

**School of Information Technology**

Course : Diploma in Business Informatics

Subject : ITP282 - Enterprise Application Development & Project

AY / Sem : 2018 S2

**Lab 5d**: Updating Data Using Databinding

OBJECTIVES:

By the end of this Practical students should be able to:

1. Use Grid View to display, update and delete data selectively

**Lab 5d: Updating Data Using Databinding**

**Updating, deleting and inserting records in Grid view**

Grid View also has the capabilities to perform updates, deletes and inserts.

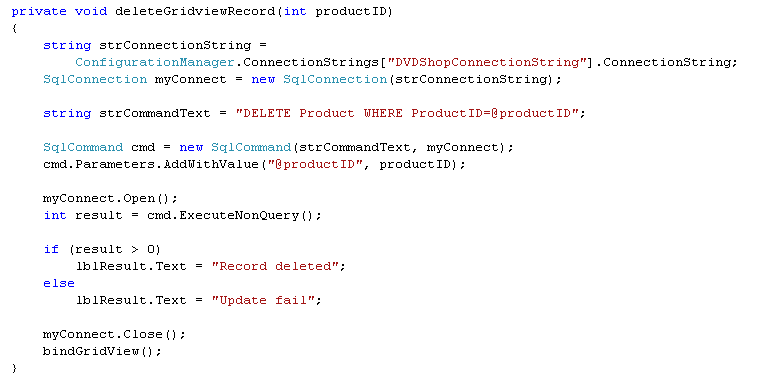
|  |  |
| --- | --- |
| **SQL Syntax** | **Example** |
|  |  |
|  |  |
|  |  |

In all three cases, we make use of the ExecuteNonQuery method of SqlCommand class to perform operation. An example for delete operation is shown here:

SQL Delete Statement

Specify which record to delete

Perform operation



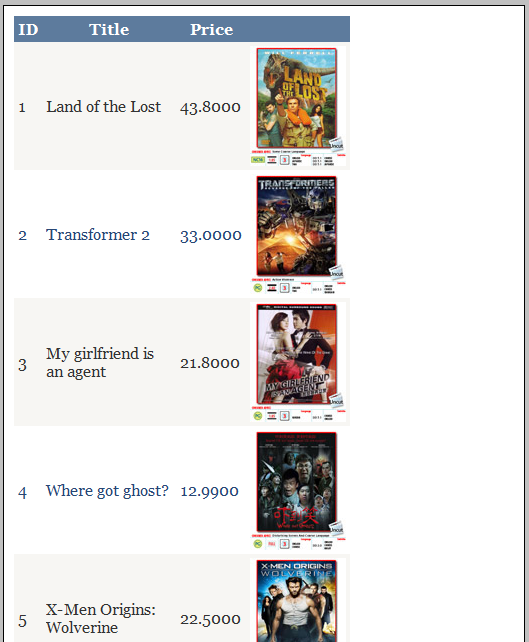
**Exercise 1: Updating and deleting records in Grid View**

* 1. Copy **Lab5d** into your computer. Open up **Ex1.aspx**. A Grid View ID=grdMovie and codes to retrieve using data reader and bind to it is already completed for you. Go to source view. Notice that the following

