

Lab 04:

LẬP TRÌNH WINDOWS FORM KẾT NỐI CSDL SQL SERVER

A. MỤC TIÊU:

- ✓ Hướng dẫn sinh viên làm quen với việc xây dựng ứng dụng Windows App có kết nối với CSDL SQL Server bằng Entity Framework của .NET.
- ✓ Sử dụng mô hình *Code First* trong *EntityFramework* và Sử dụng ADO.NET
- ✓ Thiết kế các Form nhập liệu cho các bảng trong cơ sở dữ liệu (hiện thị, thêm, xóa, sửa)
- ✓ Nâng cao: Tìm hiểu vấn đề transaction trong thao tác với cơ sở dữ liệu.

B. BÀI TẬP:

Sử dụng SQL Server tạo cơ sở dữ liệu “QuanLySinhVien” đơn giản với 2 bảng

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

và **Faculty**(*FacultyID*, *FacultyName*)

```
USE [QuanLySinhVien]
GO
/***** Object: Table [dbo].[Faculty]      Script Date: 10/07/2019 20:59:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Faculty] (
    [FacultyID] [int] NOT NULL,
    [FacultyName] [nvarchar](200) NULL,
    CONSTRAINT [PK_dbo.Faculty] PRIMARY KEY CLUSTERED
(
    [FacultyID] 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].[Student]      Script Date: 10/07/2019 20:59:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Student] (
    [StudentID] [nvarchar](20) NOT NULL,
    [FullName] [nvarchar](200) NULL,
    [AverageScore] [float] NULL,
    [FacultyID] [int] NULL,
    CONSTRAINT [PK_dbo.Student] PRIMARY KEY CLUSTERED
(
    [StudentID] 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_Student_Faculty]      Script Date: 10/07/2019
20:59:08 *****/
ALTER TABLE [dbo].[Student] WITH CHECK ADD CONSTRAINT [FK_Student_Faculty]
FOREIGN KEY ([FacultyID])
REFERENCES [dbo].[Faculty] ([FacultyID])
GO
```

```
ALTER TABLE [dbo].[Student] CHECK CONSTRAINT [FK_Student_Faculty]
GO
```

Bài Tập 1 Sử dụng EntityFrameWork với mô hình Code First

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

| 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 |
| 1711061125 | Lê Khải Hiếu | Ngôn Ngữ Anh | 10 |
| 1711061880 | Nguyễn Văn Bốn | Công Nghệ Thông Tin | 9 |
| 1711062090 | Lê Tuấn Sang | Công Nghệ Thông Tin | 6 |
| 1711062409 | Lê Xuân Quang | Công Nghệ Thông Tin | 8 |

