Database Programming

IIO10200 Tietokantaohjelmointi

Michal Zabovsky

Department of Informatics
Faculty of Management Science and Informatics
University of Zilina Slovak Republic

Presentation overview

- ❖Accessing data with ADO.NET
- ❖Connection object
- ❖Command object
- ❖DataAdapter object
- ❖Transaction
- ❖Examples

Accessing data with ADO.NET

Most of the applications that you write must access some sort of data store. ActiveX Data Objects .NET (ADO.NET) is the technology used in the .NET Framework for all database access. ADO is the set of COM components (DLLs) that allow to access databases, emails or filesystem.

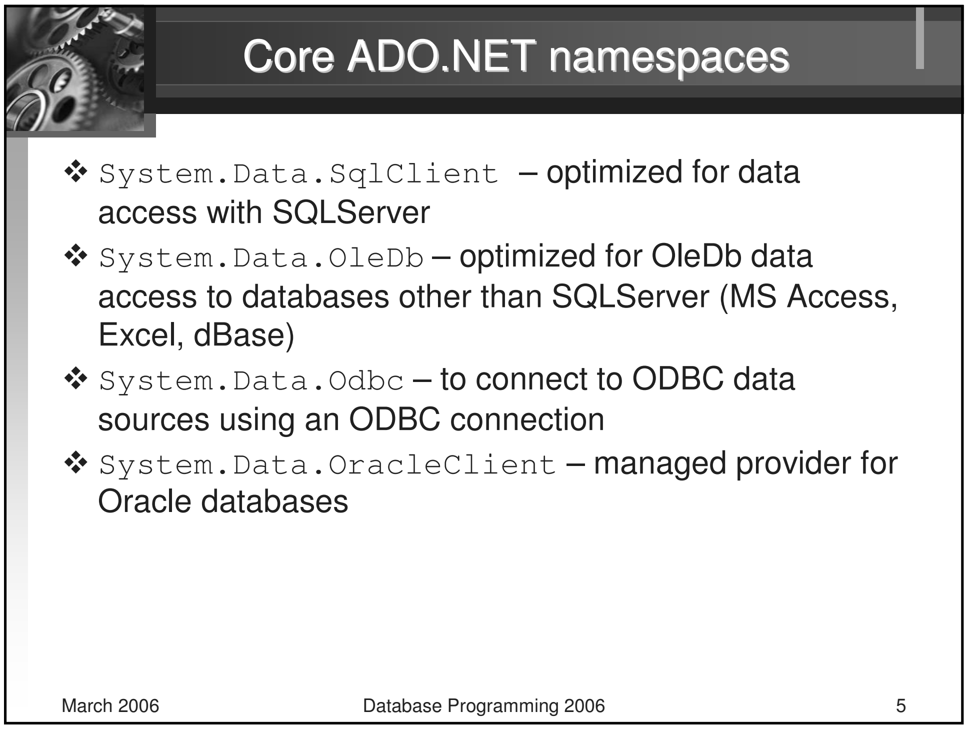
Before .NET
- ActiveX Data Objects (ADO) – designed for disconnected environment
- ODBC
- Native drivers

Note: There is still quite confusing behavior of Microsoft in the field of technology naming. You can meet different technologies for names e.g. ActiveX or COM.

.NET FCL for data access

- FCL – Framework Class Library
- `System.Data` – namespace has all the classes you need to access database or data store

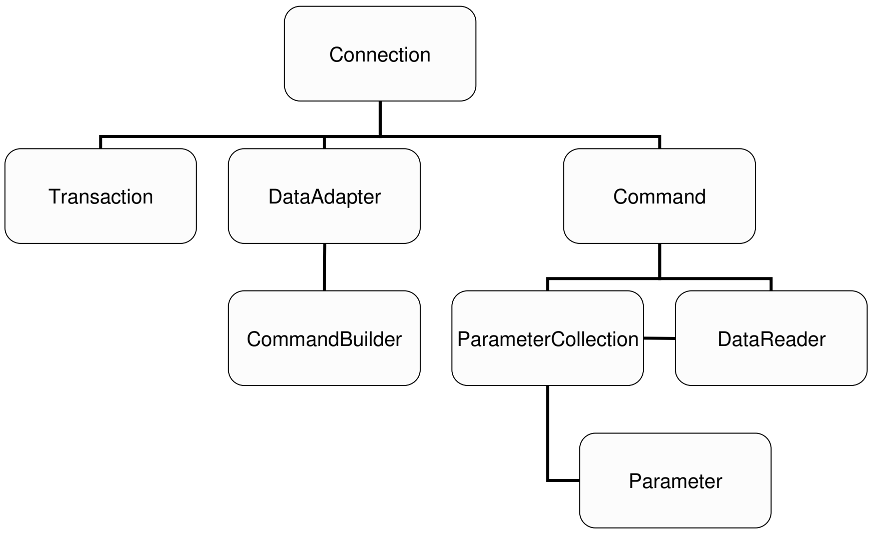
- Steps to accessing database:
  - Connecting to database
  - Selecting records from table
  - Executing action – adding, updating or deleting data



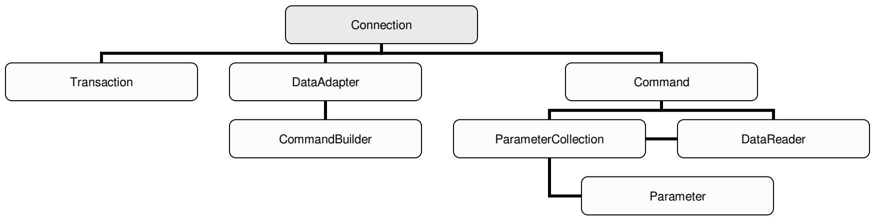
Core ADO.NET namespaces

- `System.Data.SqlClient` – optimized for data access with SQLServer
- `System.Data.OleDb` – optimized for OleDb data access to databases other than SQLServer (MS Access, Excel, dBase)
- `System.Data.Odbc` – to connect to ODBC data sources using an ODBC connection
- `System.Data.OracleClient` – managed provider for Oracle databases

Data access objects



Connection object



To work with any database, the first thing you must do is connect to it. In ADO.NET, you use the *Connection* object to connect to a database. There are three basic types of *Connection* object:

- SqlConnection
- OleDbConnection
- OdbcConnection

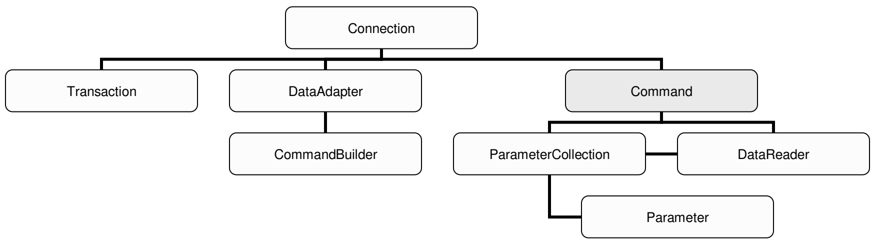
When you open a *Connection* object, you must always explicitly close it. Calling *Close* or *Dispose* on a *Connection* object ensures that the connection is sent back to the connection pool.

