


MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Cài đặt và sử dụng EF 6
- ✓ Viết ứng dụng các thao tác CRUD với EF 6

PHẦN I**Bài 1 (4 điểm)**

Thiết kế CSDL StudentInformation.mdf:

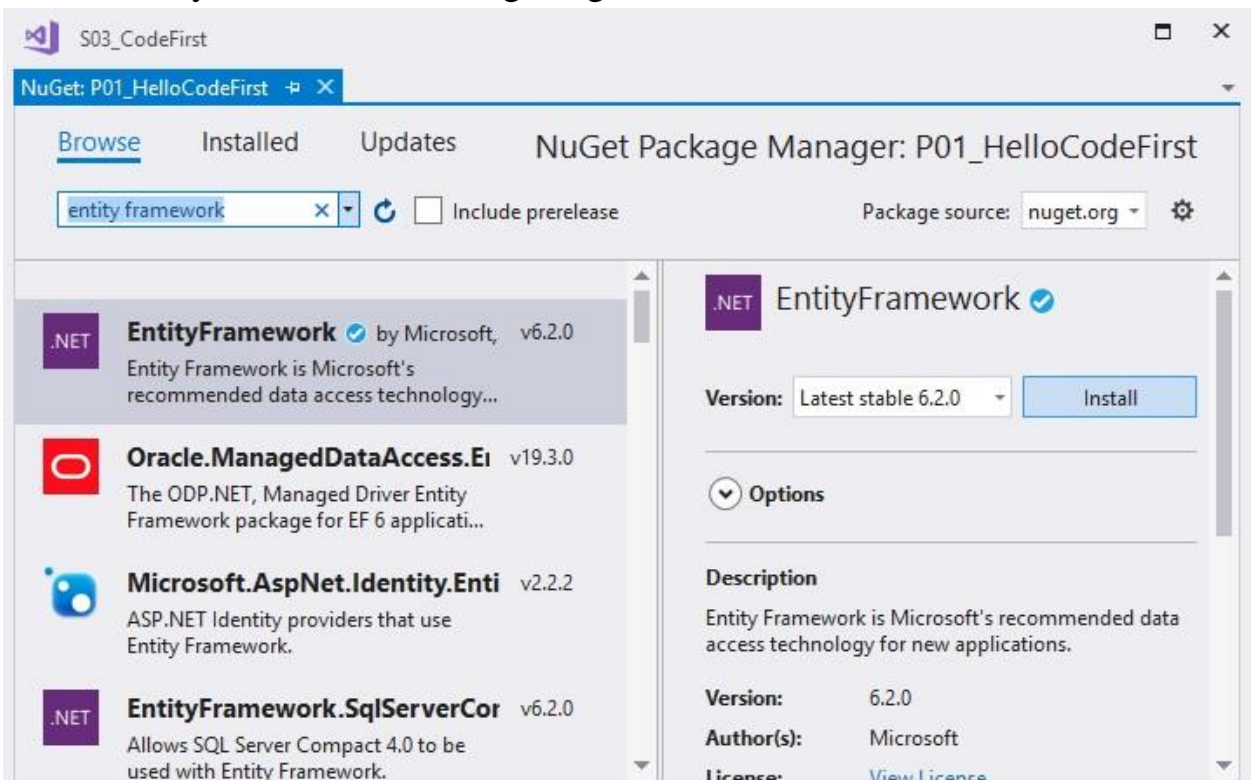
| StudentDetails | |
|---|--------|
|  | Id |
| | Name |
| | Age |
| | City |
| | Gender |

tham khảo script bên dưới:

```
create database StudentInformation;  
CREATE TABLE [dbo].[StudentDetails](  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Name] [varchar](50) NULL,  
    [Age] [int] NULL,  
    [City] [varchar](50) NULL,  
    [Gender] [varchar](50) NULL,  
    PRIMARY KEY (Id))
```