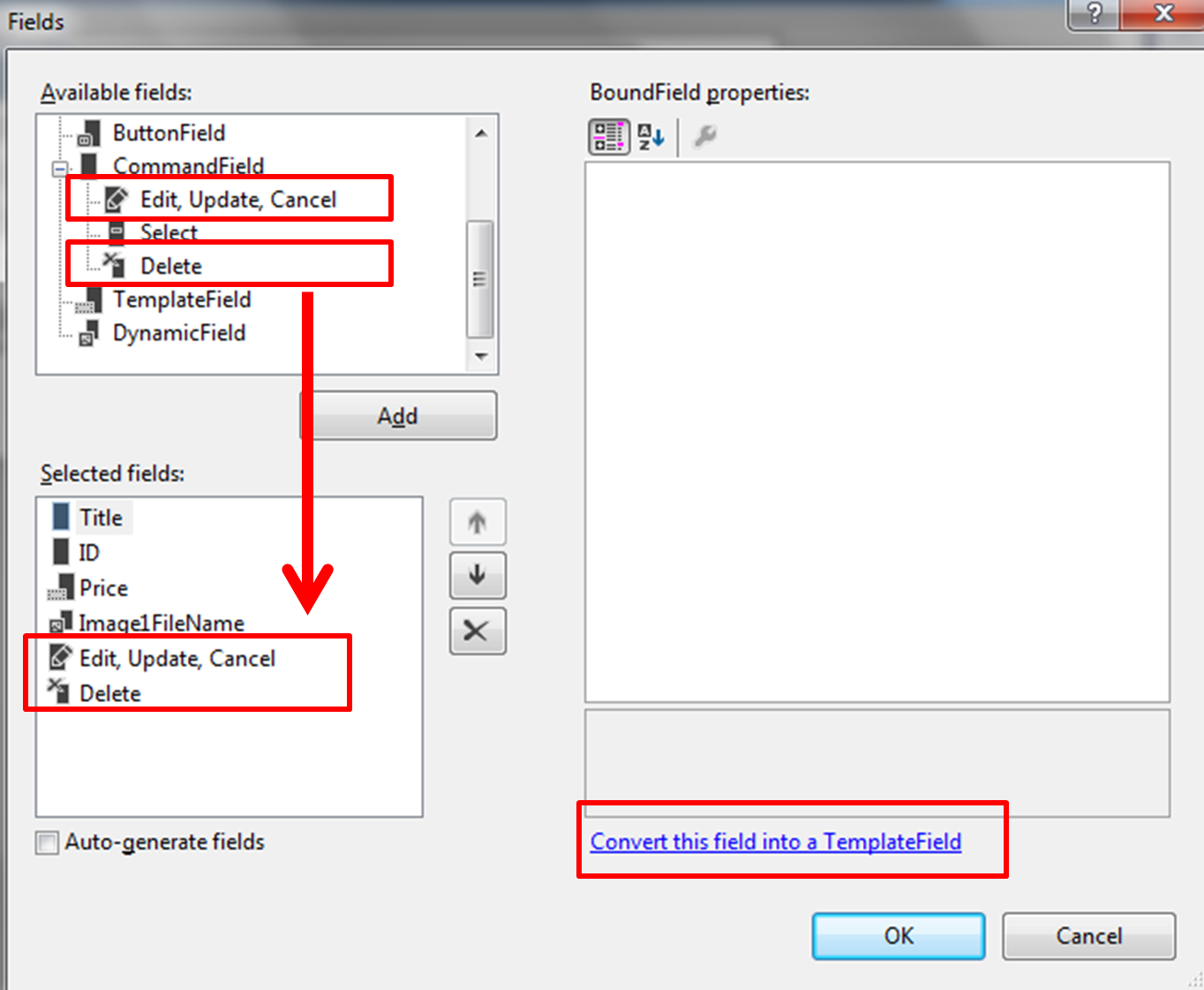
1. AutoGenerateColumns= False. Instead, There are three BoundField ( ProductID, Title and Price ) and one ImageField (Image1FileName) are manually created.
2. All the BoundFields are set to ReadOnly=True except Price. This is because, in this exercise, we only want to allow the user to edit the Price. Once a column is set to ReadOnly=True, it will not be editable, as seen in the sections below
3. DataKeyName = ProductID. This will be used later in the code to determine which record to update or delete.



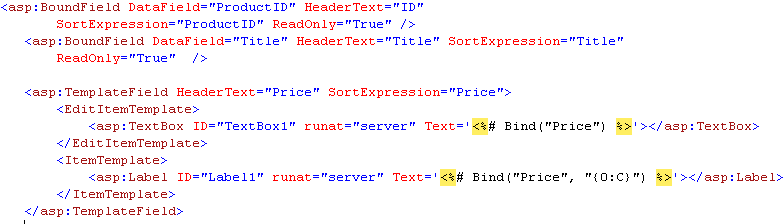
* 1. Run your program to make sure it is working.



* 1. Add two CommandFields – Edit, Update, Cancel and Delete – into the Selected fields list box.
  2. Since we want to make Price editable when the user selects the edit button, we need to convert this field into a Text Box when the edit button is selected. To do this, select Price under Selected Fields list box, and click on the link Convert this field into a TemplateField. Click OK.



* 1. Go to Source View. Notice that the BoundField Price is converted to a Text Box and Label. The Text Box will be displayed later to allow the user to edit when the Edit button is selected.



