

C# Programming

Ngôn ngữ truy vấn tích hợp - LINQ

NGUYỄN THỊ HỒNG ANH
BỘ MÔN KỸ THUẬT PHẦN MỀM

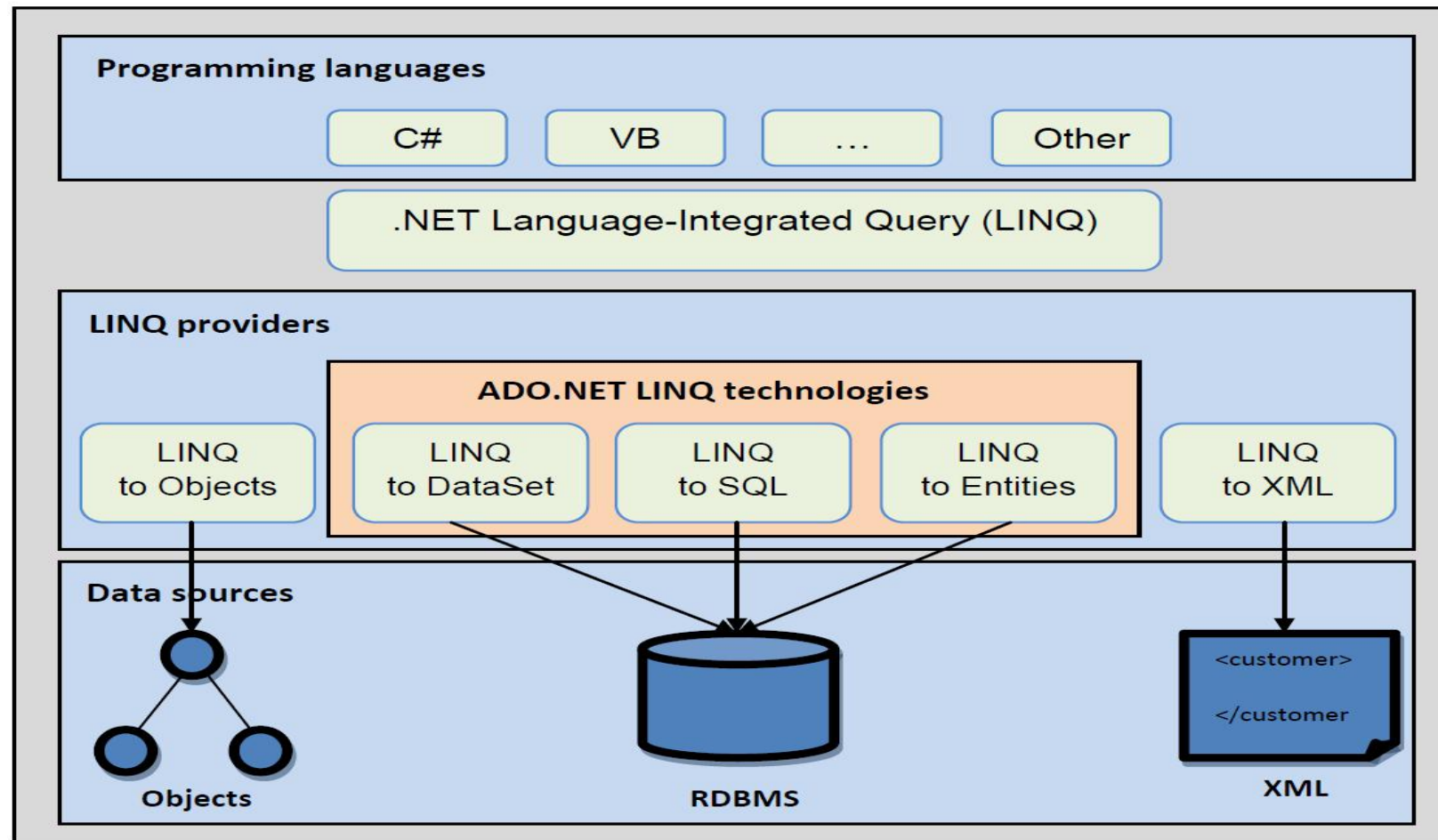
Nội dung

- Giới thiệu
- Lập trình LINQ to Objects
- Lập trình LINQ to Entities

Mục đích của LINQ

- LINQ được giới thiệu trong Visual Studio 2008 và .NET Framework 3.5 vào tháng 11 năm 2007.
- Khi LINQ chưa ra đời, các loại dữ liệu khác nhau phải truy vấn theo các phương pháp khác nhau:
 - Ví dụ cho mảng: `int[] a = {2,6,1,9,8,1,7,6,5,1,0,12,4,8};`
 - Tìm phần tử $a[i] > 4 \rightarrow$ xây dựng thuật toán duyệt các phần tử.
 - Tìm phần tử có giá trị lớn nhất \rightarrow xây dựng thuật toán duyệt phần tử.
 - Cơ sở dữ liệu quan hệ: Ngôn ngữ SQL.
 - Sử dụng lệnh SELECT
- LINQ được thiết kế cho mục đích:
Đơn giản hóa và thống nhất thực hiện truy vấn các loại dữ liệu khác nhau mà không cần quan tâm đến kiểu lưu trữ của dữ liệu.

Mô hình LINQ



- Tài liệu này đề cập đến `LINQ to Objects` và `LINQ to Entities`.

Các thư viện của LINQ

- `System.Linq`: Chứa các lớp và các giao diện cho truy vấn LINQ.
- `System.Data.Linq`: Chứa các lớp làm việc của LINQ to SQL.
- `System.Data.Entities`: Chứa các lớp làm việc của LINQ to Entities.
- `System.Xml.Linq`: Chứa các lớp làm việc của LINQ to XML.
- `System.Data.Objects`: Chứa các lớp làm việc của mô hình dữ liệu thực thể ADO.NET.

Các phương pháp truy vấn với LINQ

- Dạng biểu thức truy vấn (query expression)
- Dạng phương thức (method-based query)
- Dạng hỗn hợp (mixed query), nghĩa là kết hợp biểu thức và phương thức.

Lập trình LINQ to Objects

- LINQ to Objects được thiết kế để truy vấn các đối tượng trong bộ nhớ như mảng và danh sách.
- Nội dung:
 - Truy vấn các phần tử của mảng.
 - Truy vấn các phần tử của danh sách.

