

Nội dung thực hành

Ngôn ngữ: C#.NET

Môi trường lập trình: Visual studio .NET

Hệ quản trị CSDL: SQL Server.

Đánh giá:

Hình thức: Nộp đồ án (báo cáo + source code của phần mềm) + thi vấn đáp trực tiếp trên máy dựa trên các đề tài mà giáo viên lý thuyết đề nghị hoặc các em tự đề xuất.

Tiêu chí: Phần mềm của các bạn phải thỏa các tính chất sau:

1. Tính đúng đắn
2. Tính tiện dụng
3. Tính hiệu quả
4. Tính tiến hóa
5. Tính tương thích
6. Tính bảo mật
7. Tính an toàn

Nội dung:

1. Thiết kế dữ liệu: Tạo bảng, khóa, quan hệ.
2. Thiết kế giao diện
 - a. Màn hình chính + thực đơn
 - b. Màn hình đăng nhập, thay đổi qui định
 - c. Màn hình nhập (Danh mục, 1- Nhiều và Nhiều - nhiều)
 - d. Màn hình tra cứu
3. Kết nối CSDL
4. Debug Chương trình.
5. Cài đặt các hàm xử lý
 - a. Thêm, Xóa, Cập nhật và Lấy dữ liệu.
 - b. Tính toán theo nghiệp vụ bài toán.
6. Tạo report (Crystal report)
7. Tạo help
8. Đóng gói chương trình.

Thời gian thực hành:

- Trong 3 hoặc 4 tuần đầu giáo viên hướng dẫn sinh viên làm bài tập phần căn bản.
- Trong 7 hoặc 8 tuần tiếp theo giáo viên hướng dẫn sinh viên xây dựng phần mềm quản lý thư viện (các phần tương tự sinh viên tự thực hiện).

- Tuần cuối cùng sinh viên nộp bài và giáo viên hỏi vấn đáp trực tiếp.

Phần 1. bài tập căn bản

Bài 1. Viết chương trình theo yêu cầu sau:

Thực hiện các yêu cầu sau:

- Khi nhấn nút “Tính”, chương trình sẽ giải phương trình bậc 2, và hiển thị kết quả trong textbox kết quả X1 hoặc X2.
- Khi nhấn nút “Bỏ qua” các textbox sẽ được xóa trắng, con trỏ sẽ chuyển về textbox A.
- Khi nhấn nút “Thoát” chương trình sẽ kết thúc.

Bài 2. Thiết kế form tìm ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của 2 số A và B dưới đây:

Thực hiện các yêu cầu sau:

- Viết hàm (Nút **Tìm**) tìm USCLN và BSCNN của 2 số a và b. Nếu ta chọn Option USCLN thì xuất ra kết quả ước số chung lớn nhất hay ngược lại là bội số chung nhỏ nhất.
- Nút bỏ qua dùng để xóa rỗng các Textbox.

Bài 3. Thiết kế form để thực hiện liệt kê tất cả các số nguyên tố có giá trị nhỏ hơn N (số nguyên dương) và tính tổng cộng các số nguyên tố đã tìm được dưới đây:

Thực hiện các yêu cầu sau:

- Viết hàm (Nút **Liệt kê**) liệt kê tất cả các số nguyên tố có giá trị nhỏ hơn N và tính tổng cộng các số nguyên tố đã tìm được.
- Nút bỏ qua “Bỏ qua” dùng để xóa rỗng textbox nhập N .
- Nút thoát khỏi form.

Bài 4. Xây dựng chương trình thao tác với Listbox như sau :

Yêu cầu:

- Chỉ cho nhập số vào textbox **Nhập số**

- Khi nhập dữ liệu vào textbox **Nhập số** và nhấn Enter(hoặc click vào **Cập Nhật**) thì số mới nhập này được thêm vào Listbox đồng thời dữ liệu trong textbox bị xóa và focus được chuyển về textbox
- Khi nhấn vào các phím chức năng, yêu cầu sẽ được thực hiện trên listbox

Bài 5 Xây dựng chương trình nhập danh sách sinh viên theo yêu cầu:

Yêu cầu:

- Nút > dùng để di chuyển các mục(còn gọi là item) được chọn từ listbox trái qua listbox phải và ngược lại cho <
- Nút >> dùng để di chuyển tất cả các mục(kể cả không được chọn) từ listbox trái qua listbox phải và ngược lại cho <<
- Nút **Xóa** dùng để xóa những item được chọn trên textbox trái

Bài 6. Xây dựng chương trình thao tác với mảng

The screenshot shows a Windows application window titled "Xử lý mảng". It contains two large text boxes at the top: "Mảng nhập" (Input Array) which contains the text "0 5 6 9 14", and "Kết quả tính toán" (Calculation Results) which is empty. Below these text boxes are six buttons arranged in a 3x2 grid: "Nhập Mảng" (Input Array), "Đếm phần tử chẵn" (Count Even Elements), "Tính tổng mảng" (Calculate Array Sum), "Xuất các phần tử lẻ" (Output Odd Elements), "Tìm Max-Min của mảng" (Find Max-Min of Array), and "Xuất các số nguyên tố" (Output Prime Numbers). At the bottom center of the window is a button labeled "Kết thúc" (End).

Yêu cầu:

- Việc nhập giá trị cho các phần tử bằng hàm InputBox
`InputBox(lời nhắc nhập, tiêu đề, giá trị mặc nhiên)` => **trả về chuỗi**

The screenshot shows a small dialog box titled "Nhập Mảng". It contains a label "Nhập giá trị phần tử:" (Enter element value:), an "OK" button, a "Cancel" button, and a text input field at the bottom containing the number "5".

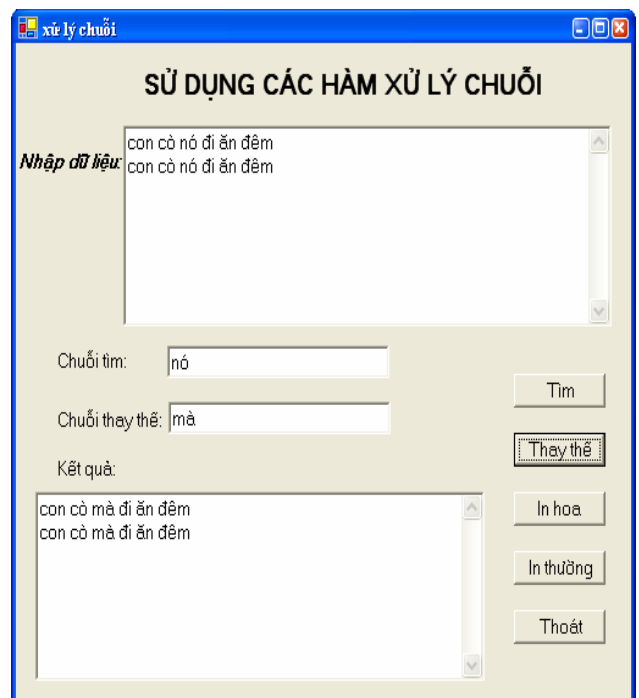
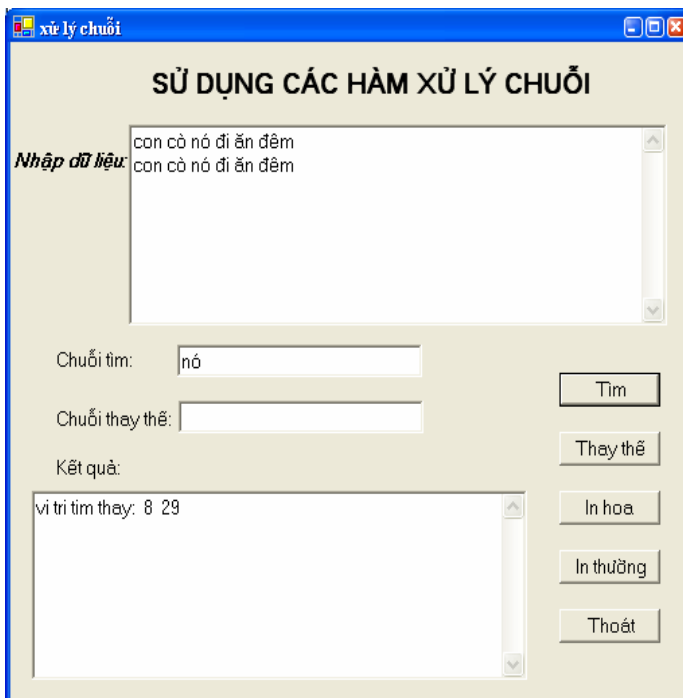
- 2 textbox để hiển thị mảng và kết quả, ko được nhập giá trị trực tiếp

Bài 7. Xây dựng chương trình xử lý chuỗi sau:



Yêu cầu:

- Khi vừa mở form chỉ có nút **Thoát** là sử dụng được, textbox nhập dữ liệu nhận focus
- Khi textbox nhập dữ liệu vừa được nhập thì các nút sáng lên, đồng thời nếu xóa hết dữ liệu trong textbox nhập thì các nút cũng mờ đi ngoại trừ nút **Thoát**=> viết trong sự kiện **change** của textbox
- Khi các nút chức năng được nhấn thì kết quả sẽ hiển thị bên textbox **Kết quả**



Hướng dẫn:

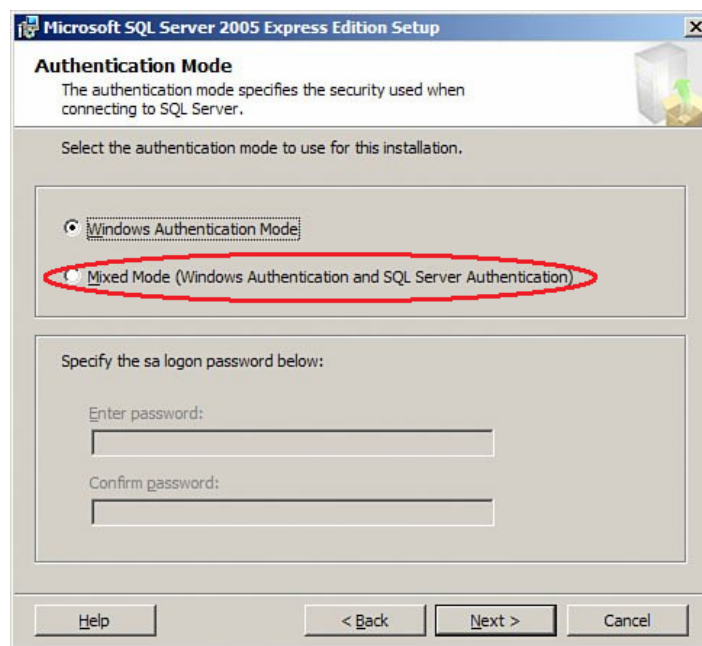
Tìm kiếm : Instr(vị trí bắt đầu tìm, chuỗi chứa, chuỗi cần tìm, phân biệt hoa thường hay ko)

Thay thế : Replace(chuỗi chứa, chuỗi tìm, chuỗi thay thế, vị trí bắt đầu, số lần tìm và thay thế, phân biệt hoa thường)
Trả về chuỗi mới đã được thay thế

Phần 2. Làm việc với cơ sở dữ liệu

Bài 1. Làm việc với Microsoft SQL Server 2005

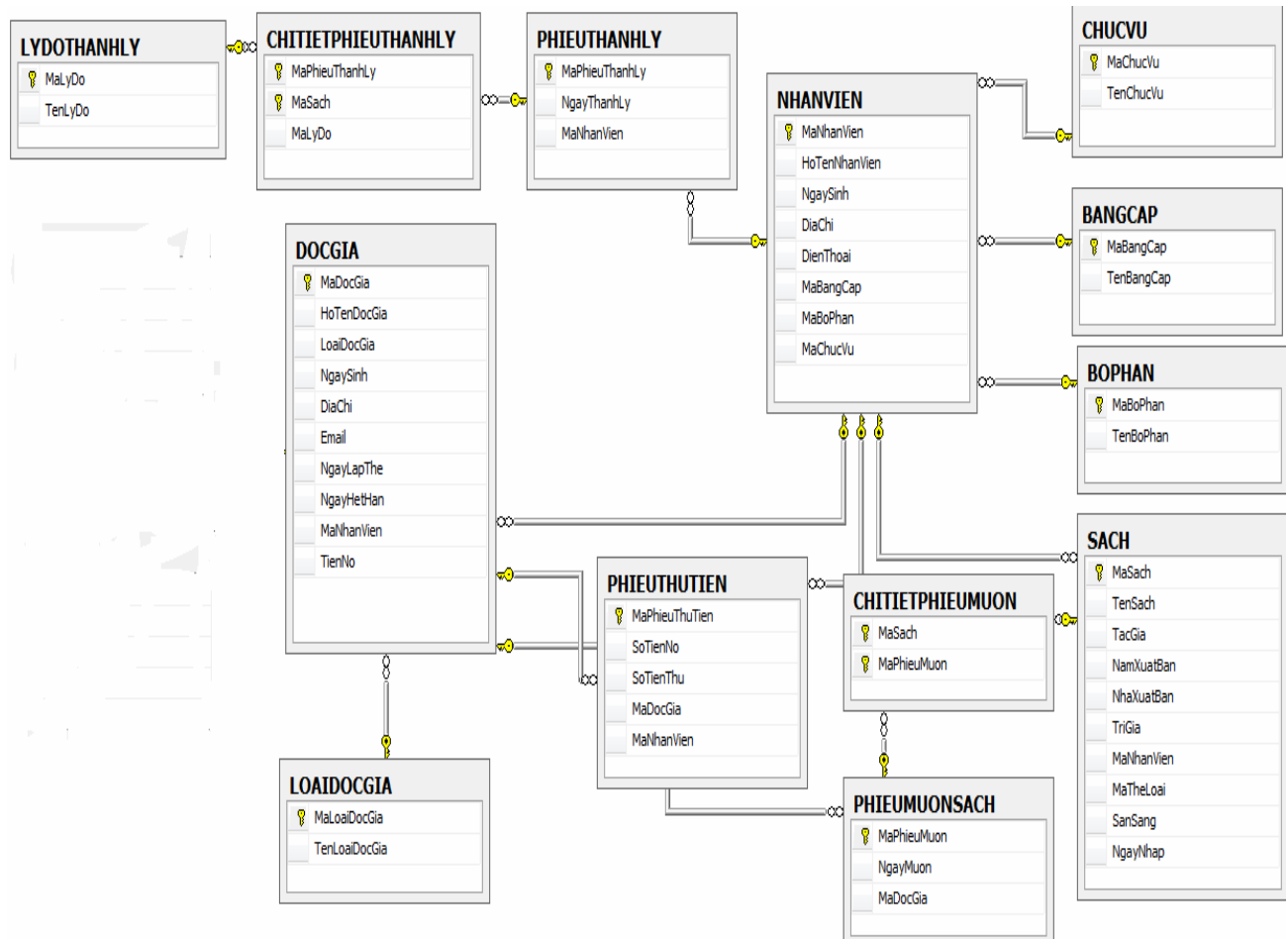
- Lưu ý:
 - Để cài đặt SQL Server 2005 chạy trên HĐH Window XP các bạn mua version SQL Server 2005 Developer Edition hoặc sử dụng version SQL Server 2005 Express Edition kèm theo đĩa cài đặt Visual Studio 2005.
 - Để có thể đăng nhập vào SQL Server bằng 2 cách Windows Authentication và SQL Server Authentication các bạn phải chọn chế độ Mixed Mode khi cài đặt.



- Các bạn xem file “Sams Teach Yourself Microsoft® SQL Server™ 2005 Express in 24 Hours.chm” để học cách cài đặt cũng như làm việc với SQL Server 2005.