SqlConnection properties

Basic properties of *SqlConnection* object are:

- *ConnectionTimeout*
- *Database* - the name of the current database
- *DataSource* – name of the instance of SQL Server to which to connect
- *PacketSize*
- *ServerVersion* – the version of the instance of SQL server
- *State* – current state of the connection
- *WorkstationId* – database client id

Command object



The *Command* object is used to execute SQL statements against a database. The SQL statements can be ad hoc text or the name of a stored procedure in SQL Server.

- SqlCommand
- OleDbCommand
- OdbcCommand

The Command object can be created i two ways – by calling the *CreateCommand* method of a *Connection* object or by creating an instance of the *SqlCommand* or *OleDbCommand* and passing a valid *Connenction* object to the *Command* instance.

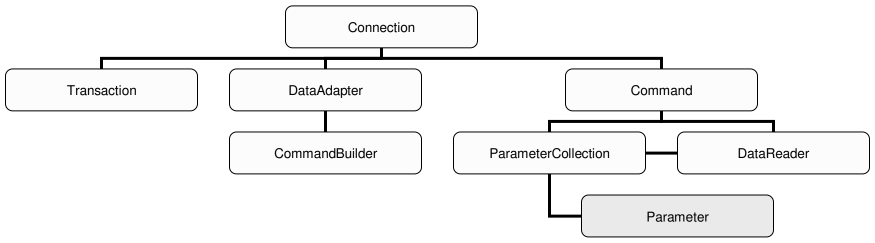
SqlCommand properties

- *CommandText* – SQL statement or stored procedure
- *CommandTimeout* – time before terminating an attempt to execute
- *CommandType* – indicates how the *CommandText* property is interpreted
- *Connection* – instance of the *Command* object
- *Parameters –SqlParameterConnection*
- *Transaction* – transaction in which the *SqlCommand* executes
- *UpdateRowSource* – how command results are applied to the *DataRow* when used by the *Update* method of *DataAdapter*

SqlCommand execute methods

- ❖ *ExecuteReader* – execute commands that return rows
- ❖ *ExecuteNonQuery* – execute command such as INSERT, DELETE, UPDATE and SET
- ❖ *ExecuteScalar* – retrieves a single value from database
- ❖ *ExecuteXmlReader* – sets *CommandText* to the *Connection* and builds an *XmlReader* object

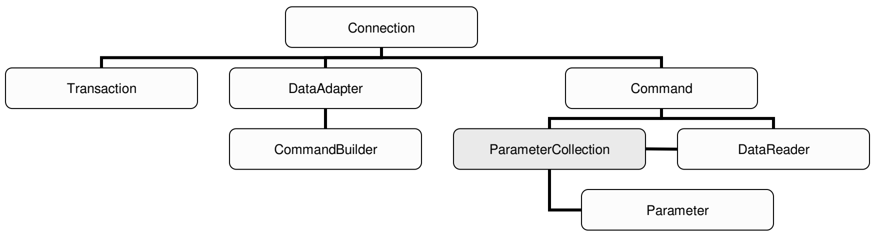
Parameter object



Object *Parameter* is used to passing parameter to a *Command* object. Parameter value can be passed to SQL command or to stored procedure.

- SqlParameter
- OleDbParameter
- OdbcParameter

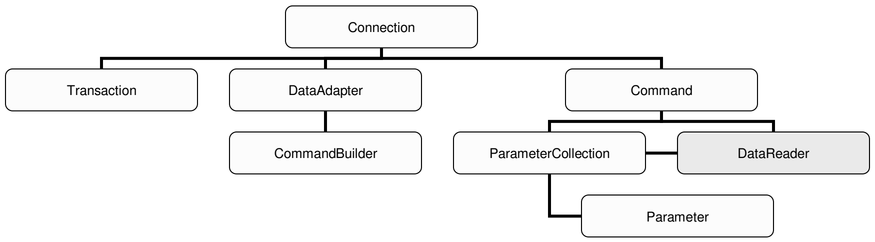
ParameterConnection object



Used for passing more than one *Parameter* object to a *Command* object.

- SqlParameterCollection
- OleDbParameterCollection
- OdbcParameterCollection

DataReader object



DataReader instance is used to read rows returned as the result of the *Command* object.

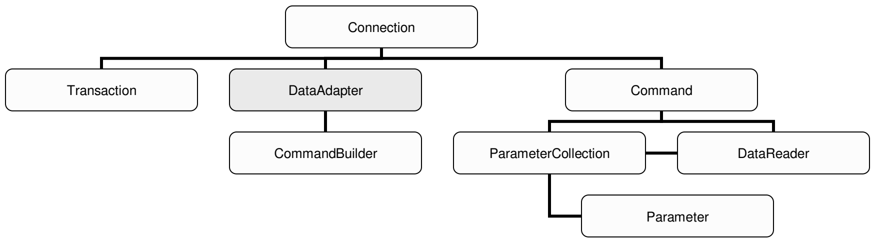
- SqlDataReader
- OleDbDataReader
- OdbcDataReader

DataReader is a forward-only set of records, so you can't move backward in the DataReader. Reading data by using *DataReader* is obviously faster than by *DataSet*.

SqlDataReader methods

- *GetSqlBinary* - Gets the value of the specified column as a *SqlBinary*
- *GetSqlBoolean* - Gets the value of the specified column as a *SqlBoolean*
- *GetSqlByte* - Gets the value of the specified column as a *SqlByte*
- *GetSqlDateTime* - Gets the value of the specified column as a *SqlDateTime*
- *GetSqlDecimal* - Gets the value of the specified column as a *SqlDecimal*
- *GetSqlDouble* - Gets the value of the specified column as a *SqlDouble*
- *GetSqlGuid* - Gets the value of the specified column as a *SqlGuid*
- *GetSqlInt16* - Gets the value of the specified column as a *SqlInt16*
- *GetSqlInt32* - Gets the value of the specified column as a *SqlInt32*
- *GetSqlInt64* - Gets the value of the specified column as a *SqlInt64*
- *GetSqlMoney* - Gets the value of the specified column as a *SqlMoney*
- *GetSqlSingle* - Gets the value of the specified column as a *SqlSingle*
- *GetSqlString* - Gets the value of the specified column as a *SqlString*

DataAdapter object



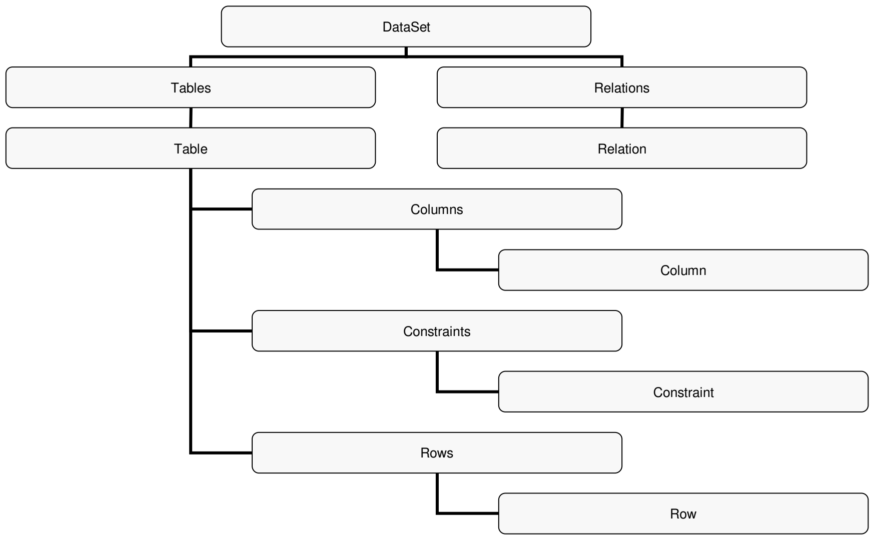
If you need more flexibility than a *DataReader* offers, you can use a *DataSet* object as a container for records from the database. The *DataSet*

- doesn't connect to a database
- simply holds data and table information in its *DataTables* collection
- data into a *DataSet* are loaded by a *DataAdapter*

The synchronization is provided by a *Connection* object.