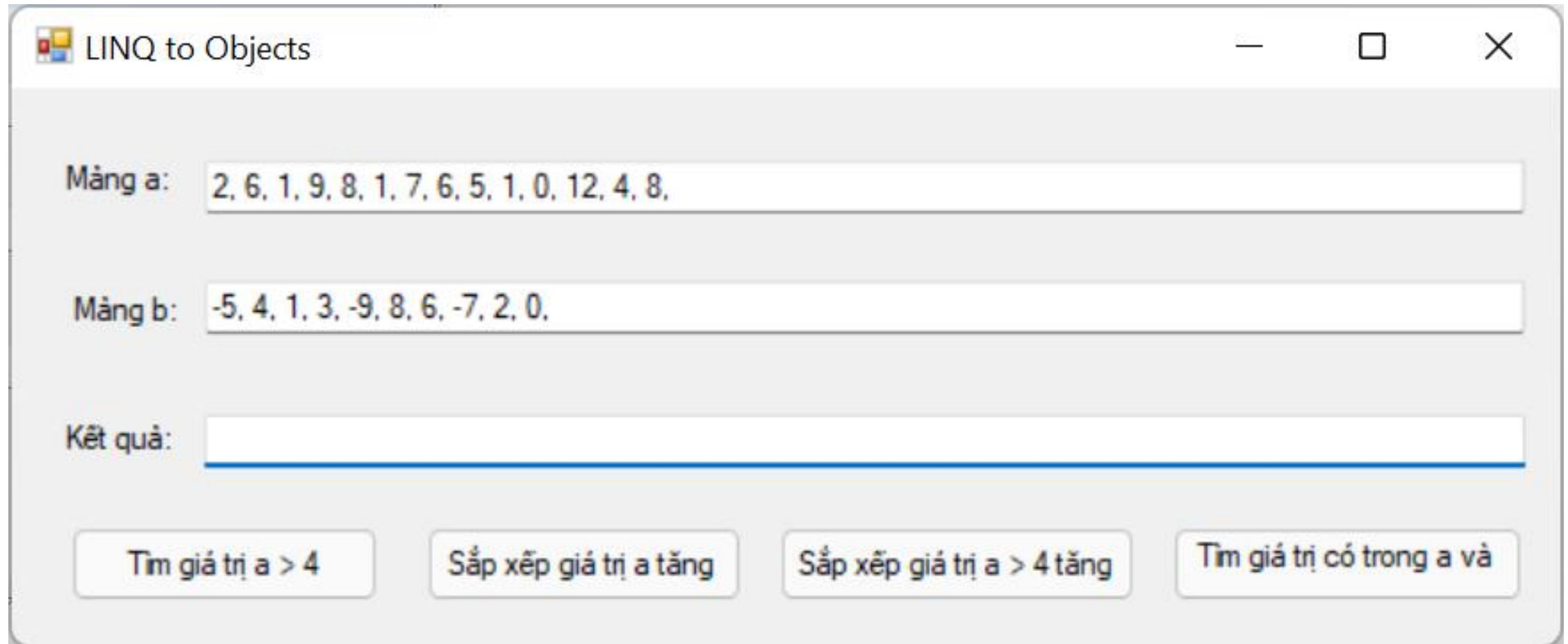
Dạng biểu thức truy vấn

```
query-expression = from itemName in srcExpr [join itemName in  
srcExpr on keyExpr equals keyExpr (into itemName)]  
[where predExpr]  
[orderby keyExpr (ascending | descending)]  
[select selExpr]  
[group selExpr by keyExpr]
```

- Mệnh đề **from**: Truy vấn dữ liệu từ dữ liệu nguồn.
- Mệnh đề **join**: Kết nối dữ liệu giữa các biểu thức chứa dữ liệu truy vấn.
- Mệnh đề **where**: Lọc dữ liệu ứng với biểu thức điều kiện có giá trị đúng.
- Mệnh đề **orderby**: Sắp xếp kết quả truy vấn tăng dần hoặc giảm dần.
- Mệnh đề **select**: Dùng để chọn biểu thức trong kết quả truy vấn.
- Mệnh đề **group**: Dùng để nhóm dữ liệu của biểu thức truy vấn.

Lập trình LINQ to Objects

- Ví dụ thiết kế Form:



The screenshot shows a Windows application window titled "LINQ to Objects". Inside the window, there are three text input fields and four buttons at the bottom. The first field, labeled "Mảng a:", contains the array "2, 6, 1, 9, 8, 1, 7, 6, 5, 1, 0, 12, 4, 8,". The second field, labeled "Mảng b:", contains the array "-5, 4, 1, 3, -9, 8, 6, -7, 2, 0,". The third field, labeled "Kết quả:", is empty. The four buttons at the bottom are: "Tìm giá trị a > 4", "Sắp xếp giá trị a tăng", "Sắp xếp giá trị a > 4 tăng", and "Tìm giá trị có trong a và".

LINQ to Objects

Mảng a: 2, 6, 1, 9, 8, 1, 7, 6, 5, 1, 0, 12, 4, 8,

Mảng b: -5, 4, 1, 3, -9, 8, 6, -7, 2, 0,

Kết quả:

Tìm giá trị a > 4 Sắp xếp giá trị a tăng Sắp xếp giá trị a > 4 tăng Tìm giá trị có trong a và

Truy vấn mảng dựa trên biểu thức

```
int[] a = {2,6,1,9,8,1,7,6,5,1,0,12,4,8};
```

```
int[] b = {-5,4,1,3,-9,8,6,-7,2,0};
```

- Truy vấn các phần tử lớn hơn 4 của mảng a:

```
var c = from i in a
        where i > 4
        select i;
```

- ☐ Từ khóa `var` tự xác định kiểu của biểu thức truy vấn.

- Hiển thị kết quả:

```
textBox3.Clear();
foreach (var i in c)
    textBox3.Text = textBox3.Text + i.ToString() + ", ";
```

- Truy vấn các phần tử của mảng a, kết quả sắp xếp tăng dần:

```
var c = from i in a
        orderby i
        select i;
```

Truy vấn mảng dựa trên biểu thức

```
int[] a = {2,6,1,9,8,1,7,6,5,1,0,12,4,8};
```

```
int[] b = {-5,4,1,3,-9,8,6,-7,2,0 };
```

- Truy vấn các phần tử của mảng a lớn 4 với kết quả sắp xếp giảm dần:

```
var c = from i in a
        where i > 4
        orderby i descending
        select i;
```

- Kết quả: c = {12,9,8,8,7,6,6,5};

- Truy vấn các phần tử của mảng a trong mảng b:

```
var c = from i in a
        join j in b on i equals j
        select i;
```

- Kết quả: c = {2,6,1,8,1,6,1,0,4,8};

Truy vấn mảng dựa trên phương thức

- Truy vấn dựa trên phương thức sử dụng biểu thức Lambda như là tham số.
- Dạng cơ bản của biểu thức Lambda:
$$(\text{input parameters}) \Rightarrow \text{expression} | \{\text{statement};\}$$
- Ví dụ: $x \Rightarrow x * x$; chỉ ra một tham số với tên x nhận giá trị là x^2 .