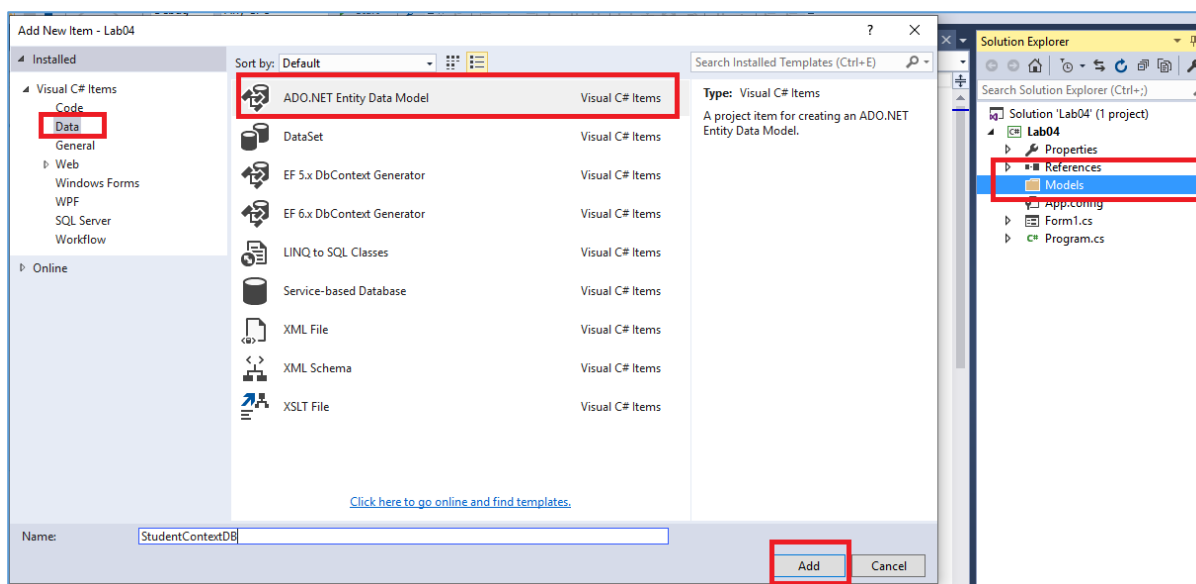
Yêu Cầu Xử Lý

- Sự kiện Form_load: Hiển thị danh sách các sinh viên đang có trong CSDL
Combobox Khoa lấy từ bảng **Faculty** và hiện thị tên khoa
- Add/ Update: Nếu Sinh Viên chưa có mã số trong cơ sở dữ liệu thì Thêm mới
Nếu đã tồn tại thì cập nhật
Thông báo thành công khi insert/update hoặc khi gặp lỗi.
- Delete: Nếu mã Sinh Viên không tồn tại thông báo thể hiện thông báo
Nếu mã Sinh viên tồn tại. Cảnh báo Yes/No trước khi thực hiện, xóa dòng dữ liệu có mã số StudentID trong cơ sở dữ liệu khi chọn Yes.

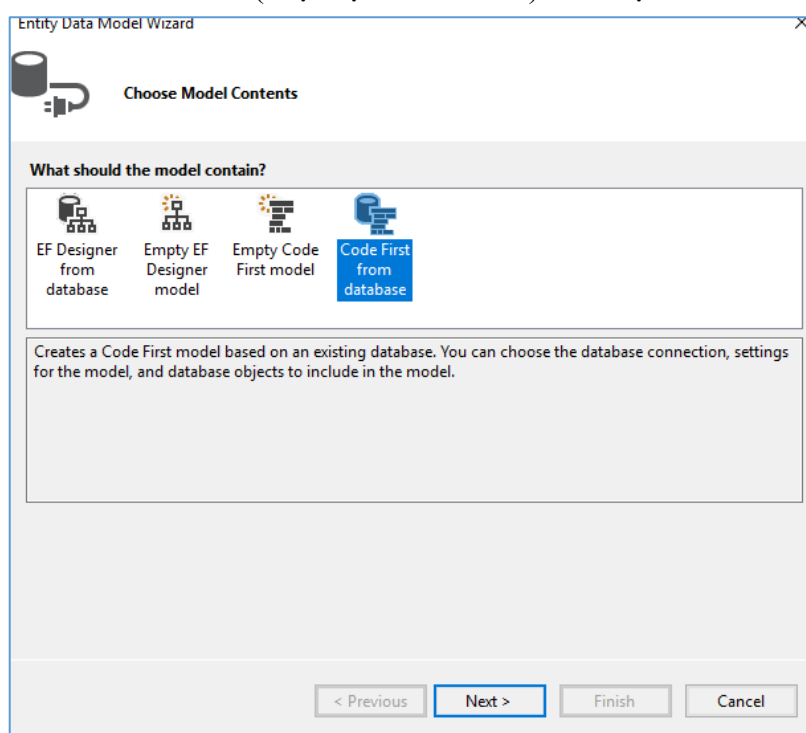
Hướng Dẫn

Bước 1: Entity FrameWork sinh ra các class chúng ta nên gom vào 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**