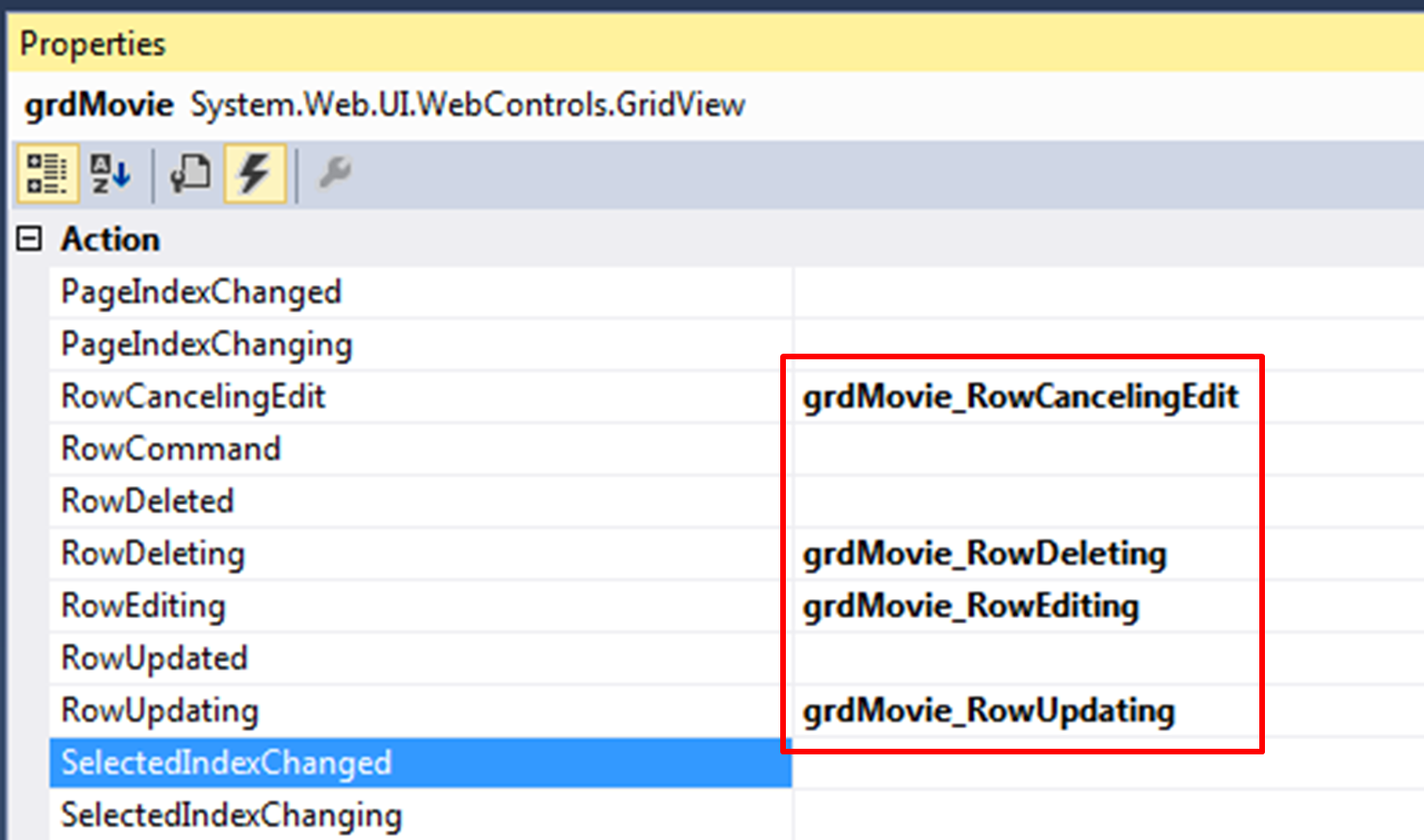
* 1. Rename the Text Box ID = **txtPrice**, and Label ID = **lblPrice**.



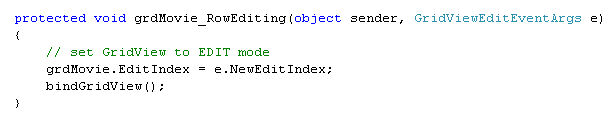
* 1. We now need to generate the respective event handlers when the Edit or Delete buttons are selected. The respective events are

|  |  |
| --- | --- |
| **Events** | **Description** |
| Row\_Editing | Raised when Edit button is selected |
| RowCancelingEdit | Raised when Cancel button is selected while in Edit mode |
| RowUpdating | Raised when Update button is selected while in Edit mode |
| RowDeleting | Raised when Delete button is selected |

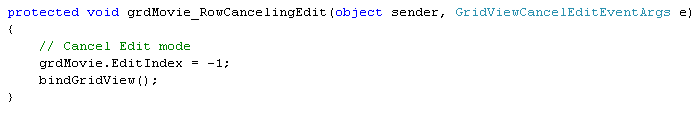
Generate these event handlers by double clicking on them in the Property Window of Grid View:



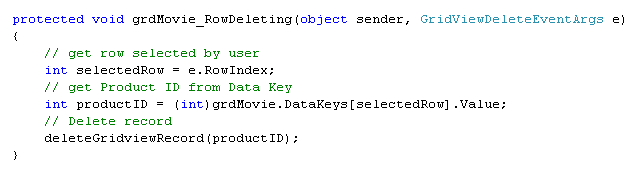
* 1. In the RowEditing event handler, set Grid View to Edit mode using following codes:



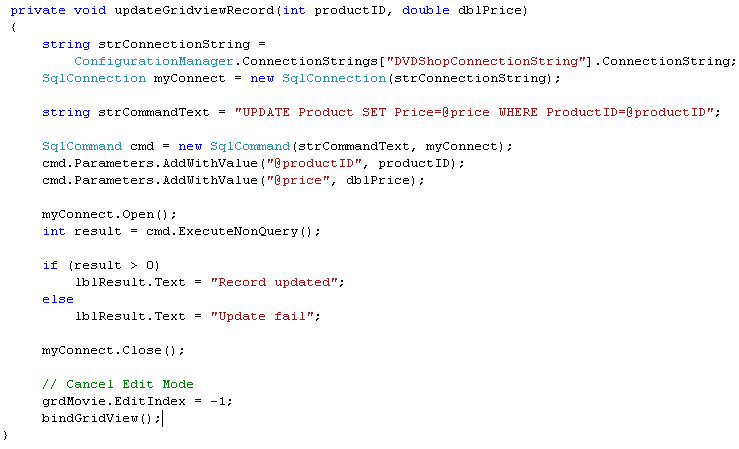
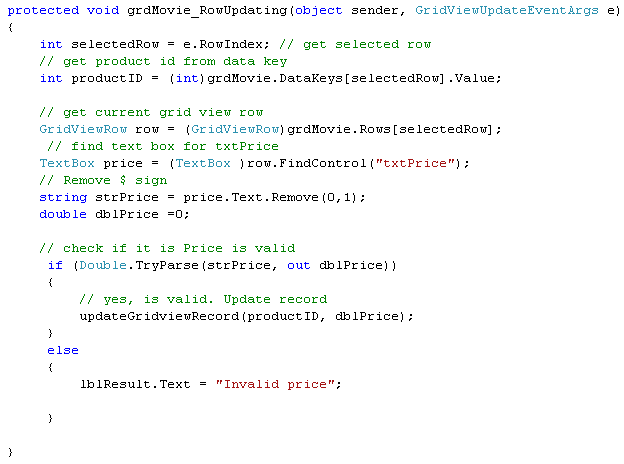
* 1. In the RowCancelingEdit event handler, cancel Edit mode using the following codes:



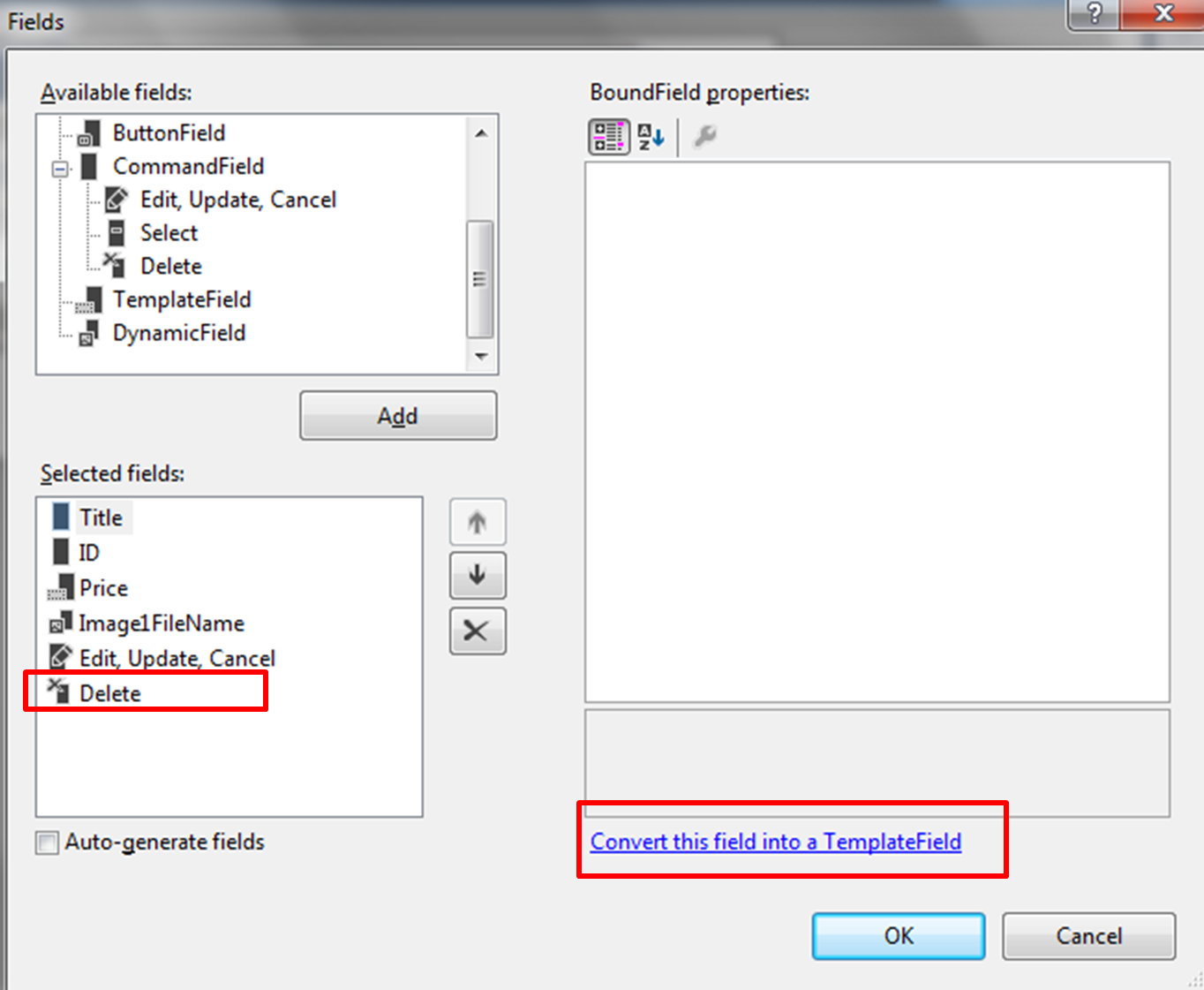
* 1. In RowDeleting event handler, use the following codes to retrieve the productID (via Data Key) to be deleted, and perform the delete operation.



