

Lab 06

Truy cập dữ liệu sử dụng ADO.NET Entity Framework

Phần I: Bài tập step by step

Mục tiêu:

- Mô hình hóa cơ sở dữ liệu sử dụng ADO.NET Entity Framework
- Thực hiện các thao tác với cơ sở dữ liệu sử dụng ADO.NET Entity Framework

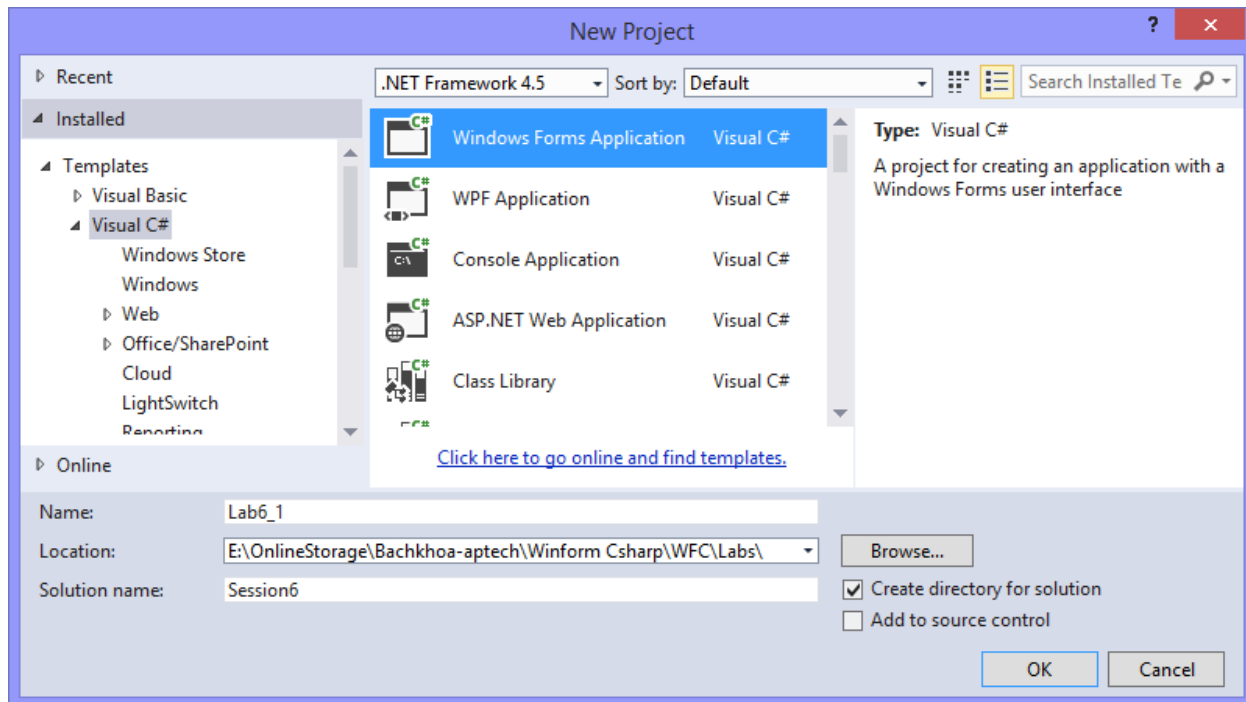
Bài 6.1

Tạo ứng dụng Windows Form cho phép thực hiện các thao tác thêm xóa sửa, hiển thị dữ liệu với bảng Employee sử dụng ADO.NET Entity Framework.

- Cho cơ sở dữ liệu HRM như sau bài Lab5_1
- Giao diện màn hình thêm nhân viên

The screenshot shows a Windows Form titled "Quản lý nhân viên". It features several input fields for employee information: "Mã NV", "Họ đệm", "Tên", "Ngày sinh" (with a calendar icon), "Email", "Giới tính" (with radio buttons for "Nam"), "Địa chỉ", "Điện thoại", and "Phòng ban" (with a dropdown arrow). Below these fields are five buttons: "Tạo mới", "Cập nhật", "Xóa", "Bỏ qua", and "Thoát". At the bottom of the form is a large, empty gray rectangular box, presumably for displaying a list of employees.

Bước 1: Mở Visual Studio 2013 -> Click menu File -> New -> Project -> chọn loại Windows Forms Appliaction -> Nhập tên Project -> Tên Solution như hình dưới -> OK



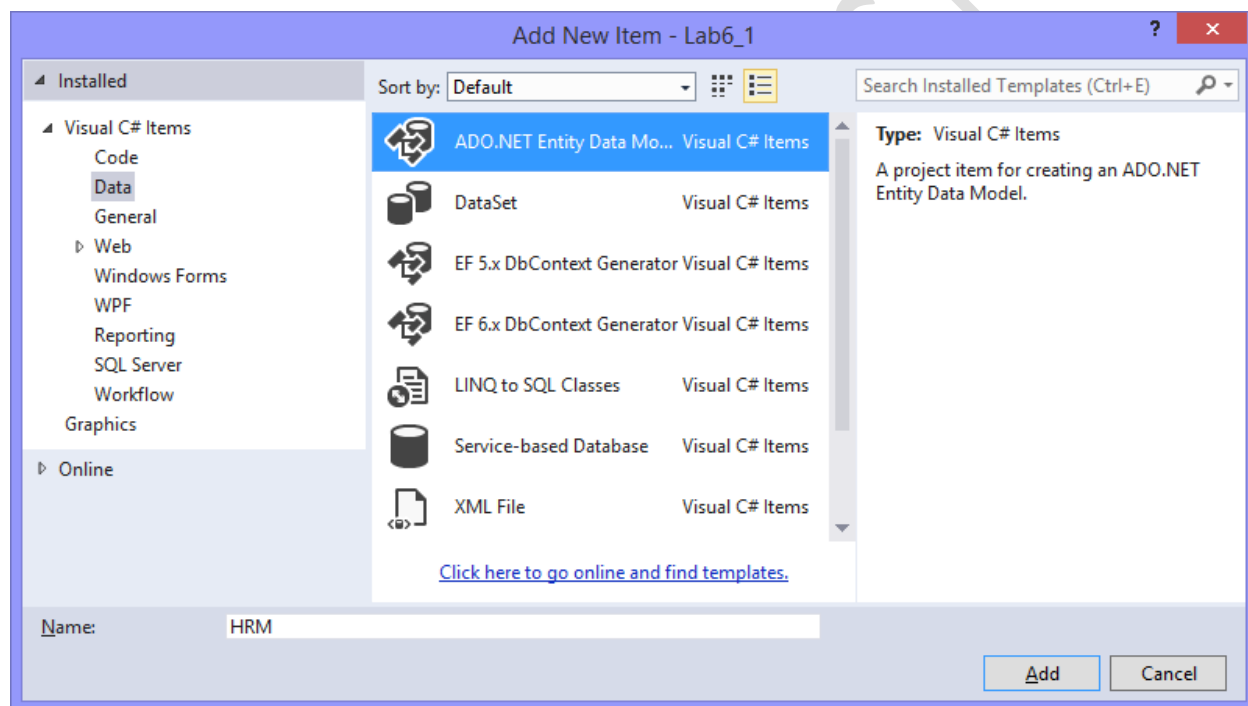
Bước 2: Thiết kế form theo đầu bài yêu cầu và đặt tên cho các control theo bảng sau (chú ý các control label không cần đặt tên):