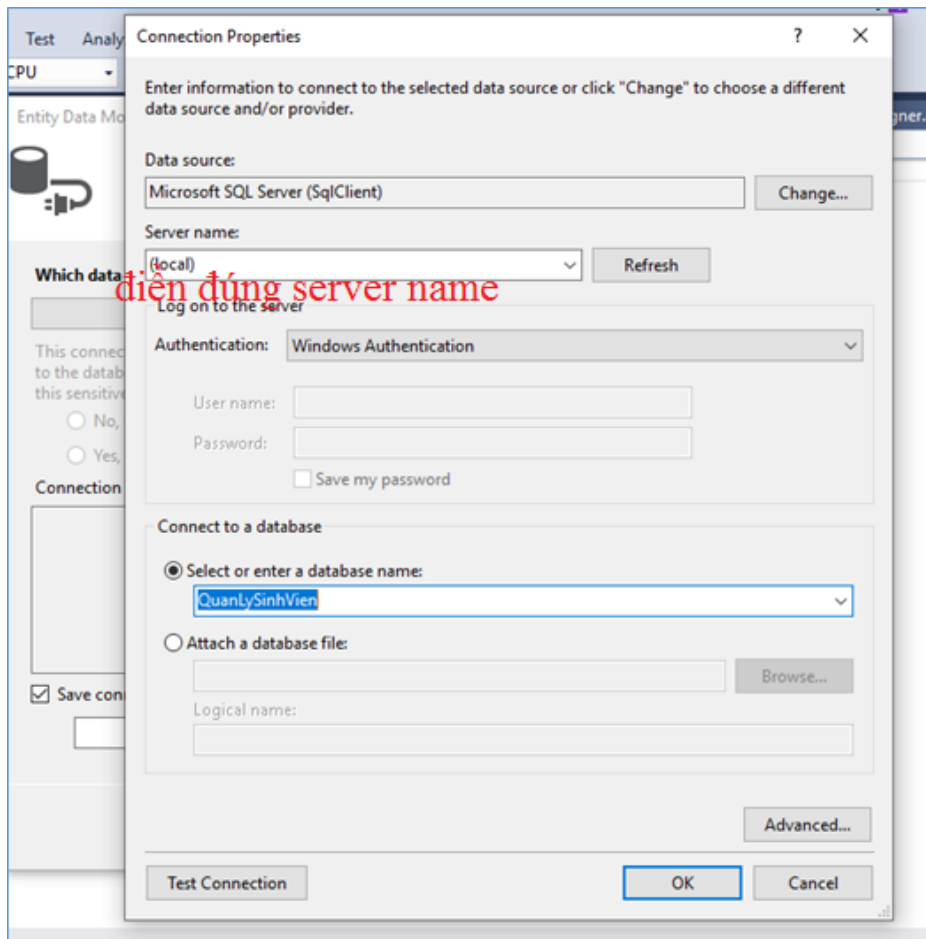


Đặt tên context là “*StudentContextDB*” (mặc định là Model1). Và chọn Add

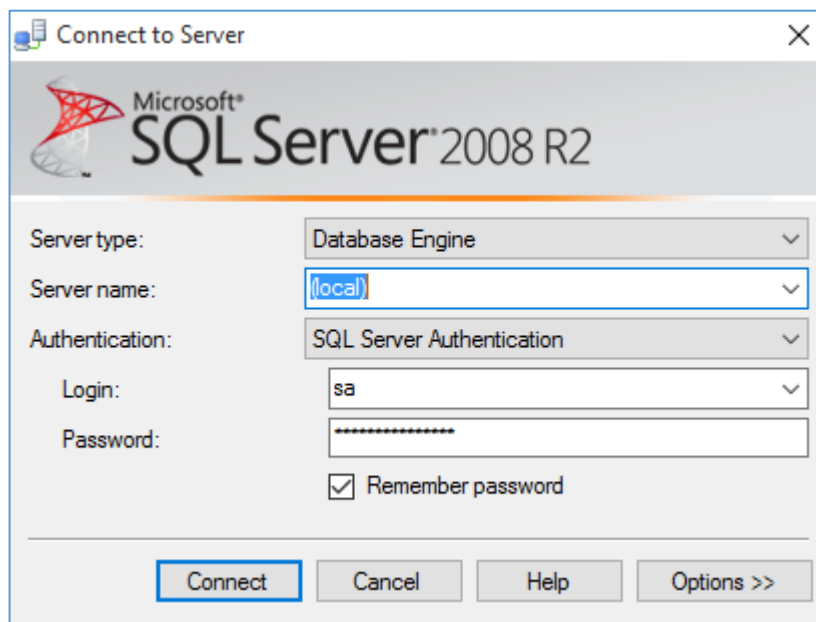


Ta chọn loại model là “*Code first from database*”. Chọn Next

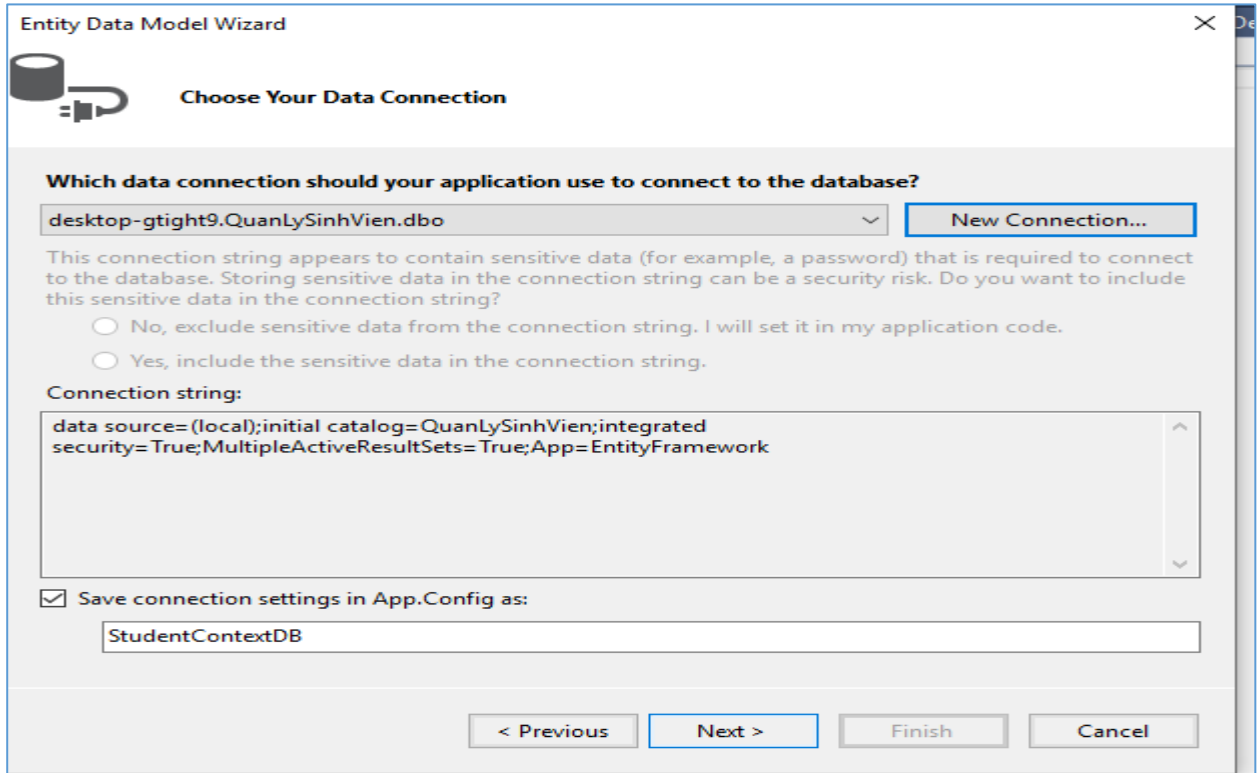
Tìm cơ sở dữ liệu Student ở SQL để trả database name vào



Chọn đúng tên server Name trên từng máy có thể khác nhau. Nếu quên 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



Entity Data Model Wizard

Choose Your Data Connection

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

desktop-gtight9.QuanLySinhVien.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:

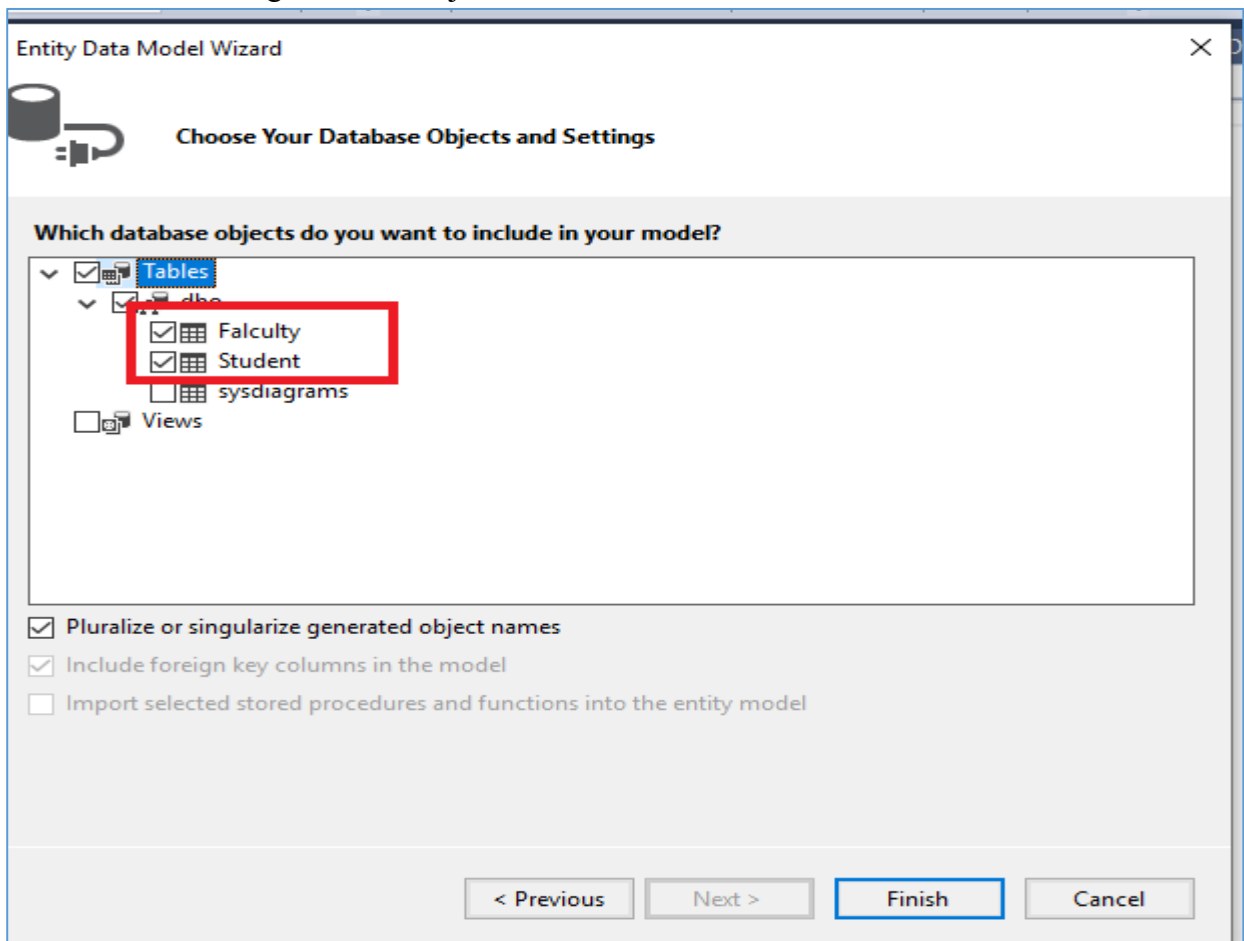
data source=(local);initial catalog=QuanLySinhVien;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework

☒ Save connection settings in App.Config as:

StudentContextDB

< Previous Next > Finish Cancel

Sau đó chọn các bảng muốn tạo object



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

Tables

dbo

☒ Faculty

☒ Student

☐ sysdiagrams

☐ Views

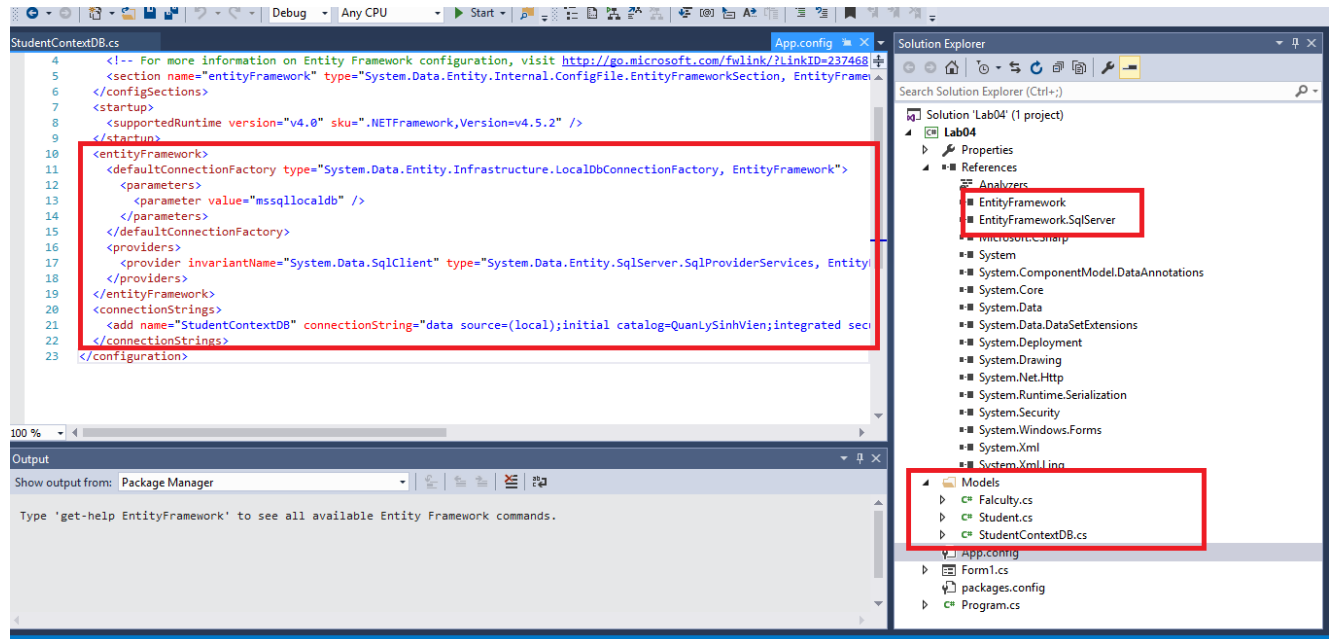
☒ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

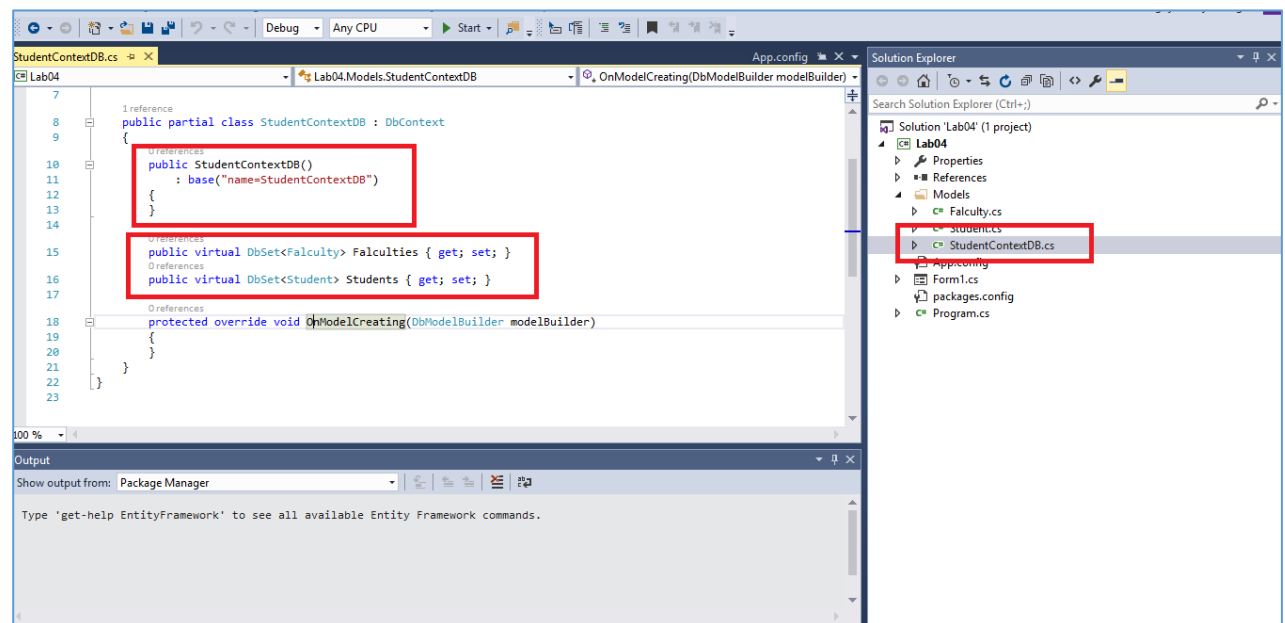
☐ Import selected stored procedures and functions into the entity model

< Previous Next > Finish Cancel

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



Ở File StudentContextDB.cs chứa tập hợp DataSet các table



VS.Net tự động sinh ra các file tương ứng map với database. Sinh viên kiểm tra từng file để hiểu cách tự động mapping.