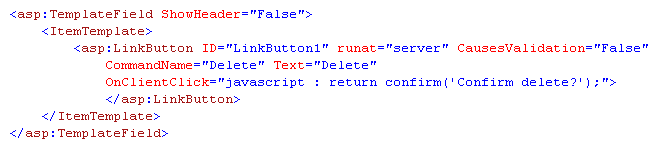
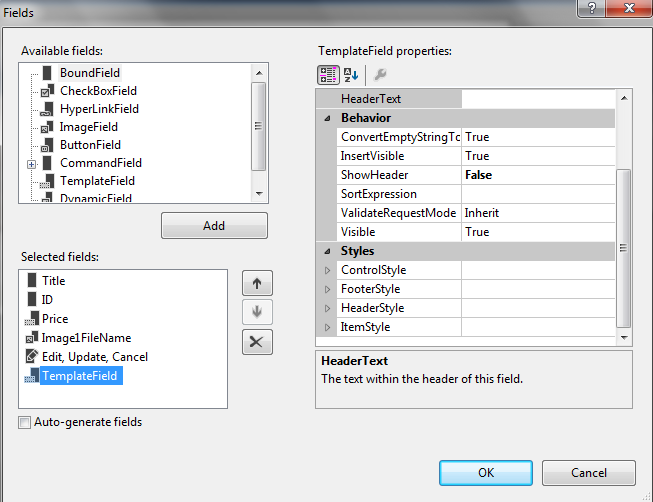
* 1. In the RowUpdating event handler, we need to get the new price value entered by the user via the Text Box and update it to the database:



* 1. Build and run your program. Test out the Edit, Delete, Update and Cancel operations.
  2. Notice that in the above program, when you delete a record, you are not prompted for a confirmation. To implement a java script confirmation, convert the Delete button into a TemplateField and add the following java script codes:



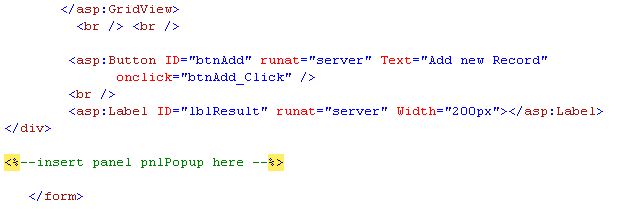
After Conversion:



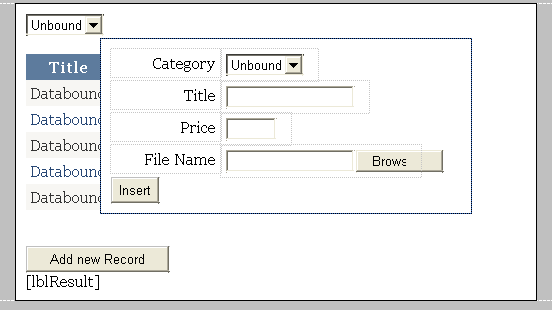
* 1. Build and run your application. Click on a Delete button and notice that a confirmation box appears.

