

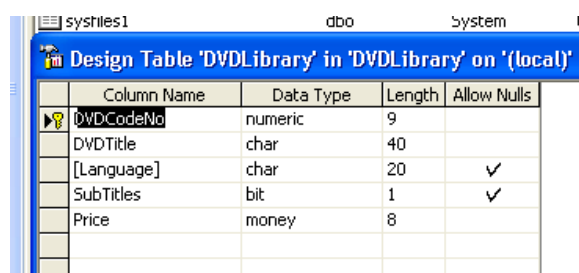
Buổi thực hành 3

TRUY CẬP CƠ SỞ DỮ LIỆU VỚI ADO.NET

Bài 1: Trong phần này chúng ta sẽ tìm hiểu cách thức làm việc với ADO.NET. Đây là một tập hợp các lớp cũng như các giao diện cho phép truy cập dữ liệu trong .NET Framework. Ở đây ta sẽ sử dụng ADO.NET để nối kết với cơ sở dữ liệu SQL Server, lấy dữ liệu và hiển thị chúng trên form.

Các bước thực hiện:

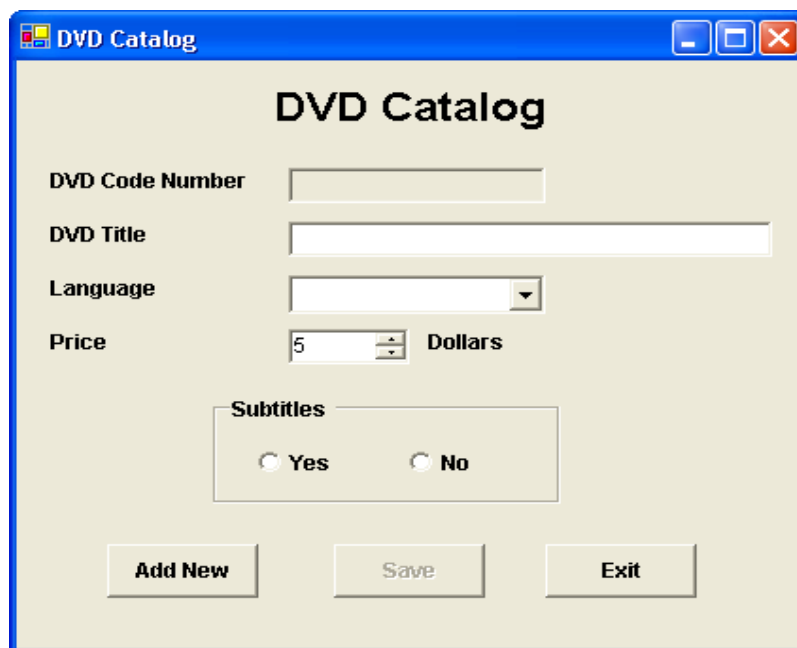
1. Trong SQL Server tạo một cơ sở dữ liệu tên là DVDLibrary.
2. Thêm một bảng tên DVDLibrary với cấu trúc như sau (Hình 4.1):



Column Name	Data Type	Length	Allow Nulls
DVDCodeNo	numeric	9	
DVDTitle	char	40	
[Language]	char	20	✓
SubTitles	bit	1	✓
Price	money	8	

Hình 4.1: Cấu trúc bảng DVDLibrary

3. Thêm vài dòng dữ liệu vào bảng.
4. Tạo một ứng dụng WinForm tên prjDVD.
5. Đổi tên của form chính thành frmDVDCatalog.
6. Thêm một số control vào và sửa đổi một số thuộc tính tương ứng (bảng 4.3) sao cho ta được giao diện:



DVD Catalog

DVD Code Number

DVD Title

Language

Price Dollars

Subtitles ☐ Yes ☐ No

Hình 4.2: Giao diện ứng dụng

Control	Property	Value
Form	Name Text	frmDVDCatalog DVD Catalog
Label	Name Text	lblHeading DVD Catalog
Label	Name Text	lblDVDCodeNo DVD Code Number
Label	Name Text	lblTitle DVD Title
Label	Name Text	lblLang Language
Label	Name Text	lblPrice Price
Label	Name Text	lblDollars Dollars
GroupBox	Name Text	grpSubTitles Subtitles
RadioButton	Name Text Checked	radYes Yes true
RadioButton	Name Text Checked	radNo No false
ComboBox	Name Collection	cboLang English Spanish French Chinese Russian
NumericUpDown	Name Maximum Minimum Value	updPrice 150 5 5
TextBox	Name ReadOnly Text	txtCodeNo true
TextBox	Name Text	txtTitle
Button	Name Text	btnAddNew AddNew
Button	Name Text	btnSave Save
Button	Name Text	btnExit Exit

Hình 4.3: Giá trị một số thuộc tính của các controls

7. Chuyển sang cửa sổ Code View, tham chiếu đến một số không gian tên:

```
using System.Data;
using System.Data.SqlClient;
```

