

# LANGUAGE INTEGRATED QUERY

- Biến cục bộ kiểu ngầm định (Implicitly Typed Local Variables)
- Lớp/Kiểu “tạm thời” (Anonymous types)
- Phương thức mở rộng (Extension Methods)
- Biểu thức Lambda (Lambda Expressions)



- TRUY CẬP DỮ LIỆU VỚI LINQ
- LINQ TO SQL



# TRUY CẬP DỮ LIỆU VỚI LINQ

- Truy vấn dữ liệu từ các nguồn dữ liệu theo mô hình hướng đối tượng bằng chính ngôn ngữ lập trình (không lệ thuộc ngôn ngữ của hệ quản trị CSDL).
- Truy vấn dữ liệu các mảng, tập hợp, XML..
- Các lớp hỗ trợ LINQ thuộc các không gian tên: **System.Linq**,  
**System.Linq.Expressions**

# TRUY CẬP DỮ LIỆU VỚI LINQ

- Lập trình truy vấn dữ liệu theo phương pháp truyền thống → tiềm ẩn nhiều lỗi.

```
using (SqlConnection connection = new SqlConnection("..."))
{
    connection.Open();
    SqlCommand command = connection.CreateCommand();
    command.CommandText =
        @"SELECT Name, Country
        FROM Customers
        WHERE City = @City";
    command.Parameters.AddWithValue("@City", "Paris");
    using (SqlDataReader reader = command.ExecuteReader())
    {
        while (reader.Read())
        {
            string name = reader.GetString(0);
            string country = reader.GetString(1);
            ...
        }
    }
}
```

1 SQL query in a string

2 Loosely bound parameters

3 Loosely typed columns

## Mục tiêu thiết kế của LINQ:

- Thống nhất cú pháp ngôn ngữ truy vấn trên các loại dữ liệu (objects, relational data, XML)
- Truy vấn dữ liệu bằng ngôn ngữ lập trình.
- Hạn chế lỗi (phát hiện lỗi ở thời điểm biên dịch).



# TRUY CẬP DỮ LIỆU VỚI LINQ

## Mục tiêu thiết kế của LINQ:

- Hỗ trợ người lập trình khi viết các câu truy vấn.
- Hỗ trợ gỡ rối (debug)
- Tương thích hoàn toàn với những công nghệ khác (standard and generic collections, data binding, web and Windows Forms controls)



# LINQ TO SQL

## LINQ TO SQL





# KẾT NỐI ĐẾN CSDL

1. View -> Server Explorer
2. Kích phải chuột lên DataConnection -> Add Connection.



# KẾT NỐI ĐẾN CSDL

Server Explorer

Data Connections

men\_pc.QLSINHVIEN.dbo

Servers

SharePoint Connections

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:  
Microsoft SQL Server (SqlClient) Change...

Server name:  
MEN\_PC Refresh

Log on to the server

☐ Use Windows Authentication

☒ Use SQL Server Authentication

User name: sa

Password: .....

☐ Save my password

Connect to a database

☒ Select or enter a database name:  
QLSINHVIEN

☐ Attach a database file:  
Browse...