**Exercise 2: Inserting records**

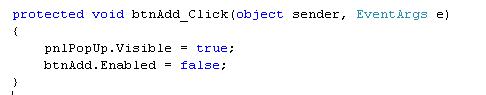
* 1. Continuing from last exercise, add in a button ID=btnAdd and a Panel ID= pnlPopup as shown below. You may find the codes for pnlPopup in pnlPopUp.txt. Insert the codes just below the comment <%--insert panel pnlPopup here --%>.



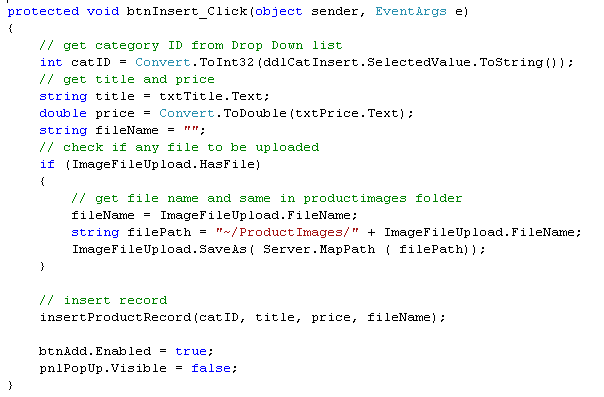
Your form should look as follows. Notice that there is a Drop Down List Box, two Text Boxes, one Button to insert a record and one File Upload to upload an image of the movie.



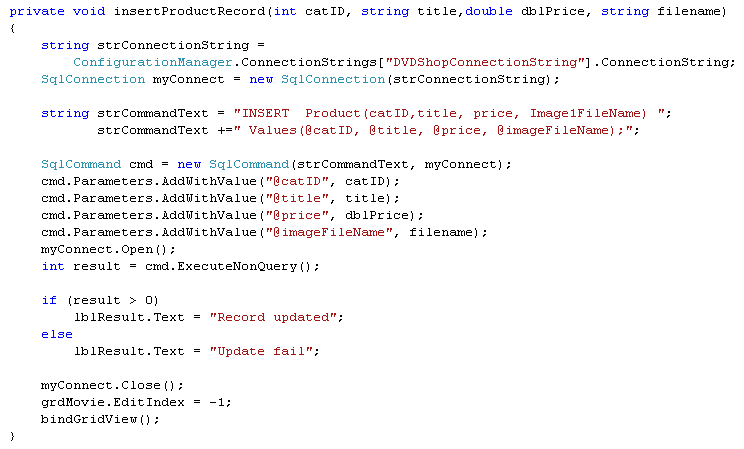
* 1. When page first loads, bind the Drop Down List Box using a Data Reader using the SQL statement "SELECT CategoryID, Name from Category". Refer to **Practical 5c Ex 1** if you are not sure.
  2. When the Add New Record button is clicked, make the Panel visible and the Add New button not visible:



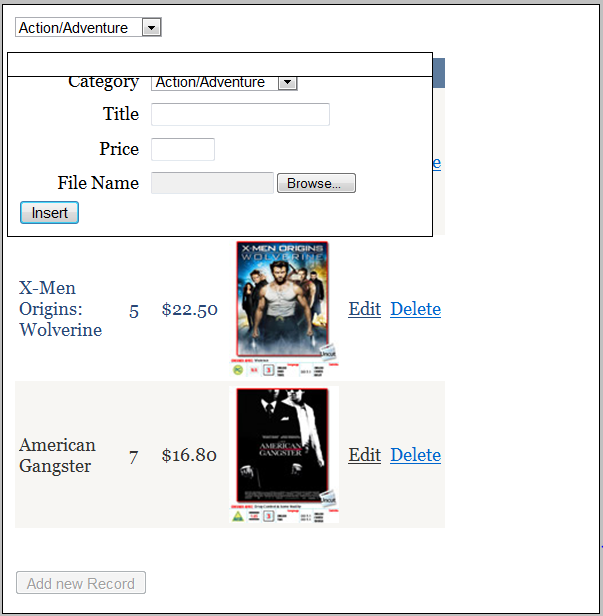
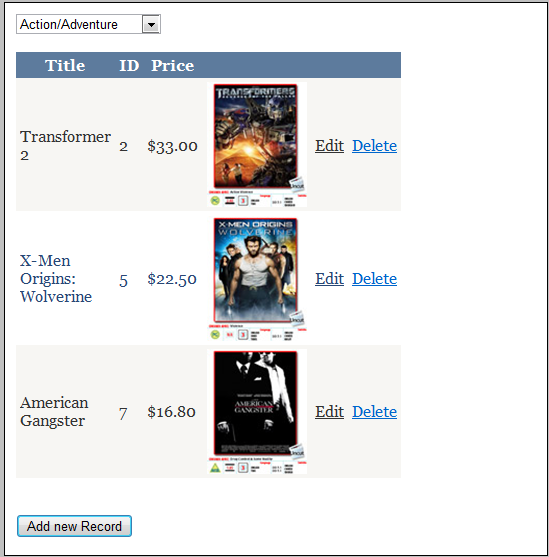
* 1. When the user clicks on the Insert button, read in all the values entered and selected by the user and insert into the database. As you can see below, the category ID is retrieved from the Drop Down List box, while the price and title are read from the Text Boxes. The image file of the movie uploaded, if any, is uploaded and saved in the ProductImages folder.