- SqlDataAdapter
- OleDbDataAdapter
- OdbcDataAdapter

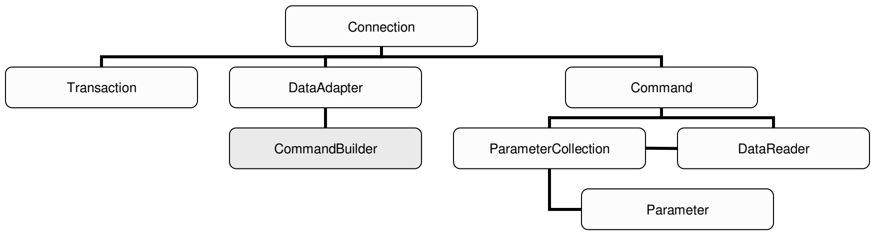
DataSet object hierarchy



Steps to fill DataSet

1. Build a connect string to database.
2. Create object *SqlConnection* and use prepared connect string with it.
3. Build a SELECT statement.
4. Create object *SqlCommand* and assign prepared SELECT statement to the *CommandText* property of this object.
5. Create object *SqlDataAdapter* and set the property *SelectedCommand* to the *SqlCommand* object.
6. Create *DataSet* object.
7. Use *Open ()* method of the *SqlConnection* object to open database connection.
8. Call *Fill ()* method of *SqlDataAdapter* object to reading rows from table and to save then into a *DataTable* object of the *DataSet* object.
9. Close the database connection by calling *Close ()* method of *SqlConnection* object.
10. Select *DataTable* object from the *DataSet* object.
11. By using *DataRow* object show columns for each row of *DataTable* object.

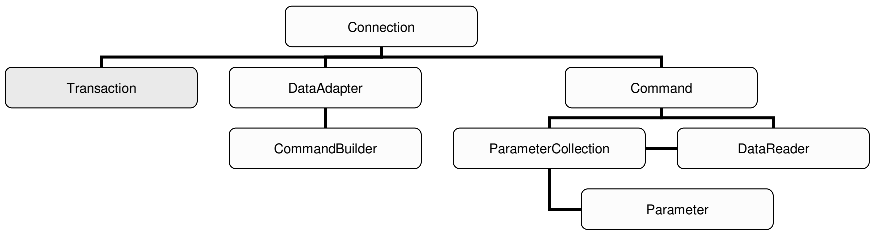
CommandBuilder object



The *CommandBuilder* object is used to create INSERT, UPDATE and DELETE commands automatically. These commands are synchronizing each change of a *DataSet* object with database. The synchronization is provided by a *DataAdapter* object.

- SqlCommandBuilder
- OleDbCommandBuilder
- OdbcCommandBuilder

Transaction object



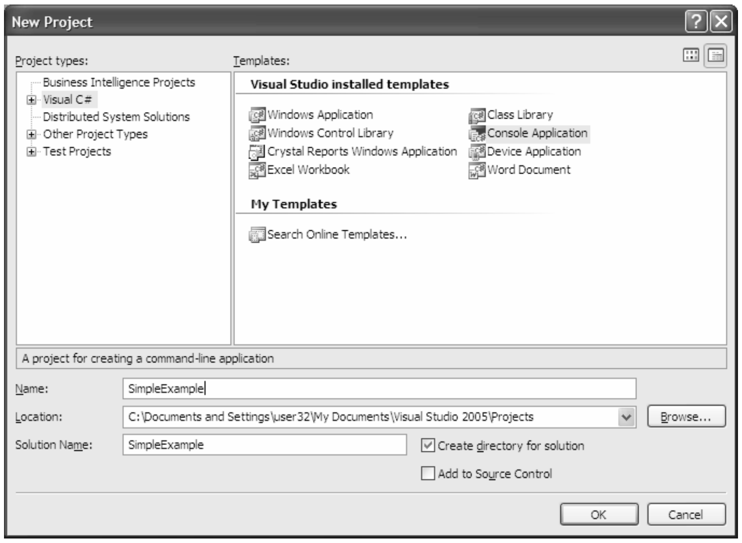
The *Transaction* object represents database transaction.

- SqlTransaction
- OleDbTransaction
- OdbcTransaction

Examples using Microsoft SQL

- *SimpleExample* – simple console based application. It reads data from table *Person.Contact* of database *AdventureWorks*. For the data reading is used instance of the *DataReader* object.
- *SelectIntoDataSet* – simple console application to demonstrate how to use *DataSet* object to store records from database.
- *DataGridApplication* – simple windows application showing data from table in the *DataGridView* component.
- *DataBindingsMsSQL* – demonstration of visual data bindings by using Visual Studio 2005 and Microsoft SQL database.

E: Simple example



A project for creating a command-line application

Name: SimpleExample

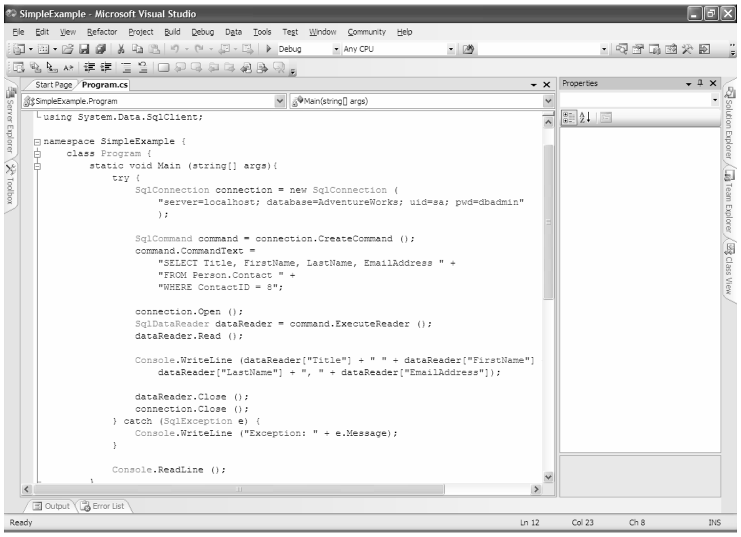
Location: C:\Documents and Settings\user32\My Documents\Visual Studio 2005\Projects