✓ Cách Sử dụng Entity để Lấy tất cả, thêm , xóa , sửa (cho sv muốn test model)

```
//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

```

        // muốn sử dụng DataGridView có style như ví dụ
        public void SetGridViewStyle(DataGridView dgvview)
        {
            dgvview.BorderStyle = BorderStyle.None;
            dgvview.AlternatingRowsDefaultCellStyle.BackColor = Color.FromArgb(238, 239,
249);
            dgvview.DefaultCellStyle.SelectionBackColor = Color.DarkTurquoise;
            dgvview.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;
            dgvview.DefaultCellStyle.SelectionForeColor = Color.WhiteSmoke;
            dgvview.BackgroundColor = Color.White;
            dgvview.EnableHeadersVisualStyles = false;
            dgvview.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;
            dgvview.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(20, 25, 72);
            dgvview.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;
            dgvview.AllowUserToDeleteRows = false;
            dgvview.AllowUserToAddRows = false;
            dgvview.AllowUserToOrderColumns = true;
            dgvview.MultiSelect = false;
            dgvview.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
        }
        private void frmStudentManagement_Load(object sender, EventArgs e)
        {
            try
            {
                SetGridViewStyle(dgvStudent);
                StudentContextDB context = new StudentContextDB();
                List<Faculty> listFaculty = context.Faculties.ToList();
                List<Student> listStudent = context.Students.ToList();

                FillFacultyCombobox(listFaculty);
                BindGrid(listStudent);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    
```

```

    }

    //Hàm binding list dữ liệu khoa vào combobox 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";
    }

    private void BindGrid(List<Student> listStudent) //Hàm binding gridView từ list sinh
viên
    {
        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;
            if(item.Faculty!= null)
                dgvStudent.Rows[index].Cells[2].Value = item.Faculty.FacultyName;
            dgvStudent.Rows[index].Cells[3].Value = item.AverageScore + "";
        }
    }

    // Sinh viên tự viết các hàm insert,update và delete sau khi đọc hướng dẫn sử dụng Entity để
    làm bài tập
    // Hướng dẫn:
    Phân biệt trường hợp Thêm Mới -> ( ID của sinh viên chưa tồn tại trong database
    trường hợp Update -> (ID của sinh viên nhập đã tồn tại trong database

```

Bài Tập 2: Sử dụng công nghệ ADO.NET (Cho Sinh viên muốn tìm hiểu công nghệ ADO.NET)

Hướng Dẫn Sử dụng công nghệ ADO.NET

- ✓ Sử dụng chuỗi connectionString trong config App.config để dễ dàng chỉnh sửa thay đổi.

```

<connectionStrings>
    <add name="DSStudentConnectionString" connectionString="data
        source=(local);initial catalog=QuanLySinhVien;integrated
        security=True;MultipleActiveResultSets=True;"
        providerName="System.Data.SqlClient"/>
</connectionStrings>

```

Để đọc được chuỗi này ta add thêm thư viện **System.Configuration** có sẵn của Framework

- ✓ Sử dụng **SqlConnection, SqlCommand, SqlDataReader...** để thao tác với cơ sở dữ liệu trong thư viện `using System.Data.SqlClient;`
- Ví dụ hàm Lấy tất cả danh sách sinh viên trong bảng Student được viết như sau

```

public static List<Student> GetAllStudent()
{
    List<Student> listStudent = new List<Student>();
    string connectionString =
        ConfigurationManager.ConnectionStrings["DSStudentConnectionString"].ConnectionString;
    string queryString = "Select * from Student";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(queryString, connection);
        command.CommandType = CommandType.Text;
        connection.Open();
        using (SqlDataReader objReader = command.ExecuteReader())
        {
            if (objReader.HasRows)

```



```

        {
            while (objReader.Read())
            {
                Student temp = new Student();
                string studentID =
                objReader.GetString(objReader.GetOrdinal("StudentID"));
                if (studentID != null)
                {
                    temp.StudentID = studentID;
                    temp.FullName = objReader["FullName"].ToString();
                    string averageScore =
                objReader["AverageScore"].ToString();
                    if (averageScore != null)
                        temp.AverageScore =
                Convert.ToDouble(averageScore);

                    string facultyID = objReader["FacultyID"].ToString();
                    if (facultyID != null)
                        temp.FacultyID = Convert.ToInt32(facultyID);
                }
                listStudent.Add(temp);
            }
        }
    }
    return listStudent;
}

```

- Viết hàm ExcuteNonQuery để thực hiện 1 lệnh câu lệnh SQL (commandText)