1. Tạo Database

- Mở “SQL Server Management Studio” & Kết nối SQL Server: Khi cài đặt các bạn chọn Mixed Mode thì bây giờ các bạn có thể “Connect to Server” bằng 2 cách: Windows Authentication và SQL Server Authentication. Bên dưới là kết nối bằng SQL Server Authentication.



```

CREATE DATABASE ThuVien
Go
USE ThuVien
GO
--Tạo bảng Bộ phận
CREATE TABLE [BOPHAN] (
    [MaBoPhan] [int] NOT NULL,
    [TenBoPhan] [nvarchar] (40) NULL,
    CONSTRAINT [PK_BOPHAN] PRIMARY KEY (MaBoPhan)
)
GO
--Tạo bảng Loại độc giả
CREATE TABLE [LOAIDOCGIA] (
    [MaLoaiDocGia] [int] NOT NULL,
    [TenLoaiDocGia] [nvarchar] (50) NULL,
    CONSTRAINT [PK_DOCGIA] PRIMARY KEY (MaLoaiDocGia)
)
GO
--Tạo bảng Thẻ loại sách
CREATE TABLE [THELOAISACH] (
    [MaTheLoai] [int] NOT NULL,
    [TenTheLoai] [nvarchar] (40) NULL,
    CONSTRAINT [PK_THELOAISACH] PRIMARY KEY (MaTheLoai)
)
GO
--Tạo bảng Sách mượn
CREATE TABLE [SACHMAT] (
    [MaSach] [int] NOT NULL,
    [NgayMat] [datetime] NULL,
    [TienPhatMatSach] [float] NULL,
    [MaDocGia] [int] NULL,
    [MaNhanVien] [int] NULL,
    CONSTRAINT [PK_SACHMAT] PRIMARY KEY (MaSach)
  
```



```

)
GO
--Tạo bảng Tham số
CREATE TABLE [THAMSO] (
    [TenThamSo] [nvarchar] (40) NOT NULL,
    [GiaTri] [int] NULL,
    CONSTRAINT [PK_THAMSO] PRIMARY KEY (TenThamSo)
)
GO
--Tạo bảng Lý do thanh lý
CREATE TABLE [LYDOTHANHLY] (
    [MaLyDo] [int] NOT NULL,
    [TenLyDo] [nvarchar] (20) NULL,
    CONSTRAINT [PK_LYDOTHANHLY] PRIMARY KEY (MaLyDo)
)
GO
--Tạo bảng Chức vụ
CREATE TABLE [CHUCVU] (
    [MaChucVu] [int] NOT NULL,
    [TenChucVu] [nvarchar] (50) NULL,
    CONSTRAINT [PK_CHUCVU] PRIMARY KEY (MaChucVu)
)
GO
--Tạo bảng Bảng cấp
CREATE TABLE [BANGCAP] (
    [MaBangCap] [int] NOT NULL,
    [TenBangCap] [nvarchar] (40) NULL,
    CONSTRAINT [PK_BANGCAP] PRIMARY KEY (MaBangCap)
)
GO
--Tạo bảng Nhân viên
CREATE TABLE [NHANVIEN] (
    [MaNhanVien] [int] NOT NULL,
    [HoTenNhanVien] [nvarchar] (50) NULL,
    [NgaySinh] [datetime] NULL,
    [DiaChi] [nvarchar] (50) NULL,
    [DienThoai] [nvarchar] (15) NULL,
    [MaBangCap] [int] NULL,
    [MaBoPhan] [int] NULL,
    [MaChucVu] [int] NULL,
    CONSTRAINT [PK_NHANVIEN] PRIMARY KEY (MaNhanVien)
)
GO
--Tạo bảng Độc giả
CREATE TABLE [DOCGIA] (
    [MaDocGia] [int] NOT NULL,
    [HoTenDocGia] [nvarchar] (40) NULL,
    [LoaiDocGia] [int] NULL,
    [NgaySinh] [datetime] NULL,
    [DiaChi] [nvarchar] (50) NULL,
    [Email] [nvarchar] (30) NULL,
    [NgayLapThe] [datetime] NULL,
    [NgayHetHan] [datetime] NULL,
    [MaNhanVien] [int] NULL,
    [TienNo] [float] NULL,
    CONSTRAINT [PK_DOCGIA_1] PRIMARY KEY (MaDocGia)
)
GO
--Tạo bảng Chi tiết phiếu thanh lý
CREATE TABLE [CHITIETPHIEUTHANHLY] (
    [MaPhieuThanhLy] [int] NOT NULL,
    [MaSach] [int] NOT NULL,
    [MaLyDo] [int] NULL,
    CONSTRAINT [PK_CHITIETPHIEUTHANHLY] PRIMARY KEY (MaPhieuThanhLy, MaSach)
)

```

```

)
GO
--Tạo bảng Phiếu thu tiền
CREATE TABLE [PHIEUTHUTIENTIEN] (
    [MaPhieuThuTien] [int] NOT NULL,
    [SoTienNo] [float] NULL,
    [SoTienThu] [float] NULL,
    [MaDocGia] [int] NULL,
    [MaNhanVien] [int] NULL,
    CONSTRAINT [PK_PHIEUTHUTIENTIEN] PRIMARY KEY (MaPhieuThuTien)
)
GO
--Tạo bảng Sách
CREATE TABLE [SACH] (
    [MaSach] [int] NOT NULL,
    [TenSach] [nvarchar](40) NULL,
    [TacGia] [nvarchar](30) NULL,
    [NamXuatBan] [int] NULL,
    [NhaXuatBan] [nvarchar](40) NULL,
    [TriGia] [float] NULL,
    [MaNhanVien] [int] NULL,
    [MaTheLoai] [int] NULL,
    [SanSang] [bit] NULL,
    [NgayNhap] [datetime] NULL,
    CONSTRAINT [PK_SACH] PRIMARY KEY (MaSach)
)
GO
--Tạo bảng Phiếu thanh lý
CREATE TABLE [PHIEUTHANHLY] (
    [MaPhieuThanhLy] [int] NOT NULL,
    [NgayThanhLy] [datetime] NULL,
    [MaNhanVien] [int] NULL,
    CONSTRAINT [PK_PHIEUTHANHLY] PRIMARY KEY (MaPhieuThanhLy)
)
GO
--Tạo bảng Phiếu mượn sách
CREATE TABLE [PHIEUMUONSACH] (
    [MaPhieuMuon] [int] NOT NULL,
    [NgayMuon] [datetime] NOT NULL,
    [MaDocGia] [int] NULL,
    CONSTRAINT [PK_PHIEUMUONSACH] PRIMARY KEY (MaPhieuMuon)
)
--Tạo bảng Chi tiết phiếu mượn
CREATE TABLE [CHITIETPHIEUMUON] (
    [MaSach] [int] NOT NULL,
    [MaPhieuMuon] [int] NOT NULL,
    CONSTRAINT [PK_CHITIETPHIEUMUON] PRIMARY KEY (MaSach, MaPhieuMuon)
)
GO
--Tạo khoá ngoại
GO
ALTER TABLE [NHANVIEN] WITH NOCHECK ADD CONSTRAINT [FK_NHANVIEN_BANGCAP] FOREIGN
KEY([MaBangCap])
REFERENCES [BANGCAP] ([MaBangCap])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [NHANVIEN] CHECK CONSTRAINT [FK_NHANVIEN_BANGCAP]
GO
ALTER TABLE [NHANVIEN] WITH NOCHECK ADD CONSTRAINT [FK_NHANVIEN_BOPHAN] FOREIGN
KEY([MaBoPhan])
REFERENCES [BOPHAN] ([MaBoPhan])
ON UPDATE CASCADE
ON DELETE CASCADE

```

```

GO
ALTER TABLE [NHANVIEN] CHECK CONSTRAINT [FK_NHANVIEN_BOPHAN]
GO
ALTER TABLE [NHANVIEN] WITH NOCHECK ADD CONSTRAINT [FK_NHANVIEN_CHUCVU] FOREIGN
KEY([MaChucVu])
REFERENCES [CHUCVU] ([MaChucVu])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [NHANVIEN] CHECK CONSTRAINT [FK_NHANVIEN_CHUCVU]
GO
ALTER TABLE [DOCGIA] WITH NOCHECK ADD CONSTRAINT [FK_DOCGIA_LOAIDOCGIA] FOREIGN
KEY([LoaiDocGia])
REFERENCES [LOAIDOCGIA] ([MaLoaiDocGia])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [DOCGIA] CHECK CONSTRAINT [FK_DOCGIA_LOAIDOCGIA]
GO
ALTER TABLE [DOCGIA] WITH CHECK ADD CONSTRAINT [FK_DOCGIA_NHANVIEN] FOREIGN
KEY([MaNhanVien])
REFERENCES [NHANVIEN] ([MaNhanVien])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [DOCGIA] CHECK CONSTRAINT [FK_DOCGIA_NHANVIEN]
GO
ALTER TABLE [CHITIETPHIEUTHANHLY] WITH CHECK ADD CONSTRAINT
[FK_CHITIETPHIEUTHANHLY_LYDOTHANHLY] FOREIGN KEY([MaLyDo])
REFERENCES [LYDOTHANHLY] ([MaLyDo])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [CHITIETPHIEUTHANHLY] CHECK CONSTRAINT
[FK_CHITIETPHIEUTHANHLY_LYDOTHANHLY]
GO
ALTER TABLE [CHITIETPHIEUTHANHLY] WITH CHECK ADD CONSTRAINT
[FK_CHITIETPHIEUTHANHLY_PHIEUTHANHLY] FOREIGN KEY([MaPhieuThanhLy])
REFERENCES [PHIEUTHANHLY] ([MaPhieuThanhLy])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [CHITIETPHIEUTHANHLY] CHECK CONSTRAINT
[FK_CHITIETPHIEUTHANHLY_PHIEUTHANHLY]
GO
ALTER TABLE [PHIEUTHUTIEN] WITH CHECK ADD CONSTRAINT [FK_PHIEUTHUTIEN_DOCGIA]
FOREIGN KEY([MaDocGia])
REFERENCES [DOCGIA] ([MaDocGia])
GO
ALTER TABLE [PHIEUTHUTIEN] CHECK CONSTRAINT [FK_PHIEUTHUTIEN_DOCGIA]
GO
ALTER TABLE [PHIEUTHUTIEN] WITH CHECK ADD CONSTRAINT [FK_PHIEUTHUTIEN_NHANVIEN]
FOREIGN KEY([MaNhanVien])
REFERENCES [NHANVIEN] ([MaNhanVien])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [PHIEUTHUTIEN] CHECK CONSTRAINT [FK_PHIEUTHUTIEN_NHANVIEN]
GO
ALTER TABLE [SACH] WITH CHECK ADD CONSTRAINT [FK_SACH_NHANVIEN] FOREIGN
KEY([MaNhanVien])
REFERENCES [NHANVIEN] ([MaNhanVien])
ON UPDATE CASCADE
ON DELETE CASCADE

```

```

GO
ALTER TABLE [SACH] CHECK CONSTRAINT [FK_SACH_NHANVIEN]
GO
ALTER TABLE [PHIEUTHANHLY] WITH CHECK ADD CONSTRAINT [FK_PHIEUTHANHLY_NHANVIEN]
FOREIGN KEY ([MaNhanVien])
REFERENCES [NHANVIEN] ([MaNhanVien])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [PHIEUTHANHLY] CHECK CONSTRAINT [FK_PHIEUTHANHLY_NHANVIEN]
GO
ALTER TABLE [PHIEUMUONSACH] WITH CHECK ADD CONSTRAINT [FK_PHIEUMUONSACH_DOCGIA]
FOREIGN KEY ([MaDocGia])
REFERENCES [DOCGIA] ([MaDocGia])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [PHIEUMUONSACH] CHECK CONSTRAINT [FK_PHIEUMUONSACH_DOCGIA]
GO
ALTER TABLE [CHITIETPHIEUMUON] WITH CHECK ADD CONSTRAINT
[FK_CHITIETPHIEUMUON_PHIEUMUONSACH] FOREIGN KEY ([MaPhieuMuon])
REFERENCES [PHIEUMUONSACH] ([MaPhieuMuon])
GO
ALTER TABLE [CHITIETPHIEUMUON] CHECK CONSTRAINT
[FK_CHITIETPHIEUMUON_PHIEUMUONSACH]
GO
ALTER TABLE [CHITIETPHIEUMUON] WITH CHECK ADD CONSTRAINT
[FK_CHITIETPHIEUMUON_SACH] FOREIGN KEY ([MaSach])
REFERENCES [SACH] ([MaSach])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [CHITIETPHIEUMUON] CHECK CONSTRAINT [FK_CHITIETPHIEUMUON_SACH]

```

2. Nhập dữ liệu cho các bảng:

a. Nhập tay

b. Cách 1: Thực hiện câu lệnh Insert

```

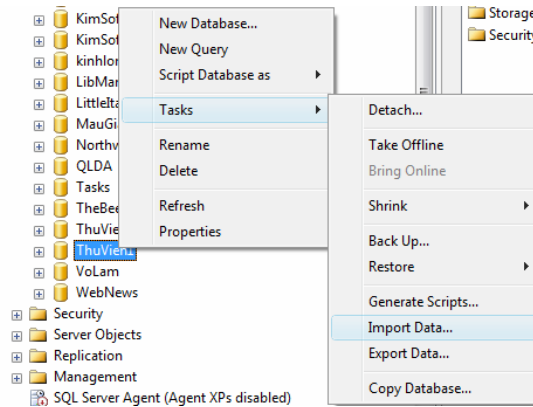
USE ThuVien
GO
INSERT INTO BANGCAP VALUES ('1',N'Tú tài')
INSERT INTO BANGCAP VALUES ('2',N'Trung cấp')
INSERT INTO BANGCAP VALUES ('3',N'Cao đẳng')
INSERT INTO BANGCAP VALUES ('4',N'Đại học')
INSERT INTO BANGCAP VALUES ('5',N'Thạc sĩ')
INSERT INTO BANGCAP VALUES ('6',N'Tiến sĩ')
INSERT INTO BANGCAP VALUES ('7',N'Nghiên cứu sinh')

```

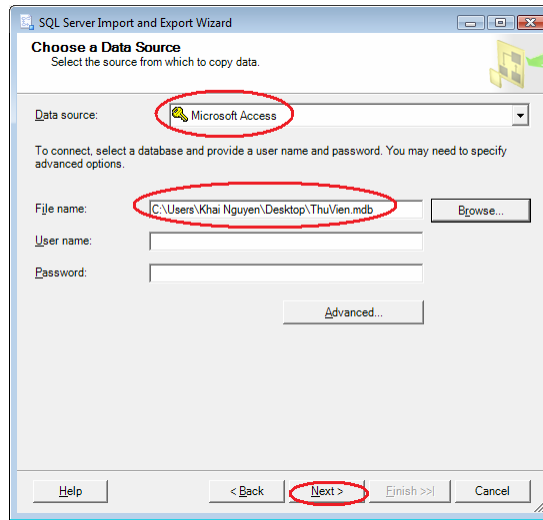
- Làm tương tự cho các bảng còn lại dựa theo file ThuVien.mdb đính kèm.

c. Cách 2:

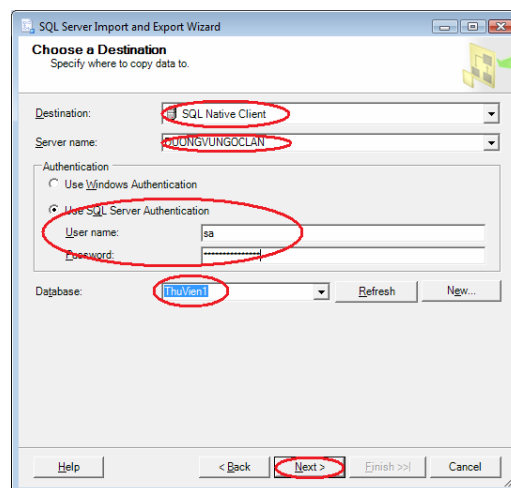
- Tạo database ThuVien1 mới (rỗng).
- Right click lên database ThuVien chọn Task\Import Data...



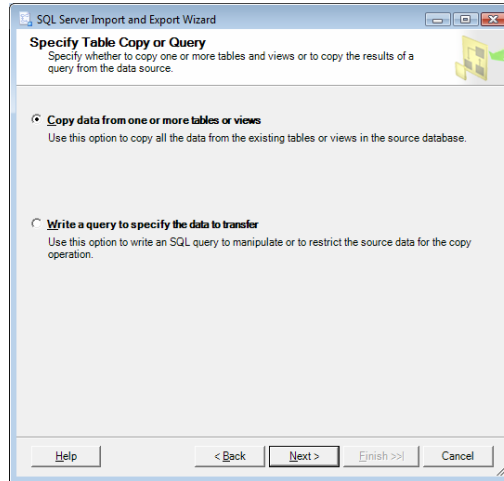
- Chọn Data Source, File name và click Next.



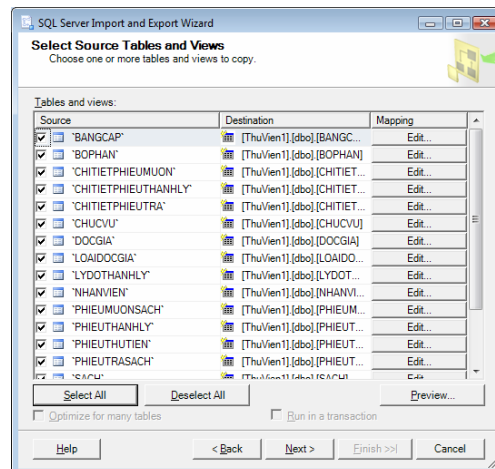
- Chọn Destination, Server name (nếu bạn dùng máy local có thể đánh vào dấu '.')
Connect bằng Windows hay SQL Server Authentication là tùy bạn và click Next.