Solution Name: SimpleExample

☑ Create directory for solution

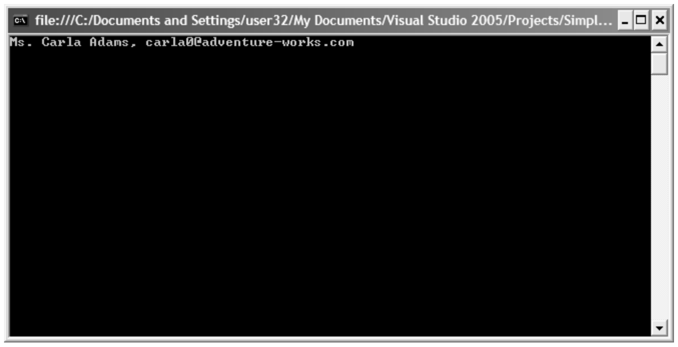
☐ Add to Source Control

E: Simple example



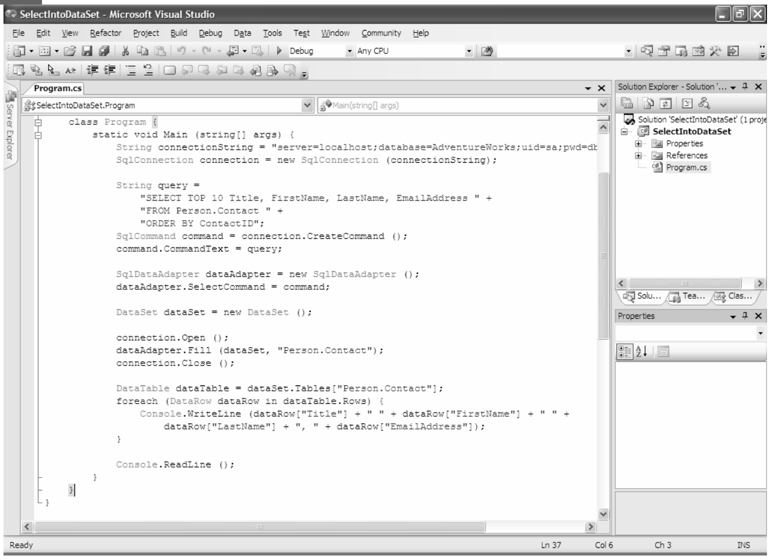
```
using System.Data.SqlClient;

namespace SimpleExample {
    class Program {
        static void Main (string[] args){
            try {
                SqlConnection connection = new SqlConnection (
                    "server=localhost; database=AdventureWorks; uid=sa; pwd=dbadmin"
                    );

                SqlCommand command = connection.CreateCommand ();
                command.CommandText =
                    "SELECT Title, FirstName, LastName, EmailAddress " +
                    "FROM Person.Contact " +
                    "WHERE ContactID = 8";

                connection.Open ();
                SqlDataReader dataReader = command.ExecuteReader ();
                dataReader.Read ();

                Console.WriteLine (dataReader["Title"] + " " + dataReader["FirstName"]
                    dataReader["LastName"] + ", " + dataReader["EmailAddress"]);

                dataReader.Close ();
                connection.Close ();
            } catch (SqlException e) {
                Console.WriteLine ("Exception: " + e.Message);
            }

            Console.ReadLine ();
        }
```

E: SimpleExample


file:///C:/Documents and Settings/user32/My Documents/Visual Studio 2005/Projects/Simpl...

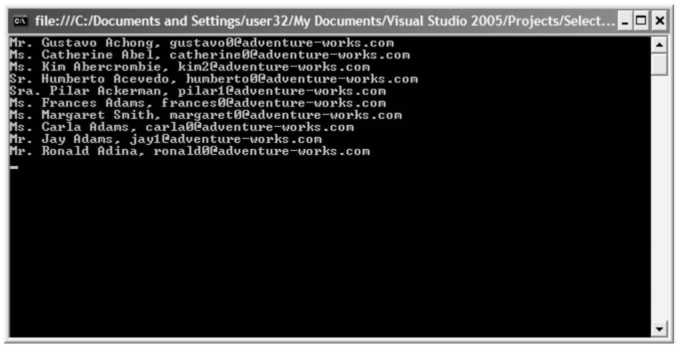
```
Ms. Carla Adams, carla0@adventure-works.com
```

E: SelectIntoDataSet



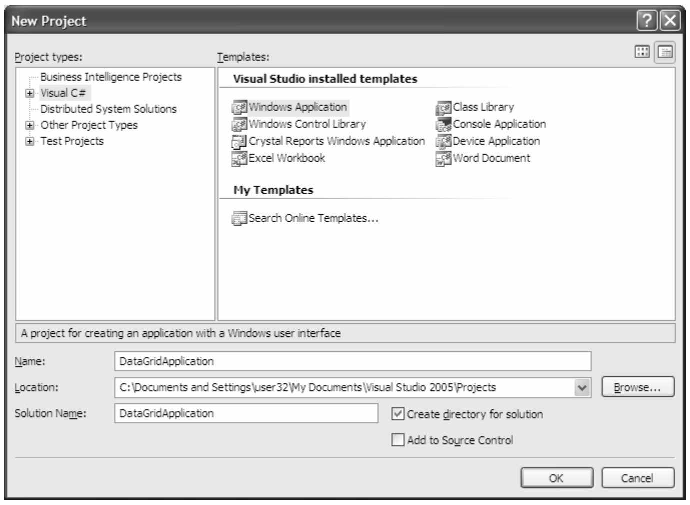
```
    class Program {
        static void Main (string[] args) {
            String connectionString = "server=localhost;database=AdventureWorks;uid=sa;pwd=dk
            SqlConnection connection = new SqlConnection (connectionString);

            String query =
                "SELECT TOP 10 Title, FirstName, LastName, EmailAddress " +
                "FROM Person.Contact " +
                "ORDER BY ContactID";
            SqlCommand command = connection.CreateCommand ();
            command.CommandText = query;

            SqlDataAdapter dataAdapter = new SqlDataAdapter ();
            dataAdapter.SelectCommand = command;

            DataSet dataSet = new DataSet ();

            connection.Open ();
            dataAdapter.Fill (dataSet, "Person.Contact");
            connection.Close ();

            DataTable dataTable = dataSet.Tables["Person.Contact"];
            foreach (DataRow dataRow in dataTable.Rows) {
                Console.WriteLine (dataRow["Title"] + " " + dataRow["FirstName"] + " " +
                    dataRow["LastName"] + ", " + dataRow["EmailAddress"]);
            }

            Console.ReadLine ();
        }
    }
}
```

E: SelectIntoDataSet



```
Mr. Gustavo Achong, gustavo0@adventure-works.com
Ms. Catherine Abel, catherine0@adventure-works.com
Ms. Kim Abercrombie, kim2@adventure-works.com
Sr. Humberto Acevedo, humberto0@adventure-works.com
Sra. Pilar Ackerman, pilar1@adventure-works.com
Ms. Frances Adams, frances0@adventure-works.com
Ms. Margaret Smith, margaret0@adventure-works.com
Ms. Carla Adams, carla0@adventure-works.com
Mr. Jay Adams, jay1@adventure-works.com
Mr. Ronald Adina, ronald0@adventure-works.com
_
```