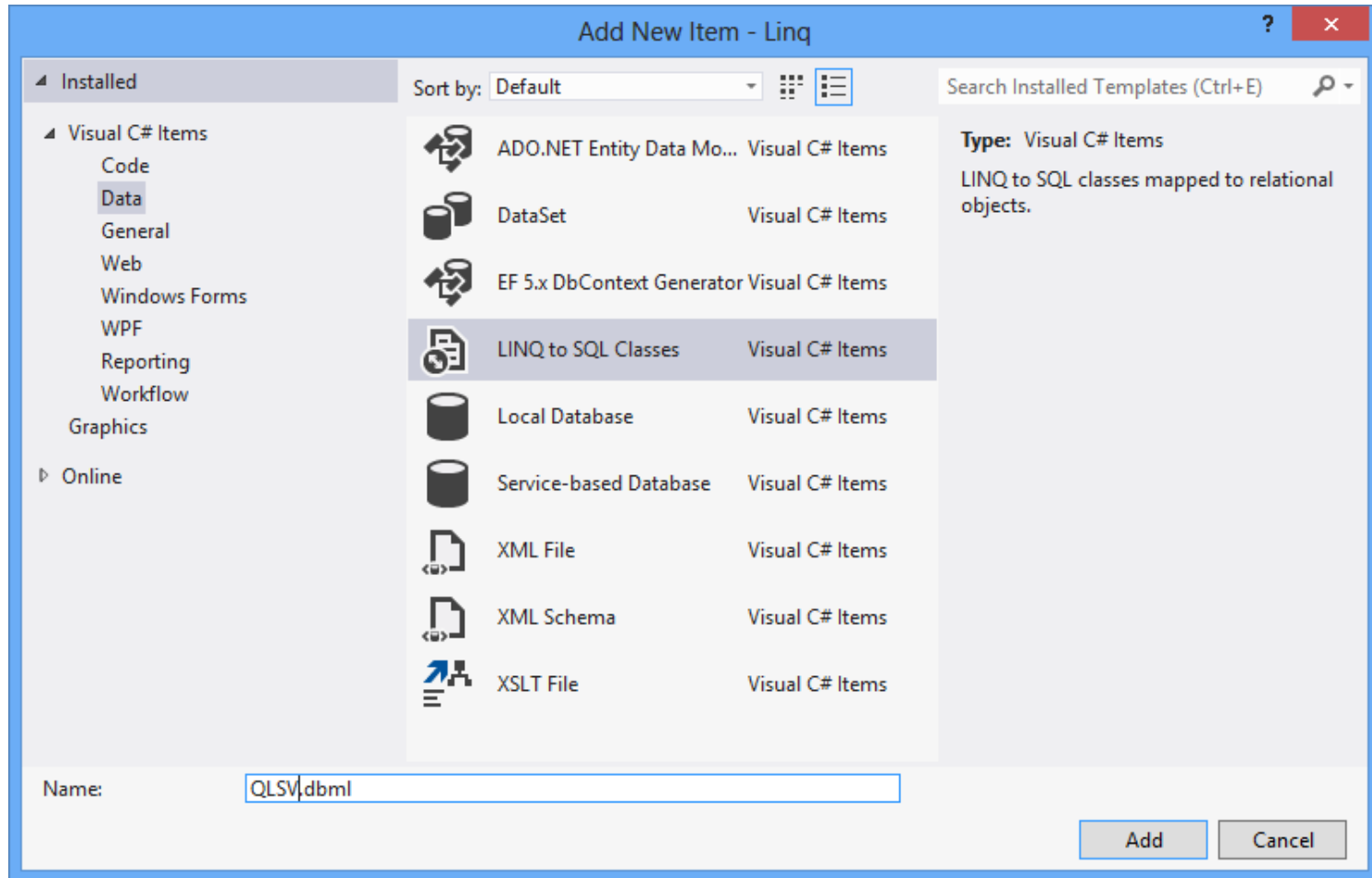
Logical name:

Advanced...