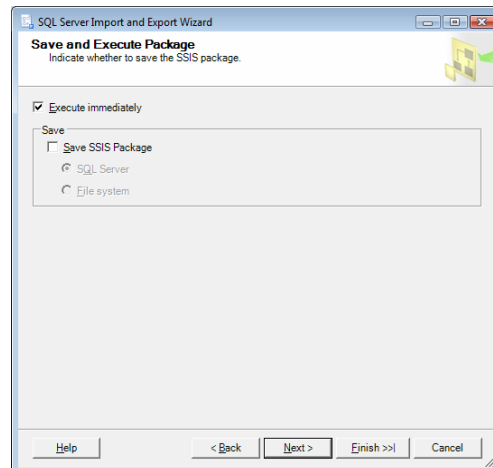
- Chọn Copy data from...click Next.



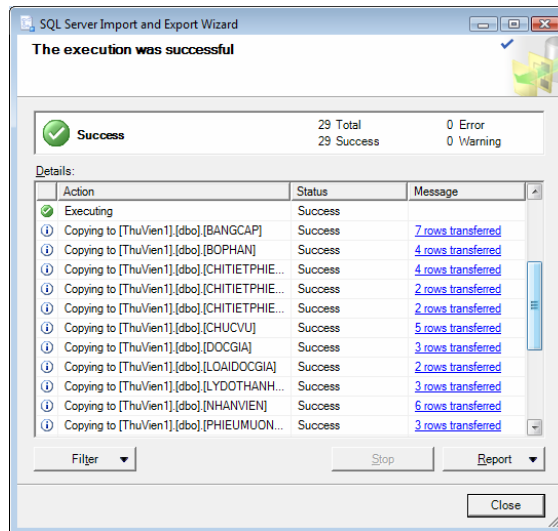
- Chọn Select All...và click Next.



- Click Next, Finish.



- Kết quả như sau:



3. Backup & Restore Database:

Việc mang 1 database từ máy này sang máy khác rất thường xuyên xảy ra (ví dụ: chép bài từ trường về nhà và chép bài từ nhà lên trường,...). Có 2 cách để làm việc này:

1. Attach database:

a. Chép 2 file :

C:\Program files\Microsoft SQL Server\MSSQL.1\Data\ThuVien.mdf

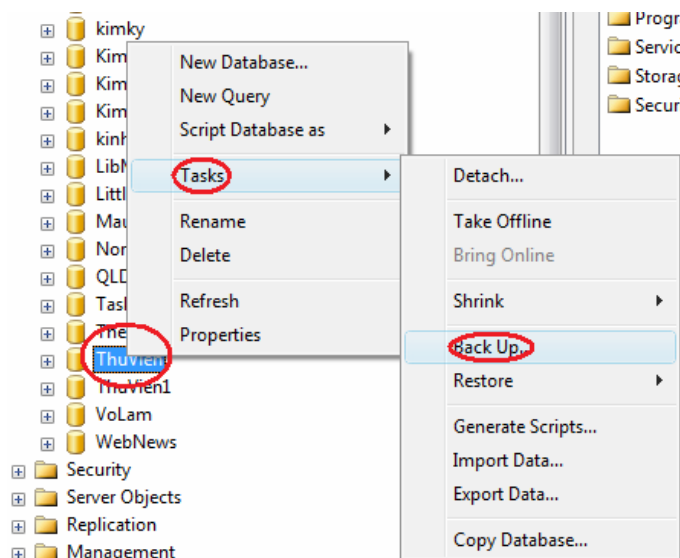
C:\Program files\Microsoft SQL Server\MSSQL.1\Data\ThuVien_log.ldf

b. Attach 2 file trên vào SQL Server khác (máy khác).

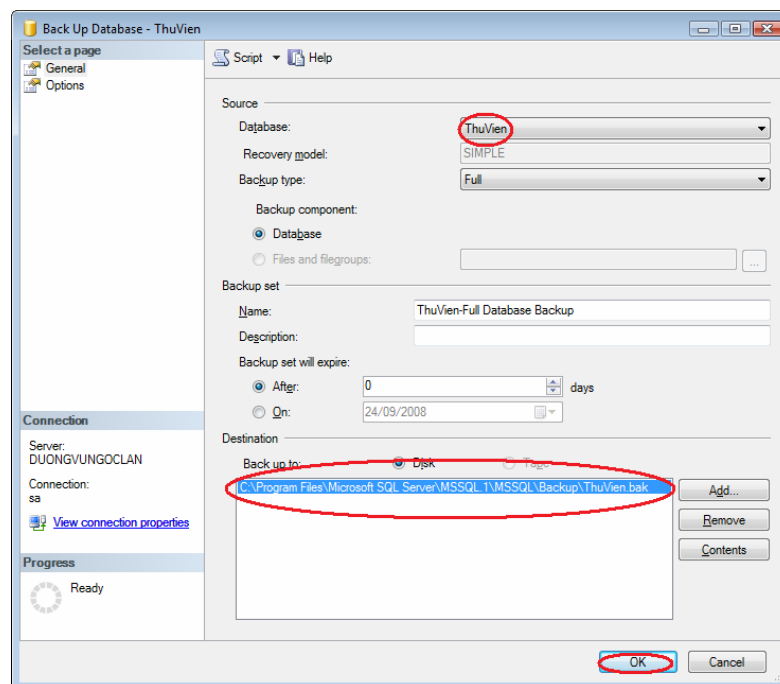
2. Backup & Restore

• Backup:

- Right click lên database ThuVien cần backup, chọn Task\Back Up...

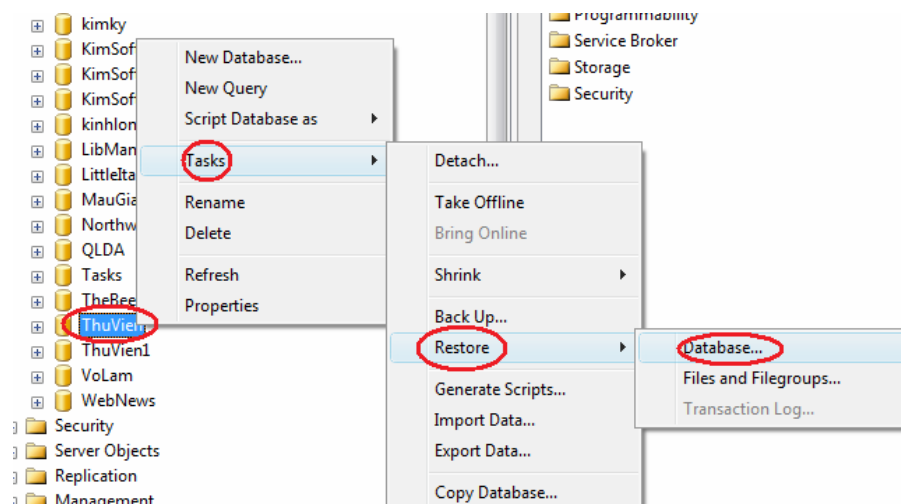


- SQL Server sẽ mặc định backup database thư viện vào file ThuVien.bak theo đường dẫn mặc định sau: C:\Program files\Microsoft SQL Server\MSSQL.1\Backup\ThuVien.bak
- Nhấn Ok để backup, sau đó chép file ThuVien.bak sang máy mới (SQL Server khác) để Restore.
- Trong trường hợp muốn back up với 1 đường dẫn khác thì remove đường dẫn trên và Add lại đường dẫn mới muốn back up đến.

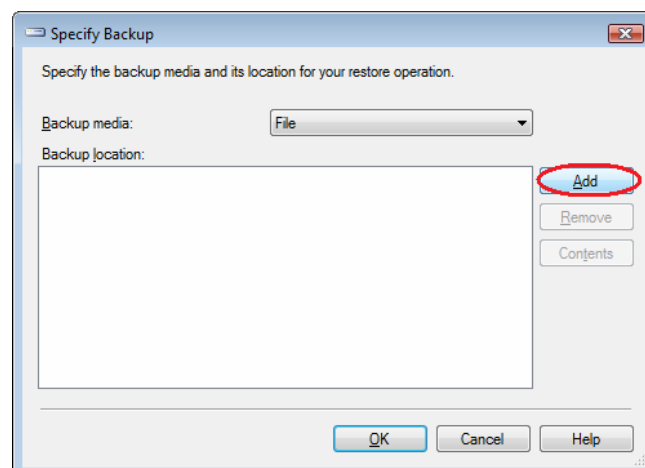
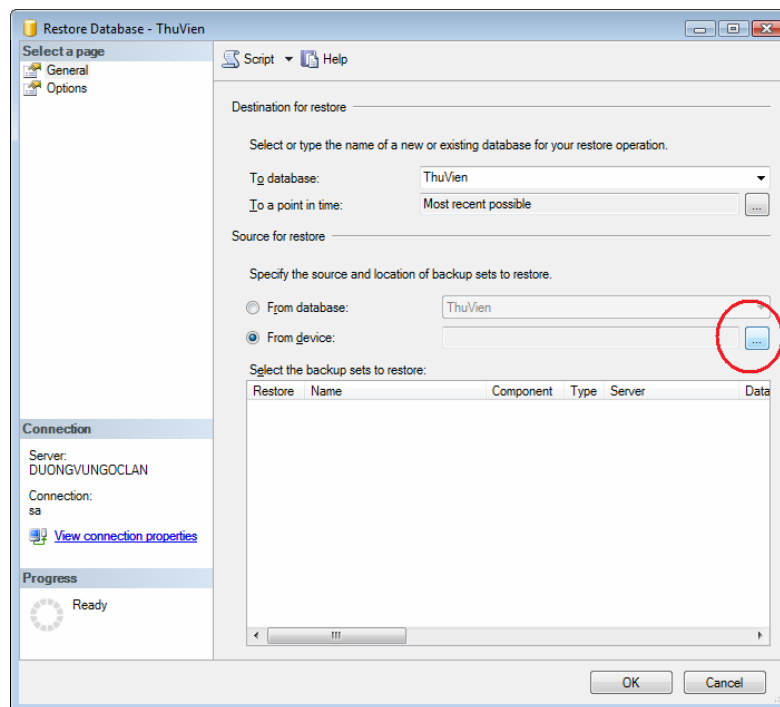


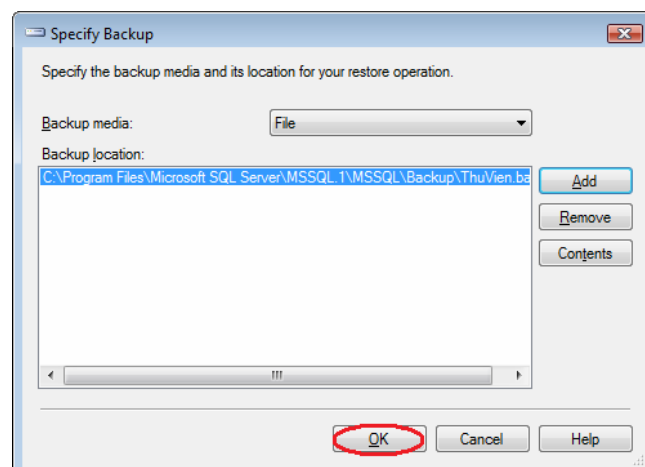
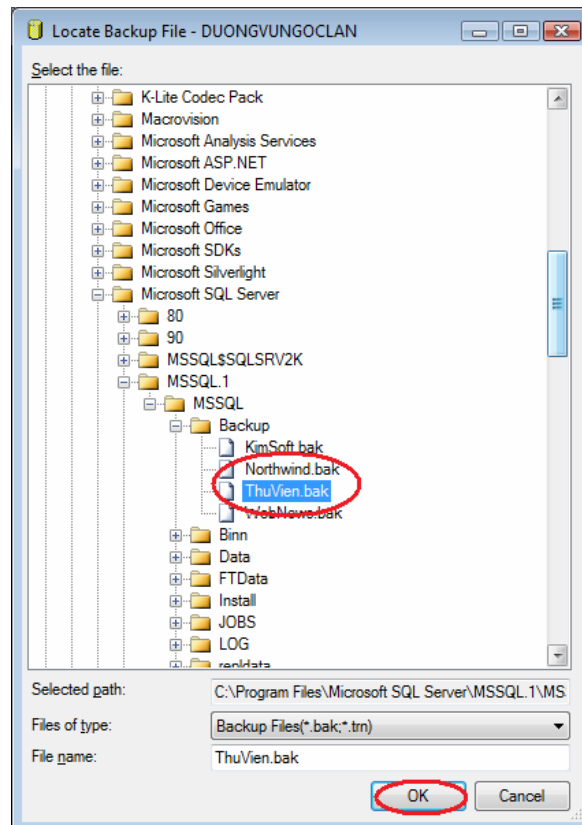
- Restore:

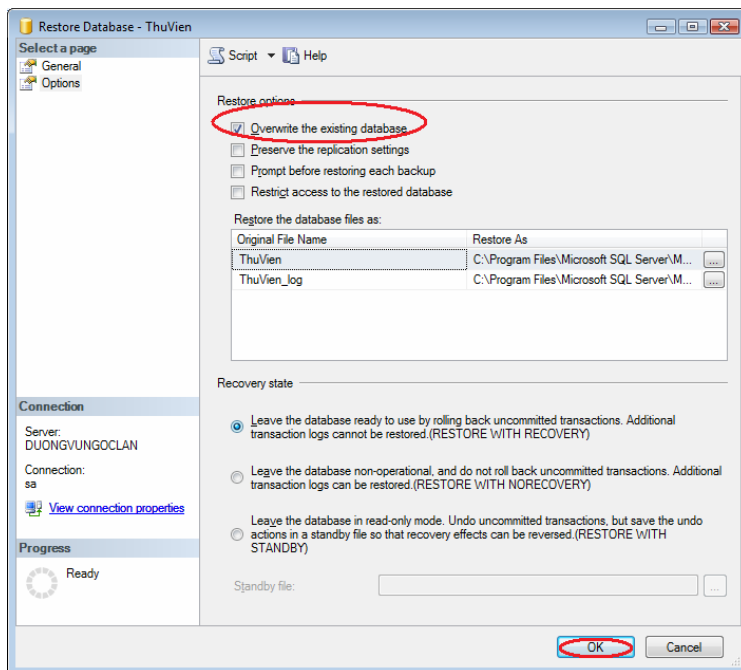
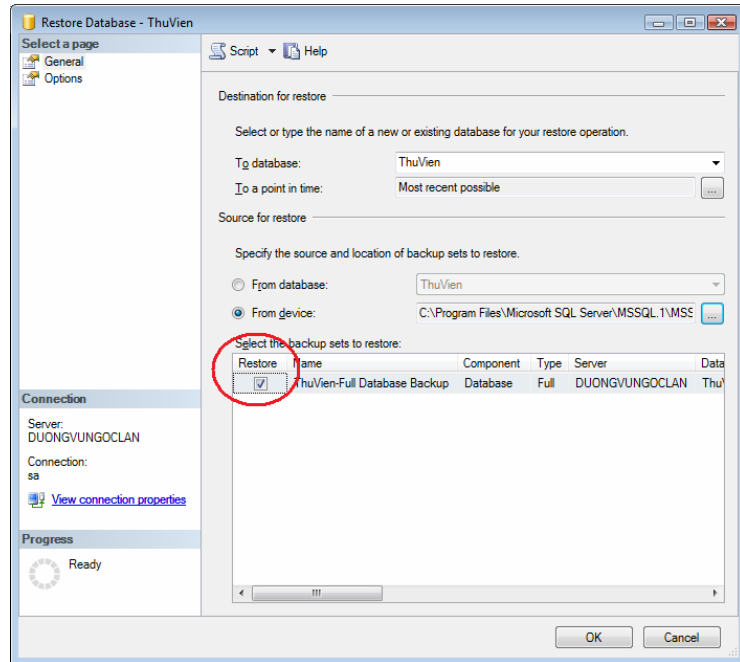
- Đầu tiên tạo database ThuVien trong SQL Management Studio
- Right click lên database vừa tạo, chọn Task\Restore\Database