Truy vấn mảng dựa trên phương thức

- Truy vấn các phần tử lớn hơn 4 của mảng a:

```
var c = a.Where(i => i > 4)
        .Select(i => i);
```

□ Kết quả: c = {6,9,8,7,6,5,12,8};

- Truy vấn các phần tử của mảng a với kết quả sắp xếp tăng dần:

```
var c = a.OrderBy(i => i)
        .Select(i => i);
```

□ Kết quả: c = {0,1,1,1,2,4,5,6,6,7,8,8,9,12};

Truy vấn mảng dựa trên phương thức

- Truy vấn các phần tử của mảng a lớn 4 với kết quả sắp xếp giảm dần:

```
var c = a.Where(i => i > 4)
        .OrderByDescending(i => i)
        .Select(i => i);
```

- Kết quả: c = {12, 9, 8, 8, 7, 6, 6, 5};

- Truy vấn các phần tử của mảng a trong mảng b:

```
var c = a.Join(b, i=>i, j => j, (i,j) =>j)
        .Select (i=>i);
```

- Kết quả: c = {2, 6, 1, 8, 1, 6, 1, 0, 4, 8}

Phương thức và mệnh đề truy vấn

Phương thức	Mệnh đề truy vấn
GroupBy()	group...by
Join()	join...in
OrderBy()	orderby
OrderByDescending()	orderby...descending
Select()	select
ThenBy()	orderby...,...
ThenByDescending()	orderby...,...descending
Where()	where

Một số phương thức phổ biến trong Linq

- Count(): đếm số phần tử của danh sách
- Average(): tính trung bình
- Max(): lấy ra giá trị lớn nhất
- Min(): lấy ra giá trị nhỏ nhất
- Sum(): tính tổng các phần tử
- First(): lấy ra phần tử đầu tiên
- Last(): lấy ra phần tử cuối cùng
- ToArray(): chuyển danh sách thành array. (ví dụ từ kiểu List sang Array)
- ToList(): chuyển danh sách thành List. (ví dụ từ kiểu Array sang List)

Truy vấn các phần tử của danh sách

- Lớp List dùng để lưu trữ các phần tử của một danh sách.
- Khai báo: `List <T> <variable>;`
trong đó `<T>` là kiểu dữ liệu và `<variable>` là một biến có kiểu danh sách.
- Một số phương thức:
 - `Add()`: Tạo một phần tử cuối danh sách.
 - `Clear()`: Xóa tất cả các phần tử của danh sách.
 - `Insert()`: Chèn phần tử ở vị trí chỉ ra.
 - `Remove()`: Xóa phần tử đầu tiên của giá trị chỉ ra.
 - `RemoveAt()`: Xóa phần tử ở vị trí chỉ ra trong danh sách.
 - `Sort()`: Sắp xếp danh sách.

Truy vấn các phần tử của danh sách

```
List <string> myList = new List <string>();  
myList.Add("One");  
myList.Add("Two");  
myList.Add("Three");  
myList.Add("Four");  
myList.Add("Five");  
...  
myList.Add("Nine");  
myList.Add("Ten");  
  
var queryList = from item in myList  
                 where item.Contains("e")  
                 orderby item  
                 select item;
```

Bài tập

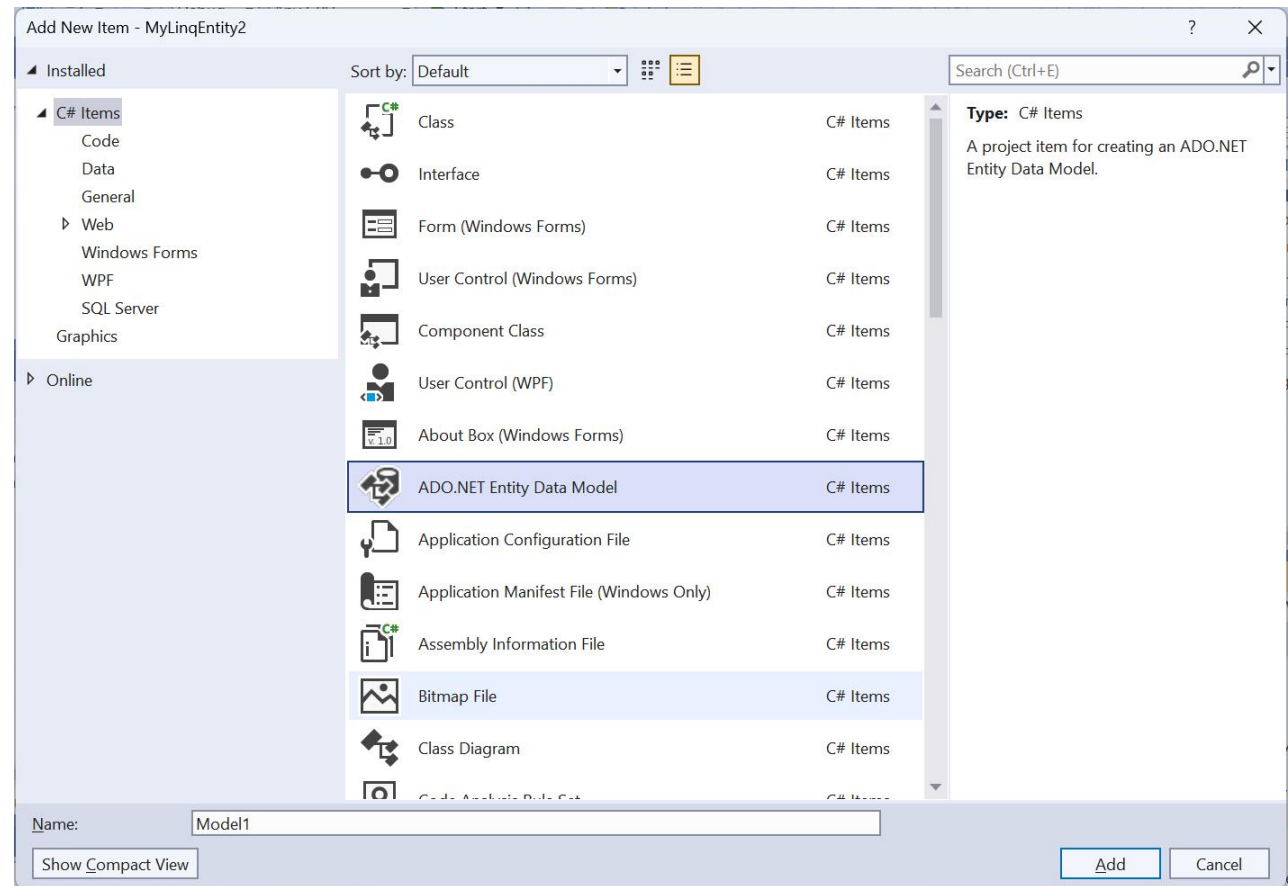
- Cho List : gồm các Student, mỗi sinh viên có các thuộc tính id, name, mark, address. Hãy viết các câu lệnh truy vấn dựa trên Linq thực hiện các công việc sau:
 - ❑ Lấy ra học sinh có id = 4
 - ❑ Lấy ra danh sách học sinh có tên chứa 'an' và sống ở thành phố 'HCM'
 - ❑ Lấy ra tên học sinh có điểm cao nhất

