

BÀI 4: LẬP TRÌNH VỚI CƠ SỞ DỮ LIỆU SỬ DỤNG ENTITY

4.1 MỤC TIÊU

- Hướng dẫn sinh viên làm quen với việc xây dựng ứng dụng Windows Application có kết nối với CSDL SQL Server bằng Entity Framework của .NET.
- Sử dụng mô hình *Code First* trong EntityFramework với hướng tiếp cận đã có sẵn CSDL
- Thiết kế các Form nhập liệu cho các bảng trong cơ sở dữ liệu (hiện thị, thêm, xóa, sửa)

4.2 BÀI TẬP

Sử dụng SQL Server tạo cơ sở dữ liệu "QuanLySinhVien" đơn giản với 2 bảng: Sinh viên và Khoa như sau

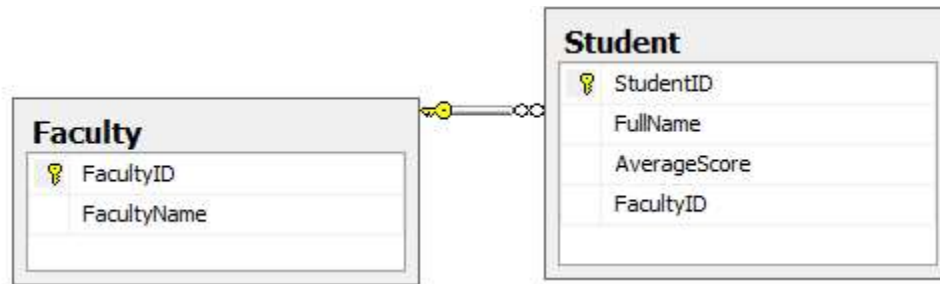
Student (**StudentID**, FullName, AverageScore, FacultyID)

Faculty(**FacultyID**, FacultyName)

DESKTOP-AA3CGL7...n - dbo.Student			
Column Name	Data Type	Allow Nulls	
StudentID	nvarchar(20)	<input type="checkbox"/>	
FullName	nvarchar(200)	<input type="checkbox"/>	
AverageScore	float	<input type="checkbox"/>	
FacultyID	int	<input type="checkbox"/>	

DESKTOP-AA3CGL7...en - dbo.Faculty			
Column Name	Data Type	Allow Nulls	
FacultyID	int	<input type="checkbox"/>	
FacultyName	nvarchar(200)	<input type="checkbox"/>	

- ✓ Tạo mối quan hệ 2 bảng như sau:



- ✓ Nhập liệu sẵn vào cơ sở dữ liệu một số dòng

DESKTOP-AA3CGL7....en - dbo.Faculty		DESKTOP-AA3CGL7....n - dbo.Student			
FacultyID	FacultyName	StudentID	FullName	AverageScore	FacultyID
1	Công Nghệ Thông Tin	1611061916	Nguyễn Trần Hoàng Lan	4.5	1
2	Ngôn Ngữ Anh	1711060596	Đàm Minh Đức	2.5	1
3	Quản trị kinh doanh	1711061004	Nguyễn Quốc An	10	2

Bài Tập 1 Sử dụng EntityFrameWork với mô hình Code First để kết nối CSDL

- ✓ Viết chương trình quản lý sinh viên có giao diện tương tự sau đây

Quản Lý Sinh Viên

Quản lý thông tin Sinh viên

Thông Tin Sinh Viên

Mã Số SV:

Họ Tên:

Khoa:

Điểm TB:

Mã Số SV	Họ Tên	Tên Khoa	Điểm TB
1611061916	Nguyễn Trần Hoàng Lan	Công Nghệ Thông Tin	4.5
1711060596	Đàm Minh Đức	Công Nghệ Thông Tin	2.5
▶ 1711061004	Nguyễn Quốc An	Ngôn Ngữ Anh	10

Yêu Cầu Xử Lý

1.1 Sự kiện Form_load:

- Hiển thị danh sách sinh viên hiện có trong CSDL (Lấy từ bảng sinh viên)
- ComboBox Khoa lấy từ bảng **Faculty** và hiển thị tên khoa

1.2 Khi nhấn vào nút "Thêm" Hoặc "Sửa"

- Kiểm tra các thông tin bắt buộc phải nhập liệu cho sinh viên như mã sinh viên, tên, và điểm trung bình. Nếu để trống sẽ xuất hiện thông báo lỗi "**Vui lòng nhập đầy đủ thông tin!**"

- Kiểm tra mã số sinh viên phải có 10 kí tự. Nếu không sẽ xuất thông báo "Mã số sinh viên phải có 10 kí tự!"
- Nếu trường hợp nhấn vào nút "Thêm" thì Thêm mới dữ liệu sinh viên vừa nhập vào CSDL, Load lại DataGridView, và thông báo "**Thêm mới dữ liệu thành công!**".
- Nếu trường hợp nhấn vào nút "Sửa". Nếu mã sinh viên đã tồn tại thì Cập nhật dữ liệu sinh viên vào CSDL, và thông báo "**Cập nhật dữ liệu thành công!**". Nếu mã sinh viên đó không tồn tại thì xuất thông báo "**Không tìm thấy MSSV cần sửa!**"
- Reset lại dữ liệu về giá trị ban đầu sau khi thêm/ sửa thành công