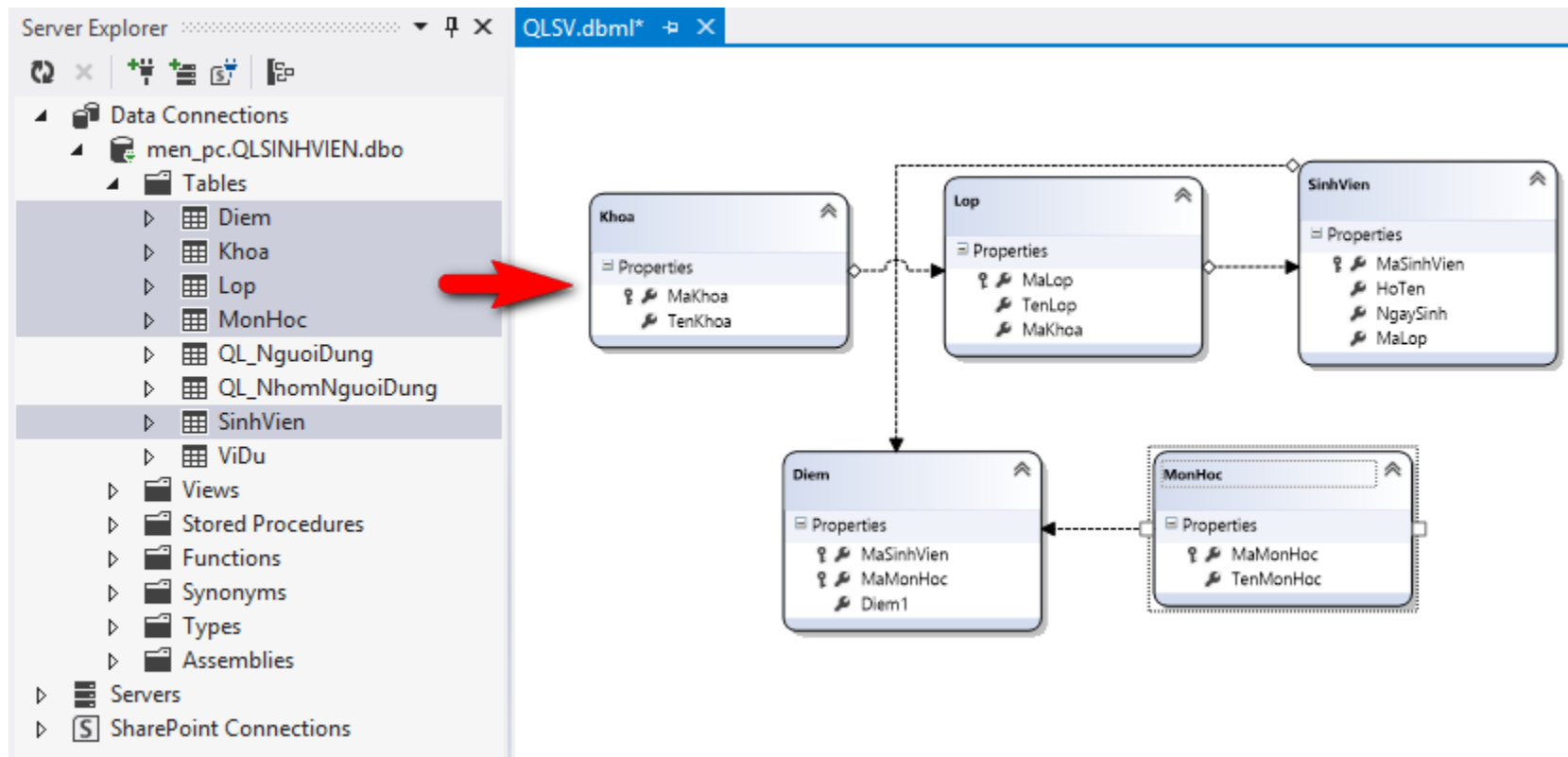
Test Connection OK Cancel

Error List

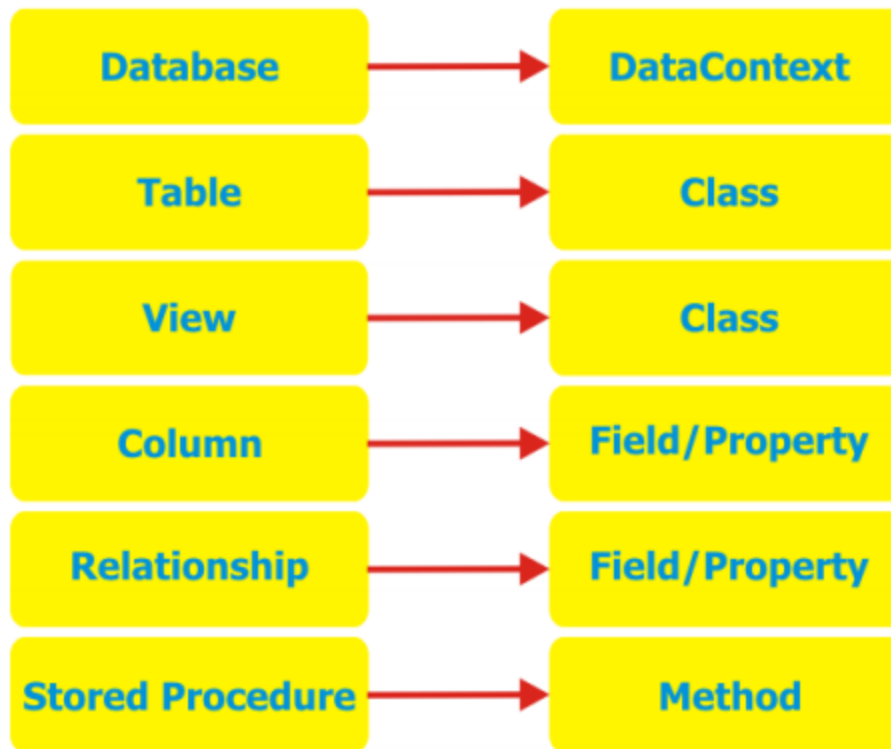
# Tạo ra mô hình dữ liệu LINQ TO SQL



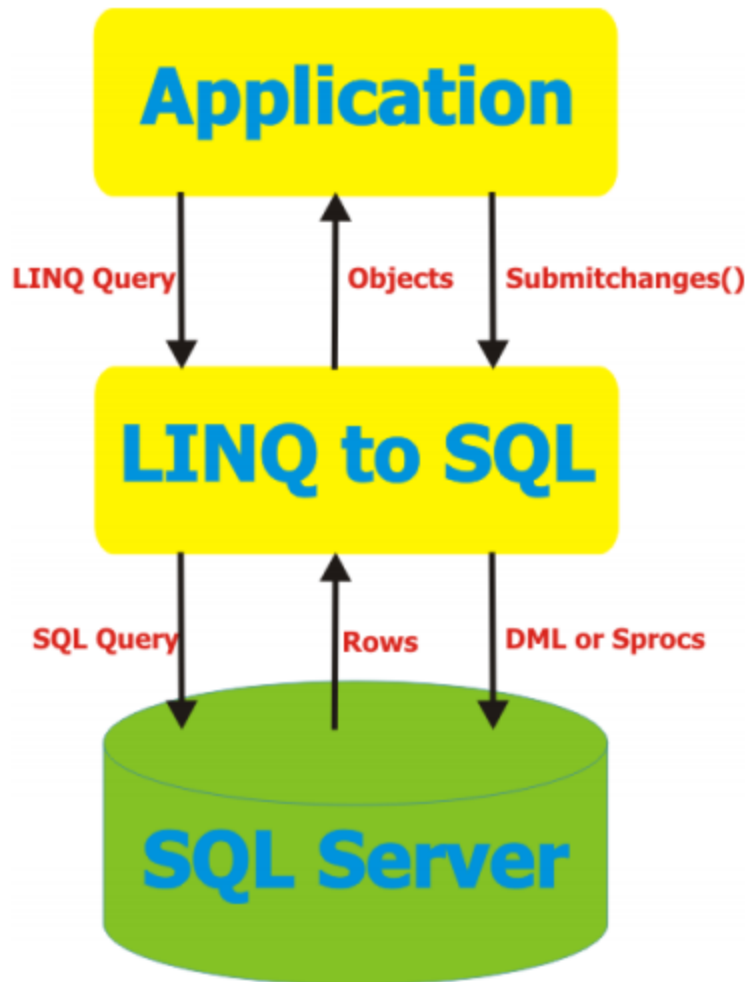
# Tạo ra mô hình dữ liệu LINQ TO SQL



## LINQ to SQL Mapping



Là cầu nối giao tiếp giữa Application và SQL Server)



## Hiển thị dữ liệu bảng Môn học

	Mã môn học	Tên môn học
▶	CTDL	Cấu trúc dữ liệu
	HDT	Hướng đối tượng
	LTW	Lập trình windows
	NNLT	Ngôn ngữ lập trình
*		



# CÁC TOÁN TỬ TRONG LINQ TO SQL **SELECT**

```
QLSVDataContext qlsv = new QLSVDataContext();  
var monhocs = from mh in qlsv.MonHocs select mh;  
dataGridView1.DataSource = monhocs;
```



Hiển thị dữ liệu bảng SinhVien (Lựa chọn cột hiển thị Mã sinh viên, Họ tên)

The screenshot shows a Windows application window titled 'frmVD2'. Inside the window, there is a table with two columns: 'Mã sinh viên' (Student ID) and 'Họ Và Tên' (Full Name). The table contains three rows of data. The first row is highlighted in blue. Below the table, there is a large gray rectangular area.

	Mã sinh viên	Họ Và Tên
▶	077023456	Nguyễn Văn A
	077023457	Trần Thị B
	077023458	Nguyễn Văn C



# CÁC TOÁN TỬ TRONG LINQ TO SQL **SELECT**

```
QLSVDataContext qlsv = new QLSVDataContext();  
    var SinhViens = from sv in qlsv.SinhViens select new {  
sv.MaSinhVien,sv.HoTen};  
    dataGridView1.DataSource = SinhViens;
```



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Take, Skip**

Lấy mẫu tin đầu tiên trong bảng (Lấy sinh viên đầu tiên trong bảng SinhVien).



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Take, Skip**

```
QLSVDataContext qlsv=new QLSVDataContext();  
    var SinhViens = (from sv in  
qlsv.SinhViens select sv).Skip(0).Take(1);  
    dataGridView1.DataSource =  
SinhViens;
```



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Take, Skip**

Tương tự trên, nhưng lấy mẫu tin cuối cùng(sinh viên cuối cùng trong bảng SinhVien).

Code????