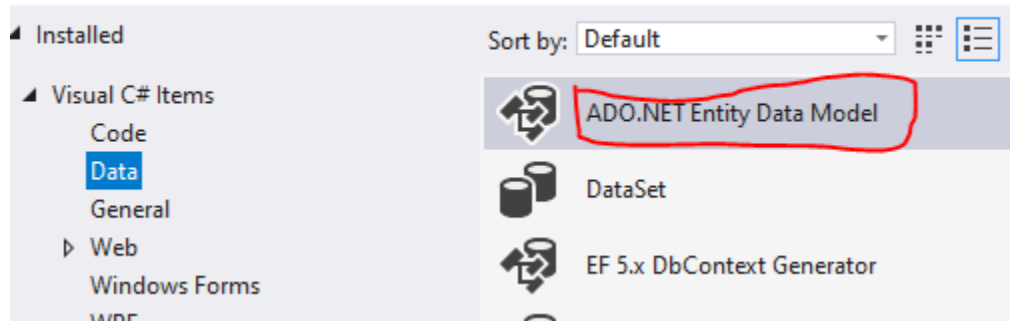
- a) Tạo ứng dụng winform với các thành phần labels, text-boxes, combo-boxes, Data GridView và buttons, chú ý ID label là hidden field:

Cài đặt Entity Framework vào ứng dụng:

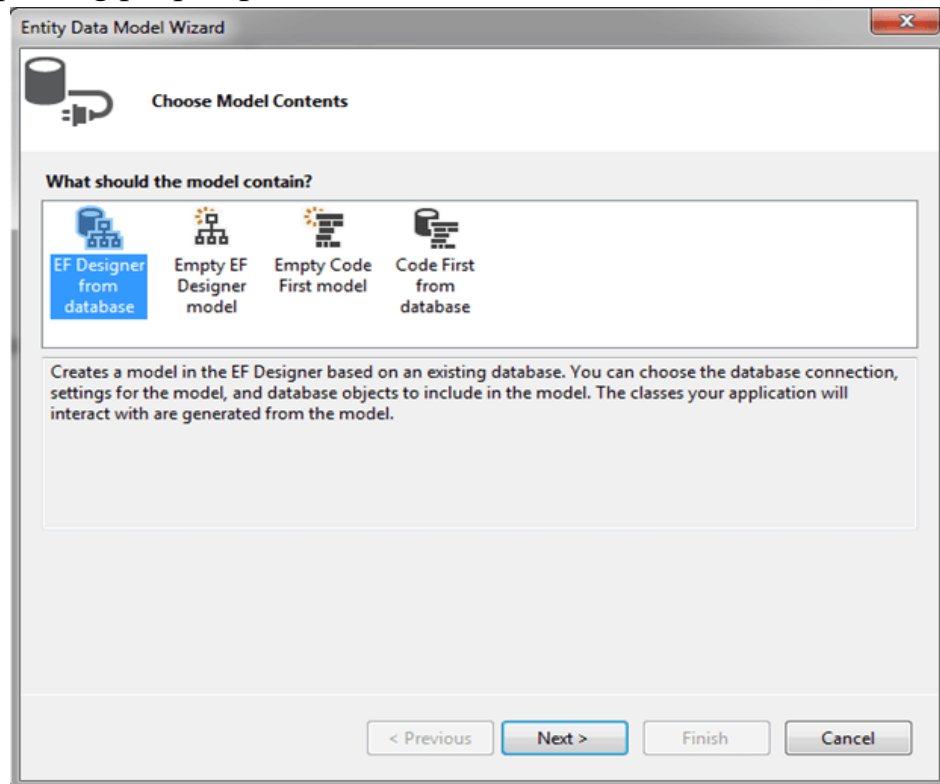


b) Thêm ADO.NET Entity Data Model với tên “Student” vào ứng dụng

Add New Item - Slide5



Chọn phương pháp tiếp cận Database-First



Tạo kết nối tới csdl với tên model: StudentInformationEntities

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

desktop-86lvi3p.StudentInformation.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.