1.3 Khi nhấn vào nút "Xóa"

- Kiểm tra nếu MSSV cần xóa không tồn tại trong CSDL thì thông báo lỗi "**Không tìm thấy MSSV cần xóa!**".
- Ngược lại thì xuất hiện cảnh báo YES/NO. Nhấn YES sẽ thực hiện xóa dòng dữ liệu sinh viên trong DataGridView và thông báo "**Xóa sinh viên thành công!**".
- Reset lại dữ liệu về giá trị ban đầu sau khi xóa thành công

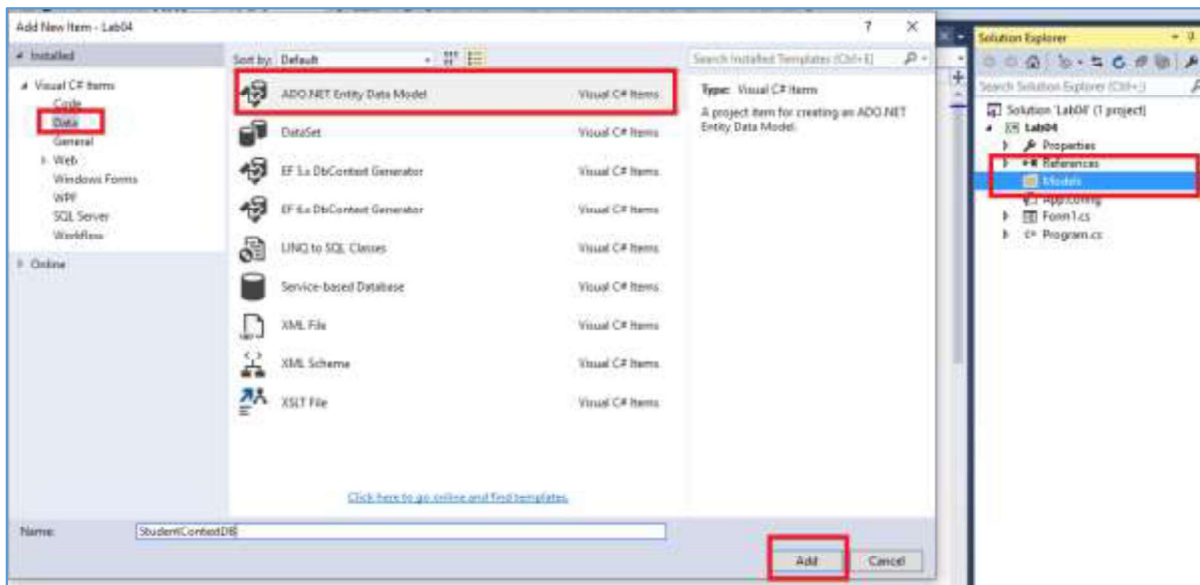
1.4 Viết code cho sự kiện ở DataGridView, người dùng chọn 1 dòng thì thể hiện ngược lại thông tin của các sinh viên đã chọn ở phần nhập liệu (bên trái).

Hướng Dẫn

Bước 1: Entity Framework sinh ra các class chúng ta nên tạo trong 1 thư mục (**Models**) để dễ dàng quản lý.

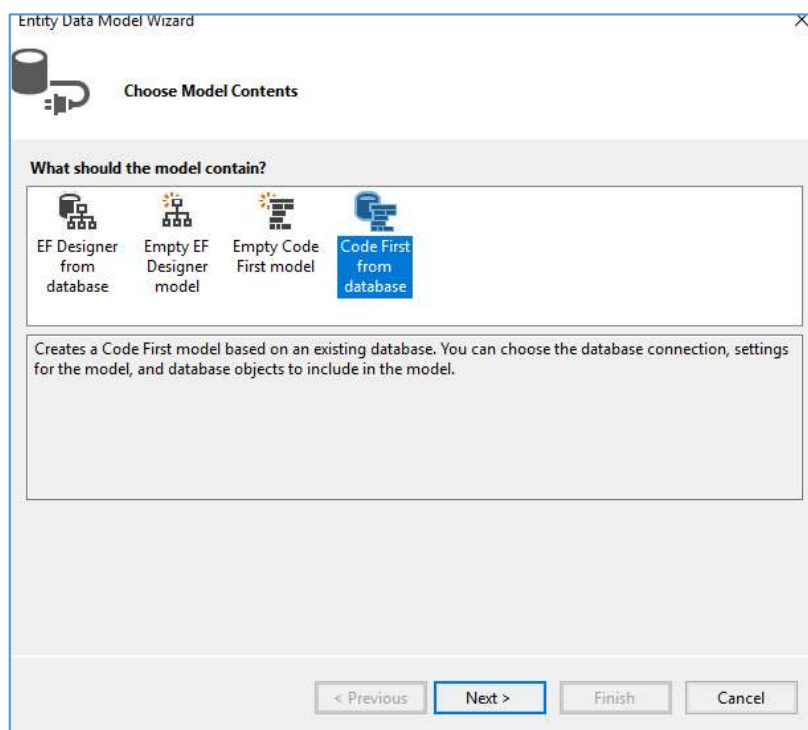
Click chuột phải vào Models chọn **New Item**. Chọn Loại **Data/ ADO.NET Entity Data Model**