E: DataGridApplication



A project for creating an application with a Windows user interface

Name: DataGridApplication

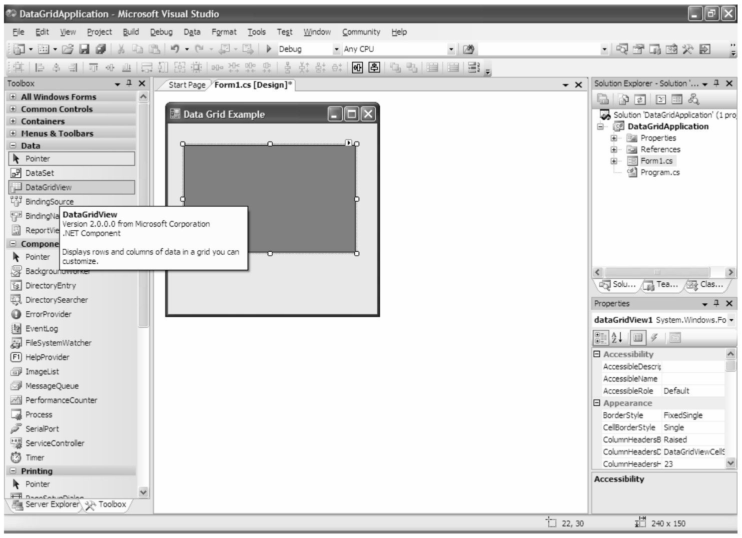
Location: C:\Documents and Settings\user32\My Documents\Visual Studio 2005\Projects

Solution Name: DataGridApplication

[✓] Create directory for solution

[ ] Add to Source Control

E: DataGridApplication



DataGridView
Version 2.0.0.0 from Microsoft Corporation
.NET Component

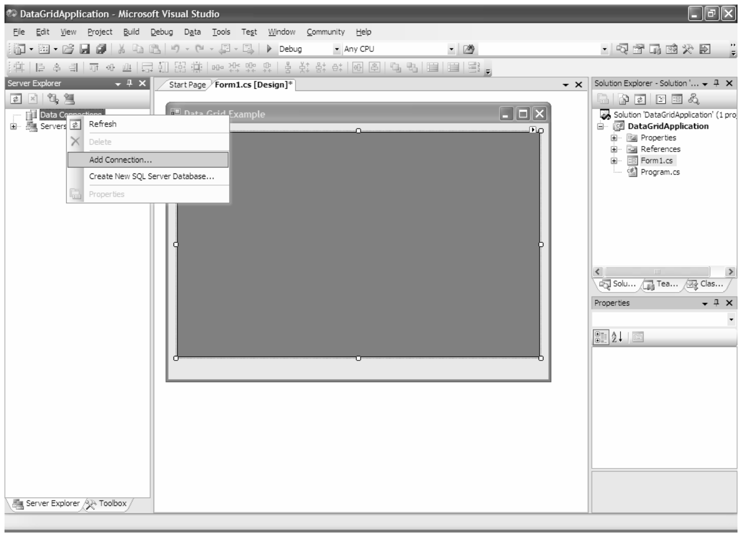
Displays rows and columns of data in a grid you can customize.

dataGridView1 System.Windows.Fo

Accessibility	
AccessibleDescrip	
AccessibleName	
AccessibleRole	Default
Appearance	
BorderStyle	FixedSingle
CellBorderStyle	Single
ColumnHeadersB	Raised
ColumnHeadersD	DataGridViewCellS
ColumnHeadersH	23

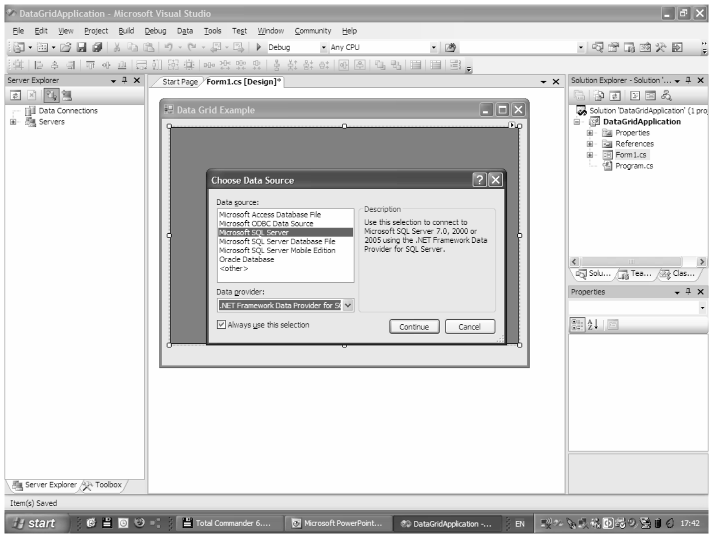
Accessibility

E: DataGridApplicaton



March 2006 Database Programming 2006 29

E: DataGridApplication



Data source:
- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Database File
- Microsoft SQL Server Mobile Edition
- Oracle Database
- <other>

Description

Use this selection to connect to Microsoft SQL Server 7.0, 2000 or 2005 using the .NET Framework Data Provider for SQL Server.

Data provider: .NET Framework Data Provider for S

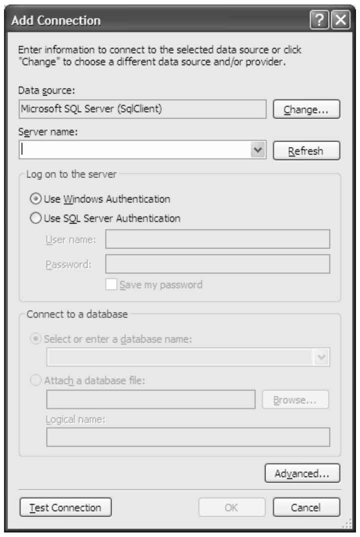
[✓] Always use this selection

Continue Cancel

Item(s) Saved

March 2006 Database Programming 2006 30

E: DataGridApplication



Add Connection

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:
[Refresh]

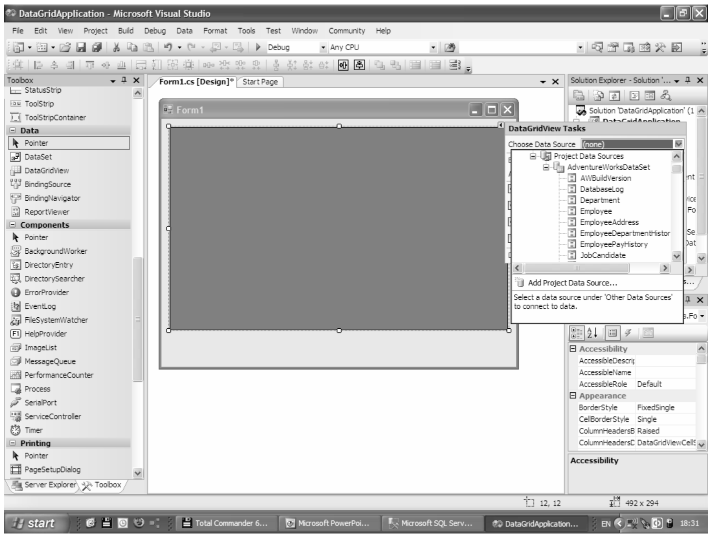
Log on to the server
(●) Use Windows Authentication
( ) Use SQL Server Authentication
User name:
Password:
☐ Save my password

Connect to a database
(●) Select or enter a database name:
( ) Attach a database file: [Browse...]
Logical name:

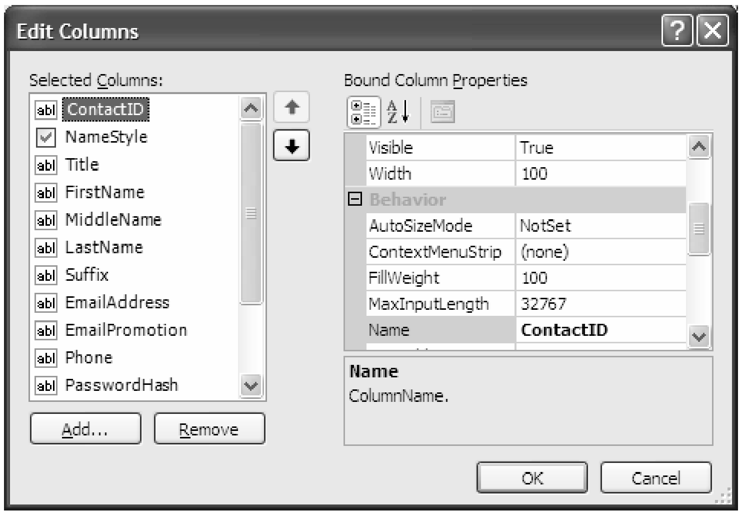
[Advanced...]

[Test Connection] [OK] [Cancel]

E: DataGridApplication


DataGridApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Community Help
Toolbox
Form1.cs [Design]* Start Page
Form1
DataGridView Tasks
Choose Data Source (none)
Project Data Sources
AdventureWorksDataSet
AWBuildVersion
DatabaseLog
Department
Employee
EmployeeAddress
EmployeeDepartmentHistor
EmployeePayHistory
JobCandidate
Add Project Data Source...
Select a data source under 'Other Data Sources' to connect to data.
Solution Explorer - Solution '...
Accessibility
Appearance
BorderStyle FixedSingle
CellBorderStyle Single
ColumnHeadersB Raised
start

E: DataGridApplication



Edit Columns

Selected Columns:

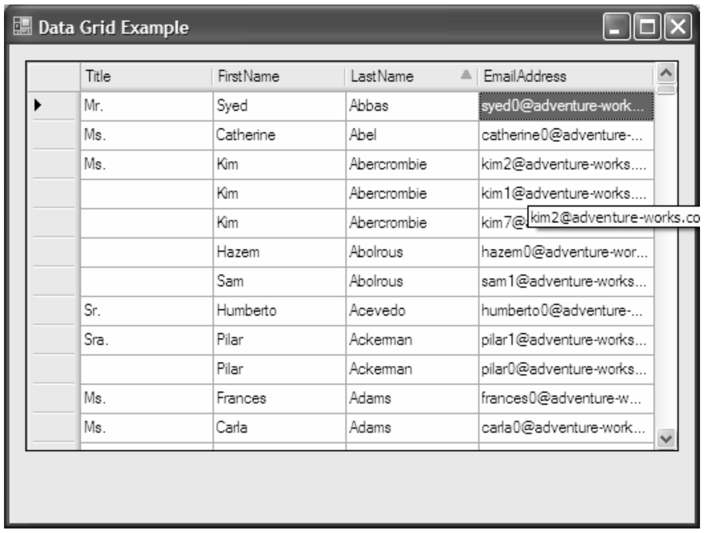
- ContactID
- NameStyle
- Title
- FirstName
- MiddleName
- LastName
- Suffix
- EmailAddress
- EmailPromotion
- Phone
- PasswordHash

Bound Column Properties

Visible	True
Width	100
Behavior	
AutoSizeMode	NotSet
ContextMenuStrip	(none)
FillWeight	100
MaxInputLength	32767
Name	**ContactID**

Name
ColumnName.

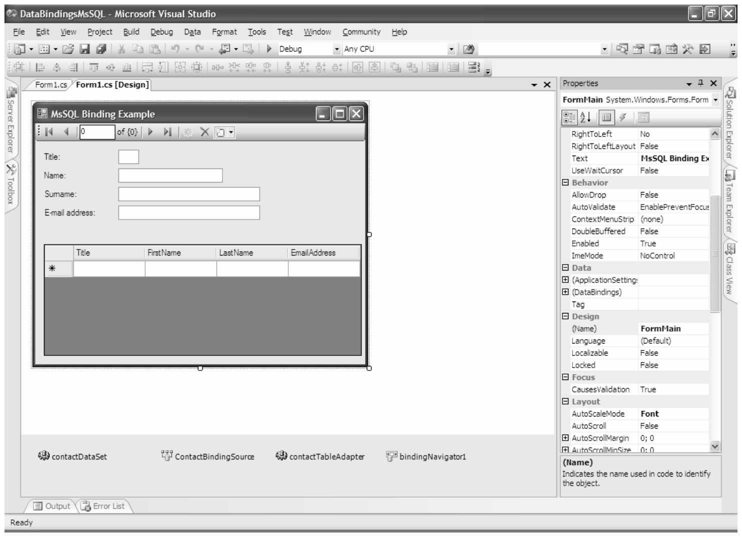
E: DataGridApplication



Data Grid Example

Title	FirstName	LastName	EmailAddress
Mr.	Syed	Abbas	syed0@adventure-work...
Ms.	Catherine	Abel	catherine0@adventure-...
Ms.	Kim	Abercrombie	kim2@adventure-works....
	Kim	Abercrombie	kim1@adventure-works....
	Kim	Abercrombie	kim7@ kim2@adventure-works.co
	Hazem	Abolrous	hazem0@adventure-wor...
	Sam	Abolrous	sam1@adventure-works...
Sr.	Humberto	Acevedo	humberto0@adventure-...
Sra.	Pilar	Ackerman	pilar1@adventure-works...
	Pilar	Ackerman	pilar0@adventure-works...
Ms.	Frances	Adams	frances0@adventure-w...
Ms.	Carla	Adams	carla0@adventure-work...

E: DataBindingsMsSQL


DataBindingsMsSQL - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Community Help
Form1.cs Form1.cs [Design]
MsSQL Binding Example
Title:
Name:
Surname:
E-mail address:
Title FirstName LastName EmailAddress
contactDataSet
ContactBindingSource
contactTableAdapter
bindingNavigator1
Output
Error List
Ready

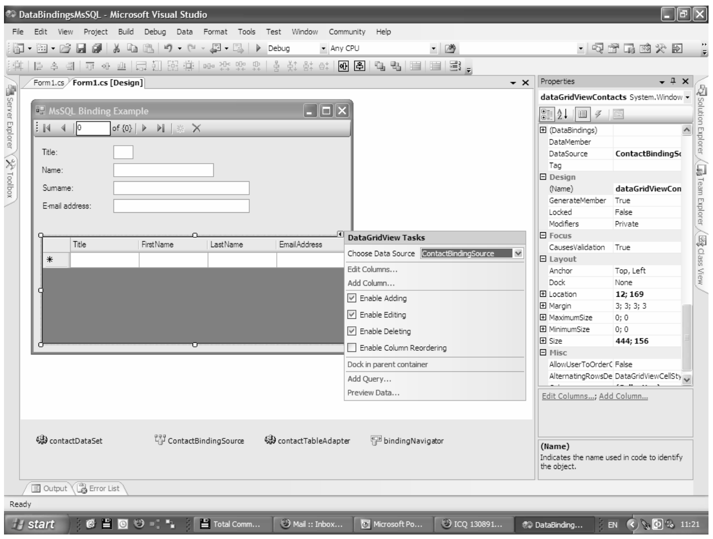
Properties

FormMain System.Windows.Forms.Form

RightToLeft	No
RightToLeftLayout	False
Text	**MsSQL Binding Ex**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocus
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	
(ApplicationSetting	
(DataBindings)	
Tag	
Design	
(Name)	**FormMain**
Language	(Default)
Localizable	False
Locked	False
Focus	
CausesValidation	True
Layout	
AutoScaleMode	**Font**
AutoScroll	False
AutoScrollMargin	0; 0
AutoScrollMinSize	0; 0

(Name)
Indicates the name used in code to identify the object.