Loại Control	Đặt tên	Thuộc tính khác
Màn hình: Form	frmEmployee	Text:Thêm mới nhân viên
Mã số: TextBox	txtId	
Họ: TextBox	txtFirstName	
Tên: TextBox	txtLastName	
Ngày sinh: DateTimePicker	txtBirthday	Format:Custom CustomFormat: dd/MM/yyyy
Giới tính: CheckBox	chkSex	Text: Male
Địa chỉ: TextBox	txtAddress	
Hòm thư: TextBox	txtEmail	
Mã phòng ban: ComboBox	cboDepartment	

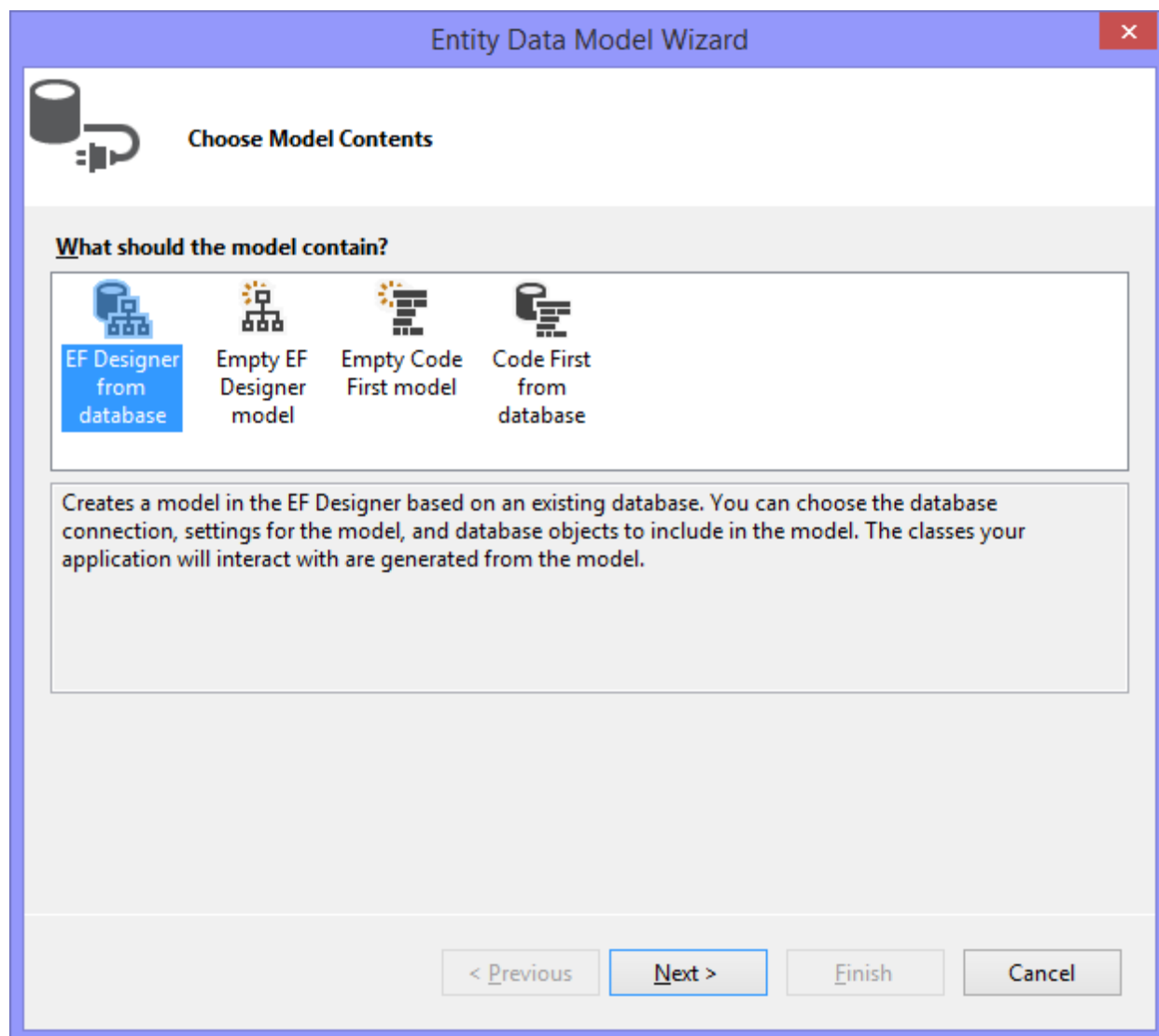
Tạo mới: Button	btnNew	Text: Tạo mới
Xóa: Button	btnDelete	Text: Xóa
Cập nhật: Button	btnUpdate	Text: Cập nhật
Bỏ qua: Button	btnCancel	Text: Bỏ qua
Thoát: Button	btnExit	Text: Thoát
Lưới: DataGridView	dgvEmployee	

Bước 3: Tạo ADO.NET Entity Data Model

- Kích chuột phải vào project “Lab6_1” -> Add -> New Item



- Chọn Add



- Chọn Next -> New Connection

Connection Properties

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:
CHUNGLD\SQLEXPRESS 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:
HRM

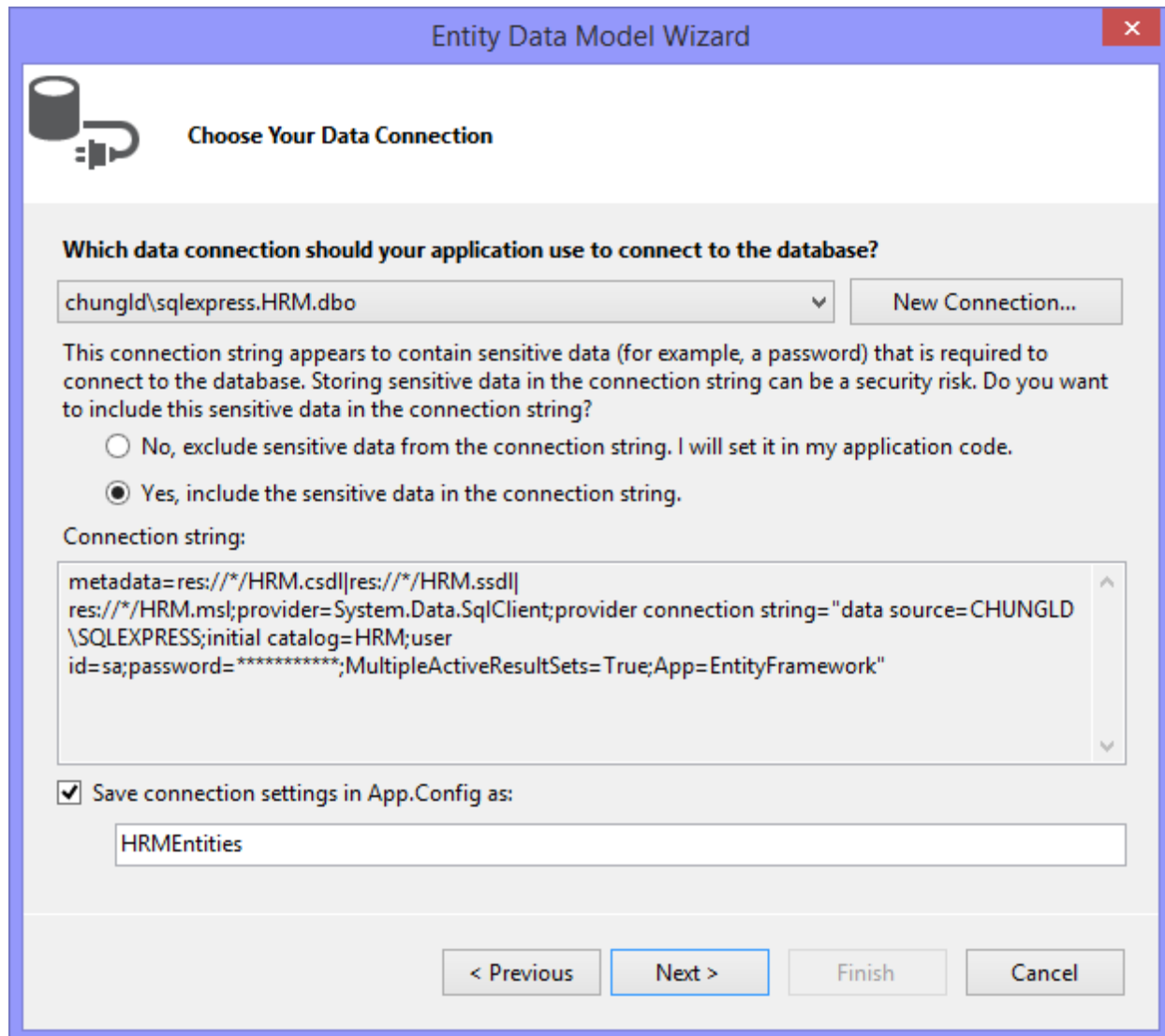
☐ Attach a database file:
Browse...