Đặt tên context là "**StudentContextDB**" (mặc định là Model1). Và chọn **Add**

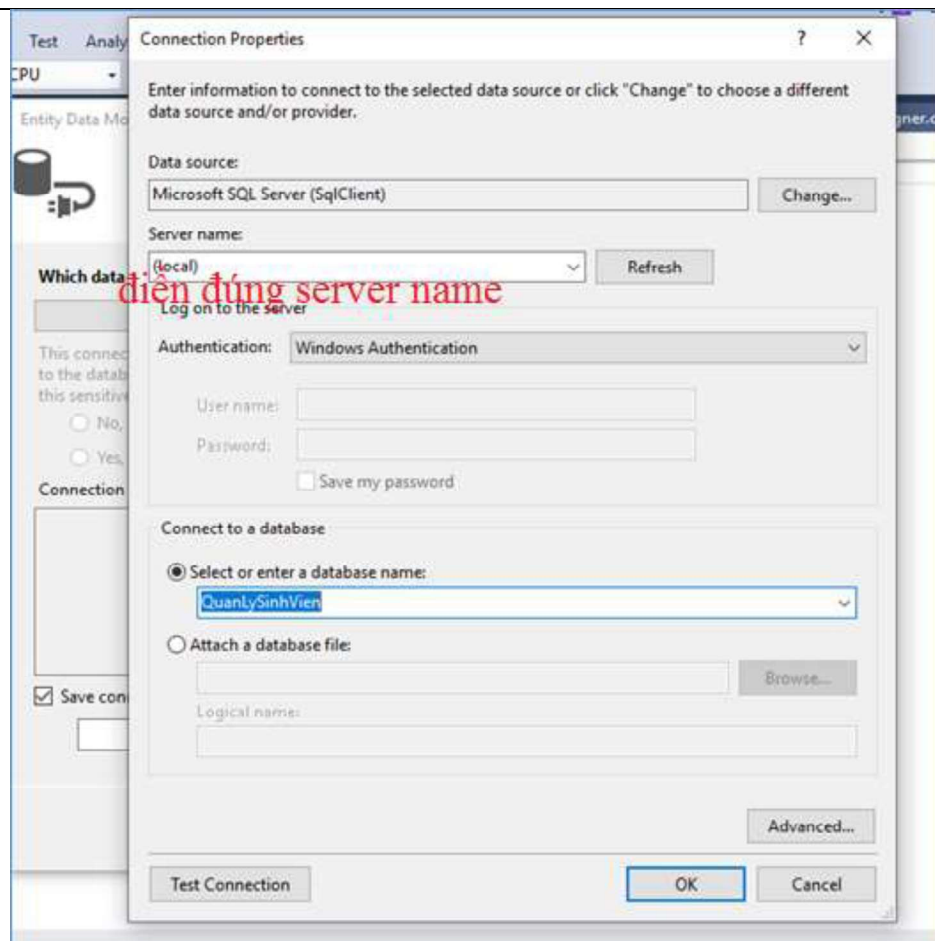


Sau khi chọn Add có 4 hướng loại Entity model để kết nối với cơ sở dữ liệu

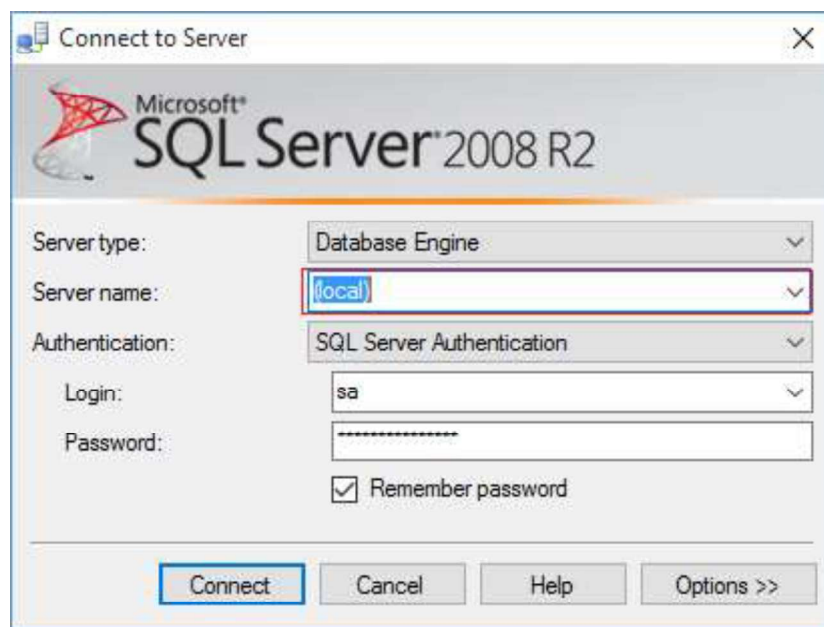
- Database First
- Model First
- Code First với hướng tiếp cận tạo ra cơ sở dữ liệu
- Code First với hướng tiếp cận đã có sẵn cơ sở dữ liệu