E: DataBindingsMsSQL


DataBindingsMsSQL - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Community Help
Form1.cs Form1.cs [Design]
MsSQL Binding Example
Title:
Name:
Surname:
E-mail address:
Title FirstName LastName EmailAddress
contactDataSet
ContactBindingSource
contactTableAdapter
bindingNavigator
Output
Error List
Ready
start
11:21

DataGridView Tasks

Choose Data Source: ContactBindingSource

Edit Columns...
Add Column...
[✓] Enable Adding
[✓] Enable Editing
[✓] Enable Deleting
[ ] Enable Column Reordering
Dock in parent container
Add Query...
Preview Data...

Properties

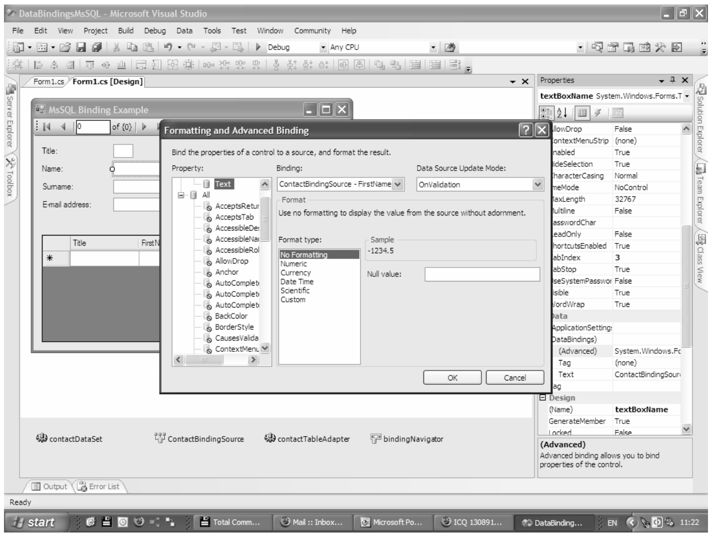
dataGridViewContacts System.Window

(DataBindings)	
DataMember	
DataSource	**ContactBindingSc**
Tag	
Design	
(Name)	**dataGridViewCon**
GenerateMember	True
Locked	False
Modifiers	Private
Focus	
CausesValidation	True
Layout	
Anchor	Top, Left
Dock	None
Location	**12; 169**
Margin	3; 3; 3; 3
MaximumSize	0; 0
MinimumSize	0; 0
Size	**444; 156**
Misc	
AllowUserToOrderC	False
AlternatingRowsDe	DataGridViewCellSty

Edit Columns...; Add Column...

(Name)
Indicates the name used in code to identify the object.

E: DataBindingsMsSQL



Formatting and Advanced Binding

Bind the properties of a control to a source, and format the result.

Property: Text

Binding: ContactBindingSource - FirstName

Data Source Update Mode: OnValidation

Format

Use no formatting to display the value from the source without adornment.

Format type: No Formatting (Numeric, Currency, Date Time, Scientific, Custom)

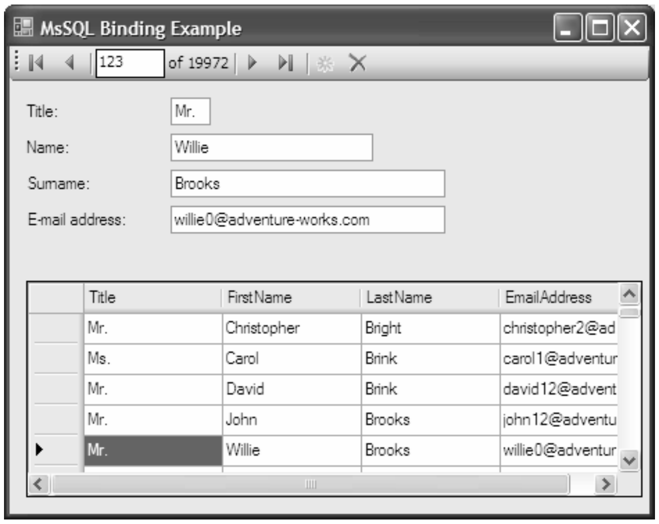
Sample: -1234.5

Null value:

OK Cancel

(Advanced)
Advanced binding allows you to bind properties of the control.

E: DataBindingsMsSQL



Title: Mr.

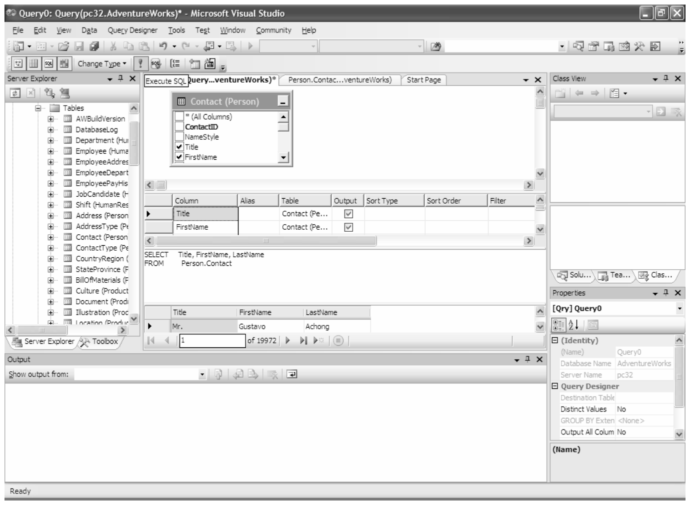
Name: Willie

Surname: Brooks

E-mail address: willie0@adventure-works.com

	Title	FirstName	LastName	EmailAddress
	Mr.	Christopher	Bright	christopher2@ad
	Ms.	Carol	Brink	carol1@adventur
	Mr.	David	Brink	david12@advent
	Mr.	John	Brooks	john12@adventu
▶	Mr.	Willie	Brooks	willie0@adventur

Working with MsSQL database in VS



```
SELECT   Title, FirstName, LastName
FROM     Person.Contact
```

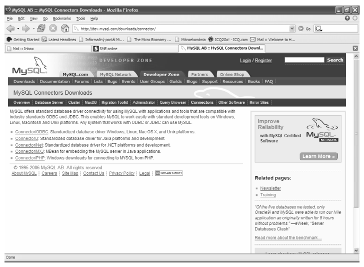
Column	Alias	Table	Output	Sort Type	Sort Order	Filter
Title		Contact (Pe...	✓			
FirstName		Contact (Pe...	✓			

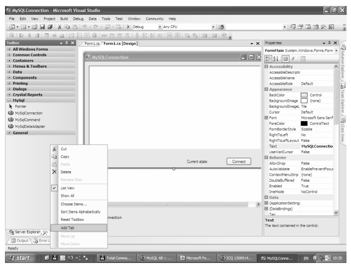
Title	FirstName	LastName
Mr.	Gustavo	Achong

Installing MySQL .NET connector

MySQL is using standardized connector for .NET platform. To develop application on Windows is necessary to download and install the connector and then to register it in the Visual Studio.