Logical name:

Advanced...

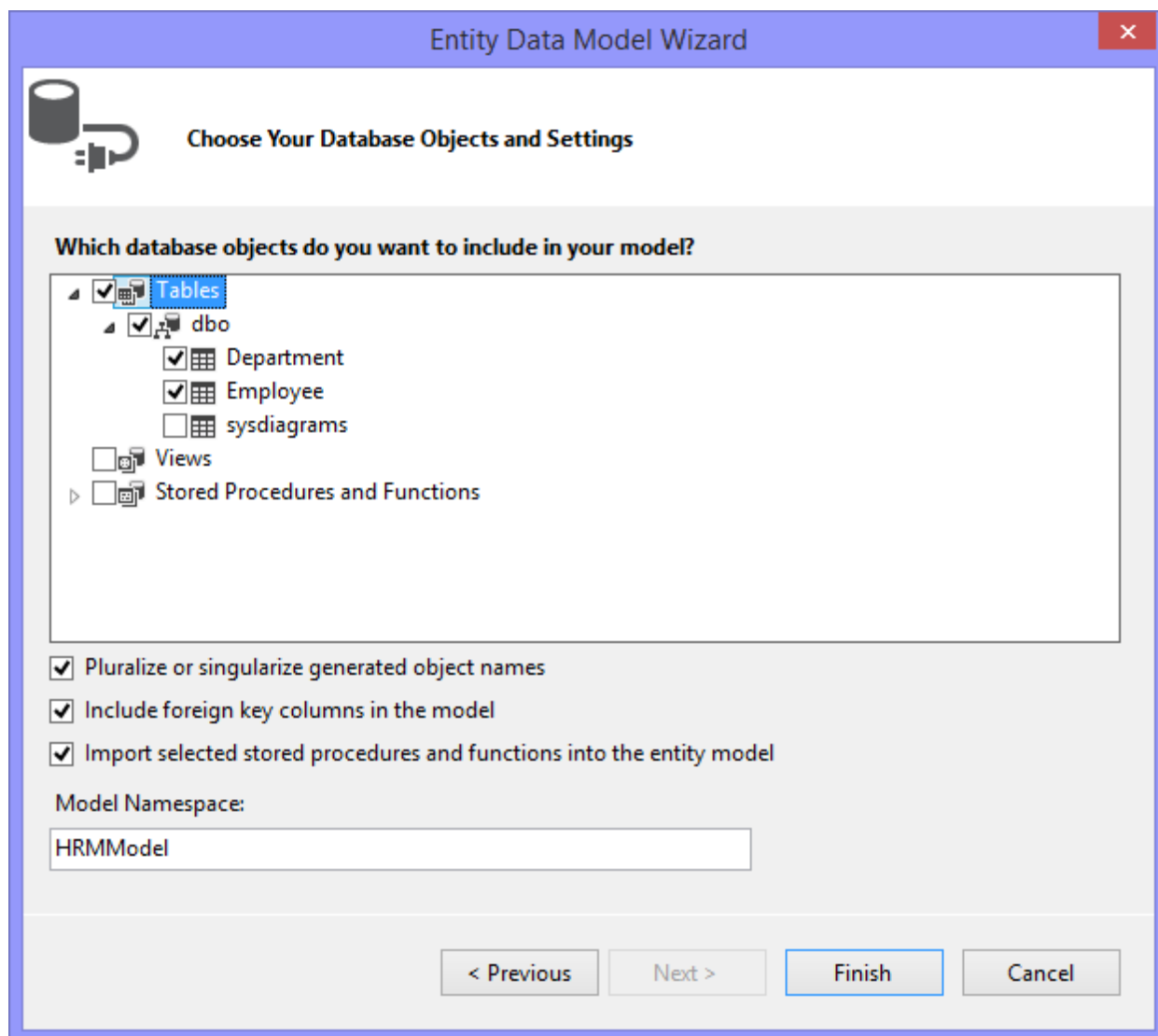
Test Connection OK Cancel

- OK

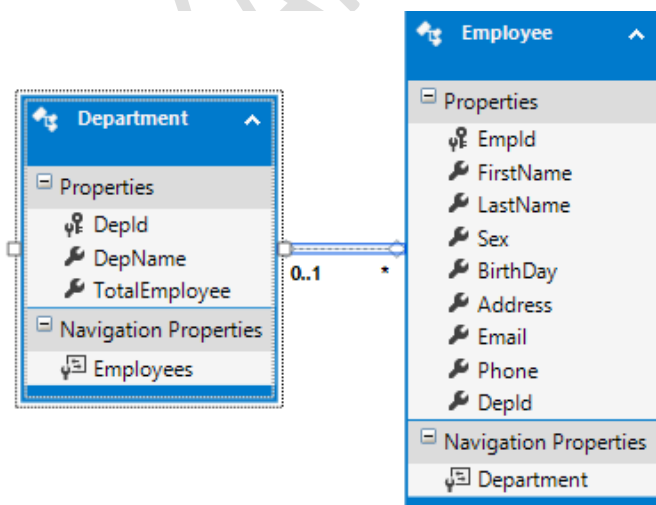


The image shows the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. The title bar reads 'Entity Data Model Wizard'. The main heading is 'Choose Your Data Connection'. Below this, a question asks: 'Which data connection should your application use to connect to the database?'. A dropdown menu shows 'chungld\sqlexpress.HRM.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.' (unselected) and 'Yes, include the sensitive data in the connection string.' (selected). Below this, the 'Connection string:' is displayed in a text box with the following content: 'metadata=res://*/HRM.csdl|res://*/HRM.ssdl|res://*/HRM.msl;provider=System.Data.SqlClient;provider connection string="data source=CHUNGLD\SQLEXPRESS;initial catalog=HRM;user id=sa;password=*****;MultipleActiveResultSets=True;App=EntityFramework"'. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked, followed by a text box containing 'HRMEntities'. Navigation buttons at the bottom include '< Previous', 'Next >' (highlighted), 'Finish', and 'Cancel'.