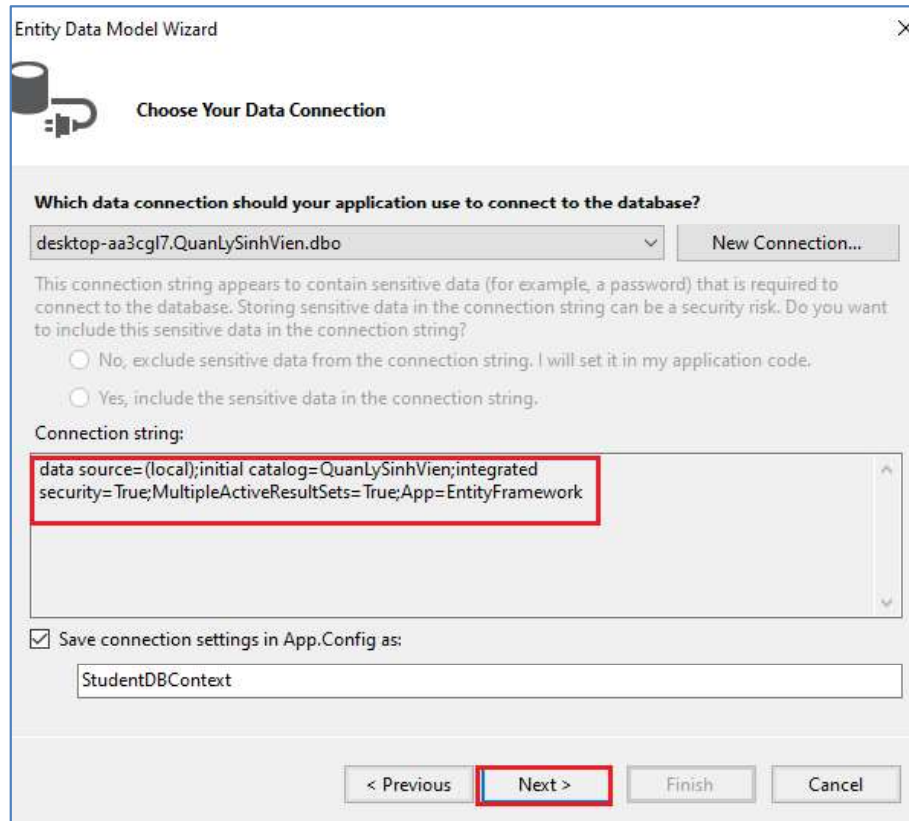
Ta chọn loại model là "**Code first from database**". Chọn **Next**
Tìm đúng cơ sở dữ liệu Student ở SQL để trở database name vào



Chọn đúng tên server Name (trên từng máy có thể khác nhau - sinh viên có thể re-connect lại database để xem đúng tên Server Name. Ở ví dụ đây là trên local máy cá nhân)

