# How to Modify data in DataTable and Update

## Introduction

We have several ways to modify the data in DataTable. In this application, we will demonstrate how to use different ways to modify data in DataTable and update to the source.

1. We use SqlDataAdapter to fill the DataTables.

2. We set DataTable Constraints in DataTables.

4. We use DataTable Edits to modify data.

3. We use DataRow.Delete Method and DataRowCollection.Remove Method to delete the rows, and then compare them.

5. We use SqlDataAdapter to update the datasource.

## Building the Sample

Before you run the sample, you need to finish the following steps:

Step1. Please choose one of the following ways to build the database:

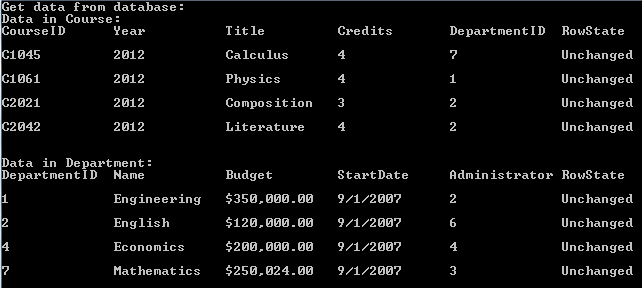
* Attach the database file MySchool.mdf under the folder \_External\_Dependecies to your SQL Server 2008 database instance.
* Run the MySchool.sql script under the folder \_External\_Dependecies in your SQL Server 2008 database instance.

Step2. Modify the connection string in the Project Properties->Settings according-> MySchoolConnectionString to your SQL Server 2008 database instance name.

## Running the Sample

Press F5 to run the sample, the following is the result.

First, we get the data from database.

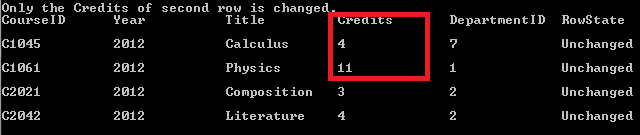


Second, we use DataTable Edits to edit the data.

a. We change two values in Credits column, and one will cause the following display and cancel the Edit.

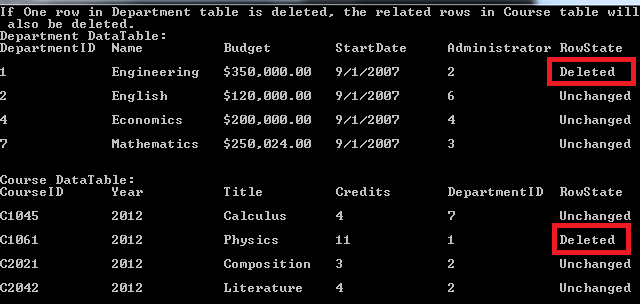


b. We change the first two values of Credits, but only the second value is changed.

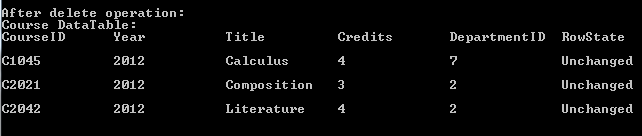


Third, we delete and remove the rows in Datatable.

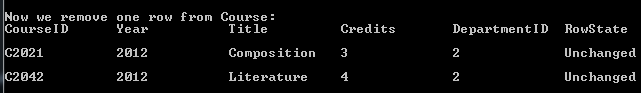
a. Because we set the foreign key constraint, the related rows in child table will be deleted when the rows in parent table are deleted.



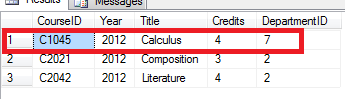
Then we can update the delete operations. Now the deleted row is removed from the DataTable and the database.



b. We can also remove the rows from DataTable, and the row isn’t exist in the table.



After update the delete operations, however, we can also find the row in database. The Remove operation only remove the rows from DataTable, and doesn’t change the value in Database.



## Using the Code

1. Use SqlDataAdapter to get data.

|  |
| --- |
| -Code block start-  --C# code snippet start--  private static void GetDataTables(String connectionString,String selectString,  DataSet dataSet,params DataTable[] tables)  {  using (SqlDataAdapter adapter = new SqlDataAdapter())  {  adapter.SelectCommand = new SqlCommand(selectString);  adapter.SelectCommand.Connection = new SqlConnection(connectionString);  adapter.Fill(0, 0,tables);  foreach (DataTable table in dataSet.Tables)  {  Console.WriteLine("Data in {0}:",table.TableName);  ShowDataTable(table);  Console.WriteLine();  }  }  }  --C# code snippet end--  --VB code snippet start--  Private Shared Sub GetDataTables(ByVal connectionString As String, ByVal selectString As String,  ByVal dataSet As DataSet, ByVal ParamArray tables() As DataTable)  Using adapter As New SqlDataAdapter()  adapter.SelectCommand = New SqlCommand(selectString)  adapter.SelectCommand.Connection = New SqlConnection(connectionString)  adapter.Fill(0, 0, tables)  For Each table As DataTable In dataSet.Tables  Console.WriteLine("Data in {0}:", table.TableName)  ShowDataTable(table)  Console.WriteLine()  Next table  End Using  End Sub  --VB code snippet end--  -Code block end- |