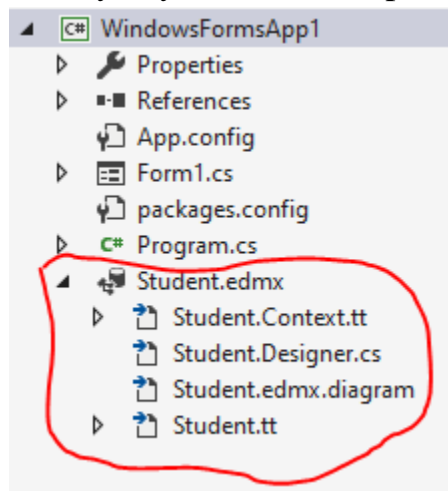
Connection string:

```
metadata=res://*/Student.csdl|res://*/Student.ssdl|
res://*/Student.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial
catalog=StudentInformation;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

StudentInformationEntities

Sau khi tạo thành công sẽ thấy đầy đủ các thành phần Entity Model:



Tạo thêm class StudentInformation trong project để có thể tương tác với các thành phần trên form:

```
namespace WindowsFormsApplication1.Models
{
    public class StudentInformation
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int? Age { get; set; }
        public string City { get; set; }
        public string Gender { get; set; }
    }
}
```

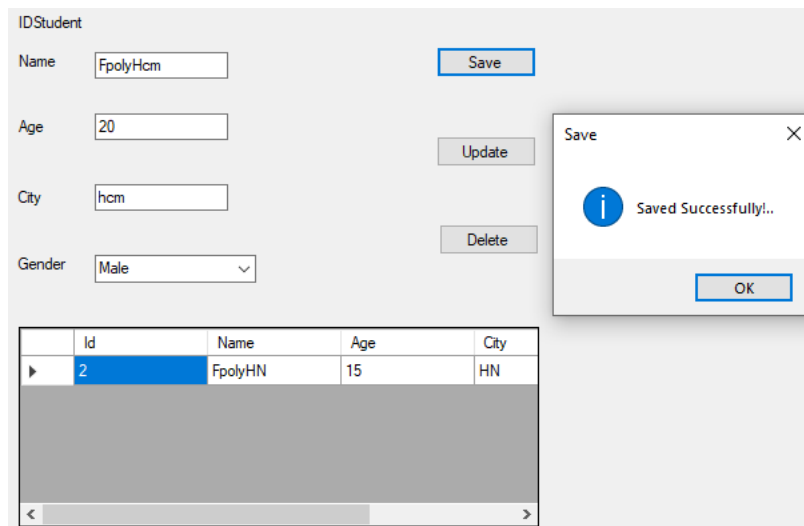
c) Lấy danh sách Student Details lên Data GridView

```
private void Form1_Load(object sender, EventArgs e) // Form load Method
{
    cmbGender.Items.Add("Male"); // Adding values for Gender Combobox
    cmbGender.Items.Add("Female");
    Display(); // calling Display Method for Bind the Student Details in
    DataGridView
}

public void Display() // Display Method is a common method to bind the
Student details in datagridview after save,update and delete operation perform.
{
    using (StudentInformationEntities _entity=new StudentInformationEntities())
    {
        List<StudentInformation> _studentList = new List<StudentInformation>
        ();
        _studentList = _entity.StudentDetails.Select(x => new StudentInformation
        {
            Id=x.Id,
            Name=x.Name,
            Age=x.Age,
            City=x.City,
            Gender=x.Gender
        }).ToList();
        dataGridView1.DataSource = _studentList;
    }
}
```

d) Xử lý khi người dùng click “Save”: Nhận các thông tin Name, Age, City, Gender từ form và truyền cho phương thức "SaveStudentDetails", phương thức này dùng SaveChanges() thông qua StudentInformationEntities để thêm mới dữ liệu xuống csdl

```
public bool SaveStudentDetails(StudentDetail Stu) // calling SaveStudentMethod for insert a new record
{
    bool result = false;
    using (StudentInformationEntities _entity = new StudentInformationEntities())
    {
        _entity.StudentDetails.Add(Stu);
        _entity.SaveChanges();
        result = true;
    }
    return result;
}
1 reference
private void BtnSave_Click(object sender, EventArgs e)
{
    StudentDetail stu = new StudentDetail();
    stu.Name = txtName.Text;
    stu.Age = Convert.ToInt32(txtAge.Text);
    stu.City = txtCity.Text;
    stu.Gender = cmbGender.SelectedItem.ToString();
    // calling SaveStudentDetails method to save the record in table. Here passing
    // a student details object as parameter
    bool result = SaveStudentDetails(stu);
    ShowStatus(result, "Save");// Xem phương thức ShowStatus bên dưới
}
```



| Id | Name | Age | City |
|----|---------|-----|------|
| 2 | FpolyHN | 15 | HN |

PHẦN 2

Bài 2 (4 điểm)

Xử lý tính năng Update & Delete:

Cần có phương thức xác định người dùng đang chọn record nào trên GridView