- Chọn Next -> Next



- Chọn Finish



Bước 4: Code cho chương trình:

- Khai báo biến

```
//khai báo biến chứa dòng hiện tại
int position;
//khai báo biến lưu trạng thái thêm hay sửa
bool edit = true;
//Khởi tạo đối tượng quản lý Database (DataContext)
HRMEntities hrm = new HRMEntities();
```

- Sự kiện Load của frmEmployee

```
private void frmEmployee_Load(object sender, EventArgs e)
{
    //Hiển thị nhân viên lên lưới
    DisplayEmployee();
    //Hiển thị phòng ban lên combobox
    DisplayDepartment();
}
```

- Sự kiện Click của nút Ghi

```
private void btnUpdate_Click(object sender, EventArgs e)
{
    if (edit)
    {
        //tìm nhân viên cần sửa có mã như trên form
        var emp = hrm.Employees.FirstOrDefault(x => x.EmpId == txtId.Text);
        //nếu tìm thấy
        if (emp != null)
```



```
{
    //gán lại thông tin cho nhân viên
    emp.FirstName = txtFirstName.Text;
    emp.LastName = txtLastName.Text;
    emp.BirthDay = txtBirthDay.Value;
    emp.Sex = chkSex.Checked;
    emp.Phone = txtPhone.Text;
    emp.Address = txtAddress.Text;
    emp.Email = txtEmail.Text;
    emp.DepId = cboDepartment.SelectedValue.ToString();
    //lưu
    hrm.SaveChanges();
    //hiển thị lại dữ liệu
    DisplayEmployee();
    //hiển thị đúng vị trí dòng đã chọn trước đó
    dgvEmployee.Rows[0].Selected = false;
    dgvEmployee.Rows[position].Selected = true;
}
else
{
    MessageBox.Show("Không tìm thấy dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
else
{
    //tạo mới nhân viên
    var emp = new Employee();
    //gán giá trị
    emp.EmpId = txtId.Text;
    emp.FirstName = txtFirstName.Text;
    emp.LastName = txtLastName.Text;
    emp.BirthDay = txtBirthDay.Value;
    emp.Sex = chkSex.Checked;
```

```
emp.Phone = txtPhone.Text;
emp.Address = txtAddress.Text;
emp.Email = txtEmail.Text;
emp.DepId = cboDepartment.SelectedValue.ToString();
hrm.Employees.Add(emp);
//lưu
hrm.SaveChanges();
//hiển thị lại dữ liệu
DisplayEmployee();
}
}
```

- Sự kiện Click của nút Thêm mới

```
private void btnNew_Click(object sender, EventArgs e)
{
    //xóa trắng dữ liệu trên form
    txtId.Text = txtFirstName.Text = txtLastName.Text = txtPhone.Text =
    txtAddress.Text = txtEmail.Text = "";
    txtId.Focus();
    edit = false;
    txtId.ReadOnly = false;
}
}
```

- Sự kiện Click của dgvEmployee

```
private void dgvEmployee_Click(object sender, EventArgs e)
{
    //hiển thị chi tiết nhân viên khi kích vào lưới
    DisplayEmployeeDetail();
}
}
```

- Sự kiện Click của nút xóa

```
private void btnDelete_Click(object sender, EventArgs e)
{
    if (dgvEmployee.CurrentRow != null)
```

```

{
    if (MessageBox.Show("Bạn có muốn xóa không?", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        //tìm nhân viên có mã như trên form
        var emp = hrm.Employees.FirstOrDefault(x => x.EmpId ==
            txtId.Text);
        if (emp != null)
        {
            //xóa dữ liệu
            hrm.Employees.Remove(emp);
            //lưu
            hrm.SaveChanges();
            //hiển thị lại dữ liệu
            DisplayEmployee();
        }
    }
}
else
{
    MessageBox.Show("Không tìm thấy dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}

```

- Sự kiện Click của nút Bỏ qua

```

private void btnCancel_Click(object sender, EventArgs e)
{
    //hiển thị lại chi tiết nhân viên
    DisplayEmployeeDetail();
}

```

- Một số các phương thức khác

```

//Phương thức hiển thị phòng ban lên combo box

```

```
private void DisplayDepartment()  
{  
    //lấy danh sách phòng ban  
    var departments = from dep in hrm.Departments select new { DepId =  
        dep.DepId, DepName = dep.DepName };  
    cboDepartment.DataSource = departments.ToList();  
    cboDepartment.DisplayMember = "DepName";  
    cboDepartment.ValueMember = "DepId";  
}  
//Phương thức hiển thị dữ liệu lên lưới  
private void DisplayEmployee()  
{  
    //truy vấn lấy các thông tin cần thiết trong bảng Employee  
    var employess = from emp in hrm.Employees  
        select new { Ma_nhan_vien = emp.EmpId, Ho_dem = emp.FirstName, Ten =  
            emp.LastName, Gioi_tinh = emp.Sex, Ngay_sinh = emp.BirthDay, Dia_chi =  
            emp.Address, Email = emp.Email, Dien_thoai = emp.Phone, Ma_phong = emp.DepId  
        };  
    //hiển thị lên lưới  
    dgvEmployee.DataSource = employess.ToList();  
    DisplayEmployeeDetail();  
}  
//phương thức hiển thị chi tiết nhân viên của dòng hiện tại trên lưới lên form  
private void DisplayEmployeeDetail()  
{  
    //nếu dòng hiện tại trên lưới khác null  
    if (dgvEmployee.CurrentRow != null)  
    {  
        DataGridViewRow row = dgvEmployee.CurrentRow;  
        txtId.Text = row.Cells[0].Value.ToString();  
        txtFirstName.Text = row.Cells[1].Value.ToString();  
        txtLastName.Text = row.Cells[2].Value.ToString();  
        chkSex.Checked = (bool)row.Cells[3].Value;  
        txtBirthDay.Value = (DateTime)row.Cells[4].Value;
```

```

        txtAddress.Text = row.Cells[5].Value.ToString();
        txtEmail.Text = row.Cells[6].Value.ToString();
        txtPhone.Text = row.Cells[7].Value.ToString();
        cboDepartment.SelectedValue = row.Cells[8].Value.ToString();
        position = dgvEmployee.CurrentRow.Index;
        edit = true;
        txtId.ReadOnly = true;
    }
}

```

Bước 5: F5 để chạy và kiểm thử

The screenshot shows a Windows Forms application titled "Quản lý nhân viên". The form includes the following fields and controls:

- Mã NV:** Text box containing "NV002".
- Họ đệm:** Text box containing "Nguyễn Văn".
- Tên:** Text box containing "Tùng".
- Ngày sinh:** Date picker showing "19/11/1983".
- Email:** Text box containing "tungnv@gmail.com".
- Giới tính:** Checkboxes for "Nam" (checked) and "Nữ".
- Địa chỉ:** Text box containing "Hà Nội".
- Điện thoại:** Text box containing "0985578567".
- Phòng ban:** Dropdown menu showing "Quản lý nhân sự".
- Buttons:** "Tạo mới", "Cập nhật", "Xóa", "Bỏ qua", and "Thoát".
- Table:** A table with 7 columns: "Ma_nhan_vien", "Ho_dem", "Ten", "Gioi_tinh", "Ngay_sinh", and "Dia_chi". It lists 7 employees, with the second row (NV002, Nguyễn Văn, Tùng) selected.