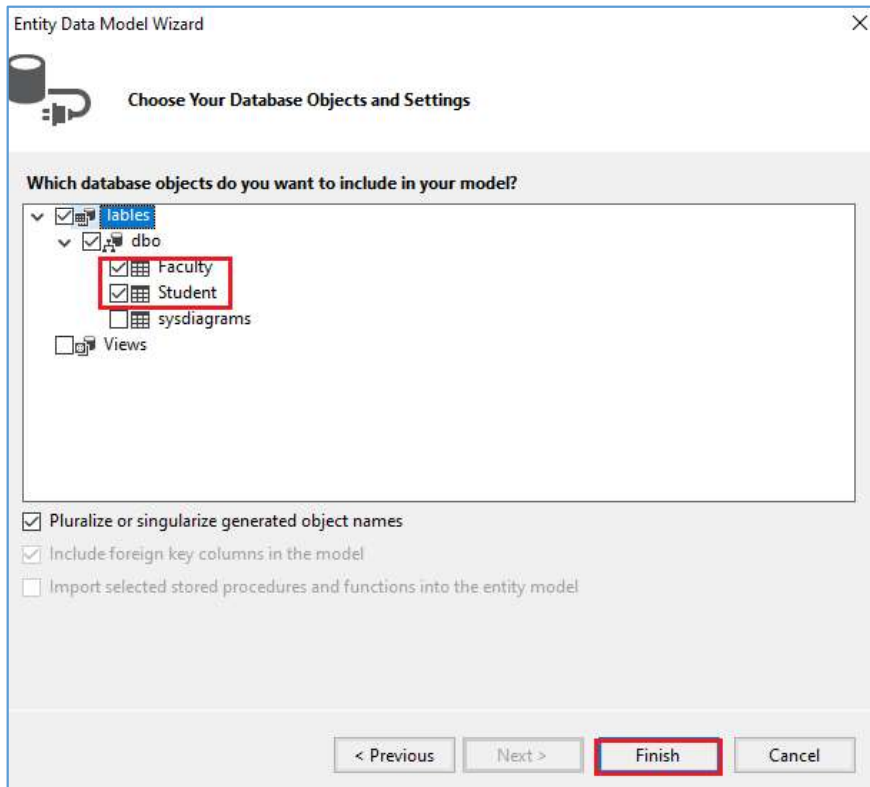


Chọn Next để tiếp tục tạo. Connect String sẽ được lưu ở file App.Config



The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The title bar reads 'Entity Data Model Wizard'. The main heading is 'Choose Your Data Connection'. Below this, a question asks: 'Which data connection should your application use to connect to the database?'. A dropdown menu shows 'desktop-aa3cgl7.QuanLySinhVien.dbo', and a 'New Connection...' button is to its right. A warning message states: '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?'. Two radio buttons are present: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (selected) and 'Yes, include the sensitive data in the connection string.'. Below this, the 'Connection string:' field contains the text: 'data source=(local);initial catalog=QuanLySinhVien;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework'. This text is highlighted with a red rectangle. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked, and a text box containing 'StudentDBContext'. At the very bottom, there are four buttons: '< Previous', 'Next >' (highlighted with a red rectangle), 'Finish', and 'Cancel'.

Sau đó chọn các bảng muốn tạo object (ở đây chọn 2 bảng từ CSDL)



The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Database Objects and Settings' step. The title bar reads 'Entity Data Model Wizard'. The main heading is 'Choose Your Database Objects and Settings'. Below this, a question asks: 'Which database objects do you want to include in your model?'. A tree view shows the following structure: 'tables' (checked), 'dbo' (checked), 'Faculty' (checked), 'Student' (checked), 'sysdiagrams' (unchecked), and 'Views' (unchecked). The 'Faculty' and 'Student' checkboxes are highlighted with a red rectangle. Below the tree view, there are three checkboxes: 'Pluralize or singularize generated object names' (checked), 'Include foreign key columns in the model' (checked), and 'Import selected stored procedures and functions into the entity model' (unchecked). At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish' (highlighted with a red rectangle), and 'Cancel'.

Sau khi Finish, Entity FrameWork đã tạo các class tương ứng như trong cơ sở dữ liệu



Một số thông tin cần lưu ý:

- App.Config: Sẽ lưu trữ connection String và tên của Context sử dụng
- StudentContextDB: Lớp chứa tập hợp DataSet cho các table được chọn.

VS.Net tự động sinh ra các file cs tương ứng map với CSDL. Sinh viên kiểm tra từng file (Student, Faculty) để hiểu cách mapping tương ứng với CSDL.

- ✓ Cách Sử dụng Entity để thao tác với cơ sở dữ liệu như: Lấy tất cả, thêm, xóa, sửa với CSDL (Sinh viên đọc kĩ và thử để thực hiện bài tập)

```
//luôn luôn sử dụng context để làm việc với các class
StudentContextDB context = new StudentContextDB();

//1. lấy tất cả các sinh viên từ bảng Student
List<Student> listStudent = context.Students.ToList();
//2. lấy sinh viên đầu tiên có StudentID = ID cho trước
Student db = context.Students.FirstOrDefault(p => p.StudentID == ID);
//3. insert 1 đối tượng sinh viên s vào database
Student s = new Student() { StudentID = "99", FullName = "test insert", AverageScore = 100 };
context.Students.Add(s);
context.SaveChanges();
//4. Update sinh viên -> lấy item ra và cần update thuộc tính nào thì set thuộc tính đó
Student dbUpdate = context.Students.FirstOrDefault(p => p.StudentID == ID);
if(dbUpdate!= null){
    dbUpdate.FullName = "Update FullName"; //....
    context.SaveChanges(); //lưu thay đổi
```

```

    }

    //5. Xóa Student có ID cho trước , tương tự update
    Student dbDelete = context.Students.FirstOrDefault(p => p.StudentID == ID);
    if (dbDelete != null) {
        context.Students.Remove(db);
        context.SaveChanges(); // lưu thay đổi
    }
    //6. Lưu ý: Nếu sử dụng using System.Data.Entity.Migrations; có thể sử dụng hàm AddOrUpdate để thay thế Add và Update từ EntityFramework 6.0.0.0
    context.Students.AddOrUpdate(s); //Add or Update sinh viên s
    context.SaveChanges();