2. Use DataTable Edits to modify the data.

|  |
| --- |
| -Code block start-  --C# code snippet start--  row.BeginEdit();  row["Credits"] = credits;  row.EndEdit();  --C# code snippet end--  --VB code snippet start--  row.BeginEdit()  row("Credits") = credits  row.EndEdit()  --VB code snippet end--  -Code block end- |

The following method will be invoked when the value in table is changed. If the new value of Credits is negative, we’ll reject the modify.

|  |
| --- |
| -Code block start-  --C# code snippet start--  private static void OnColumnChanged(Object sender, DataColumnChangeEventArgs args)  {  Int32 credits = 0;  // If Credits is changed and the value is negative, we'll cancel the edit.  if ((args.Column.ColumnName == "Credits")&&  (!Int32.TryParse(args.ProposedValue.ToString(),out credits)||credits<0))  {  Console.WriteLine("The value of Credits is invalid. Edit canceled.");  args.Row.CancelEdit();  }  }  --C# code snippet end--  --VB code snippet start--  Private Shared Sub OnColumnChanged(ByVal sender As Object,  ByVal args As DataColumnChangeEventArgs)  Dim credits As Int32 = 0  ' If Credits is changed and the value is negative, we'll cancel the edit.  If (args.Column.ColumnName = "Credits") AndAlso  ((Not Int32.TryParse(args.ProposedValue.ToString(), credits)) OrElse credits < 0) Then  Console.WriteLine("The value of Credits is invalid. Edit canceled.")  args.Row.CancelEdit()  End If  End Sub  --VB code snippet end--  -Code block end- |

3. Delete and remove rows from DataTable.

a. Delete rows

Create the foreign key constraint, and set the DeleteRule with Cascade.

|  |
| --- |
| -Code block start-  --C# code snippet start--  ForeignKeyConstraint courseDepartFK =  new ForeignKeyConstraint("CourseDepartFK",  department.Columns["DepartmentID"],  course.Columns["DepartmentID"]);  courseDepartFK.DeleteRule = Rule.Cascade;  courseDepartFK.UpdateRule = Rule.Cascade;  courseDepartFK.AcceptRejectRule = AcceptRejectRule.None;  course.Constraints.Add(courseDepartFK);  --C# code snippet end--  --VB code snippet start--  Dim courseDepartFK As New ForeignKeyConstraint("CourseDepartFK",  department.Columns("DepartmentID"),  course.Columns("DepartmentID"))  courseDepartFK.DeleteRule = Rule.Cascade  courseDepartFK.UpdateRule = Rule.Cascade  courseDepartFK.AcceptRejectRule = AcceptRejectRule.None  course.Constraints.Add(courseDepartFK)  --VB code snippet end--  -Code block end- |

We delete one row in department, and the related rows in Course table will also be deleted. And then we update Course table with the deleted operation, the row in database is also be deleted.

|  |
| --- |
| -Code block start-  --C# code snippet start--  department.Rows[0].Delete();  --C# code snippet end--  --VB code snippet start--  department.Rows(0).Delete()  --VB code snippet end--  -Code block end- |

b. Remove rows

We remove one row from course table. And then we update Course table with the deleted operation, we can still find the row in database. The Remove operation only remove the rows from DataTable, and doesn’t change the value in Database.

|  |
| --- |
| -Code block start-  --C# code snippet start--  course.Rows.RemoveAt(0);  --C# code snippet end--  --VB code snippet start--  course.Rows.RemoveAt(0)  --VB code snippet end--  -Code block end- |

## More Information

[DataTable Edits](http://msdn.microsoft.com/en-us/library/ch2aw0w6(VS.110).aspx)

[DataTable Constraints](http://msdn.microsoft.com/en-us/library/st1t2c35(VS.110).aspx)

[DataRow Deletion](http://msdn.microsoft.com/en-us/library/03c7a3zb(VS.110).aspx)

[DataRowCollection.Remove Method](http://msdn.microsoft.com/en-us/library/system.data.datarowcollection.remove(VS.110).aspx)

[DataRowCollection.IndexOf Method](http://msdn.microsoft.com/en-us/library/system.data.datarowcollection.indexof.aspx)

[DbDataAdapter.Update Method (DataSet)](http://msdn.microsoft.com/en-us/library/at8a576f(VS.100).aspx)