- Chọn From device chỉ đến file ThuVien.bak



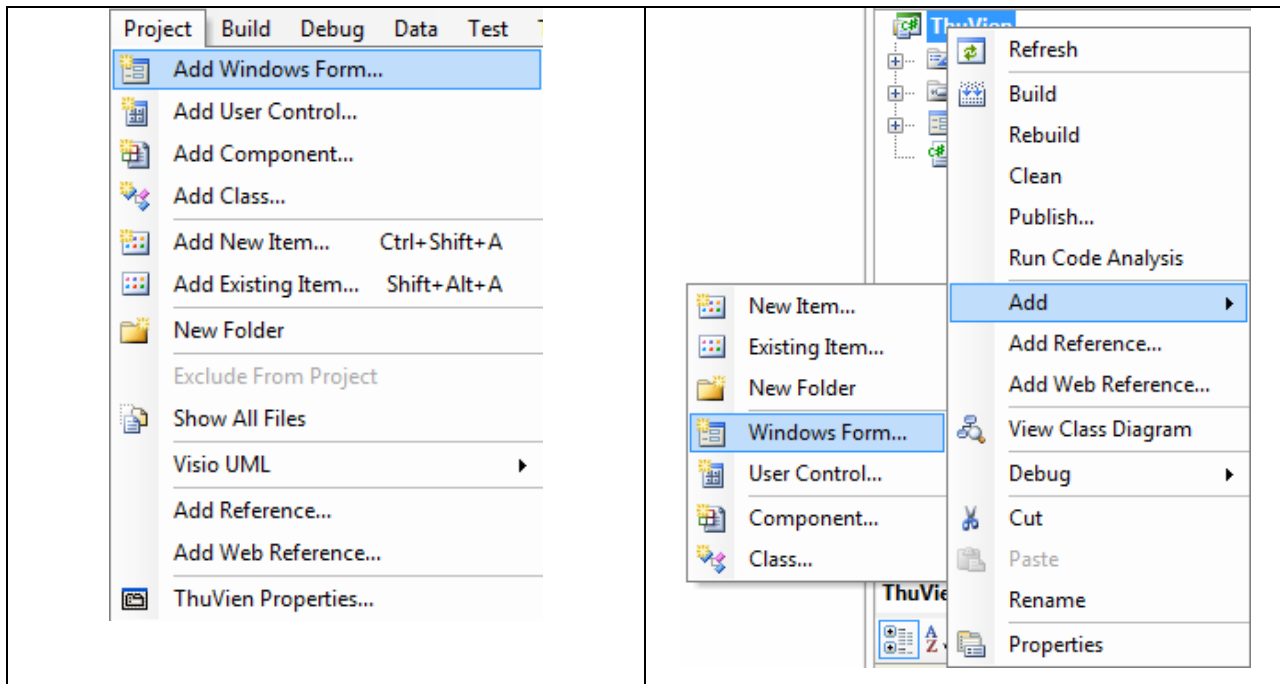




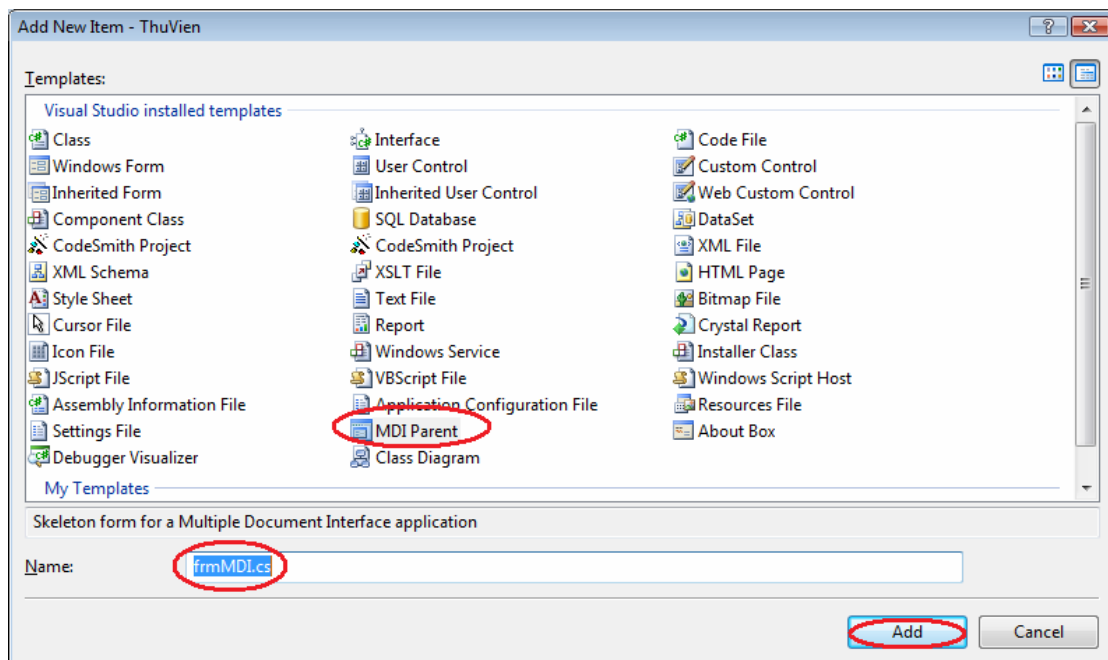
Bài 2. Xây dựng màn hình chính

1. Tạo màn hình chính:

- Có thể thực hiện theo 2 cách sau:
 - Từ menu, chọn project ➔ Add Windows Form...
 - Right-Click project thư viện (trong Solution Explorer), chọn Add ➔ Windows Form.

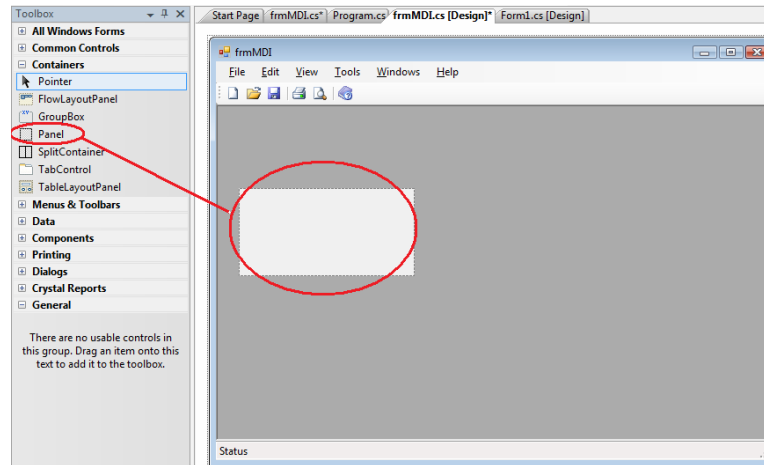


- Chọn Templates: **MDIParent** và đặt tên form là **frmMDI.cs**

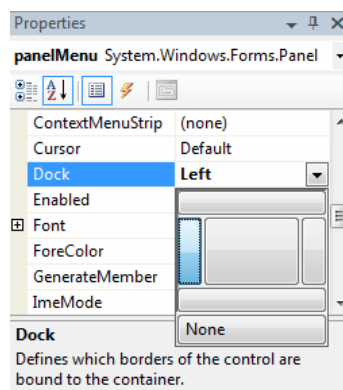


2. Thiết kế giao diện cho màn hình chính:

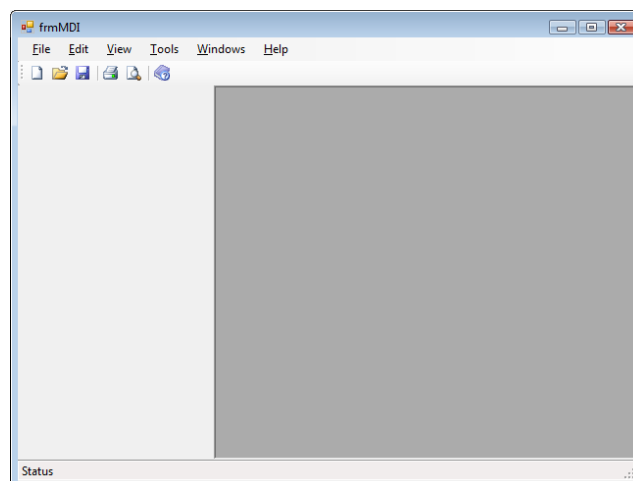
- Kéo thả control Panel vào frmMDI.



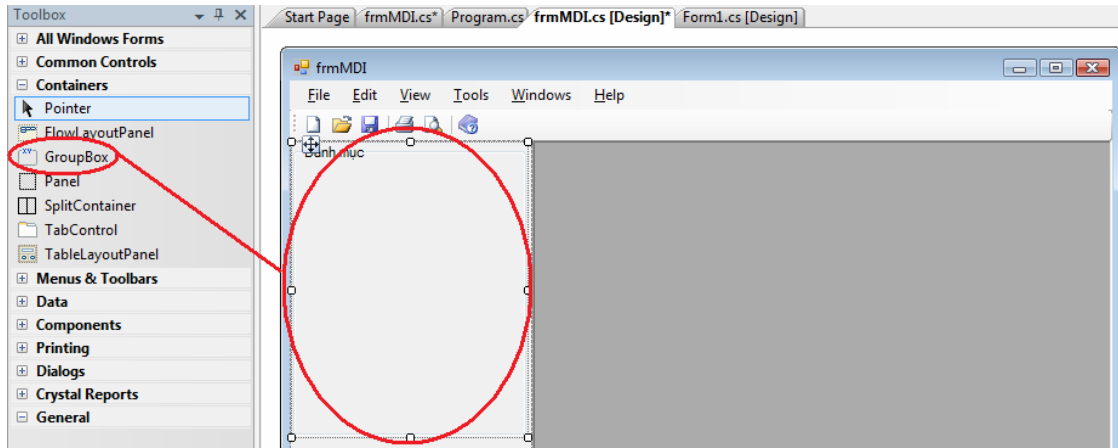
- Kéo thả control Panel vào frmMDI và đặt tên là panelMenu
- Chọn Left cho thuộc tính Dock của panelMenu.



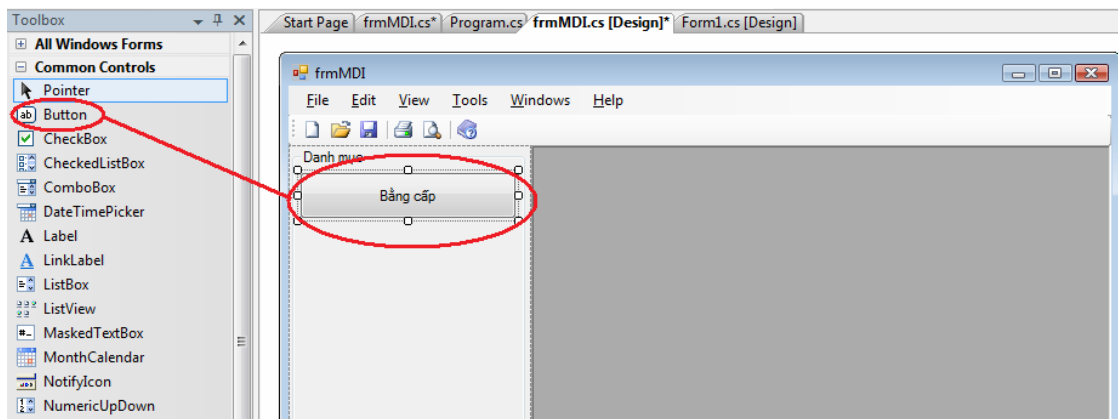
- Kết quả



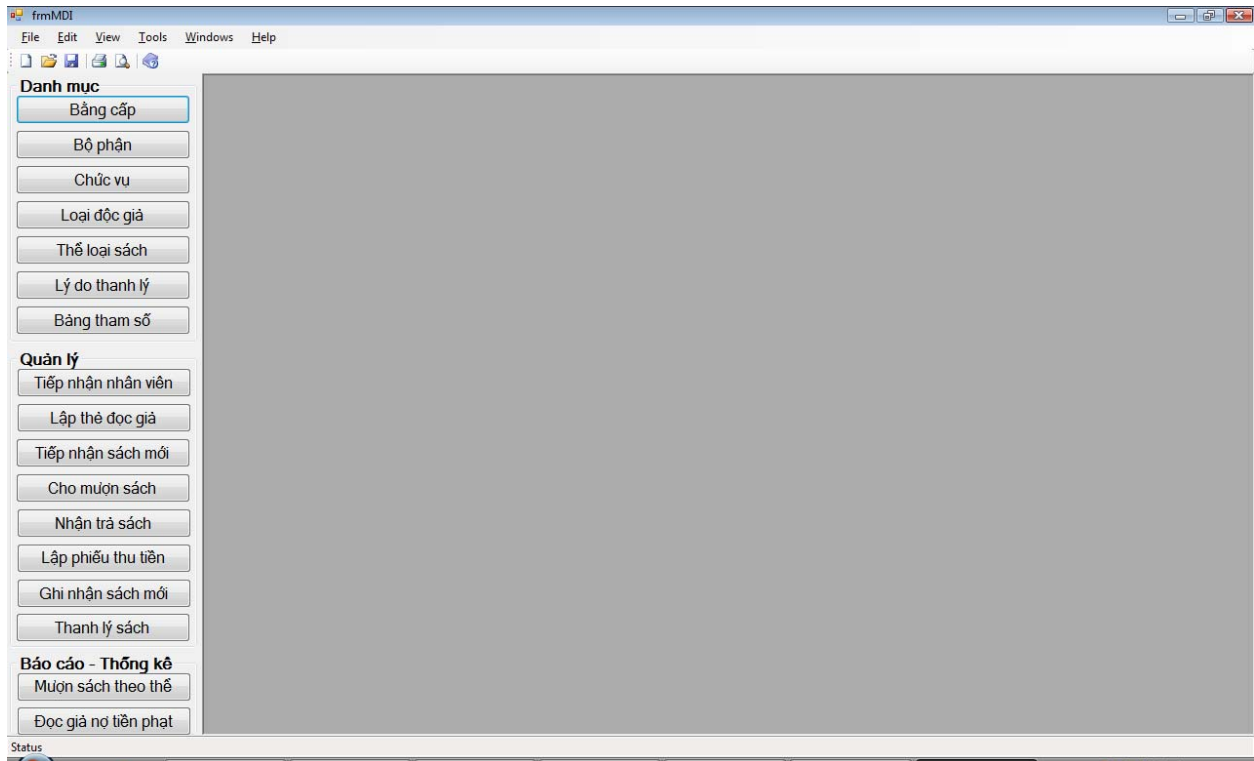
- Kéo thả control GroupBox vào frmMDI và đặt tên (Name) là groupBoxDanhMuc và text là Danh mục.



- Kéo thả control Button vào groupBoxDanhMuc và đặt tên (Name) là btnBangCap và text là Bảng cấp.



- Làm tương tự cho các groupbox và các button còn lại như sau:



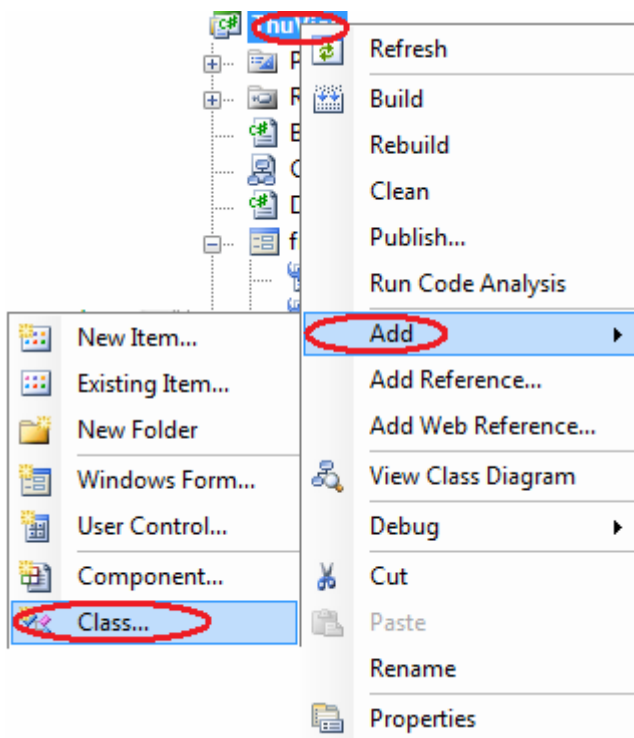
- Như vậy ta đã có được phần thiết kế cho màn hình chính. Bước tiếp theo, ta sẽ viết hàm xử lý sự kiện cho các nút chức năng. Để viết hàm xử lý cho sự kiện click của Button, ta chỉ cần double click lên button, hay chọn sự kiện trong danh sách sự kiện:

Bài 3. Xây dựng lớp giao tiếp CSDL

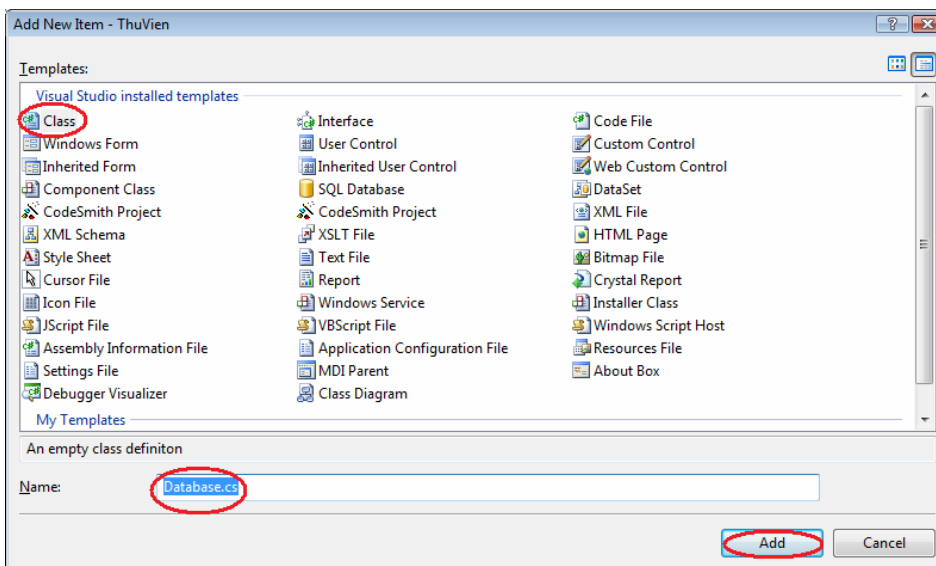
- Lớp Database gồm có các thành phần sau:

	Thuộc tính	Ý nghĩa
	sqlconn	Thuộc lớp SqlConnection
	Phương thức	Ý nghĩa
	Database	Hàm khởi tạo (Constructor)
	Execute	Thực thi một câu lệnh truy vấn và trả về kết quả là một DataTable. Dùng cho các câu lệnh Select * From ...
	ExecuteNonQuery	Thực thi một câu lệnh truy vấn không quan tâm đến kết quả trả về. Dùng cho các câu lệnh Insert, Delete, Update ...