LINQ to Entities

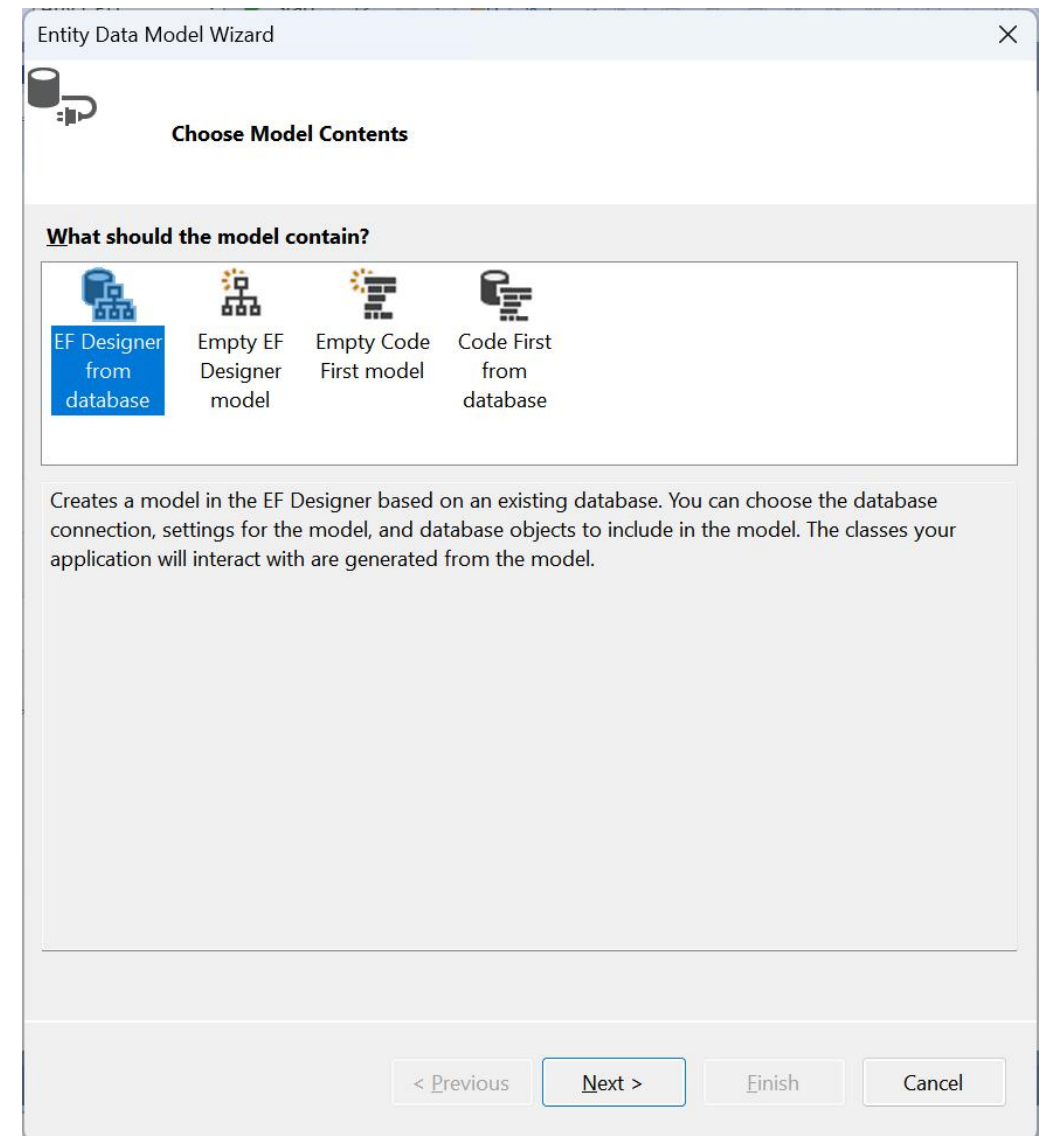
- **LINQ to Entities** được thiết kế để cung cấp một giao diện làm việc với ADO.NET Entity Framework.
- **LINQ to Entities** làm việc với cơ sở dữ liệu như thực hiện các truy vấn, cập nhật và xóa dữ liệu thông qua một mô hình dữ liệu thực thể (EDM-entity data model).
- **LINQ to Entities** được thiết kế để làm việc với mọi cơ sở dữ liệu quan hệ.

Xây dựng mô hình thực thể ADO.NET

- Tạo tệp mô hình dữ liệu thực thể ADO.NET: Chọn menu Project, chọn Add > New Item.
- Chọn ADO.NET Entity Data Model, nhập tên tệp (ví dụ BooksEntities.edmx) và chọn Add.



- Chọn kiểu tạo mô hình dữ liệu thực thể: Trong cửa sổ Entity Data Model Wizard, chọn EF Designer from database và chọn Next.



- Thiết lập kết nối tới cơ sở dữ liệu:
Chọn New Connection, xuất hiện cửa sổ Connection Properties.

The screenshot shows the 'Connection Properties' dialog box. It has a title bar with a question mark and a close button. The main text says: 'Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.'

The 'Data source:' section has a dropdown menu showing 'Microsoft SQL Server (SqlClient)' and a 'Change...' button.