8. Khai báo những biến sau trong phần khai báo của form:

```
private SqlConnection conn;
private SqlCommand commd;
private string sqlstr, lang;
private int codeno, subtitles;
decimal price;
```

9. Viết đoạn mã sau vào phương thức xử lý sự kiện Load của form:

```
private void frmCatalog_Load(object sender, System.EventArgs e)
{
    try
    {
        conn = new SqlConnection("Server=HoaiBao\\HoaiBao;Initial
            Catalog=DVDLibrary;User Id=user;pwd=user;");
        commd = new SqlCommand("Select * From DVDLibrary",conn);
        conn.Open();
        btnSave.Enabled = false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + ex.StackTrace);
    }
}
```

10. Khi button AddNew được click, một đối tượng SqlCommand được tạo ra thực hiện câu truy vấn lấy về mã số lớn nhất của DVD Code No từ bảng from the DVDLibrary. Nếu bảng rỗng, một ngoại lệ được tạo ra. Vì thế ta phải bắt lại ngoại lệ này nhằm kiểm tra khả năng bảng rỗng (lúc đó giá trị của biến codeno được gán là 0). Sau đó giá trị của codeno tăng 1 để có giá trị hiện hành của codeno. Kế tiếp các trường của form được xóa trắng nhờ vào việc gọi thực thi phương thức ResetFields() để một mẫu tin mới được thêm vào và kích hoạt button Save.

Sự kiện btnAddNew_Click:

```
private void btnAddNew_Click(object sender, System.EventArgs e)
{
    //calculate new Code
    commd = new SqlCommand("Select max(DVDCodeNo) From
        DVDLibrary",conn);
    try
    {
        codeno = Convert.ToInt32(commd.ExecuteScalar());
        btnAddNew.Enabled = false;
        btnSave.Enabled=true;
    }
    catch (Exception ex)
    {
        codeno=0;
        MessageBox.Show(ex.Message);
    }

    codeno++;
    txtCodeNo.Text = codeno.ToString();
    ResetFields();
}
```

11. Sự kiện Click của button btnExit được xử lý để đóng lại nối kết dữ liệu và kết thúc ứng dụng:

```
private void btnExit_Click(object sender, System.EventArgs e)
{
    conn.Close();
    Application.Exit();
}
```

12. Sự kiện btnSave_Click nhằm thêm một dòng dữ liệu mới vào bảng. Bởi vì đây không phải là một câu truy vấn lấy về dữ liệu mà là một câu truy vấn hành động nên ta sử dụng phương thức ExecuteNonQuery của lớp SqlCommand:

```
private void btnSave_Click(object sender, System.EventArgs e)
{
    try
    {
        sqlstr = "Insert Into DVDLibrary Values(" +
            codeno.ToString() + ", '" + txtTitle.Text + "', '" +
            lang + "', " + subtitles.ToString() +
            ", " + price.ToString() + ")";
        MessageBox.Show(sqlstr);
        commd = new SqlCommand(sqlstr, conn);
        commd.ExecuteNonQuery();
        MessageBox.Show("Record has been add successfully");
        ResetFields();
        btnExit.Focus();
        btnAddNew.Enabled = true;
        btnSave.Enabled=false;
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message + ex.StackTrace);
    }
}
```

13. Sự kiện ValueChanged của NumericUpDown được xử lý:

```
private void updPrice_ValueChanged(object sender,
    System.EventArgs e)
{
    price = updPrice.Value;
}
```

14. Sự kiện SelectedIndexChanged của ComboBox được xử lý:

```
private void cboLang_SelectedIndexChanged(object sender,
    System.EventArgs e)
{
    lang = cboLang.SelectedItem.ToString();
}
```

15. Sự kiện CheckedChanged của RadioButton được xử lý:

```
private void radYes_CheckedChanged(object sender,
    System.EventArgs e)
{
    subtitles = 1;
}

private void radNo_CheckedChanged(object sender,
    System.EventArgs e)
{
    subtitles = 0;
}
```

16. Đoạn mã cho phương thức ResetFields:

```
private void ResetFields()
{
    txtTitle.Text = "";
    price = updPrice.Value=5;
    lang=cboLang.Items[0].ToString();
    cboLang.Text = "";
    radYes.Checked = true;
    radNo.Checked= false;
    btnSave.Enabled=true;
}
```

17. Lưu, biên dịch và thực thi dự án. Một thí dụ mẫu cho việc thêm vào một mẫu tin như trong hình 4.4:

Hình 4.4: Kết quả thực thi chương trình

Bài 2: Sửa đổi ứng dụng ở bài 1 để thêm chức năng cập nhật và xóa các mẫu tin. Giao diện chương trình trở thành:



Hình 4.5: Giao diện mới của chương trình

Ứng dụng sửa đổi có một số chức năng:

1. Ban đầu các buttons Save & Cancel bị vô hiệu hóa (disable).
2. Khi người dùng click buttons Update hay Delete, textbox DVD Code Number sẽ bị thay thế bởi 1 Combobox để người dùng có thể lựa chọn giá trị các mã DVD (DVDCode) từ ComboBox. Nếu buttons Update (AddNew hay Delete) được click; các buttons AddNew, Update, Delete bị vô hiệu hóa; các button Save & Cancel trở nên có hiệu lực (enable).
3. Dựa trên mã DVD (DVDCode) từ Combobox, thông tin chi tiết về đĩa DVD có mã được lựa chọn sẽ được hiển thị chi tiết trên form.
4. Khi người dùng thay đổi giá trị của đĩa DVD hiện hành và click Save, các thông tin sửa đổi sẽ được lưu vào cơ sở dữ liệu; các buttons AddNew và Delete trở nên có hiệu lực (buttons Save & Cancel mất hiệu lực).
5. Tương tự khi người dùng click Delete, một thông điệp để người dùng xác nhận có muốn xóa thông tin của DVD hiện hành không? Nếu người dùng xác nhận là muốn xóa, thông tin về DVD hiện hành bị xóa khỏi cơ sở dữ liệu.
6. Khi người dùng click button AddNew sau khi đã thực hiện các thao tác như Update hay Delete; Combobox DVD Code No sẽ bị thay thế bởi textbox DVD Code Number.
7. Khi người dùng click button Cancel; không có sự thay đổi nào xảy ra; mọi thứ trở lại bình thường như cũ.

Gợi ý

- Tạo 1 Combobox và 1 Textbox cho mã DVD (DVD Code No) ở cùng 1 vị trí trên form để chúng chồng lấp lên nhau.
- Dùng thuộc tính `Visible` để thiết lập việc ẩn/hiện của 2 controls này.
- Dùng thuộc tính `Enabled` để cho phép các buttons có hiệu lực (enable) hay mất hiệu lực (disable).

Hình 4.6: Kết quả mới của ứng dụng

Hình 4.6 là một thí dụ khi người dùng click button Update và chọn mã DVD từ combobox (Các buttons AddNew, Update, Delete bị mất hiệu lực và các buttons Save, Cancel có hiệu lực).

Bài 3: Tạo một cơ sở dữ liệu tên Results trong SQLServer và 03 bảng có cấu trúc như trong hình 4.7, 4.8, 4.9. Tạo một ứng dụng Winforms để hiển thị thông tin từ các bảng này trong 1 DataGridView. Bên cạnh đó, ứng dụng còn có chức năng tìm kiếm thông tin của 1 sinh viên với mã sinh viên nhập từ bàn phím

Column Name	Data Type	Length	Allow Nulls
StudentCode	char	4	
StudentName	char	40	✓

Hình 4.7: Cấu trúc bảng Students

Column Name	Data Type	Length	Allow Nulls
ExamCode	char	10	
ExamName	char	20	✓
ExamDate	datetime	8	✓

Hình 4.8: Cấu trúc bảng Exams

Column Name	Data Type	Length	Allow Nulls
StudentCode	char	4	✓
ExamCode	char	10	✓
Score	int	4	✓

Hình 4.9: Cấu trúc bảng Result

Giao diện của ứng dụng có thể như hình 4.10:

The screenshot shows a window titled "Exam Results". It has a search bar with a dropdown menu labeled "Student Code" and a "Search" button. Below the search bar is a table with the following data:

	StudentCode	StudentName	ExamCode	ExamName	Score
	03	Alice	NET01	Introduction to NET	33
	03	Alice	NET02	Winform	55
	06	Zidane	JAV01	Java Basics	50
	03	Alice	JAV01	Java Basics	60
	06	Zidane	JAV02	Advanced Java	80
	06	Zidane	NET01	Introduction to NET	57
	06	Zidane	NET02	Winform	67
	07	Ballack	JAV01	Java Basics	50
	07	Ballack	JAV02	Advanced Java	45
	07	Ballack	NET01	Introduction to NET	55
	07	Ballack	NET02	Winform	75
*					

Hình 4.10: DataGridView hiển thị thông tin từ 03 bảng

Khi người dùng lựa chọn một mã sinh viên trong combobox, kết quả của sinh viên đó sẽ được hiển thị như hình 4.11

The screenshot shows the same "Exam Results" window, but the "Student Code" dropdown is now set to "07". The table displays only the results for student code 07:

	StudentCode	StudentName	ExamCode	ExamName	Score
▶	07	Ballack	JAV01	Java Basics	50
	07	Ballack	JAV02	Advanced Java	45
	07	Ballack	NET01	Introduction to NET	55
	07	Ballack	NET02	Winform	75
*					

Hình 4.11: Tìm kiếm kết quả 1 sinh viên

Gợi ý: Dùng lớp DataAdapter và DataSet.