- Right-Click project ThuVien, chọn Add → Class...



- Nhập vào tên lớp là Database.cs và click Add.



- Viết code cho Database.cs như sau:

• C#.NET

```
using System;
using System.Collections.Generic;
using System.Text;

//Khai báo thư viện cho các lớp
using System.Data.SqlClient; //SqlConnection,...
using System.Data; //DataSet, DataTable,...

namespace ThuVien
{
    class Database
```



```

{
    //Thuộc tính
    SqlConnection sqlconn;
    //Constructor khởi tạo
    public Database(string svrName, string dbName, bool intergratedMode, string
usrName, string pwd)
    {
        string connStr;
        if (intergratedMode == true)
        {
            //Đăng nhập SQL Server sử dụng Windows Authentication Mode.
            connStr = "server=" + svrName + "; database=" + dbName + "; Integrated
Security = True";
        }
        else
        {
            //Đăng nhập SQL Server sử dụng SQL Server Authentication Mode.
            connStr = "server=" + svrName + "; uid=" + usrName + "; pwd=" + pwd +
" ;database=" + dbName;
        }
        //Câu lệnh dùng để thiết lập kết nối đến SQL Server là server=tên_svr;
database = tên_db;
        //Integrated Security = True hay server=tên_svr; database = tên_db; uid
= tên_đăng_nhập;
        //pwd = mật_khẩu; database = tên_db .
        sqlconn = new SqlConnection(connStr);
    }

    public DataTable Execute(string strQuery)
    {
        //Câu lệnh này tạo mới một đối tượng SqlDataAdapter và
        //cung cấp cho nó câu truy vấn cùng với kết nối hiện hành
        SqlDataAdapter da = new SqlDataAdapter(strQuery, sqlconn);

        //DataSet đại diện cho một danh sách các DataTable (mỗi DataTable là 1
bảng dữ liệu)
        DataSet ds = new DataSet();

        //Dùng SqlDataAdapter đã tạo ra ở trên thực thi câu lệnh truy vấn đã cung
cấp,
        //và đưa kết quả trả về và ds
        da.Fill(ds);

        //Chúng ta chỉ lấy bảng đầu tiên
        return ds.Tables[0];
    }

    public void ExecuteNonQuery(string strquery)
    {
        //SqlCommand là đối tượng chuyên đảm nhận việc thực hiện các câu lệnh truy
vấn
        SqlCommand sqlcom = new SqlCommand(strquery, sqlconn);

        sqlconn.Open(); //Mở kết nối
        sqlcom.ExecuteNonQuery(); //Thực hiện câu lệnh truy vấn đã cung cấp ở trên
        sqlconn.Close(); //Đóng kết nối
    }
}

```

Bài 4. Xây dựng các màn hình danh mục

1. Màn hình danh mục Bằng cấp

- Thiết kế theo màn hình sau:

STT	Tên
1	Tú tài
2	Trung cấp
3	Cao đẳng
4	Đại học
5	Thạc sĩ
6	Tiến sĩ
7	Nghiên cứu sinh

Tên bảng cấp: Tú tài

Thêm Xoá Cập nhật Soạn lại

STT	Tên Control	Loại	Tên hàm xử lý
1	lvBangCap	ListView	lvBangCap_SelectedIndexChanged
2	txtTenBangCap	TextBox	
3	btnThem	Button	btnThem_Click
4	btnXoa	Button	btnXoa_Click
5	btnCapNhat	Button	btnCapNhat_Click
6	btnSoanLai	Button	btnSoanLai_Click
7	frmBangCap	Form	frmBangCap_Load

- Vào thuộc tính Columns của ListView lvBangCap, click chọn Collection để Add 2 cột như sau:

ColumnHeader Collection Editor

Members:

- 0 colSTT
- 1 colTenBangCap

colSTT properties:

- Behavior: DisplayIndex: 0
- Data: (ApplicationSetting): Tag:
- Design: (Name): colSTT, GenerateMember: True, Modifiers: Private
- Misc: ImageIndex: (none), ImageKey: (none), Text: STT, TextAlign: Left

Add Remove OK Cancel

- Màn hình bảng cấp được cài đặt thành 2 lớp:
 - frmBangCap.cs
 - BangCap.cs

```
BangCap.vb
using System;
using System.Collections.Generic;
```

```

using System.Text;

//Khai báo thư viện cho các lớp
using System.Data;

namespace ThuVien
{
    class BangCap
    {
        //Khai báo thuộc tính
        Database db;
        DataTable bc_dt;

        //Hàm Constructor khởi tạo giá trị cho thuộc tính
        public BangCap()
        {
            //SvrName và DbName là 2 biến tính (shared) của lớp frmMDI
            db = new Database(frmMDI.svrName, frmMDI.dbName,
frmMDI.intergratedMode, frmMDI.usrName, frmMDI.pwd);

        }
        public DataTable LayDanhSachBangCap()
        {
            bc_dt = db.Execute("Select * from BANGCAP order by MaBangCap");
            //Chạy câu lệnh truy vấn và trả về DataTable kết quả
            //Lưu ý: Các loại bằng cấp được đưa lên ListView trên màn hình hiển
thị
            //phải có thứ tự giống với thứ tự chúng được lấy ra bằng câu lệnh
Select
            //ở đây (ListView không được có thuộc tính Sort), để khi người dùng
chọn
            //1 dòng trong ListView và nhấn Delete, chúng ta có thể dễ dàng lấy
được
            //Mã loại tương ứng với loại bằng cấp muốn xóa.
            return bc_dt;
        }
        public void ThemBangCap(string ten)
        {
            DataTable dt;
            dt = db.Execute("select MaBangCap from BANGCAP where MaBangCap >=
all (select MaBangCap from BANGCAP)");
            //Chạy câu lệnh truy vấn lấy mã loại bằng cấp có giá trị lớn nhất
            //Kết quả trả về là một DataTable chỉ có 1 dòng duy nhất chứa mã
loại
            //bằng cấp có giá trị lớn nhất này
            int MaBC;
            MaBC = int.Parse(dt.Rows[0][0].ToString()) + 1;
            //Mã loại độc giả mới = mã loại độc giả lớn nhất + 1
            string str;
            str = "INSERT INTO BANGCAP VALUES(" + MaBC.ToString() + ",N'" + ten
+ "')";

            //Chuẩn bị câu lệnh truy vấn thêm vào CSDL
            db.ExecuteNonQuery(str);
            //Cập nhật bc_dt
            DataRow dr;
            dr = bc_dt.NewRow();
            dr[0] = MaBC;
            dr[1] = ten;
            bc_dt.Rows.Add(dr);
        }
        public void XoaBangCap(int index)
        {
            string str;
            str = "delete from BANGCAP where MaBangCap = " +
bc_dt.Rows[index]["MaBangCap"].ToString();
            db.ExecuteNonQuery(str);
            //Cập nhật lại danh sách loại độc giả
            bc_dt.Rows.RemoveAt(index);
        }
        public void CapNhatBangCap(int index, string ten)
        {

```

```

        string str;
        str = "Update BANGCAP set TenBangCap = N'" + ten + "' where
MaBangCap = " + bc_dt.Rows[index]["MaBangCap"].ToString();
        db.ExecuteNonQuery(str);
        //Cập nhật lại danh sách bằng cấp
        DataRow dr;
        dr = bc_dt.Rows[index];
        dr[1] = ten;
    }
}
}

```

- Viết code cho frmBangCap như sau:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace ThuVien
{
    public partial class frmBangCap : Form
    {
        //Khai báo thuộc tính
        BangCap bc = new BangCap();

        //Constructor
        public frmBangCap()
        {
            InitializeComponent();
        }

        private void frmBangCap_Load(object sender, EventArgs e)
        {
            HienThiDanhSachBangCap();
        }

        public void HienThiDanhSachBangCap()
        {
            lvBangCap.View = View.Details;
            lvBangCap.FullRowSelect = true;
            lvBangCap.Columns[1].Width = 273;

            lvBangCap.Items.Clear();
            int i;
            string str;
            DataTable dt;
            dt=bc.LayDanhSachBangCap();
            for (i = 0; i < dt.Rows.Count -1; i++)
            {
                ListViewItem lvi;
                lvi=lvBangCap.Items.Add((i + 1).ToString());
                str = dt.Rows[i]["TenBangCap"].ToString();
                lvi.SubItems.Add(str);
            }
        }

        private void btnThem_Click(object sender, EventArgs e)
        {
            if (txtTenBangCap.Text != "")
            {
                //Dùng biến thuộc lớp LoaiDocGia để thực hiện việc thêm loại
                bc.ThemBangCap(txtTenBangCap.Text);
                //Cập nhật lại màn hình hiển thị
                HienThiDanhSachBangCap();
                MessageBox.Show("Thêm thành công", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
}

```

```

        txtTenBangCap.Text = "";
    }
    else
    {
        MessageBox.Show("Bạn chưa nhập tên bảng cấp", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void btnXoa_Click(object sender, EventArgs e)
{
    if (lvBangCap.SelectedIndices.Count > 0)
    {
        //lvBangCap.SelectedIndices.Count là số lượng item của list
view được chọn
        string str="";
        str
=lvBangCap.Items[lvBangCap.SelectedIndices[0]].SubItems[1].Text;
        if (MessageBox.Show("Bạn có chắc chắn là muốn xóa bảng cấp '" +
str + "' không ?", "Hỏi lại", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
== DialogResult.Yes)
        {
            bc.XoaBangCap(lvBangCap.SelectedIndices[0]);
            //Xóa loại bảng cấp này khỏi danh sách, việc xóa khỏi CSDL
            //được thực hiện qua biến bc chứ không chạy truy vấn trực
tiếp
            lvBangCap.Items.Remove(lvBangCap.SelectedItems[0]);
            //Thông báo xóa thành công
            MessageBox.Show("Bảng cấp '" + txtTenBangCap.Text + "' đã
được xóa.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtTenBangCap.Text = "";
        }
    }
    else
    {
        //Nếu không có item nào được chọn thì hiển thị thông báo
        MessageBox.Show("Bạn phải chọn 1 bảng cấp trong danh sách
trước", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnCapNhat_Click(object sender, EventArgs e)
{
    if (lvBangCap.SelectedIndices.Count > 0)
    {
        if (txtTenBangCap.Text != "")
        {
            bc.CapNhatBangCap(lvBangCap.SelectedIndices[0],
txtTenBangCap.Text);
            //Cập nhật loại bảng cấp trong CSDL và trên list view
            HienThiDanhSachBangCap();
            MessageBox.Show("Bảng cấp '" + txtTenBangCap.Text + "' đã
được cập nhật thành công", "Thông báo", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            txtTenBangCap.Text = "";
        }
    }
    else
    {
        //Nếu không có loại bảng cấp nào được chọn để tiến hành cập
nhật thì hiển thị thông báo
        MessageBox.Show("Bạn phải chọn 1 bảng cấp trong danh sách
trước", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnSoanLai_Click(object sender, EventArgs e)
{
    txtTenBangCap.Text = "";
}

```

```

private void lvBangCap_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lvBangCap.SelectedIndices.Count > 0)
    {
        txtTenBangCap.Text =
lvBangCap.Items[lvBangCap.SelectedIndices[0]].SubItems[1].Text;
    }
}
}

```

- Thêm đoạn code sau vào phần khai báo thuộc tính của frmMDI:

```

public partial class frmMDI : Form
{
    public static string svrName = ".";
    public static string dbName = "ThuVien";
    public static bool intergratedMode = false;
    public static string usrName = "sa";
    public static string pwd = "123";
    ...
}

```

- Chạy và kiểm tra chương trình.
- Làm tương tự cho các màn hình danh mục sau:
 - Bộ phận
 - Chức vụ
 - Loại độc giả
 - Thẻ loại sách
 - Lý do thanh lý
 - Bảng tham số

Bài 5. Xây dựng các màn hình quản lý

- Thiết kế frmNhanVien.vb như sau:

frmNhanVien

Thông tin chung

STT	Họ & tên	Bằng cấp	Bộ phận	Chức vụ
1	Trần Thị B	Trung Cấp	Thủ Quỹ	Phó Giám Đốc
2	Nguyễn Đức Trí	Tú Tài	Thủ Thư	Giám Đốc
3	Trần Hoàng Hiếu	Cao Đẳng	Thủ Thư	Phó Phòng
4	Phan Minh Tuyền	Đại học	Thủ Quỹ	Phó Phòng

Tổng số nhân viên: 5

Thông tin chi tiết

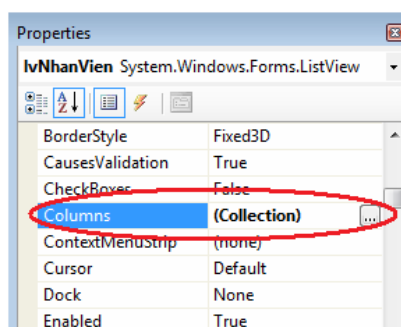
Họ tên: Bằng cấp:

Bộ phận: Chức vụ:

Ngày sinh: Điện thoại:

Địa chỉ:

- Chọn thuộc tính Columns của lvNhanVien:



- Add 5 cột, đặt tên, text, width tương ứng như sau:

ColumnHeader Collection Editor

Members:

- 0 colSTT
- 1 colHoTen
- 2 colBangCap
- 3 colBoPhan
- 4 colChucVu

colSTT properties:

DisplayIndex 0

Data (ApplicationSetting)

Tag

Design (Name) **colSTT**

GenerateMember True

Modifiers Private

Misc ImageIndex ☐ (none)

ImageKey ☐ (none)

Text **STT**

TextAlign Left

Width **60**

STT	Tên Control	Loại	Tiêu đề
1	tbLenh	Toolbar	
2	gbThongTinChung	Groupbox	Thông tin chung
3	lvNhanVien	ListView	
4	lblTongnv	Label	Tổng số nhân viên
5	lblTong	Label	Giá trị tổng số nhân viên
6	gbThongTinChiTiet	Groupbox	Thông tin chi tiết
7	lblHoTen	Label	Họ tên
8	txtHoTen	Textbox	
9	lblBangCap	Label	Bằng cấp
10	cbBangCap	ComboBox	
11	lblBoPhan	Label	Bộ phận
12	cbBoPhan	ComboBox	
13	lblChucVu	Label	Chức vụ
14	cbChucVu	ComboBox	
15	lblNgaySinh	Label	Ngày sinh
16	dtNgaySinh	DateTimePicker	
17	lblDienThoai	Label	Điện thoại
18	txtDienThoai	Textbox	
19	lblDiaChi	Label	Địa chỉ
20	txtDiaChi	Textbox	

- Màn hình nhân viên được cài đặt thành 2 lớp:
 - frmNhanVien.vb
 - NhanVien.vb