```

Bước 2: Thiết kế và lập trình – Viết sự kiện Form-Load

```

private void frmStudentManagement_Load(object sender, EventArgs e)
{
    try
    {
        StudentContextDB context = new StudentContextDB();
        List<Faculty> listFaculty = context.Faculties.ToList(); //lấy các khoa
        List<Student> listStudent = context.Students.ToList(); //lấy sinh viên
        FillFacultyCombobox(listFaculty);
        BindGrid(listStudent);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//Hàm binding list có tên hiển thị là tên khoa, giá trị là Mã khoa
private void FillFacultyCombobox(List<Faculty> listFaculty)
{
    this.cmbFaculty.DataSource = listFaculty;
    this.cmbFaculty.DisplayMember = "FacultyName";
    this.cmbFaculty.ValueMember = "FacultyID";
}

//Hàm binding gridView từ list sinh viên
private void BindGrid(List<Student> listStudent)
{
    dgvStudent.Rows.Clear();
    foreach (var item in listStudent)
    {
        int index = dgvStudent.Rows.Add();
        dgvStudent.Rows[index].Cells[0].Value = item.StudentID;
    }
}

```



```

        dgvStudent.Rows[index].Cells[1].Value = item.FullName;
        dgvStudent.Rows[index].Cells[2].Value = item.Faculty.FacultyName ;
        dgvStudent.Rows[index].Cells[3].Value = item.AverageScore;
    }
}
// Sinh viên tự viết các sự kiện thêm, xóa, sửa sau khi đọc hướng dẫn sử dụng Entity để thao
tác với CSDL.

```

Bài tập 2: Tạo Form quản lý thông tin các khoa ở cùng project trong bài tập 1

- Thêm 1 cột **TotalProfessor** (tổng số giáo sư) kiểu INT cho phép *NULL* vào bảng **Faculty**

Column Name	Data Type	Allow Nulls
FacultyID	int	<input type="checkbox"/>
FacultyName	nvarchar(200)	<input type="checkbox"/>
TotalProfessor	int	<input checked="" type="checkbox"/>

- Thêm 1 form mới là **frmFaculty** có đủ các chức năng thêm, xóa, sửa, hiện thị thông tin khoa tương tự như quản lý Sinh viên ở bài tập 1.

Mã Khoa	Tên Khoa	Tổng số GS
1	Công Nghệ Thông Tin	
2	Ngôn Ngữ Anh	
3	Quản trị kinh doanh	

- Tạo 1 Button ở Form Quản lý sinh viên để khi click đó sẽ gọi sang Form quản lý thông tin các khoa (Hoặc sinh viên có thể dùng MenuStrip – tạo ra Sub menu)
- Thực hiện các yêu cầu trên form quản lý khoa: Lấy dữ liệu vào DataGridView, Thêm/Sửa, Xóa và Đóng form.

Chú ý: Khi CSDL có thay đổi sinh viên nên cập nhật lại phần models **được thay đổi** bằng cách tương tự như lúc tạo ra ban đầu (xóa đi – tạo lại hoặc đưa phần thay đổi chèn vào models hiện tại). Mục đích để đảm bảo CSDL đó đúng với Models.

Bài tập 3: Tìm kiếm sinh viên

Thiết kế chương trình quản lý sinh viên như giao diện sau (Thiết kế thêm Chức năng, Quản lý khoa, tìm kiếm và form tìm kiếm sinh viên từ bài tập 1-2)

Mã Số SV	Họ Tên	Tên Khoa	Điểm TB
1611061916	Nguyễn Trần Hoàng Lan	Công Nghệ Thông Tin	4.5
1711060596	Đàm Minh Đức	Công Nghệ Thông Tin	2.5
1711061004	Nguyễn Quốc An	Ngôn Ngữ Anh	10

- Sử dụng ToolStrip, 2 button Quản lý khoa và Tìm kiếm
- Ở Menu chức năng: thể hiện các chức năng có phím tắt để tới các form: Quản lý khoa (F2) và Tìm Kiếm (Ctrl +F)

- Thiết kế form Tìm kiếm và thực hiện tìm kiếm thông tin sinh viên

Thông Tin Tìm Kiếm

Mã Số SV

Họ Tên

Khoa

Mã Số SV	Họ Tên	Tên Khoa	Điểm TB
----------	--------	----------	---------

Kết quả tìm kiếm

Các khoa được lấy từ CSDL. Mặc định chưa chọn là Empty

- ✓ Khi người dùng click vào button tìm kiếm sẽ tìm kiếm thông tin sinh viên thỏa các điều kiện tìm kiếm (không nhập có nghĩa là bỏ qua điều kiện tìm kiếm đó)
- ✓ Khi người dùng click vào button xóa: Trả lại giá trị mặc định như khi load form tìm kiếm.