	Ma_nhan_vien	Ho_dem	Ten	Gioi_tinh	Ngay_sinh	Dia_chi
	NV001	Lê Thị	Thủy	False	12/23/1989	Hà Nội
▶	NV002	Nguyễn Văn	Tùng	True	11/19/1983	Hà Nội
	NV003	Hoàng	Hà	False	5/12/1987	Nghệ An
	NV004	Nguyễn Tiến	Hải	True	10/18/1982	Quảng Ninh
	NV005	Trần Văn	Trưởng	True	4/23/1983	Hà Nội
	NV006	Lê Huy	Long	True	11/29/1984	Hà Nam
	NV007	Nguyễn Quang	Huy	True	4/12/1983	Hà Nội

Bài 6.2

Phát triển tiếp bài Lab6_1 bổ sung thêm màn hình tra cứu nhân viên theo phòng ban và theo tên

- *Giao diện màn hình cập nhật thông tin nhân viên*

Bước 1: Mở frmEmployee bổ sung nút Tra cứu (btnSearch)

Bước 2: Thiết kế form theo đầu bài yêu cầu và đặt tên cho các control theo bảng sau (chú ý các control label không cần đặt tên):

Loại Control	Đặt tên	Thuộc tính khác
Màn hình: Form	frmSearch	Text: Tìm kiếm nhân viên
Nhập tên: TextBox	txtName	
Tìm: Button	btnSearch	Text: Tìm
Tên: TextBox	txtLastName	
SplitContainer	SplitContainer1	Chia làm 2 phần theo dạng cột, phần 1 chứa TreeView, phần 2 chứa textbox,

		button và ListView hiển thị kết quả tìm kiếm
ImageList	imglcon	ColorDepth: 32Bit ImageSize: 48,48 Chọn 4 các ảnh tương ứng
TreeView	trvDepartment	ImageList: imglcon Anchor: Top, Bottom, Left, Right
ListView	lstEmployee	View: Details Largelcon: imglcon SmallIcon: imglcon Anchor: : Top, Bottom, Left, Right Columns: Thêm các cột như hình minh họa

Bước 3: Code cho chương trình:

- Khai báo biến

```
//Khai báo biến DataContext
HRMEntities hrm = new HRMEntities ();
```

- Sự kiện Load của frmSearch

```
private void frmSearch_Load(object sender, EventArgs e)
{
    //lấy thông tin phòng ban
    var departments = from dep in hrm.Departments select new { DepId =
        dep.DepId, DepName = dep.DepName };
    //Tạo nút gốc
    TreeNode root = new TreeNode("Danh mục phòng ban", 0, 0);
    root.Tag = 0;
    //đọc dữ liệu và đưa lên TreeView
```

```
foreach (var dep in departments)
{
    //tạo nút con
    TreeNode depnode = new TreeNode(dep.DepName, 1, 1);
    depnode.Tag = dep.DepId;
    //đưa nút con vào nút gốc
    root.Nodes.Add(depnode);
}
//đưa nút gốc vào treeview
trvDepartment.Nodes.Add(root);
//mở hết các nút trên cây
trvDepartment.ExpandAll();
}
```

- Sự kiện *AfterSelect* của *TreeView*

```
private void trvDepartment_AfterSelect(object sender, TreeViewEventArgs e)
{
    //gọi phương thức tìm
    SearchEmployee();
}
```

- Sự kiện *Click* của *Tìm*

```
private void btnSearch_Click(object sender, EventArgs e)
{
    //gọi phương thức tìm
    SearchEmployee();
}
```

- Code cho phương thức tìm kiếm

```
public void SearchEmployee()
{
    //xóa ListView
    lstEmployee.Items.Clear();
}
```

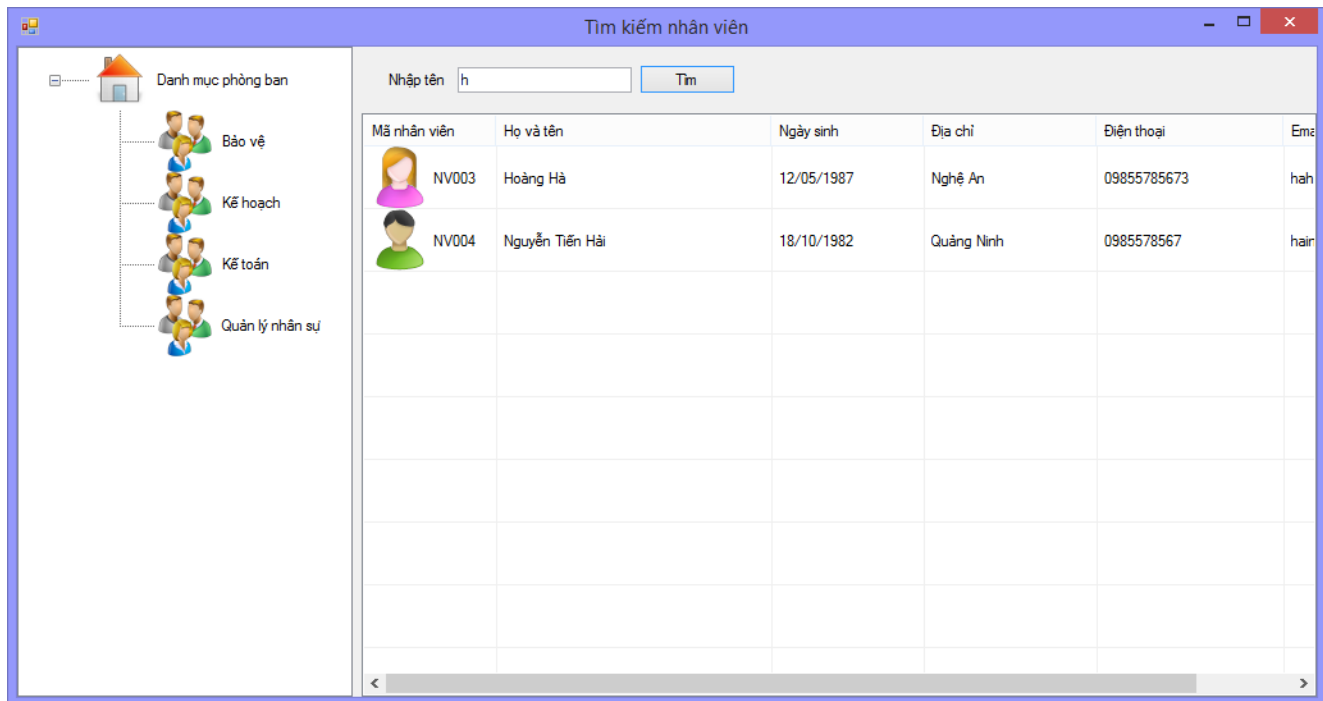


```
//lấy nút được chọn
TreeNode node =trvDepartment.SelectedNode;
//Tìm và hiển thị kết quả
//lấy nhân viên theo phòng được chọn
var employees = from emp in hrm.Employees
where emp.DepId == node.Tag.ToString()
&& emp.LastName.Contains(txtName.Text)
select new { Id = emp.EmpId, FullName = emp.FirstName + " " + emp.LastName,
Birthday = emp.BirthDay, Address = emp.Address, Email = emp.Email, Phone =
emp.Phone, Sex = emp.Sex };
//duyet và hiển thị lên ListView
foreach (var emp in employees)
{
    ListViewItem item = new ListViewItem(new string[] { emp.Id,
emp.FullName, emp.Birthday.Value.ToString("dd/MM/yyyy"), emp.Address,
emp.Phone, emp.Email });
    item.ImageIndex = (emp.Sex.Value) ? 2 : 3;
    lstEmployee.Items.Add(item);
}
}
```