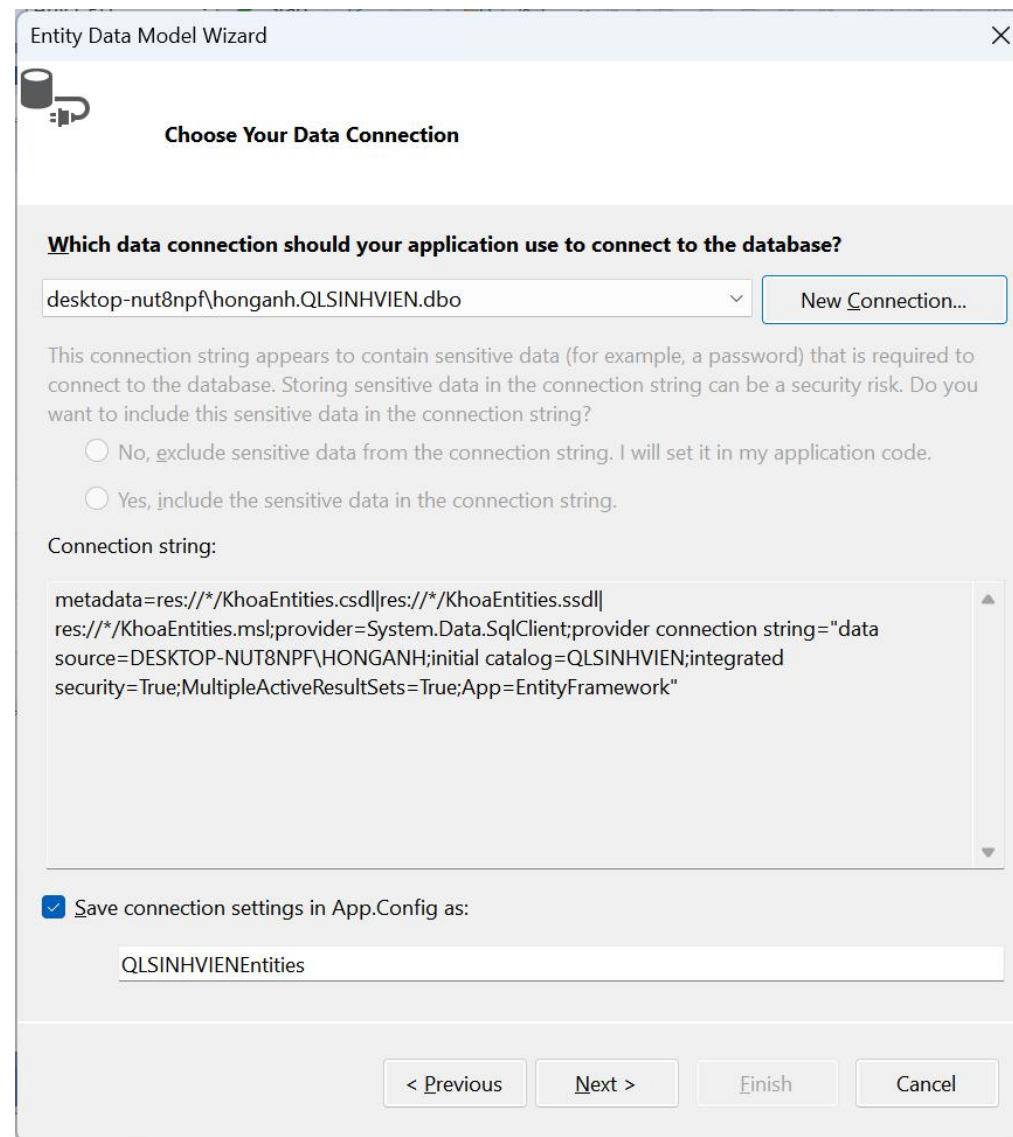
The 'Server name:' section has a text box and a 'Refresh' button.

The 'Log on to the server' section has an 'Authentication:' dropdown menu set to 'Windows Authentication'. Below it are 'User name:' and 'Password:' text boxes, and a 'Save my password' checkbox.

The 'Connect to a database' section has two radio buttons. The first is 'Select or enter a database name:' with a dropdown menu. The second is 'Attach a database file:' with a 'Browse...' button and a 'Logical name:' text box.

At the bottom right is an 'Advanced...' button. At the bottom are three buttons: 'Test Connection', 'OK', and 'Cancel'.

- Chọn Server name >
Authentication >
Database name > chọn Ok.



The image shows a screenshot of the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. The dialog has a title bar with the text 'Entity Data Model Wizard' and a close button. Below the title bar is a header area with a database icon and the text 'Choose Your Data Connection'. The main content area contains the question 'Which data connection should your application use to connect to the database?'. Below this is a dropdown menu showing 'desktop-nut8npf\honganh.QLSINHVIEN.dbo' and a 'New Connection...' button. A warning message follows: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (which is selected) and 'Yes, include the sensitive data in the connection string.'. Below this is a 'Connection string:' label and a text area containing the connection string: 'metadata=res://*/KhoaEntities.csdl|res://*/KhoaEntities.ssdl|res://*/KhoaEntities.msl;provider=System.Data.SqlClient;provider connection string="data source=DESKTOP-NUT8NPF\HONGANH;initial catalog=QLSINHVIEN;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"'. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked, and a text box containing 'QLSINHVIENEntities'. The bottom of the dialog has four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

desktop-nut8npf\honganh.QLSINHVIEN.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

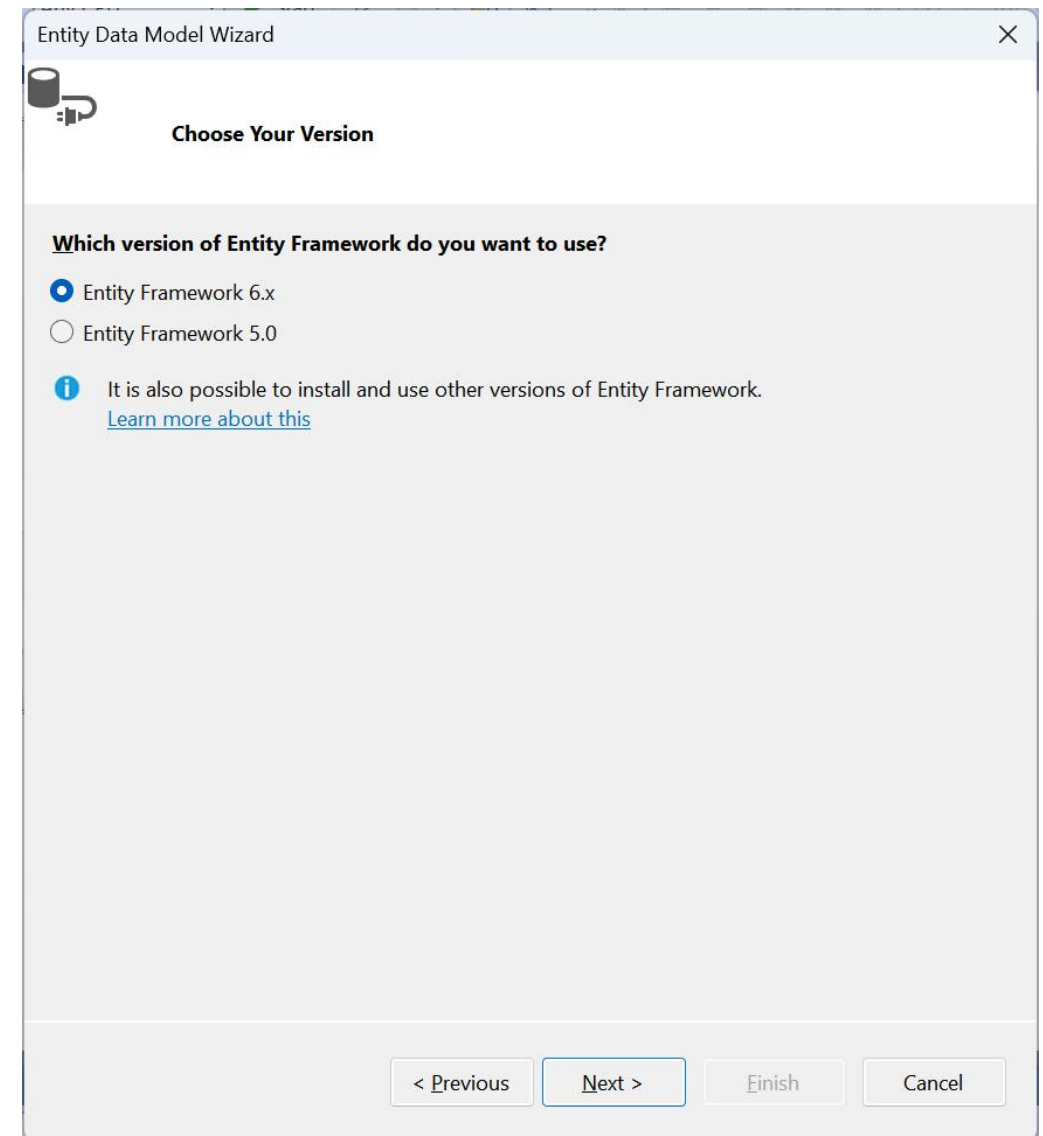
metadata=res://*/KhoaEntities.csdl|res://*/KhoaEntities.ssdl|
res://*/KhoaEntities.msl;provider=System.Data.SqlClient;provider connection string="data
source=DESKTOP-NUT8NPF\HONGANH;initial catalog=QLSINHVIEN;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in App.Config as:

QLSINHVIENEntities

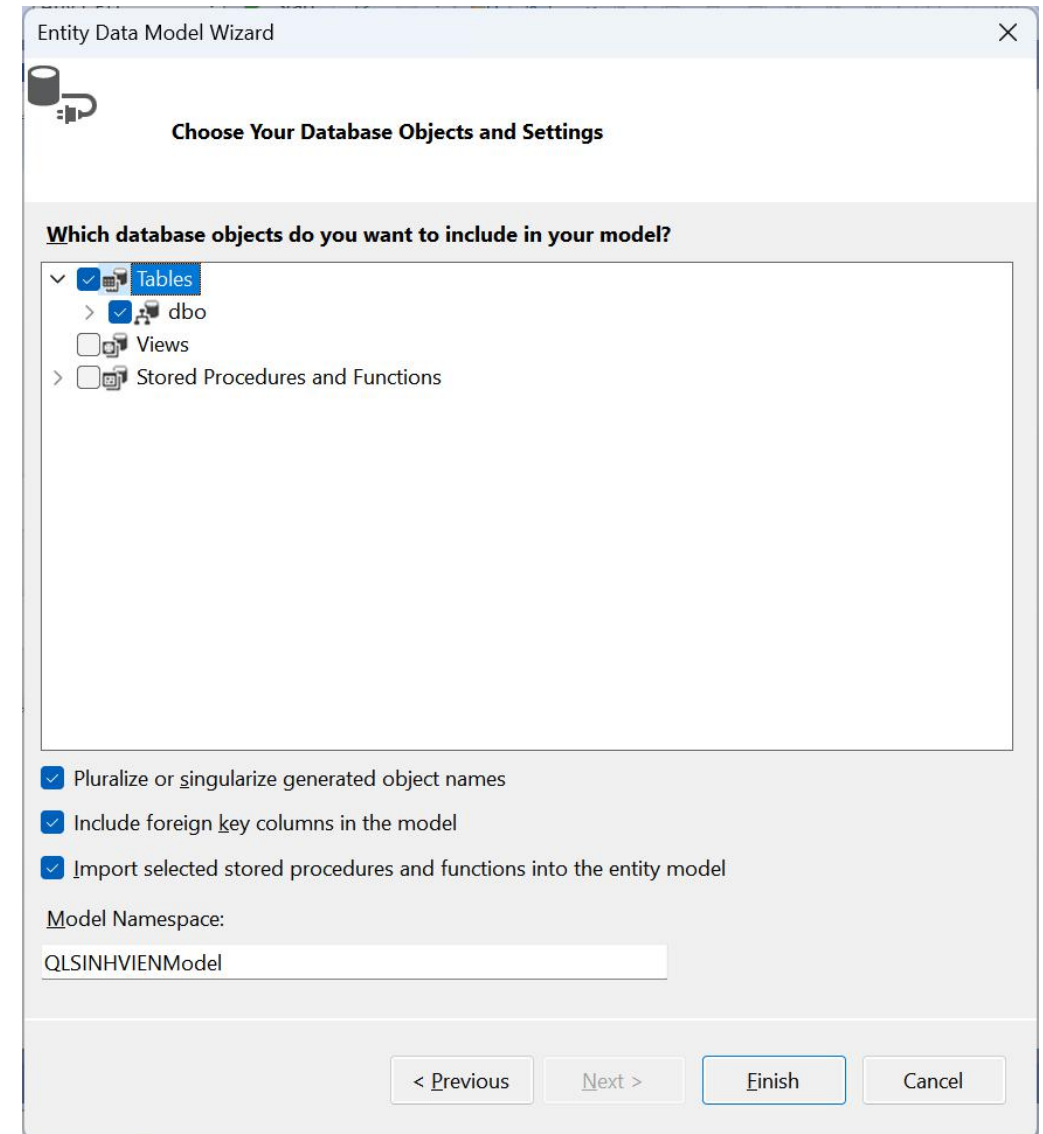
< Previous Next > Finish Cancel

- Chọn kiểu lưu trữ xâu kết nối cơ sở dữ liệu: Chọn Yes, include the sensitive data và chọn Next.
- Nhập tên mô hình thực thể: Chọn Save entity connection settings, nhập tên mô hình thực thể (ví dụ BookEntities) và chọn Next.
- ❑ Chọn Entity Framework 6.x/
Entity Framework 5.x.