Ví dụ: Tìm các sinh viên thuộc khoa công nghệ thông tin

Thông Tin Tìm Kiếm

Mã Số SV

Họ Tên

Khoa

Mã Số SV	Họ Tên	Tên Khoa	Điểm TB
1611061916	Nguyễn Trần Hoàng Lan	Công Nghệ Thông Tin	4.5
1711060596	Đàm Minh Đức	Công Nghệ Thông Tin	2.5

Kết quả tìm kiếm

Bài tập 4: Cho cơ sở dữ liệu quản lý sản phẩm và đơn hàng như sau

- ✓ Sử dụng cơ sở dữ liệu SQL server có 3 bảng **Product**, **Order**, **Invoice** lần lượt như sau

Product: Lưu trữ thông tin sản phẩm (Mã sản phẩm, Tên Sản phẩm, Đơn vị Tính, Giá Mua, Giá Bán)

Order: Lưu trữ chi tiết thông tin đơn hàng (Số HĐ, Số TT, Mã SP, Tên SP, ĐVT, Đơn giá, Số lượng)

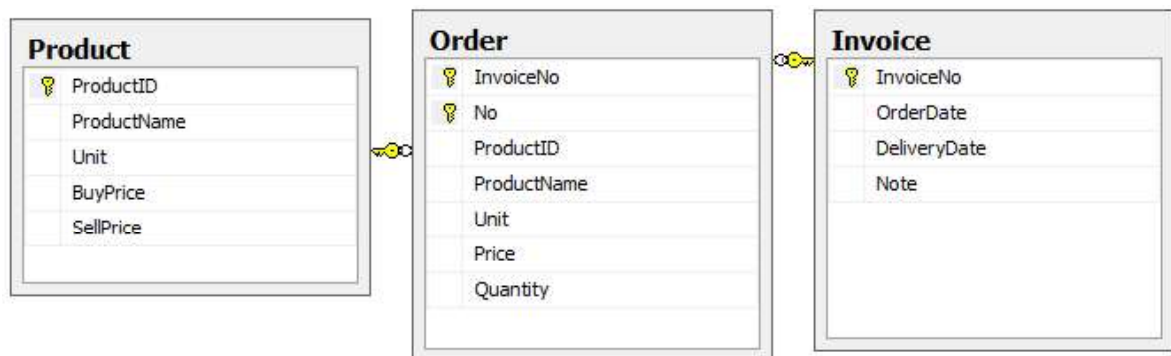
Invoice: Lưu trữ thông tin hóa đơn đặt hàng (Số HĐ, Ngày đặt hàng, ngày giao hàng, ghi chú)

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
ProductName	nvarchar(100)	<input type="checkbox"/>
Unit	nvarchar(20)	<input type="checkbox"/>
BuyPrice	decimal(18, 0)	<input checked="" type="checkbox"/>
SellPrice	decimal(18, 0)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
InvoiceNo	nvarchar(20)	<input type="checkbox"/>
No	int	<input type="checkbox"/>
ProductID	nvarchar(20)	<input type="checkbox"/>
ProductName	nvarchar(100)	<input checked="" type="checkbox"/>
Unit	nvarchar(20)	<input checked="" type="checkbox"/>
Price	decimal(18, 0)	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
InvoiceNo	nvarchar(20)	<input type="checkbox"/>
OrderDate	datetime	<input type="checkbox"/>
DeliveryDate	datetime	<input type="checkbox"/>
Note	nvarchar(255)	<input checked="" type="checkbox"/>

- ✓ Sơ đồ diagrams



Sử dụng script để tạo ra nhanh CSDL, và dữ liệu tương ứng

```

USE [ProductOrder]
GO
/***** Object:  Table [dbo].[Invoice]      Script Date: 07/04/2020 23:13:15
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Invoice] (
    [InvoiceNo] [nvarchar] (20) NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [DeliveryDate] [datetime] NOT NULL,
    [Note] [nvarchar] (255) NULL,
    CONSTRAINT [PK_Invoice] PRIMARY KEY CLUSTERED
  