```

using System;
using System.Collections.Generic;
using System.Text;

using System.Data;
namespace ThuVien
{
    class NhanVien
    {
        //Khai báo biến
        Database db;

        DataTable bc_dt;
        DataTable bp_dt;
        DataTable cv_dt;
        DataTable nv_dt;

        BangCap bc;
    }
}

```



```

        BoPhan bp;
        ChucVu cv;

        //Constructor
        public NhanVien()
        {
            db = new Database(frmMDI.svrName, frmMDI.dbName, frmMDI.intergratedMode,
frmMDI.usrName, frmMDI.pwd);
            //Khởi tạo lớp
            bc = new BangCap();
            bp = new BoPhan();
            cv = new ChucVu();
            //Lấy danh sách các đối tượng
            bc_dt = bc.LayDanhSachBangCap();
            bp_dt = bp.LayDanhSachBoPhan();
            cv_dt = cv.LayDanhSachChucVu();
        }
        public DataTable LayDanhSachNhanVien()
        {
            nv_dt = db.Execute("Select * from NHANVIEN");
            return nv_dt;
        }

        public bool ThemNhanVien(string ten, string ngaysinh, string diachi, string
dienthoai, int index_bc, int index_bp, int index_cv)
        {
            //Mã nhân viên mới = Mã nhân viên lớn nhất hiện tại + 1
            DataTable dt;
            dt = db.Execute("select MaNhanVien from NHANVIEN where MaNhanVien >= all
(select MaNhanVien from NHANVIEN)");
            //dt chỉ có 1 dòng chứa mã nhân viên lớn nhất
            int MaNV;
            MaNV = (int)dt.Rows[0][0] + 1;
            string str;
            //Tạo câu lệnh truy vấn
            str = "Insert into NHANVIEN values(" + MaNV.ToString() + ",N'" + ten + "','" +
ngaysinh + "','" + diachi + "','" + dienthoai + "','" +
bc_dt.Rows[index_bc]["MaBangCap"].ToString() + "," +
bp_dt.Rows[index_bp]["MaBoPhan"].ToString() + "," +
cv_dt.Rows[index_cv]["MaChucVu"].ToString() + ")";
            db.ExecuteNonQuery(str);
            //Cập nhật lại nv_dt
            LayDanhSachNhanVien();
            return true;
        }
        public bool XoaNhanVien(int index)
        {
            string str;
            //Chuẩn bị câu lệnh truy vấn
            str = "Delete from NHANVIEN where MaNhanVien = " +
nv_dt.Rows[index]["MaNhanVien"].ToString();
            db.ExecuteNonQuery(str);
            //Cập nhật lại nv_dt
            nv_dt.Rows.RemoveAt(index);
            return true;
        }
        public void CapNhatNhanVien(int index, string hoten, string ngaysinh, string
diachi, string dienthoai, int index_bc, int index_bp, int index_cv)
        {
            string str;
            //Chuẩn bị câu lệnh truy vấn
            str = "Update NHANVIEN set HoTenNhanVien = N'" + hoten + "', NgaySinh ='" +
ngaysinh + "', diachi = N'" + diachi + "', dienthoai = '" + dienthoai + "', MaBangCap = " +
bc_dt.Rows[index_bc]["MaBangCap"].ToString() + ", MaBoPhan = " +
bp_dt.Rows[index_bp]["MaBoPhan"].ToString() + ", MaChucVu = " +
cv_dt.Rows[index_cv]["MaChucVu"].ToString() + " where MaNhanVien = " +
nv_dt.Rows[index]["MaNhanVien"].ToString();
            db.ExecuteNonQuery(str);
            //Cập nhật lại data table nv_dt
            LayDanhSachNhanVien();
        }

```

```

    }
}

```

- Viết code cho frmNhanVien.cs như sau:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace ThuVien
{
    public partial class frmNhanVien : Form
    {
        //Khai báo biến
        NhanVien nv;
        BangCap bc;
        BoPhan bp;
        ChucVu cv;

        //Constructor
        public frmNhanVien()
        {
            InitializeComponent();
            //Khởi tạo lớp
            nv = new NhanVien();
            bc = new BangCap();
            bp = new BoPhan();
            cv = new ChucVu();
        }

        private void frmNhanVien_Load(object sender, EventArgs e)
        {
            //Load dữ liệu lên Form.
            HienThiDanhSachBangCap();
            HienThiDanhSachBoPhan();
            HienThiDanhSachChucVu();
            HienThiDanhSachNhanVien();
        }

        private void HienThiDanhSachBangCap()
        {
            DataTable dt = bc.LayDanhSachBangCap();
            //Việc lấy dữ liệu được ủy nhiệm cho biến thuộc lớp xử lý (lớp NhanVien) chứ
            //không trực tiếp thực hiện ở đây
            int i;
            string str;
            //Nạp dữ liệu lên combobox
            for(i = 0 ; i < dt.Rows.Count - 1;i++)
            {
                str = dt.Rows[i]["TenBangCap"].ToString();
                cbBangCap.Items.Add(str);
            }
            if (cbBangCap.Items.Count > 0)
                cbBangCap.SelectedIndex = 0;
        }

        private void HienThiDanhSachBoPhan()
        {
            DataTable dt = bp.LayDanhSachBoPhan();
            int i;
            string str;
            for(i = 0; i<dt.Rows.Count - 1;i++)
            {
                str = dt.Rows[i]["TenBoPhan"].ToString();
                cbBoPhan.Items.Add(str);
            }
            if (cbBoPhan.Items.Count > 0)
                cbBoPhan.SelectedIndex = 0;
        }
    }
}

```

```

private void HienThiDanhSachChucVu()
{
    DataTable dt = cv.LayDanhSachChucVu();
    int i;
    string str;
    for (i = 0; i < dt.Rows.Count - 1; i++)
    {
        str = dt.Rows[i]["TenChucVu"].ToString();
        cbChucVu.Items.Add(str);
    }
    if (cbChucVu.Items.Count > 0)
        cbChucVu.SelectedIndex = 0;
}

private void HienThiDanhSachNhanVien()
{
    lvNhanVien.Items.Clear();
    DataTable dt = nv.LayDanhSachNhanVien();
    int i;
    string str;
    int ma;
    for (i = 0; i < dt.Rows.Count - 1; i++)
    {
        ListViewItem lvi;
        lvi = lvNhanVien.Items.Add((i + 1).ToString()); //Số thứ tự của nhân viên
        str = dt.Rows[i]["HoTenNhanVien"].ToString();
        lvi.SubItems.Add(str);
        ma = (int)dt.Rows[i]["MaBangCap"];
        lvi.SubItems.Add(cbBangCap.GetItemText(cbBangCap.Items[ma - 1]));
        ma = (int)dt.Rows[i]["MaBoPhan"];
        lvi.SubItems.Add(cbBoPhan.GetItemText(cbBoPhan.Items[ma - 1]));
        ma = (int)dt.Rows[i]["MaChucVu"];
        lvi.SubItems.Add(cbChucVu.GetItemText(cbChucVu.Items[ma - 1]));
    }
    //Hiển thị tổng số nhân viên
    lblTong.Text = dt.Rows.Count.ToString();
}

private void lvNhanVien_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lvNhanVien.SelectedIndices.Count > 0)
        HienThiThongTinChiTiet(lvNhanVien.SelectedIndices[0]);
}

private void HienThiThongTinChiTiet(int index)
{
    DataTable dt;
    dt = nv.LayDanhSachNhanVien();
    txtHoTen.Text = dt.Rows[index]["HoTenNhanVien"].ToString();
    dtNgaySinh.Value = (DateTime)dt.Rows[index]["NgaySinh"];
    txtDiaChi.Text = dt.Rows[index]["Diachi"].ToString();
    txtDienThoai.Text = dt.Rows[index]["DienThoai"].ToString();
    cbBangCap.SelectedIndex =
cbBangCap.FindString(lvNhanVien.Items[index].SubItems[2].Text);
    cbBoPhan.SelectedIndex =
cbBoPhan.FindString(lvNhanVien.Items[index].SubItems[3].Text);
    cbChucVu.SelectedIndex =
cbChucVu.FindString(lvNhanVien.Items[index].SubItems[4].Text);
}

private void btnThem_Click(object sender, EventArgs e)
{
    if (KiemTraNhap() == true)
    {
        nv.ThemNhanVien(txtHoTen.Text, dtNgaySinh.Value.ToShortDateString(),
txtDiaChi.Text, txtDienThoai.Text, cbBangCap.SelectedIndex, cbBoPhan.SelectedIndex,
cbChucVu.SelectedIndex);
        //Hiển thị lại danh sách nhân viên
        HienThiDanhSachNhanVien();
        MessageBox.Show("Nhân viên '" + txtHoTen.Text + "' đã được thêm thành
công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        SoanLai();
    }
}

```

```

    }
    public bool KiemTraNhap()
    {
        if (txtHoTen.Text == "")
        {
            MessageBox.Show("Bạn chưa nhập tên nhân viên", "Lỗi nhập",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return false;
        }
        return true;
    }
    private void SoanLai()
    {
        txtHoTen.Text = "";
        dtNgaySinh.Value = DateTime.Today;
        txtDiaChi.Text = "";
        txtDienThoai.Text = "";
    }

    private void btnCapNhat_Click(object sender, EventArgs e)
    {
        if (lvNhanVien.SelectedIndices.Count > 0)
        {
            if (KiemTraNhap() == true)
            {
                nv.CapNhatNhanVien(lvNhanVien.SelectedIndices[0], txtHoTen.Text,
                dtNgaySinh.Value.ToString(), txtDiaChi.Text, txtDienThoai.Text, cbBangCap.SelectedIndex,
                cbBoPhan.SelectedIndex, cbChucVu.SelectedIndex);
                //Hiển thị lại danh sách nhân viên
                HienThiDanhSachNhanVien();
                //Thông báo cập nhật thành công
                MessageBox.Show("Nhân viên '" + txtHoTen.Text + "' đã được cập nhật
                thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                MessageBox.Show("Bạn phải chọn 1 nhân viên trong danh sách trước", "Thông
                báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }

        private void btnXoa_Click(object sender, EventArgs e)
        {
            if (lvNhanVien.SelectedIndices.Count > 0 )
            {
                if(MessageBox.Show("Bạn có chắc chắn là muốn xóa nhân viên '" +
                txtHoTen.Text + "' không ?", "Hỏi lại", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
                == DialogResult.Yes)
                {
                    nv.XoaNhanVien(lvNhanVien.SelectedIndices[0]);
                    HienThiDanhSachNhanVien();
                    //Thông báo xóa thành công
                    MessageBox.Show("Nhân viên '" + txtHoTen.Text + "' đã được xóa.", "Thông
                    báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    SoanLai();
                }
                else
                {
                    MessageBox.Show("Bạn phải chọn 1 nhân viên trong danh sách trước", "Thông
                    báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                }
            }

            private void btnSoanLai_Click(object sender, EventArgs e)
            {
                SoanLai();
            }
        }
    }
}

```

- Chạy và kiểm tra chương trình.

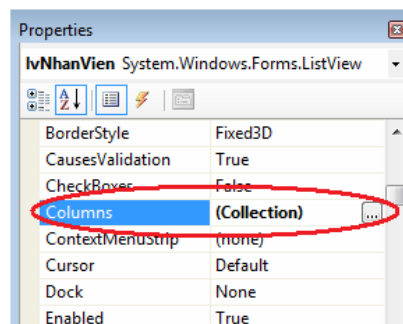
- Làm tương tự cho các màn hình quản lý sau:

- Lập thẻ độc giả
- Tiếp nhận sách mới
- Cho mượn sách
- Nhận trả sách
- Lập phiếu thu tiền phạt
- Ghi nhận sách mất
- Thanh lý sách

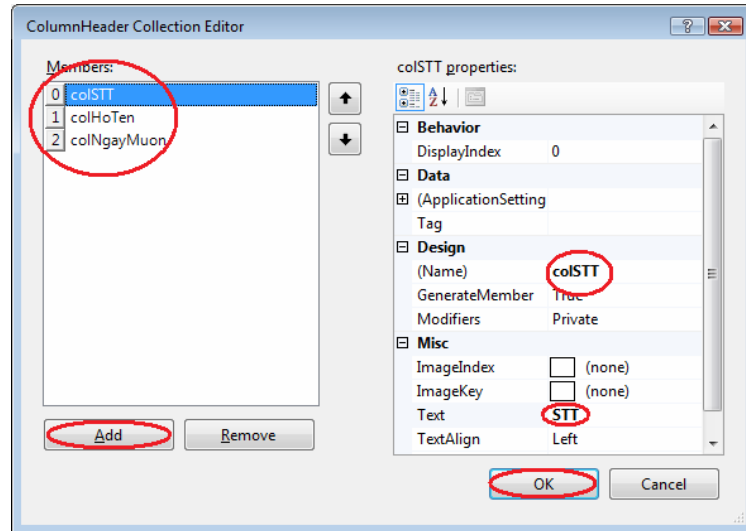
Bài 6. Màn hình Cho mượn sách

- Tạo mới Windows Form và đặt tên là frmChoMuonSach.cs.
- Thiết kế frmChoMuonSach.cs như sau:

- Chọn thuộc tính Columns của lvPhieuMuon:



- Add 3 cột, đặt tên, text, width tương ứng như sau:



STT	Tên Control	Loại	Tiêu đề
1.	gbThôngTinChung	Groupbox	Thông tin chung
2.	lvPhieuMuon	ListView	
3.	lblTongPM	Label	Tổng số phiếu mượn
4.	lblTong	Label	Giá trị tổng số phiếu mượn
5.	gbThôngTinChiTiet	Groupbox	Thông tin chi tiết
6.	lblTenDocGia	Label	Tên đọc giả
7.	cbDocGia	DropDownList	
8.	lblNgayMuon	Label	Ngày mượn
9.	dtNgayMuon	DateTimePicker	
10.	btnChiTietPhieuMuon	Button	Chi tiết phiếu mượn
11.	btnThem	Button	Thêm
12.	btnXoa	Button	Xoá
13.	btnCapNhat	Button	Cập nhật
14.	btnSoanLai	Button	Soạn lại

Bài 7. Màn hình Thống kê tình hình mượn sách theo thể loại

- Màn hình này giúp người quản lý theo dõi thông tin mượn sách trong một tháng cụ thể. Người quản lý sẽ nhập tháng/năm vào và nhấn nút xem, tình hình mượn sách thuộc tất cả các thể loại sẽ được hiển thị trong listview.
- Xây dựng màn hình theo hình sau:

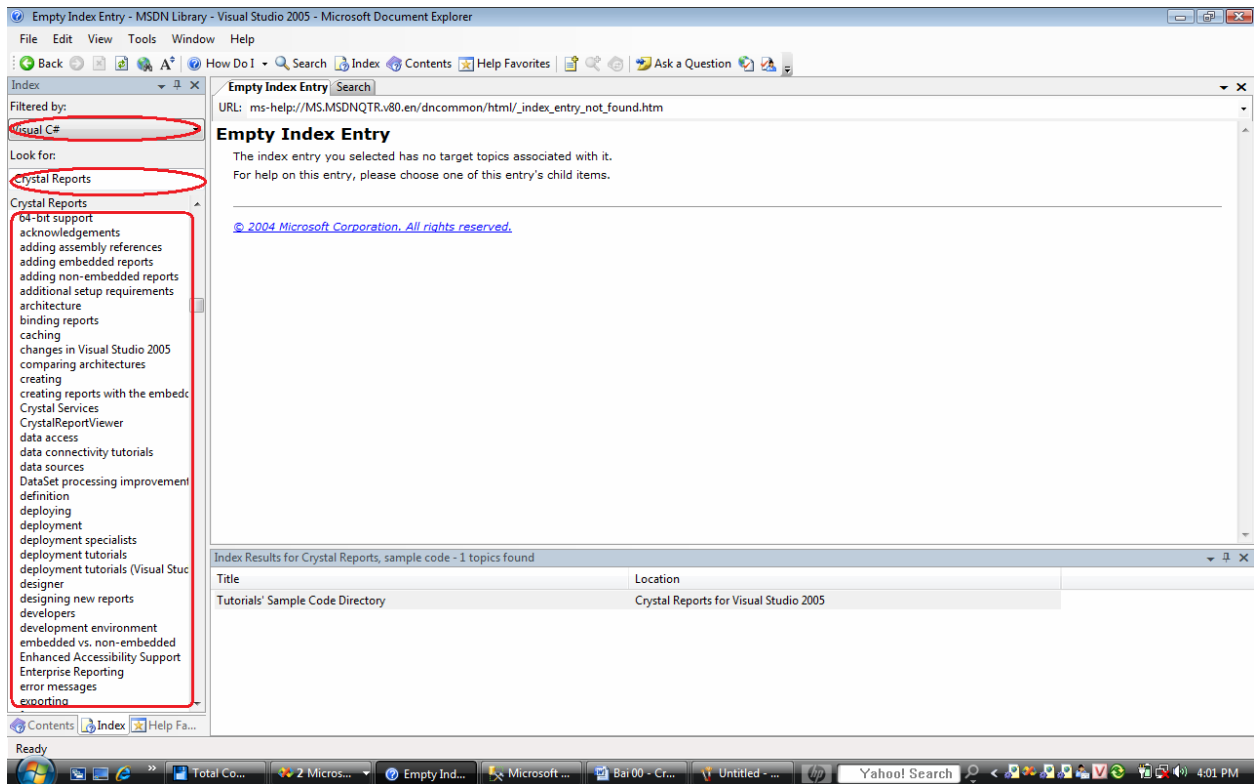
[illegible]