The new record will be created by calling insertProductRecord. Note that, to keep the code short, error handling is not shown here.

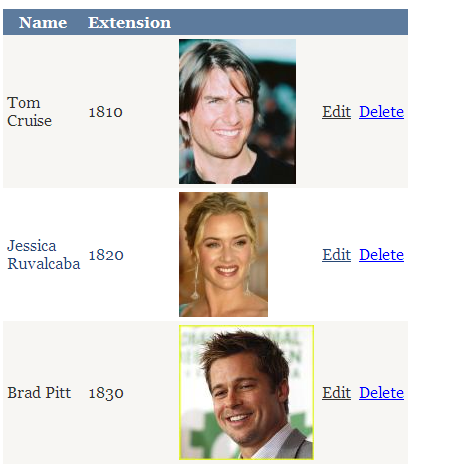


* 1. Build and run your application. Test your program by inserting a record and confirm from Server Explorer.

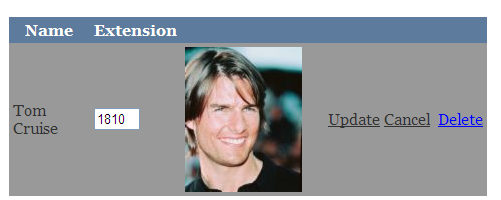


**(Optional) Exercise 3: Applying update, delete and insert using Grid View**

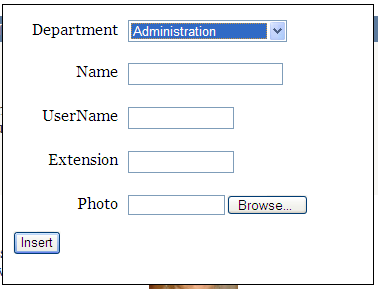
* 1. Open up **Ex3.aspx**. The relevant controls is already created for you. Create an application to display the Name, Extension and photo of all users. The database EmployeeDB.mdf is given to you.



* 1. When the user clicks on the Edit button, only the Extension is editable. Once in Edit mode, the user may update or cancel the operation. Check to make sure that only positive whole numbers are entered. The user may also delete a record.