```

public void ExcuteNonQuery(string commandText)
{
    string connectionString =
    ConfigurationManager.ConnectionStrings["DSStudentConnectionString"].ConnectionString;

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(commandText, connection);
        command.CommandType = CommandType.Text;
        connection.Open();
        command.ExecuteNonQuery();
    }
}

```

- Viết hàm Insert, Update, Delete cho Sinh viên

```

public void InsertStudent()
{
    string queryString = @"INSERT INTO Student
        ([StudentID]
        ,[FullName]
        ,[AverageScore]
        ,[FacultyID])
        VALUES ('{0}' , '{1}',{2},{3})";
    queryString = string.Format(queryString, this.StudentID, this.FullName,
    this.AverageScore, this.FacultyID);
    ExcuteNonQuery(queryString);
}
public void UpdateStudent(string studentUpdateID)
{
    string queryString = @"UPDATE Student
        Set [FullName] = '{1}'
        ,[AverageScore] = {2}
        ,[FacultyID] = {3}

```

```
WHERE StudentID= {0}";
queryString = string.Format(queryString, this.StudentID, this.FullName,
this.AverageScore, this.FacultyID);
ExcuteNonQuery(queryString);
}
public void DeleteStudent()
{
    string queryString = @"DELETE FROM Student
WHERE StudentID= {0}";
    queryString = string.Format(queryString, this.StudentID);
    ExcuteNonQuery(queryString);
}
```

Bài tập 3: Viết tiếp ở bài tập 1 hoặc bài tập 2

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

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

-Ở Form Quản lý sinh viên thêm 1 Menu có tên **Chức Năng** và có 2 SubMenu là Thông Tin Khoa và Thoát. Khi click vào Thông tin Khoa sẽ tới Form quản lý khoa (frmFaculty)

- 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.

-Viết sự kiện ở DataGridView khi người dùng chọn vào dòng nào ở GridView (bên phải) thì thông tin sẽ được hiện thị lại ngay bên cạnh ở phần thông tin bên trái (cho cả 2 form Quản lý Khoa và Quản lý Sinh Viên)

- Ở form Quản lý sinh viên, Khi người dùng tự nhập vào 1 tên khoa *chưa tồn tại trong cơ sở dữ liệu* trong combobox. Khi Insert/Update có cảnh báo sẽ thêm thông tin khoa mới

Click Yes sẽ insert thêm 1 khoa mới vào cơ sở dữ liệu, với mã khoa mới FacultyID = max(FacultyID) + 1, các trường khác là giá trị NULL mặc định.

Chú ý:

Khi CSDL có thay đổi sinh viên nên cập nhật lại phần models được thay đổi bằng cách tương tự lúc tạo ra ban đầu (xóa đi – tạo lại hoặc đưa phần thay đổi chèn vào models hiện tại).

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)

```
USE [ProductOrder]
GO
/***** Object: Table [dbo].[Product]      Script Date: 10/02/2019 00:52:26 *****/
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: 10/02/2019 00:52:26 *****/
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:  Table [dbo].[Invoice]      Script Date: 10/02/2019 00:52:26
*****/
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

```

Thêm 1 số dữ liệu vào database giả sử như sau

```

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)))
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].[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')

```

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

Thông Tin Đơn Hàng

☐ Xem tất cả trong tháng

Thời Gian Giao Hàng

 ~

| STT | Số HĐ | Ngày Đặt Hàng | Ngày Giao Hàng | Thành Tiền |
|-----|-------|---------------|----------------|------------|
| | | | | |

Tổng Cộng:

0

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 thay đổi theo

Thông Tin Đơn Hàng

☐ Xem tất cả trong tháng

Thời Gian Giao Hàng

02/10/2019

 ~

04/10/2019

| | 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 hiện hành

Thông Tin Đơn Hàng

☒ Xem tất cả trong tháng

Thời Gian Giao Hàng ~

| | 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 |

-----Hết Lab 04-----