Bước 4: Code cho nút tra cứu của frmEmployee

```
private void btnSearch_Click(object sender, EventArgs e)
{
    frmSearch fs = new frmSearch();
    fs.ShowDialog();
}
```

Bước 5: F5 để chạy và kiểm thử



Bài 6.3

Viết chương trình minh họa hiển thị dữ liệu phân cấp trong database lên treeview và thực hiện các thao tác thêm xóa sửa trên cây.

- Cho cơ sở dữ liệu LabWinform với bảng sau

Person			
	Column Name	Data Type	Allow Nulls
🔑	PersonId	int	<input type="checkbox"/>
	PersonName	nvarchar(100)	<input checked="" type="checkbox"/>
	ParentId	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- Code script mẫu

```
Create table Person
(
    PersonId int identity primary key,
    PersonName nvarchar(100),
```

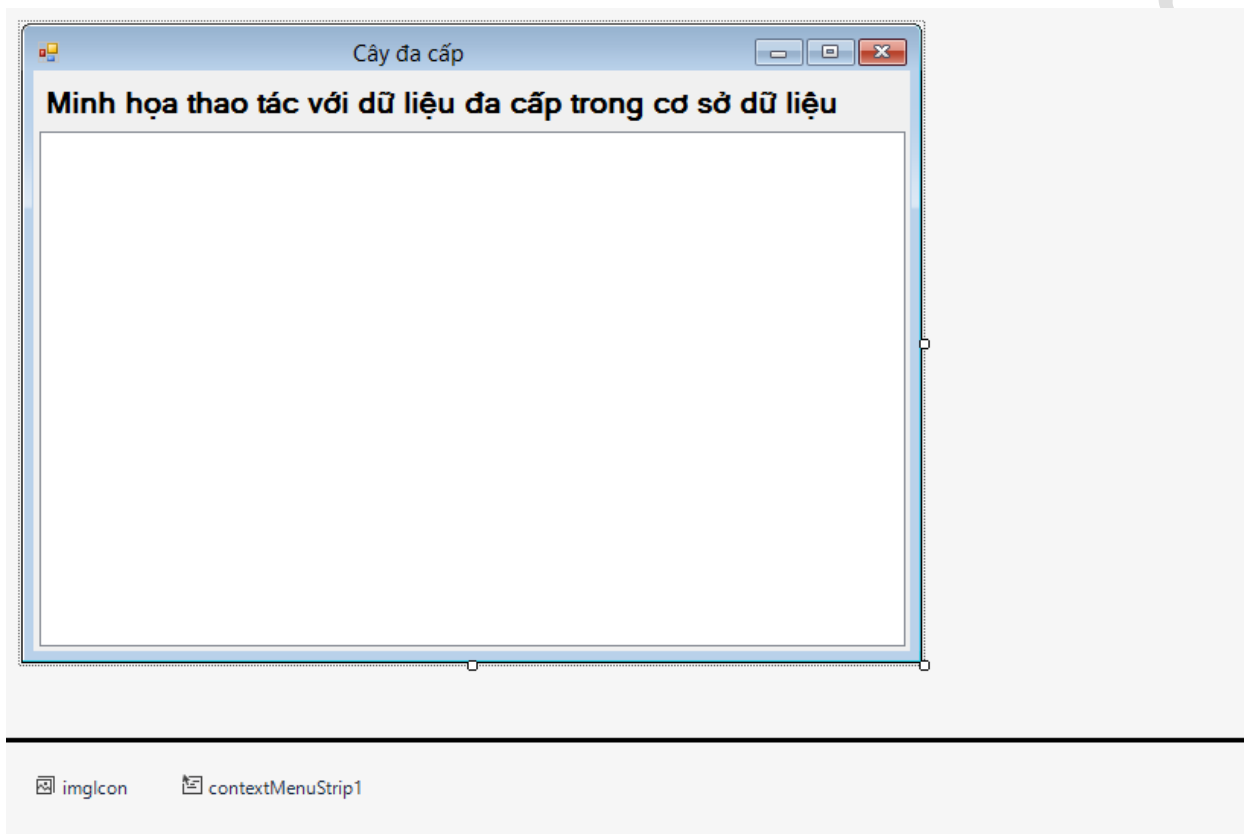
```
        ParentId int
    )
go
insert into Person values(N'Nguyễn Văn A',0)
insert into Person values(N'Nguyễn Văn B',1)
insert into Person values(N'Nguyễn Văn C',1)
insert into Person values(N'Nguyễn Văn D',2)
insert into Person values(N'Nguyễn Văn E',2)
insert into Person values(N'Nguyễn Văn F',2)
insert into Person values(N'Nguyễn Văn G',6)
insert into Person values(N'Nguyễn Văn A1',0)
go
--thủ tục thêm 1 Person
Create proc AddPerson
@PersonName nvarchar(100),
@ParentId int,
@Id int output
as
    insert into Person values(@PersonName, @ParentId)
    set @Id=@@IDENTITY
go
--thủ tục sửa
Create proc UpdatePerson
@PersonId int,
@PersonName nvarchar(100)
as
    update Person set PersonName=@PersonName where PersonId=@PersonId
go
--thủ tục xóa
Create proc DeletePerson
@PersonId int
as
    if(exists(select PersonId from Person where ParentId=@PersonId))
    begin
```

```

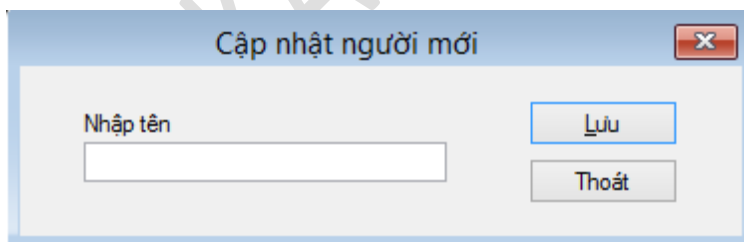
        raiserror (N'Không xóa được',16,1)
    end
    delete from Person where PersonId=@PersonId
go