```
private void Dgv_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // foreach datagridview selected rows values
    foreach (DataGridViewRow row in dgv.SelectedRows)
    {
        lblID.Text = row.Cells[0].Value.ToString();
        txtName.Text = row.Cells[1].Value.ToString();
        txtAge.Text = row.Cells[2].Value.ToString();
        txtCity.Text = row.Cells[3].Value.ToString();
        cmbGender.SelectedItem = row.Cells[4].Value.ToString();
    }
}
```

Xử lý khi người dùng chọn button “Update”

```
public bool UpdateStudentDetails(StudentDetail Stu)
{
    bool result = false;
    using (StudentInformationEntities _entity = new StudentInformationEntities())
    {
        StudentDetail _student = _entity.StudentDetails.Where(x => x.Id == Stu.Id).Select(x => x).FirstOrDefault();
        _student.Name = Stu.Name;
        _student.Age = Stu.Age;
        _student.City = Stu.City;
        _student.Gender = Stu.Gender;
        _entity.SaveChanges();
        result = true;
    }
    return result;
}

1 reference
private void BtnUpdate_Click(object sender, EventArgs e)
{
    // Binding values to StudentInformationModel
    StudentDetail stu = SetValues(Convert.ToInt32(lblID.Text), txtName.Text,
    Convert.ToInt32(txtAge.Text), txtCity.Text, cmbGender.SelectedItem.ToString());
    bool result = UpdateStudentDetails(stu); // calling UpdateStudentDetails Method
    ShowStatus(result, "Update");
}
```

| Id | Name | Age | City |
|----|----------|-----|------|
| 2 | FpolyHN | 15 | HN |
| 3 | FpolyHcm | 20 | hcm |

e) Xử lý khi người dùng chọn button “Delete”

```
// DeleteStudentDetails method to delete record from table
1 reference
public bool DeleteStudentDetails(StudentDetail Stu)
{
    bool result = false;
    using (StudentInformationEntities _entity = new StudentInformationEntities())
    {
        StudentDetail _student = _entity.StudentDetails.Where(x => x.Id == Stu.Id).Select(x => x).FirstOrDefault();
        _entity.StudentDetails.Remove(_student);
        _entity.SaveChanges();
        result = true;
    }
    return result;
}
1 reference
private void BtnDelete_Click(object sender, EventArgs e)
{
    // Binding values to StudentInformationModel
    StudentDetail stu = SetValues(Convert.ToInt32(lblID.Text), txtName.Text,
    Convert.ToInt32(txtAge.Text), txtCity.Text, cmbGender.SelectedItem.ToString());
    bool result = DeleteStudentDetails(stu); //Calling DeleteStudentDetails Method
    ShowStatus(result, "Delete");
}
```

- f) Các phương thức hỗ trợ kiểm tra, hiển thị thông báo cho các chức năng thêm xóa sửa : ClearFields(), ShowStatus()


```
// validate the Operation Status and Show the Messages To User
3 references
public void ShowStatus(bool result, string Action)
{
    if (result)
    {
        if (Action.ToUpper() == "SAVE")
        {
            MessageBox.Show("Saved Successfully!..", "Save", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
        else if (Action.ToUpper() == "UPDATE")
        {
            MessageBox.Show("Updated Successfully!..", "Update", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("Deleted Successfully!..", "Delete", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
    }
    else
    {
        MessageBox.Show("Something went wrong!. Please try again!..", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    ClearFields();
    Display();
}

// Clear the fields after Insert or Update or Delete operation
1 reference
public void ClearFields()
{
    txtName.Text = "";
    txtAge.Text = "";
    txtCity.Text = "";
    cmbGender.SelectedIndex = -1;
}
```

Bài 3 (2 điểm)

Thực hiện các validate cần thiết trên form (không nhập liệu, các dữ liệu số, trùng Id...)