STT	Tên Control	Loại	Tiêu đề
1.	txtThang	Textbox	Tháng
2.	txtNam	Textbox	Năm
3.	lvMuonSach	Listview	ListView thống kê tình hình mượn sách theo thể loại
4.	lblTong	Label	Tổng số lượt mượn
5.	btnXem Button	btnXem Button	Xem

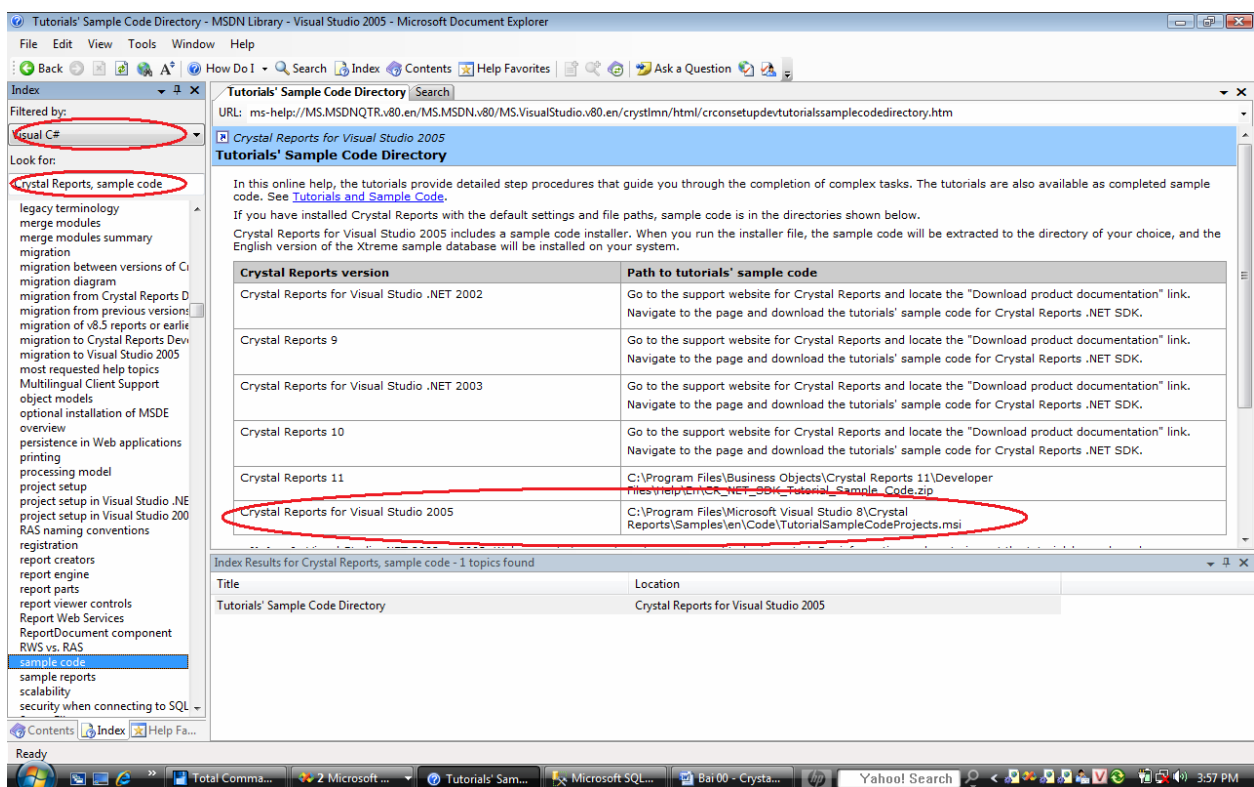
Bài 8. Tạo báo cáo thống kê với Crystal Report

2. Tham khảo MSDN 2005

- Sau khi mở MSDN 2005
 - Chọn Filtered by: Visual C#
 - Look for: Crystal Report ➔Enter



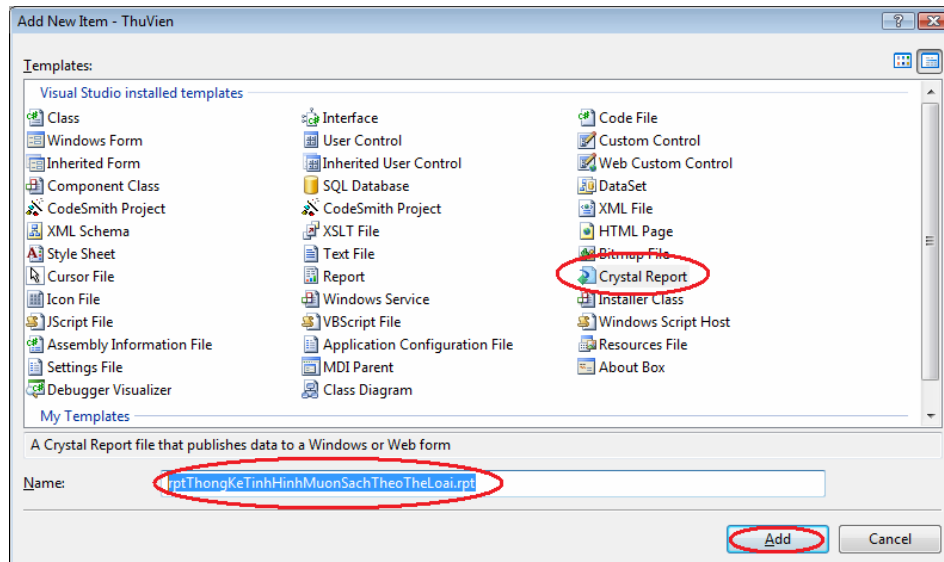
- Các bạn có thể tham khảo bằng cách click chọn các mục liên quan về Crystal Report hoặc tham khảo Sample code. Chọn mục sample code → Enter



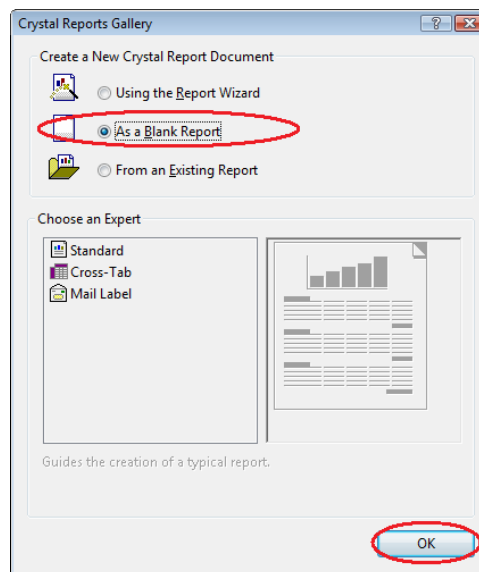
- Chạy file `C:\Program Files\Microsoft Visual Studio 8\Crystal Reports\Samples\en\Code\TutorialSampleCodeProjects.msi` sẽ pháp sinh ra các rất nhiều Sample cho ta tham khảo.

3. Tạo báo cáo thông kê tình hình độc giả nợ tiền phạt.

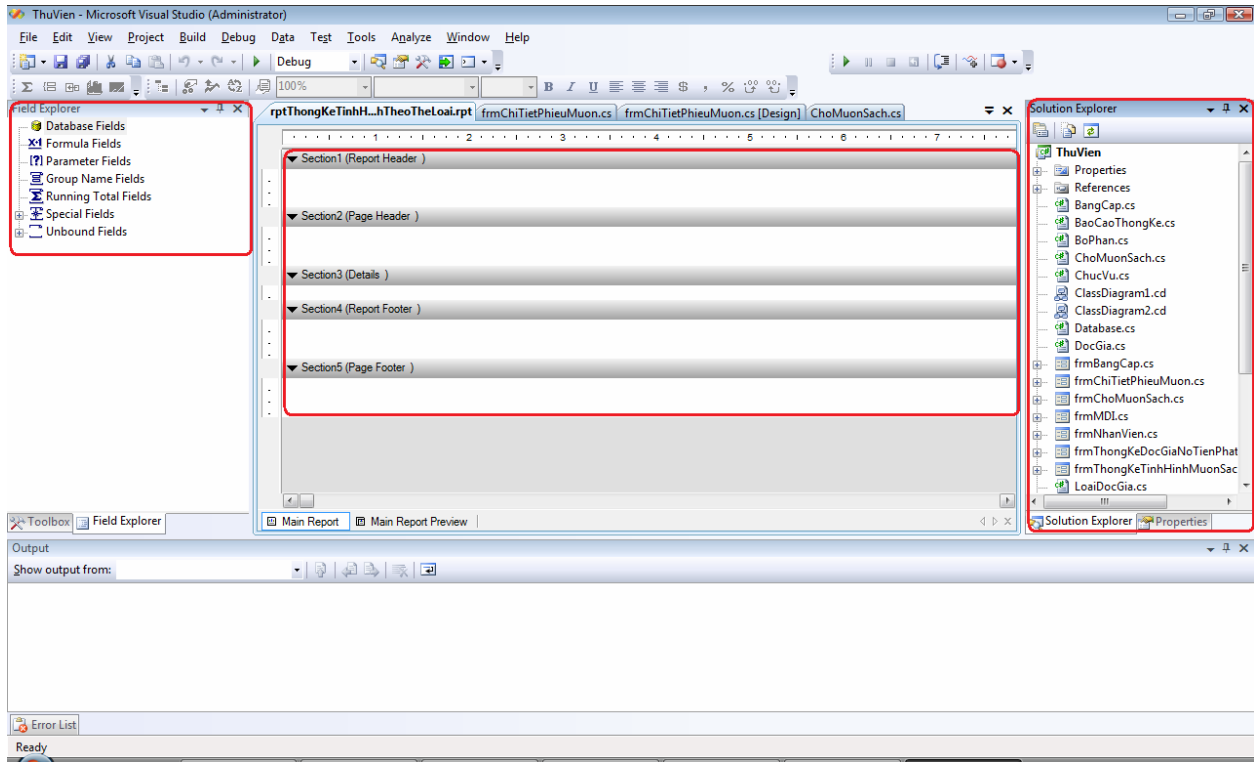
- Mở project ThuVien
- Thêm vào CrystalReports rptTinhHinhDocGiaNoTienPhat.rpt như sau:
 - Từ menu Project, chọn Add New Item...
 - Nhập vào thông tin như sau:



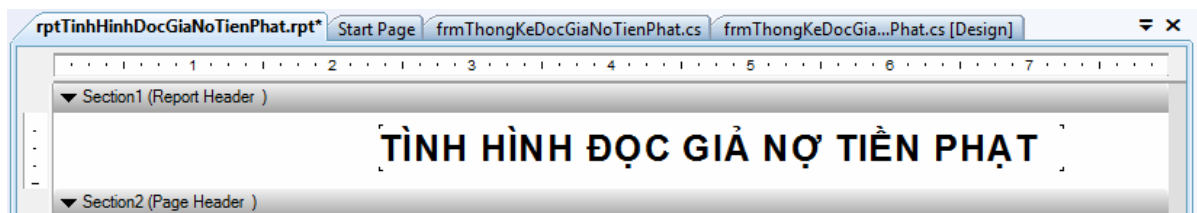
- Sau khi tạo xong dialog sau sẽ xuất hiện



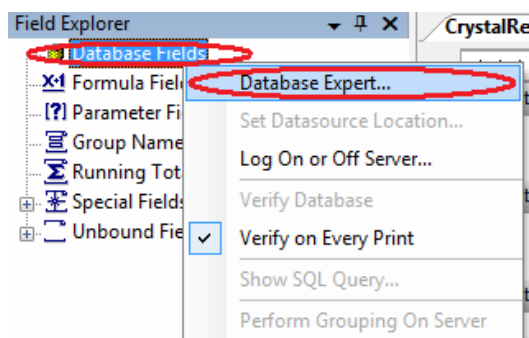
- Chúng ta sẽ tạo 1 Blank Report. Click Ok → Kết quả như sau:



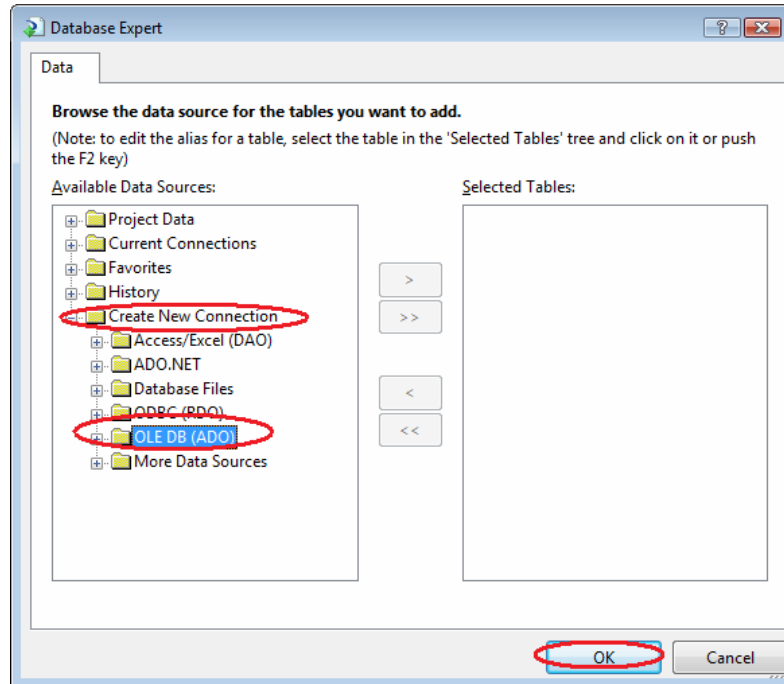
- Thêm vào dòng tiêu đề của report
 - Right click lên Section1 chọn Insert\Text object từ popup menu
 - Nhập vào nội dung và định dạng như sau.



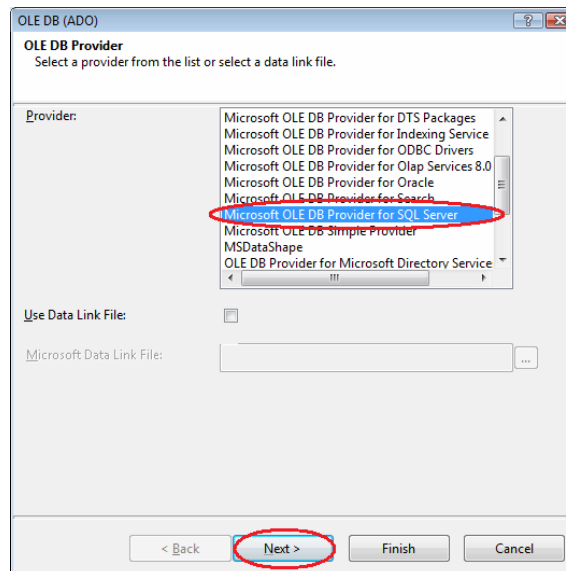
- Kết nối cơ sở dữ liệu



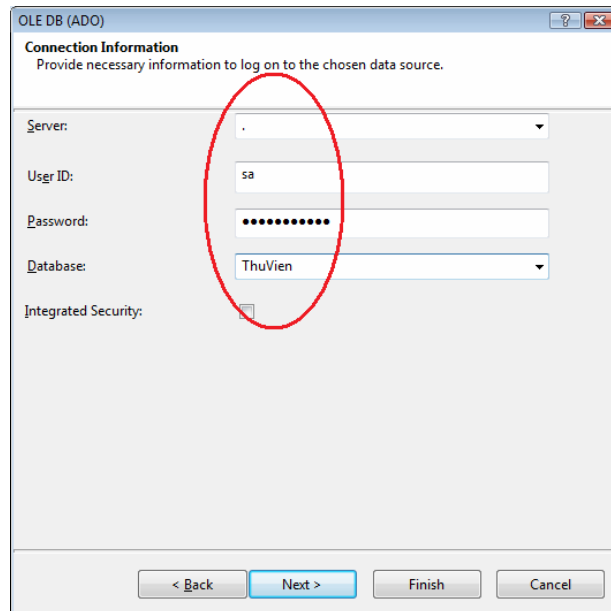
- Chọn Database Fields/ Database Expert...