```

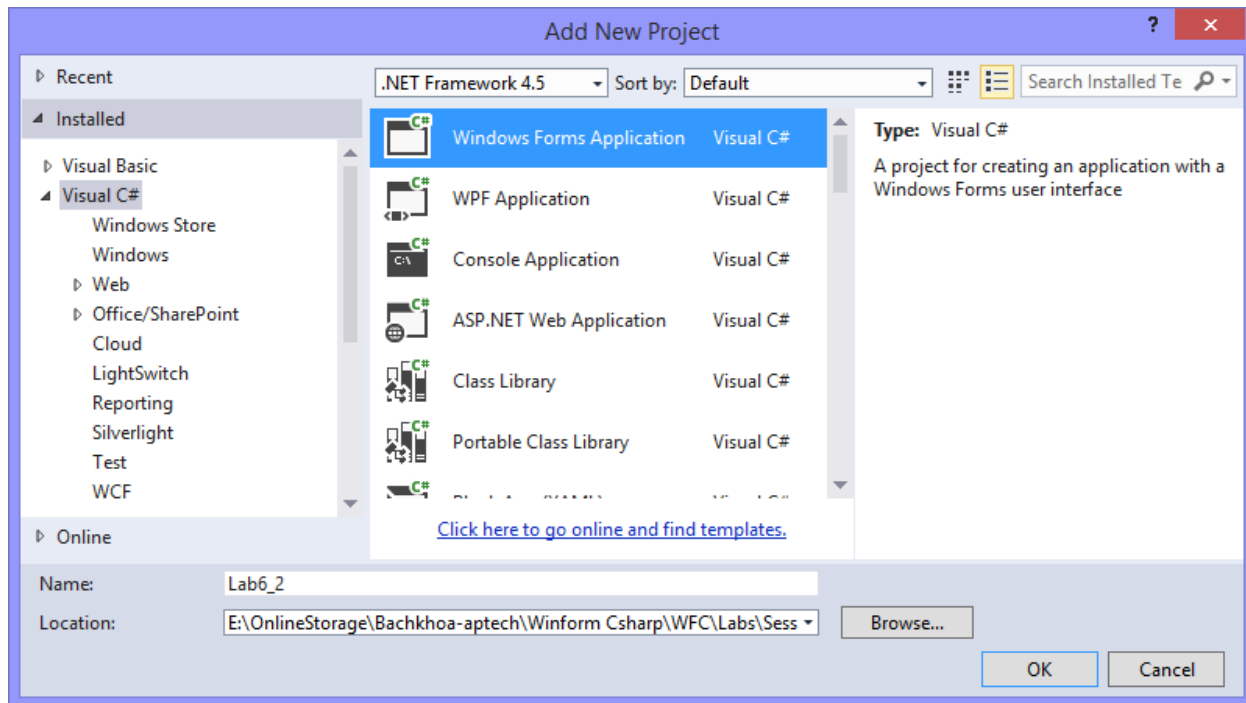
- Form hiển thị cây phân cấp thiết kế như sau



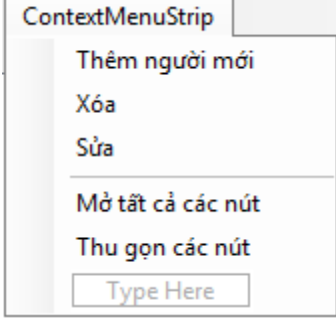
- Form thêm người mới



Bước 1: Kích chuột phải vào Solution “Session6” Add -> New Project -> chọn loại Windows Forms Appliaction -> Nhập tên Project -> OK



Bước 2: Thiết kế form hiển thị cây phân cấp theo đầu bài yêu cầu và đặt tên cho các control theo bảng sau (chú ý các control label không cần đặt tên):

Loại Control	Đặt tên	Thuộc tính khác
Màn hình: Form	frmMultiLevel	Text:Cây đa cấp
ImageList	imglcon	Chọn 3 icon làm mẫu kích thước 32,32
ContextMenu	contextMenu1	
TreeView	trvPerson	ImageList:imglcon ContextMenuStrip: contextMenu1

Bước 3: Thiết kế form thêm người mới theo đầu bài yêu cầu và đặt tên cho các control theo bảng sau (chú ý các control label không cần đặt tên):

Loại Control	Đặt tên	Thuộc tính khác
Màn hình: Form	frmAddPerson	Text:Cây đa cấp AcceptButton: btnSave
Nhập tên: TextBox	txtName	
Lưu: Button	btnSave	
Thoát: Button	btnExit	

Bước 4: Code cho frmMultiLevel

- Khai báo biến

```
//Khởi tạo Entities
LabWinformEntities lab = new LabWinformEntities();
```

- Code cho sự kiện form Load

```
private void frmMultiLevel_Load(object sender, EventArgs e)
{
    //Tạo nút gốc
    TreeNode root = new TreeNode("Hệ thống đa cấp");
    //Mã số mặc định là 0
    root.Tag = 0;
    //Biểu tượng 0
    root.ImageIndex = 0;
    root.SelectedImageIndex = 0;
    //Thêm nút gốc vào cây
    trvPerson.Nodes.Add(root);
    //Gọi phương thức lấy nút con của nút root
    GetChildPerson(root);
}

//định nghĩa phương thức lấy các nút con của một nút
private void GetChildPerson(TreeNode rootnode)
{

```

```
//lấy danh sách các nút con của rootnode
var childs = from p in lab.People where p.ParentId == ((int)rootnode.Tag)
select p;
//duyệt các nút con để add vào rootnode
foreach (var child in childs)
{
    TreeNode childnode = new TreeNode(child.PersonName);
    childnode.Tag = child.PersonId;
    rootnode.Nodes.Add(childnode);
    childnode.ImageIndex = 1;
    childnode.SelectedImageIndex = 1;
}
}
```

- Sự kiện *AfterExpand* của *treeview*

```
//sau khi mở 1 nút trên cây
private void trvPerson_AfterExpand(object sender, TreeViewEventArgs e)
{
    //lấy nút được mở
    TreeNode selectednode = e.Node;
    //duyệt các nút con của nút vừa mở
    foreach (TreeNode node in selectednode.Nodes)
    {
        //xóa các nút con của nút mở
        node.Nodes.Clear();
        //gọi phương thức lấy các nút con và đưa vào node
        GetChildPerson(node);
        //hiển thị hình ảnh phù hợp
        if (node.Nodes.Count > 0)
        {
            node.ImageIndex = 1;
            node.SelectedImageIndex = 1;
        }
        else
    }
}
```

```
        {  
            node.ImageIndex = 2;  
            node.SelectedImageIndex = 2;  
        }  
    }  
}
```

- Sự kiện *BeforeCollapse* của *treeview*

//khi đóng nút lại thì đóng tất cả các nút con

```
private void trvPerson_BeforeCollapse(object sender, TreeViewCancelEventArgs e)  
{  
    foreach (TreeNode node in e.Node.Nodes)  
    {  
        node.Collapse();  
    }  
}
```

- Sự kiện *Click* của menu *Thêm mới*

//thêm nút mới

```
private void mnuAdd_Click(object sender, EventArgs e)  
{  
    //tạo form thêm  
    frmAddPerson frmadd = new frmAddPerson();  
    //gán nút chọn trên cây sang form thêm  
    frmadd.SelectedNode = trvPerson.SelectedNode;  
    //tắt trạng thái sửa  
    frmadd.Edit = false;  
    //hiển thị  
    frmadd.ShowDialog();  
}
```

- Sự kiện *Click* của menu *Sửa*

//sửa nút


```
private void mnuEdit_Click(object sender, EventArgs e)
{
    //nếu là nút gốc thì không cho sửa
    if (trvPerson.SelectedNode.Tag.ToString() == "0")
        return;
    //tạo form thêm
    frmAddPerson frmadd = new frmAddPerson();
    //chuyển nút chọn trên tree sang form thêm
    frmadd.SelectedNode = trvPerson.SelectedNode;
    //bật trạng thái sửa của form thêm
    frmadd.Edit = true;
    //hiển thị form
    frmadd.ShowDialog();
}
```