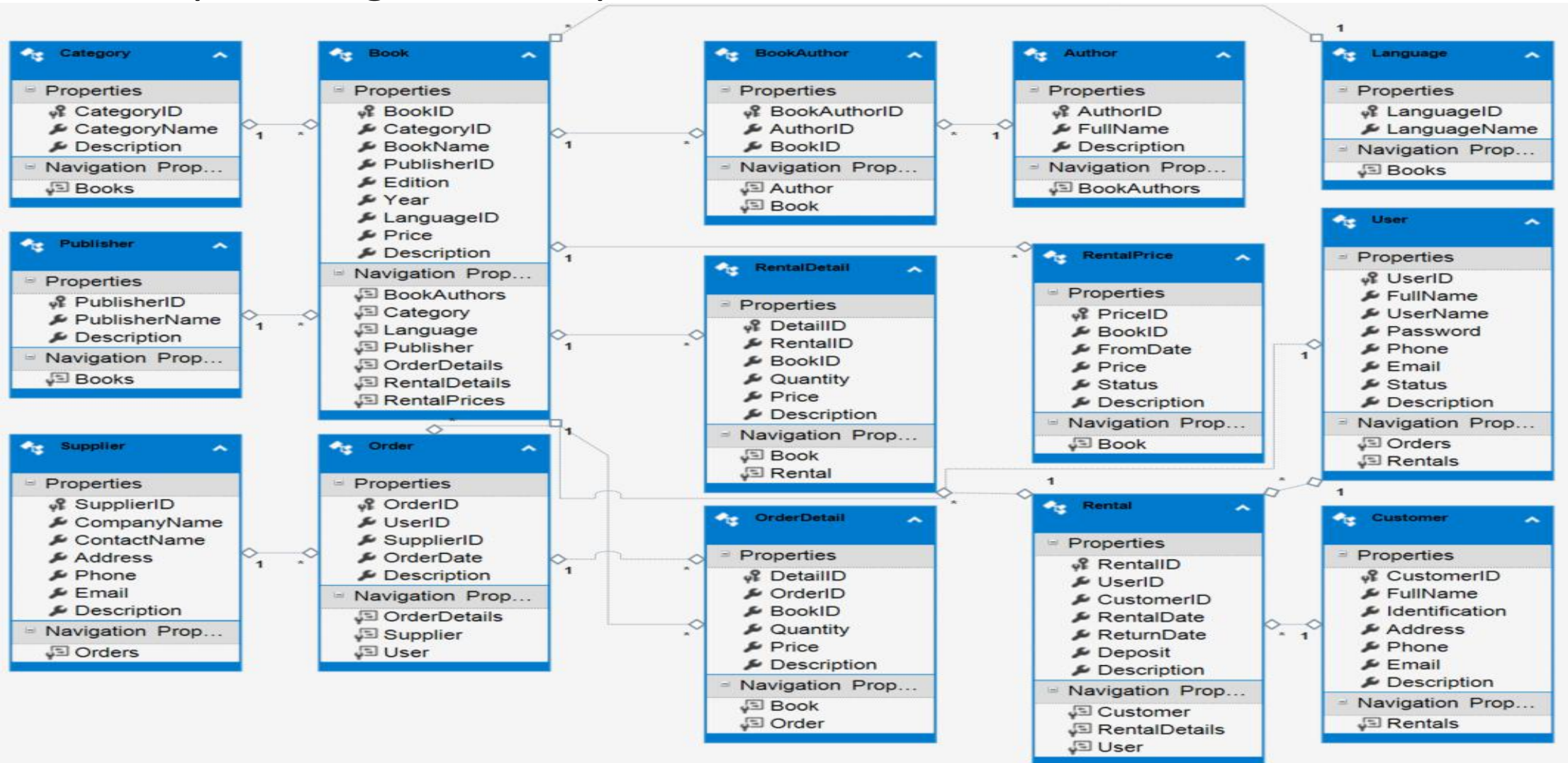
Xây dựng mô hình thực thể ADO.NET

- **Chọn các đối tượng của cơ sở dữ liệu:** Chọn Tables, chọn các bảng dữ liệu, nhập không gian tên chứa mô hình thực thể (ví dụ BooksModel) và chọn Finish.
- **Biên dịch mô hình:** Chọn thực đơn Build > Build Solution.



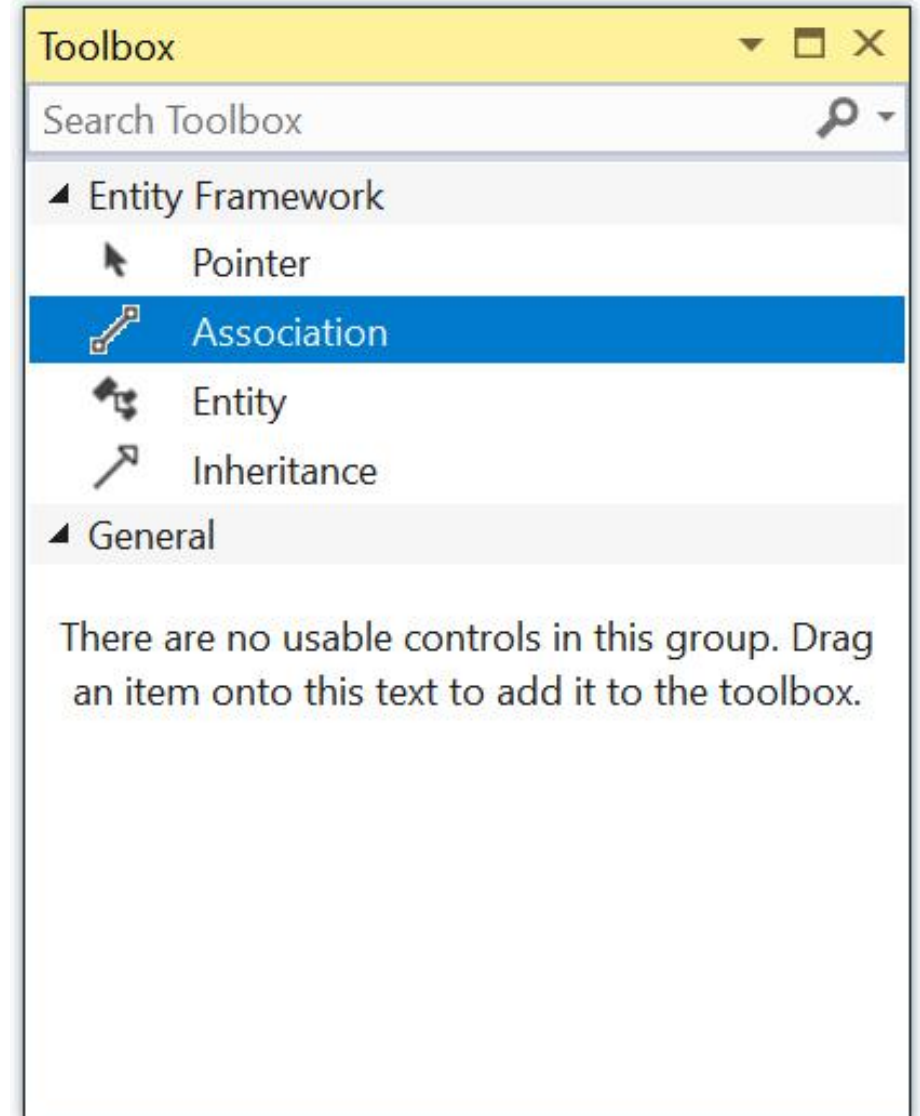
Xây dựng mô hình thực thể ADO.NET

- Mô hình quan hệ giữa các lớp thực thể:



Xây dựng mô hình thực thể ADO.NET

- Sửa quan hệ các lớp trong mô hình thực thể:
 - ❑ Chọn cửa sổ Toolbox, chọn Association.
 - ❑ Tạo quan hệ giữa các lớp: kéo thuộc tính khóa của lớp thực thể nối với nhau.
 - ❑ Sửa quan hệ giữa các lớp: chọn đường liên kết các thực thể và chọn cửa sổ Properties.
 - ❑ Xóa quan hệ giữa các lớp: chọn đường liên kết và ấn phím delete.



Xây dựng ứng dụng với LINQ to Entities

■ Xây dựng Form:

The screenshot shows a Windows Forms application window titled "frmPublishers (LINQ to Entities)". The window contains the following elements:

- Text Input:** A label "Tên nhà xuất bản (*):" followed by a text box.
- Description Input:** A label "Mô tả thêm:" followed by a larger text box.
- Action Buttons:** A grid of six buttons: "Thêm mới", "Sửa", "Xóa", "Ghi", "Hủy bỏ", and "Kết thúc".
- List View:** A panel titled "Danh sách nhà xuất bản" containing a large, empty gray rectangular area, likely intended for a list of publishers.

Truy vấn dữ liệu với LINQ to Entities

- Tạo đối tượng lớp mô hình dữ liệu thực thể ADO.NET, ví dụ tạo đối tượng lớp BooksEntities:

```
BooksEntities myBooksEntities = new BooksEntities();
```

- Định nghĩa và thực hiện câu lệnh truy vấn dữ liệu.

- Ví dụ truy vấn tất cả các nhà xuất bản:

```
var queryPublishers = from item in myBooksEntities.Publishers  
                        orderby item.PublisherName  
                        select item;
```

- Hiển thị lên dataGridView1:

```
dataGridView1.DataSource = queryPublishers.ToList();
```

Nhập dữ liệu với LINQ to Entities

- Tạo đối tượng lớp thực thể:

```
Publisher itemPublisher = new Publisher();
```

- Gán giá trị cho đối tượng lớp thực thể:

```
itemPublisher.PublisherName = txtPublisherName.Text;
```

```
itemPublisher.Description = txtDescription.Text;
```

- Thêm `itemPublisher` vào `Publisher`:

```
myBooksEntities.Publishers.Add(itemPublisher);
```

- Cập nhật lại bảng dữ liệu trong CSDL:

```
myBooksEntities.SaveChanges();
```


Sửa dữ liệu với LINQ to Entities

- Tìm bản ghi cần sửa, ví dụ tìm theo PublisherID:

```
var queryPublishers = from item in myBooksEntities.Publishers
                       where (item.PublisherID == PublisherID)
                       select item;
```

- Lấy đối tượng đầu tiên trong bảng dữ liệu truy vấn, ví dụ:

```
Publisher itemPublisher = queryPublishers.First();
```

- Gán giá trị mới cho đối tượng lớp thực thể, ví dụ:

```
itemPublisher.PublisherName = txtPublisherName.Text;
itemPublisher.Description = txtDescription.Text;
```

- Cập nhật lại bảng dữ liệu trong CSDL:

```
myBooksEntities.SaveChanges();
```


Xóa dữ liệu với LINQ to Entities

- Lấy mã PublisherID của dòng được chọn:

```
int r = dataGridView1.CurrentRow.Index;  
int PublisherID = Convert.ToInt32(dataGridView1.Rows[r].Cells[0].Value.ToString());
```

- Tìm bản ghi cần xóa:

```
var queryPublishers = from item in myBooksEntities.Publishers  
                       where (item.PublisherID == PublisherID)  
                       select item;
```

- Xóa đối tượng lớp thực thể trong lớp thực thể:

```
myBooksEntities.Publishers.Remove(queryPublishers.First());
```

- Cập nhật lại bảng trong CSDL:

```
myBooksEntities.SaveChanges();
```

Bài tập

Câu 1: Tạo cơ sở dữ liệu QLThietbi (quản lý thiết bị) trong hệ quản trị cơ sở dữ liệu SQL Server. Cơ sở dữ liệu gồm các bảng:

a) Bảng Donvi để lưu các đơn vị sử dụng các thiết bị. Bảng gồm các trường:

- Madv int (mã đơn vị - trường khóa đánh mã tự động)
- Tendv nvarchar(35) (tên đơn vị, ví dụ Khoa CNTT, Khoa Toán, Khoa Lý)

 Nhập 3 bản ghi cho bảng dữ liệu.

b) Bảng Loathietbi để lưu các loại thiết bị. Bảng gồm các trường:

- Maloi int (mã loại thiết bị - trường khóa đánh mã tự động)
- Tenloai nvarchar(35) (tên loại thiết bị, ví dụ máy in, máy tính, ...)
- Ghichu nvarchar(50) (ghi chú)

c) Bảng Thietbi để lưu các thiết bị của đơn vị sử dụng. Bảng gồm các trường:

- Matb: int (mã thiết bị - trường khóa đánh mã tự động)
- Madv: int (mã đơn vị)
- Maloi: int (mã loại thiết bị)
 - Tentbnvarchar(35)(tên thiết bị)
 - Nuocsxnvarchar(30)(nước sản xuất thiết bị)
 - Namsxint(năm sản xuất thiết bị)
 - Ngaysdnvarchar(10)(ngày/tháng/năm sử dụng thiết bị)
 - Ghichunvarchar(50)(ghi chú)

Bài tập

- Câu 2 : Sử dụng kỹ thuật LINQ to Entity xây dựng các Forms sau:
- a) Nhập, sửa, xóa dữ liệu cho bảng Loathietbi.
- b) Nhập dữ liệu cho bảng Thietbi.
- c) Chọn tên đơn vị và hiển thị thông tin về thiết bị gồm: tên loại thiết bị, tên thiết bị, nước sản xuất, năm sản xuất và ngày sử dụng.
- d) Tìm kiếm thiết bị theo tên thiết bị, sửa, xóa dữ liệu trong bảng Thietbi.