```

```
(
    [InvoiceNo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Product]      Script Date: 07/04/2020 23:13:15
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Product] (
    [ProductID] [nvarchar](20) NOT NULL,
    [ProductName] [nvarchar](100) NOT NULL,
    [Unit] [nvarchar](20) NOT NULL,
    [BuyPrice] [decimal](18, 0) NULL,
    [SellPrice] [decimal](18, 0) NULL,
    CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
    [ProductID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Order]      Script Date: 07/04/2020 23:13:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Order] (
    [InvoiceNo] [nvarchar](20) NOT NULL,
    [No] [int] NOT NULL,
    [ProductID] [nvarchar](20) NOT NULL,
    [ProductName] [nvarchar](100) NULL,
    [Unit] [nvarchar](20) NULL,
    [Price] [decimal](18, 0) NOT NULL,
    [Quantity] [int] NOT NULL,
    CONSTRAINT [PK_Order] PRIMARY KEY CLUSTERED
(
    [InvoiceNo] ASC,
    [No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: ForeignKey [FK_Order_Invoice]      Script Date: 07/04/2020
23:13:15 *****/
ALTER TABLE [dbo].[Order] WITH CHECK ADD CONSTRAINT [FK_Order_Invoice] FOREIGN
KEY([InvoiceNo])
REFERENCES [dbo].[Invoice] ([InvoiceNo])
GO
ALTER TABLE [dbo].[Order] CHECK CONSTRAINT [FK_Order_Invoice]
GO
/***** Object: ForeignKey [FK_Order_Product]      Script Date: 07/04/2020
23:13:15 *****/
ALTER TABLE [dbo].[Order] WITH CHECK ADD CONSTRAINT [FK_Order_Product] FOREIGN
KEY([ProductID])
REFERENCES [dbo].[Product] ([ProductID])
GO
```

```
ALTER TABLE [dbo].[Order] CHECK CONSTRAINT [FK_Order_Product]
GO
```

Thêm 1 số dữ liệu vào database như sau

```
USE [ProductOrder]
GO

/***** Object: Table [dbo].[Invoice]      Script Date: 07/04/2020 23:14:16
*****/

INSERT [dbo].[Invoice] ([InvoiceNo], [OrderDate], [DeliveryDate], [Note]) VALUES
(N'HDX001', CAST(0x0000AAD900000000 AS DateTime), CAST(0x0000AADA00000000 AS
DateTime), N'Giao hàng trước 9h')

INSERT [dbo].[Invoice] ([InvoiceNo], [OrderDate], [DeliveryDate], [Note]) VALUES
(N'HDX002', CAST(0x0000AADA00000000 AS DateTime), CAST(0x0000AADA00000000 AS
DateTime), N'Gọi điện trước khi giao')

INSERT [dbo].[Invoice] ([InvoiceNo], [OrderDate], [DeliveryDate], [Note]) VALUES
(N'HDX003', CAST(0x0000AADA00000000 AS DateTime), CAST(0x0000AADC00000000 AS
DateTime), N'giao tu 1-3h')

/***** Object: Table [dbo].[Product]      Script Date: 07/04/2020 23:14:16
*****/

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product1', N'Sản phẩm 1', N'Cái', CAST(100000 AS
Decimal(18, 0)), CAST(120000 AS Decimal(18, 0)))

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product2', N'Sản phẩm 2', N'Cái', CAST(90000 AS
Decimal(18, 0)), CAST(120000 AS Decimal(18, 0)))

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product3', N'Sản phẩm 3', N'Cái', CAST(40000 AS
Decimal(18, 0)), CAST(70000 AS Decimal(18, 0)))

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product4', N'Sản phẩm 4', N'Hộp', CAST(200000 AS
Decimal(18, 0)), CAST(300000 AS Decimal(18, 0)))

/***** Object: Table [dbo].[Order]      Script Date: 07/04/2020 23:14:16 *****/

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX001', 1, N'Product1', N'Sản phẩm 1', N'Cái',
CAST(120000 AS Decimal(18, 0)), 20)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX001', 2, N'Product2', N'Sản phẩm 2', N'Cái',
CAST(120000 AS Decimal(18, 0)), 4)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX001', 3, N'Product4', N'Sản phẩm 4', N'Hộp',
CAST(300000 AS Decimal(18, 0)), 10)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX002', 1, N'Product4', N'Sản phẩm 1', N'Hộp',
CAST(300000 AS Decimal(18, 0)), 10)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX002', 2, N'Product2', N'Sản phẩm 3', N'Cái',
CAST(300000 AS Decimal(18, 0)), 12)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX003', 1, N'Product1', N'Sản phẩm 1', N'Cái',
CAST(120000 AS Decimal(18, 0)), 40)
```

```
INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX003', 4, N'Product2', N'Sản phẩm 2', N'Cái',
CAST(120000 AS Decimal(18, 0)), 60)
```

Viết chương trình phần mềm xem thông tin đơn hàng như sau

Khi load Form

- Thời gian giao hàng được thể hiện trong ngày hiện hành và tự động tìm kiếm dữ liệu có Hóa Đơn phát sinh trong ngày hiện hành này
- Người dùng có thể thay đổi thời gian giao hàng trong 1 khoảng thời gian bất kì, khi đó dữ liệu cũng được tự động thay đổi theo

STT	Số HĐ	Ngày Đặt Hàng	Ngày Giao Hàng	Thành Tiền
1	HDX001	01/10/2019	02/10/2019	5880000
2	HDX002	02/10/2019	02/10/2019	6600000

- Khi check vào CheckBox Xem tất cả trong tháng, thì thời gian giao hàng sẽ được thể hiện từ **ngày đầu tháng hiện hành** đến **cuối tháng** và hiện thị thông tin giao hàng trong thời gian đó.