- Sự kiện Click của menu Xóa

```
//xóa nút
private void mnuDelete_Click(object sender, EventArgs e)
{
    //nếu nút hiện tại có nút con thì không xóa được
    if (trvPerson.SelectedNode.Nodes.Count > 0)
        MessageBox.Show("Không thể xóa nút cha khi các nút con chưa xóa",
            "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        //hỏi khi xóa
        if (MessageBox.Show("Bạn có muốn xóa không", "Thông báo",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
            DialogResult.Yes)
        {
            //nếu có 1 nút con thì đổi icon
            if (trvPerson.SelectedNode.Parent.Nodes.Count == 1)
            {
                trvPerson.SelectedNode.Parent.ImageIndex = 2;
            }
        }
    }
}
```

```

        trvPerson.SelectedNode.Parent.SelectedIndex = 2;
    }
    //lấy mã số cần xóa
    int personid = (int)trvPerson.SelectedNode.Tag;
    try
    {
        //xóa dữ liệu
        lab.DeletePerson(personid);
        //xóa nút
        trvPerson.SelectedNode.Remove();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
}
}

```

- Sự kiện Click của menu mở rộng tất cả

```

//mở tất cả các nút
private void mnuExpandAll_Click(object sender, EventArgs e)
{
    trvPerson.SelectedNode.ExpandAll();
}

```

- Sự kiện Click của menu thu lại tất cả

```

//đóng tất cả các nút
private void mnuCollapseAll_Click(object sender, EventArgs e)
{
    trvPerson.SelectedNode.Collapse();
}

```

Bước 5: Code cho frmAddPerson

- Khai báo biến

```
//khai báo thuộc tính Treenode chứa nút chọn bên frmMultiLevel
public TreeNode SelectedNode { get; set; }
//khai báo thuộc tính Edit chứa trạng thái sửa hay thêm mới
public bool Edit { get; set; }
```

- Sự kiện form Load

```
private void frmAddPerson_Load(object sender, EventArgs e)
{
    //nếu là sửa thì hiển thị tên lên ô text
    if (Edit)
        txtName.Text = SelectedNode.Text;
}
```

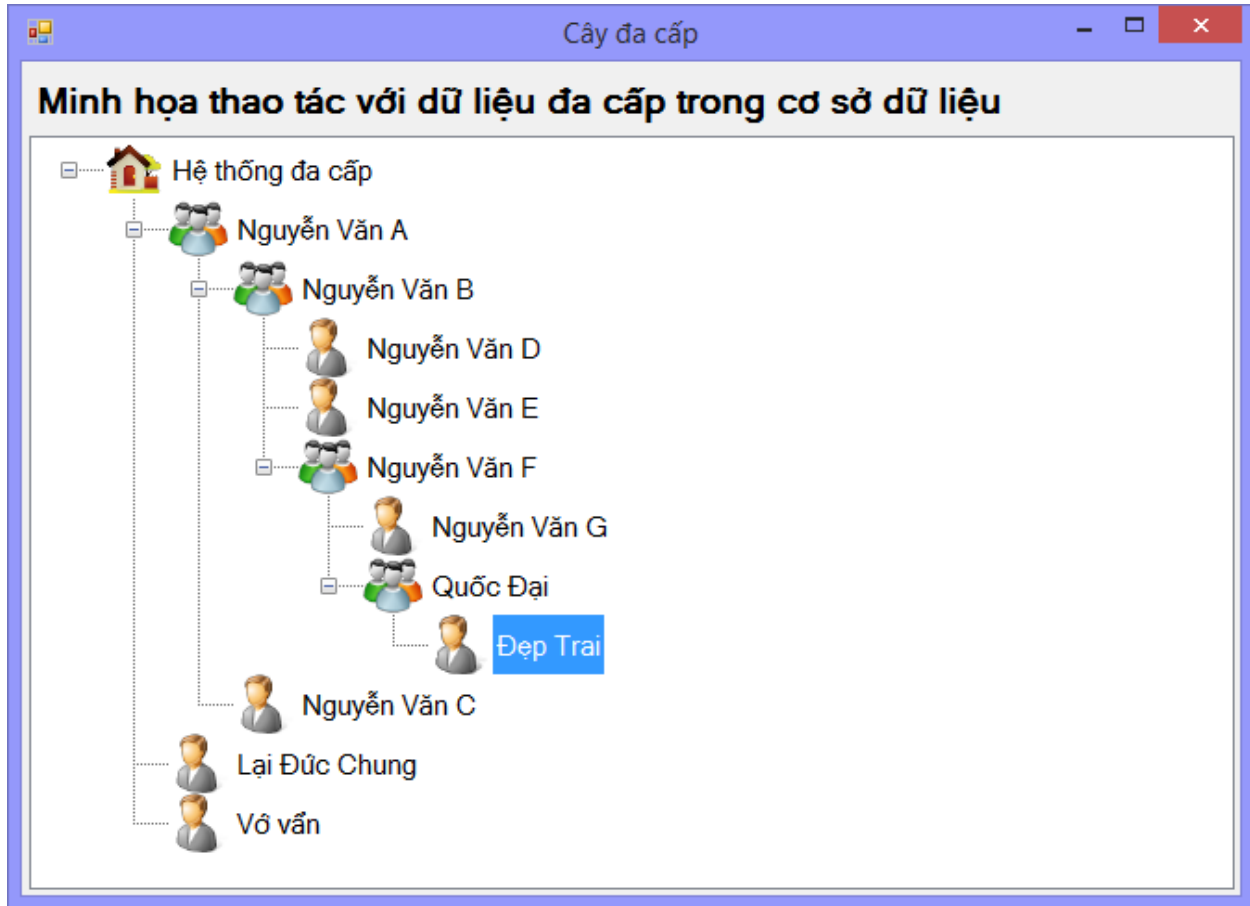
- Sự kiện Click của nút Lưu

```
private void btnSave_Click(object sender, EventArgs e)
{
    //khởi tạo entities
    LabWinformEntities lab = new LabWinformEntities();
    //tạo tham số id
    ObjectParameter id = new ObjectParameter("Id", typeof(int));
    //nếu là sửa
    if (Edit)
    {
        //cập nhật lại dữ liệu vào database
        lab.UpdatePerson((int)SelectedNode.Tag, txtName.Text);
        //hiển thị lại nhãn trên nút của cây
        SelectedNode.Text = txtName.Text;
    }
    else //nếu là thêm
    {

```

```
//thêm mới dữ liệu
lab.AddPerson(txtName.Text, (int)SelectedNode.Tag, id);
//tạo nút mới
TreeNode node = new TreeNode(txtName.Text);
node.Tag = id.Value;
node.ImageIndex = 2;
node.SelectedImageIndex = 2;
//thêm vào nút được chọn
SelectedNode.Nodes.Add(node);
if (SelectedNode.Tag.ToString() != "0")
{
    SelectedNode.ImageIndex = 1;
    SelectedNode.SelectedImageIndex = 1;
}
}
this.Close();
}
```

Bước 6: F5 chạy và kiểm tra kết quả



Phần I: Bài tập tự làm

HẾT