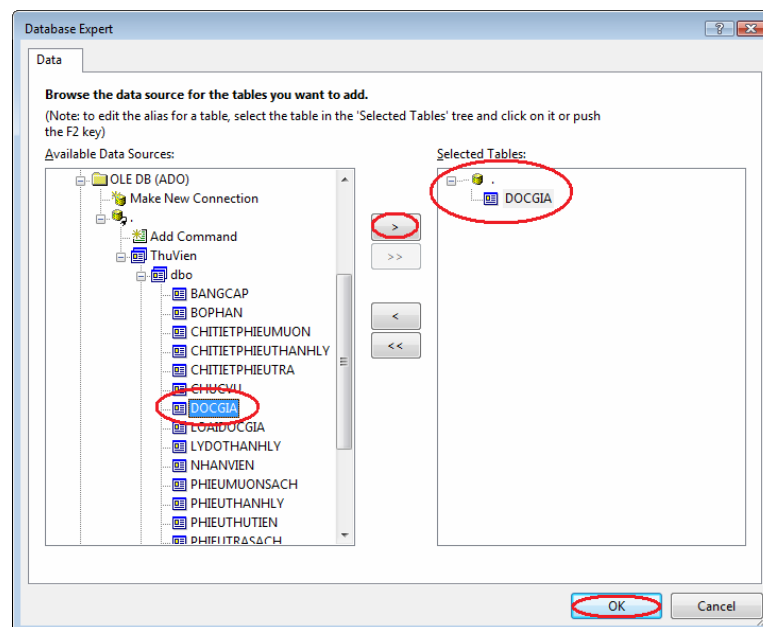
- Chọn Create New Connection/ OLEDB (ADO) để kết nối.



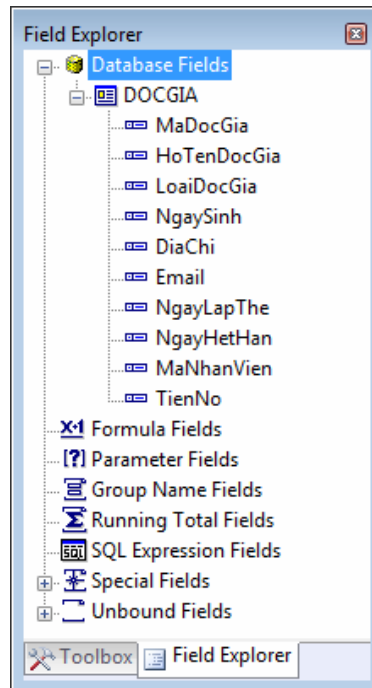
- Chọn Provider là Microsoft OLE DB Provider for SQL Server



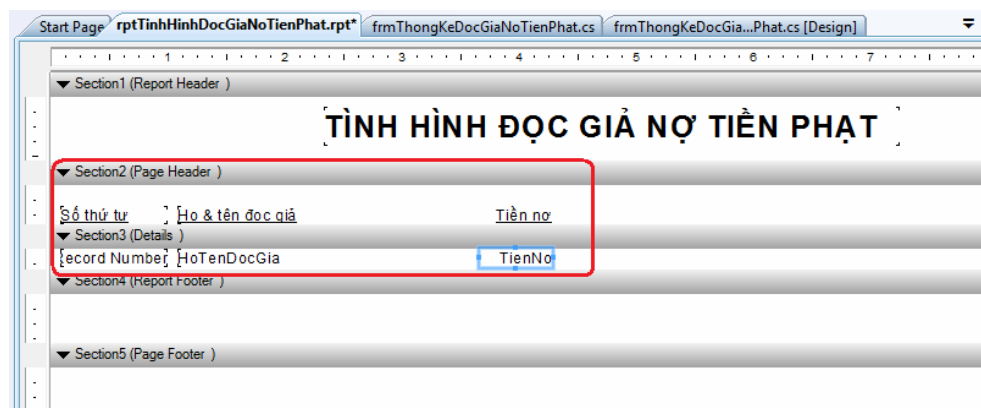
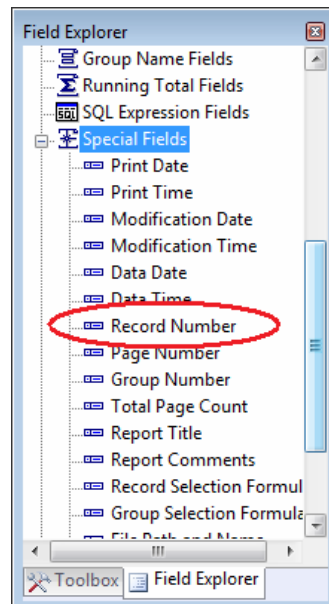
- Chọn dữ liệu cho Report như sau:



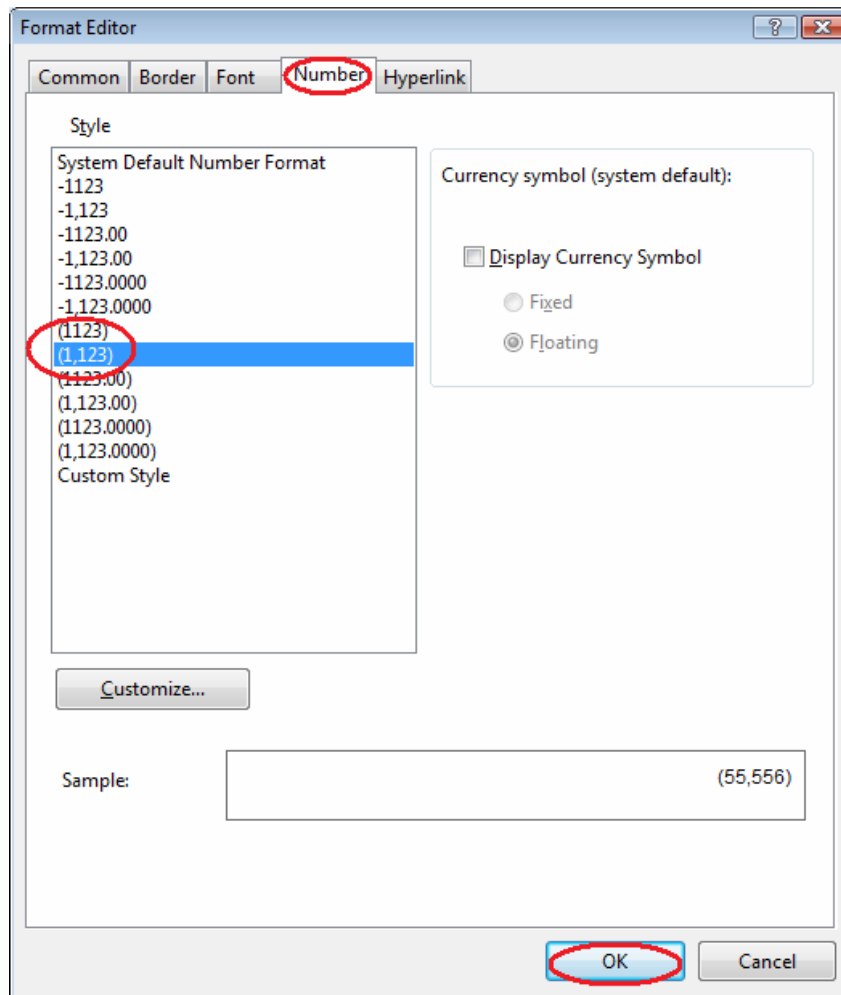
- Kết quả:



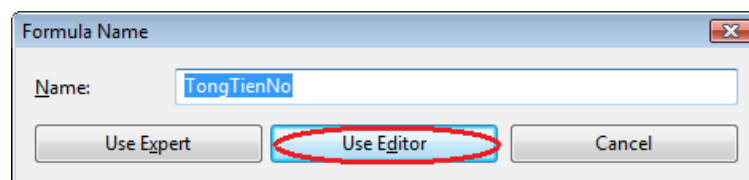
- Kéo thả các field cần thiết vào report như hình sau:



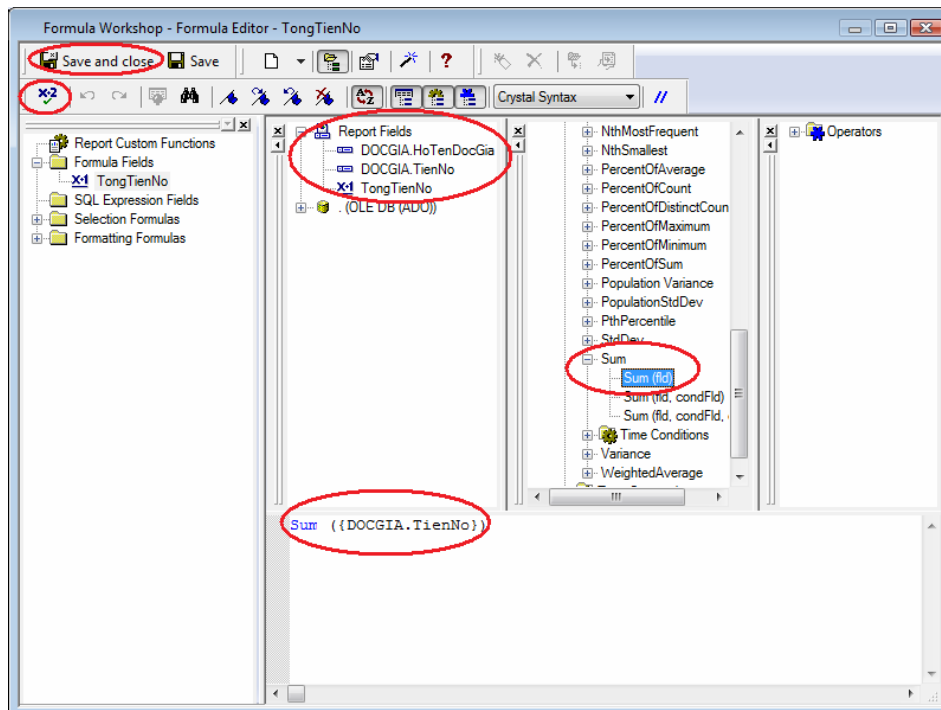
- Định dạng cho Field tiền nợ và tổng tiền nợ:



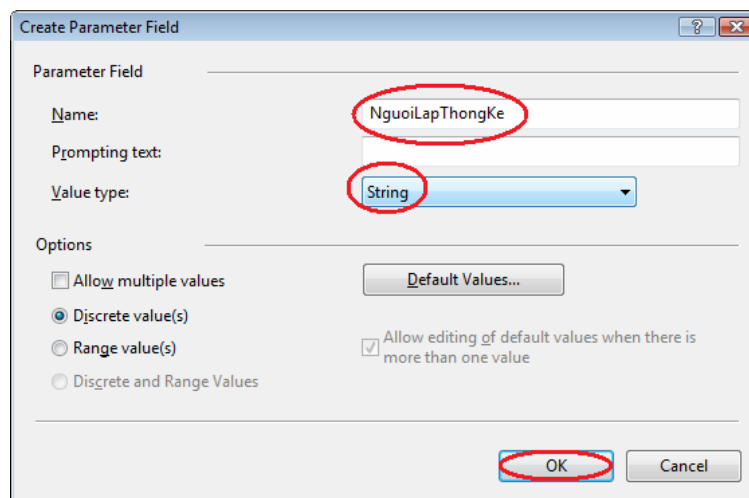
- Tạo Formula Field Tổng tiền nợ:
 - Right click lên Formula Fields chọn New...
 - Nhập vào tên Formula Field: TongTienNo



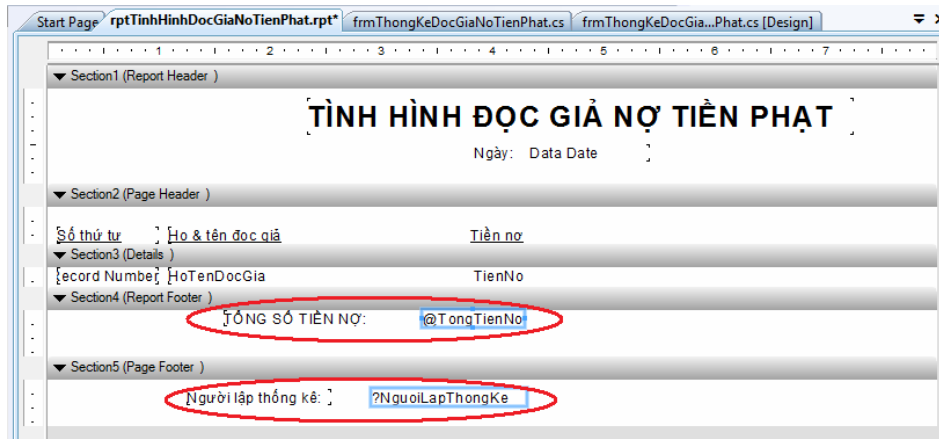
- Nhập vào công thức tính cho TongTienNo



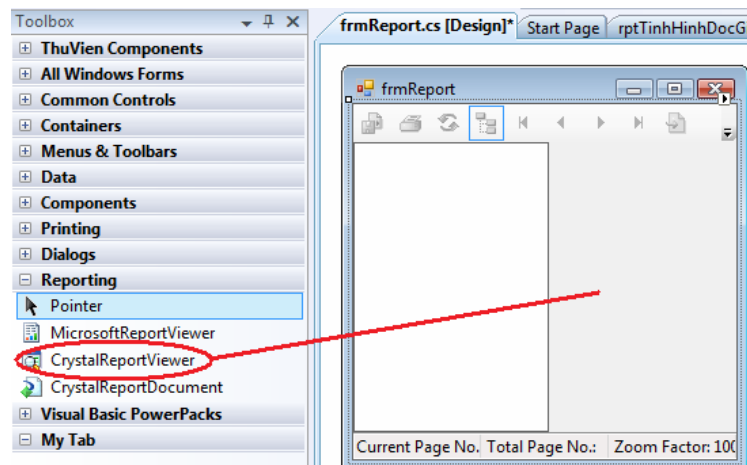
- Tạo Parameter Fields Người lập thống kê:
 - Right click lên Parameter Fields chọn New...
 - Nhập vào tên Parameter Field: NguoiLapThongKe



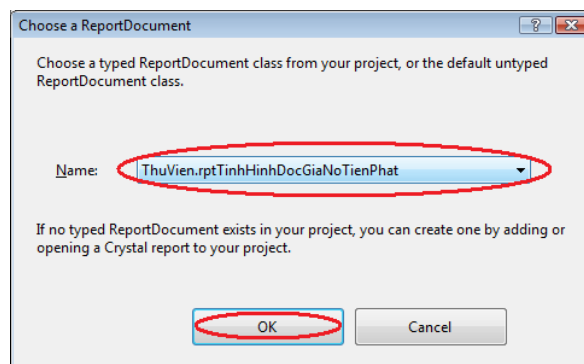
- Kéo thả Formula và Parameter + thêm 2 Text Object (TỔNG SỐ TIỀN NỢ:, Người lập thống kê:) vào report như hình sau:



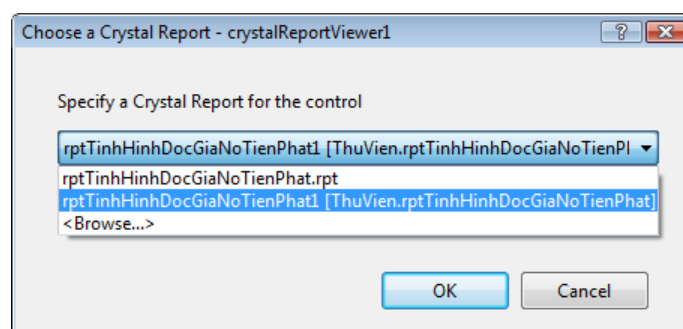
- Tạo Form frmReport để hiển thị report
- Kéo thả control CrystalReportViewer vào frmReport.



- Kéo thả control CrystalReportDocument vào frmReport và chọn Report Document như sau:



- Chọn Report Source cho CrystalReportViewer:



- Thêm button IN vào cuối frmThongKeTinhHinhDocGiaNoTienPhat.frm và viết code như sau:

frmThongKeDocGiaNoTienPhat

BÁO CÁO THỐNG KÊ TÌNH HÌNH ĐỌC GIẢ NỢ TIỀN PHẠT

Ngày: Tháng:

STT	Tên đọc giả	Tiền nợ

Tổng tiền nợ: 0

IN