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Take, Skip**

Tương tự trên, nhưng lấy mẫu tin thứ 2 và 3 (sinh viên thứ 2 và 3 trong bảng SinhVien).

Code????



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Orderby**

Sắp xếp giảm dần theo cột tên (mặc định tăng dần)



# CÁC TOÁN TỬ TRONG LINQ TO SQL **OrderBy**

```
QLSVDataContext qlsv = new  
QLSVDataContext();  
    var SinhViens = from sv in  
qlsv.SinhViens orderby sv.HoTen select new {  
sv.MaSinhVien, sv.HoTen };  
    dataGridView1.DataSource =  
SinhViens;
```





# CÁC TOÁN TỬ TRONG LINQ TO SQL **Distinct**

Loại bỏ các phần tử trùng nhau

```
QLSVDataContext qlsv = new  
QLSVDataContext();  
    var SinhViens = (from sv in  
qlsv.SinhViens orderby sv.HoTen select new {  
sv.MaSinhVien, sv.HoTen }).Distinct();  
    dataGridView1.DataSource =  
SinhViens;
```



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Where**

Lấy điều kiện theo MaKhoa



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Where**

```
QLSVDataContext q1sv = new  
QLSVDataContext();  
        var Khoas = from kh in  
q1sv.Khoas where kh.MaKhoa=="CNTT" select  
new { kh.MaKhoa, kh.TenKhoa };  
        dataGridView1.DataSource =  
Khoas;
```



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Where**

Lấy điều kiện theo Họ (Liên quan đến chuỗi)



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Where**

```
QLSVDataContext qlsv = new  
QLSVDataContext();  
        var SinhViens = from sv in  
qlsv.SinhViens where  
sv.HoTen.Contains("Nguyễn") select new {  
sv.MaSinhVien, sv.HoTen };  
        dataGridView1.DataSource =  
SinhViens;
```



- qwer



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Join**

Liên kết nhiều bảng Thực hiện truy vấn lấy MaSV, Ten, TenLop



# CÁC TOÁN TỬ TRONG LINQ TO SQL **Join**

```
QLSVDataContext qlsv = new  
QLSVDataContext();  
        var SinhViens = from sv in  
qlsv.SinhViens join  
        k in qlsv.Lops  
        on sv.MaLop equals k.MaLop  
select new { sv.MaSinhVien, sv.HoTen,  
k.TenLop };  
        dataGridView1.DataSource =  
SinhViens;
```





# CÁC TOÁN TỬ TRONG LINQ TO SQL **Group**

Nhóm dữ liệu (Tính tổng điểm của SV theo Môn học)

```
QLSVDataContext qlsv = new  
QLSVDataContext();  
var tongdiem = from diem in qlsv.Diems  
                group diem by  
                diem.MaMonHoc into kq  
select new  
{MaMonHoc=kq.Key,TongDiem=kq.Sum(t=>t.Diem1)  
};  
dataGridView1.DataSource = tongdiem;
```



# THÊM, XÓA, CẬP NHẬT DỮ LIỆU TRONG LINQ TO SQL

Thêm:

```
Khoa _khoa = new Khoa();  
    _khoa.MaKhoa = textBox1.Text;  
    _khoa.TenKhoa = textBox2.Text;  
    _qlsv.Khoas.InsertOnSubmit(_khoa);  
    _qlsv.SubmitChanges();
```



# THÊM, XÓA, CẬP NHẬT DỮ LIỆU TRONG LINQ TO SQL

Xóa:

```
Khoa _khoaXoa = _qlsv.Khoas.Where(kh => kh.MaKhoa ==  
dataGridView1.CurrentRow.Cells[0].Value.ToString()).FirstOrDefault();
```

```
if (_khoaXoa != null)  
{  
    _qlsv.Khoas.DeleteOnSubmit(_khoaXoa);  
    _qlsv.SubmitChanges();  
    MessageBox.Show("Thành công");  
}
```



# THÊM, XÓA, CẬP NHẬT DỮ LIỆU TRONG LINQ TO SQL

Xóa nhiều?



# THÊM, XÓA, CẬP NHẬT DỮ LIỆU TRONG LINQ TO SQL

Cập nhật:

```
Khoa _khoaSua = _qlsv.Khoas.Where(kh => kh.MaKhoa ==  
dataGridView1.CurrentRow.Cells[0].Value.ToString()).FirstOrDefault();
```

```
if (_khoaSua != null)  
{  
    _khoaSua.TenKhoa = textBox2.Text;  
    _qlsv.SubmitChanges();  
    MessageBox.Show("Thành công");  
}
```