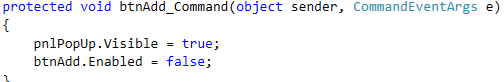
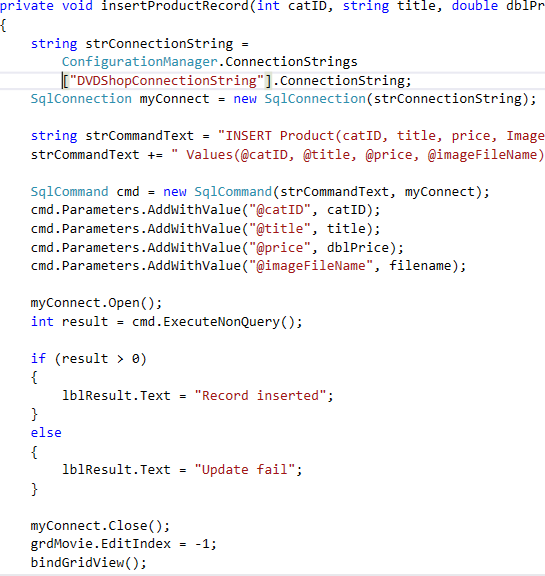
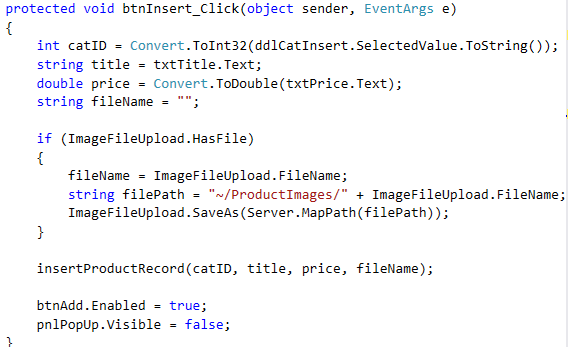
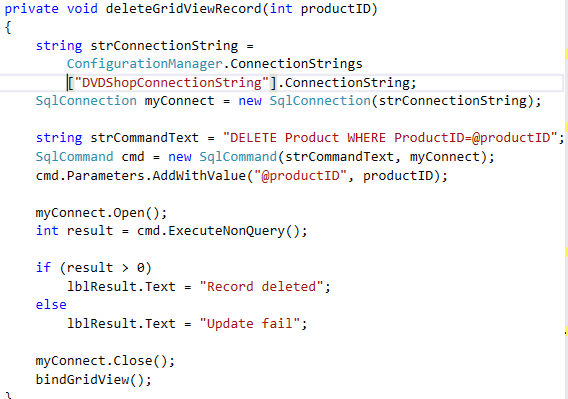
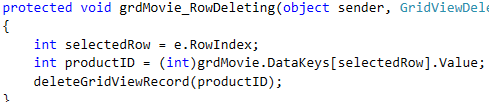
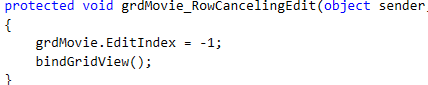
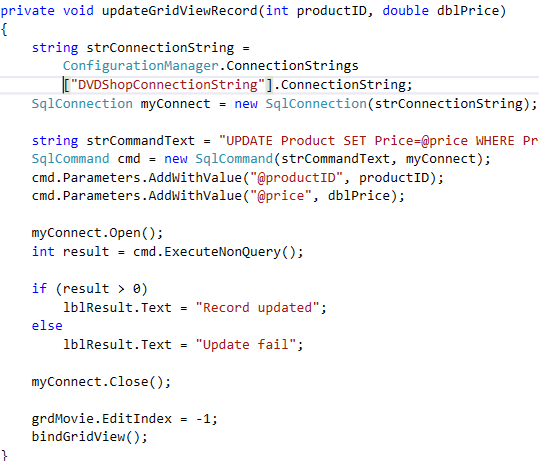
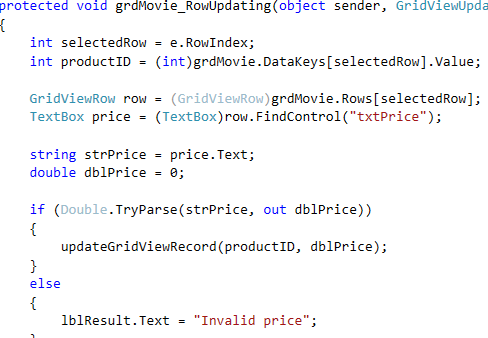
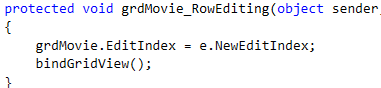
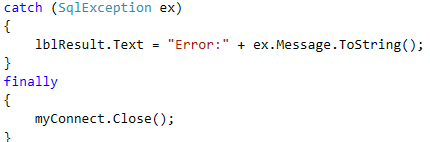
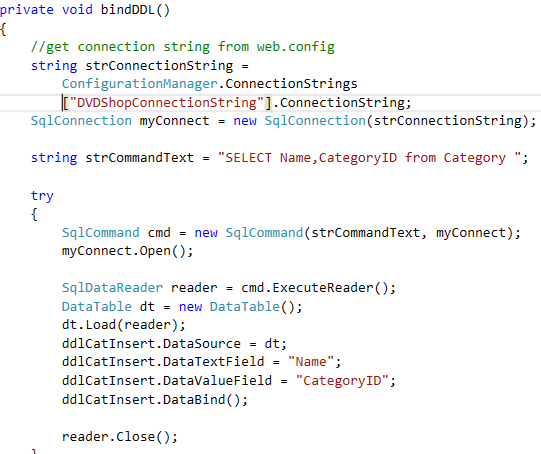
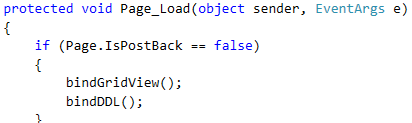
* 1. When the Insert button is clicked, a form is pop up to allow the user to enter the following to insert a new record:



* 1. Build and run your program. Test that all the functions are working.

**========== End ==========**

**EX1 & 2.aspx.cs**



**EX3aspx.cs**

