sql, con);
    DataTable table = new DataTable();
    MyData.Fill(table);
    if (table.Rows.Count > 0)
    {
        MessageBox.Show("Mã sinh viên này đã tồn tại");
        return;
    }

    //Thực hiện chèn thêm mới
    sql = "INSERT INTO tblSinhVien VALUES ('" + txtMaSV.Text + "', '" + txtHoten.Text + "', '" +
    txtNgaysinh.Text + "', '" + txtKhoa.Text + "', '" + txtLop.Text + "', '" + txtDiachi.Text +
    "')";

    RunSQL(sql);
    LoadDataGridView();
}
```

### 3. Xây dựng ứng dụng minh họa

- Bước 4: Xử lý các sự kiện
  - Nhấn nút Xóa

```
private void btnXoa_Click(object sender, EventArgs e)
{
    string sql;
    if (MessageBox.Show("Bạn có muốn xóa không?", "Thông báo",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)
    {
        sql = "DELETE tblSinhVien WHERE MaSV='" + txtMaSV.Text + "'";
        RunSQL(sql);
        LoadDataGridView();
        ResetValue();
    }
}
```

# Tài liệu tham khảo

- <https://docs.microsoft.com/en-us/visualstudio/data-tools/accessing-data-in-visual-studio?view=vs-2019>
- <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/windows-forms-data-binding?view=netframeworkdesktop-4.8>
- <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/controls/load-save-and-cancel-bindingnavigator?view=netframeworkdesktop-4.8>
- <https://docs.microsoft.com/en-us/dotnet/api/system.data.datatable?view=net-5.0>