In the Visual Studio choose *Toolbox* (if not visible use menu *View-Toolbox*), right click into it and choose *Add Tab* from the menu. Inside the created box write e.g. *MySQL* as the caption for the new tab.

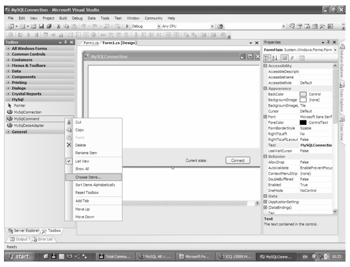


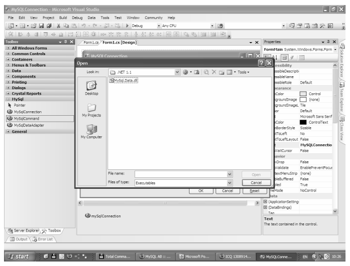


Registering .NET connector

Right click into the new tab from Toolbox and click *Choose Item* menu. After couple of seconds *Choose Toolbox Items* dialog appear and you can easily *Browse .dll* file with the .NET connector (in our case in *c:\Program Files\MySQL\MySQL Connector Net 1.0.7\bin\.NET 1.1.\MySQL.Data.dll*). Finally three new items appear inside the Toolbox's MySQL tab:

- *MySqlConnection*
- *MySqlCommand*
- *MySqlDataAdapter*

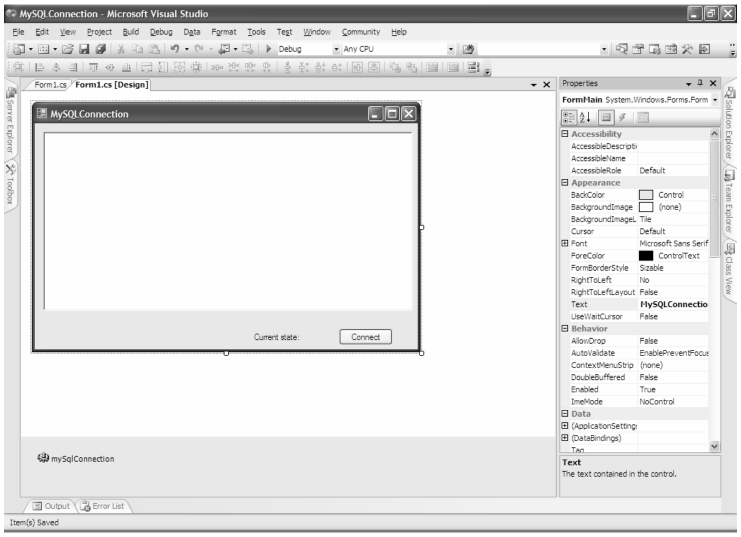




Examples using MySQL

- *MySQLConnection* – simple windows application using *MySqlConnenction* object and demonstrating basic windows based programming techniques such as event oriented programming.
- *MySQLApplication* - simple windows application demonstrating how to use *DataGriView* component with MySQL.
- *FormElementBinding* – example of binding form element by using MySQL and demonstration of a Master-Detail relationship.
- *MySQLTableEditor* – complex MySQL example (official example) demonstrating basic features of the .NET connector.
- *MySQLModifyData* – application shows how to use INSERT, UPDATE and DELETE statements in C# code. Also concept of transactional processing is introduced here.

E: MySQLConnection


MySQLConnection - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Community Help
Debug
Any CPU
Form1.cs
Form1.cs [Design]
MySQLConnection
Current state:
Connect
mySqlConnection
Output
Error List
Item(s) Saved

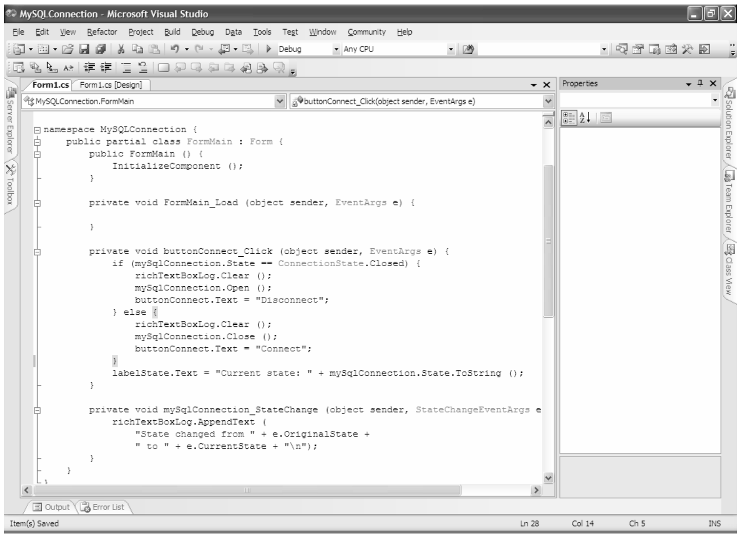
Properties

FormMain System.Windows.Forms.Form

Accessibility	
AccessibleDescripti	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImageL	Tile
Cursor	Default
Font	Microsoft Sans Serif
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**MySQLConnectio**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocus
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	
(ApplicationSetting:	
(DataBindings)	
Tag	

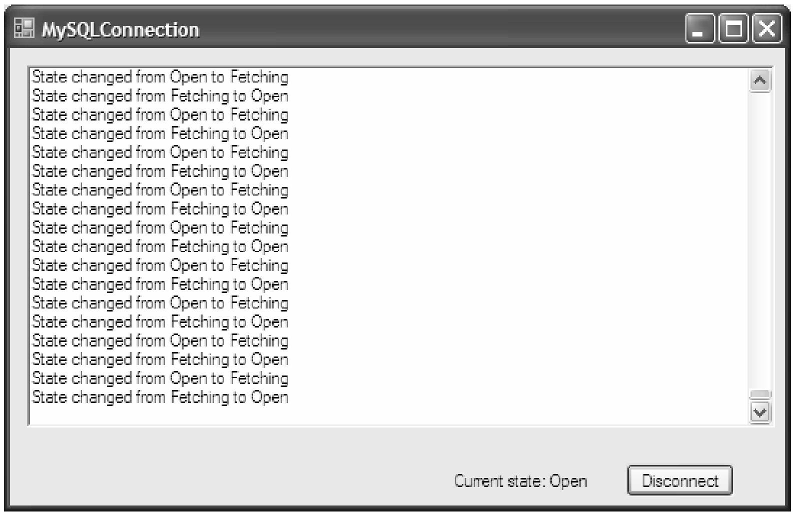
Text
The text contained in the control.

E: MySQLConnection


MySQLConnection - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Window Community Help
Debug
Any CPU
Form1.cs
Form1.cs [Design]
MySQLConnection.FormMain
buttonConnect_Click(object sender, EventArgs e)
Properties
Output
Error List
Item(s) Saved
Ln 28
Col 14
Ch 5
INS

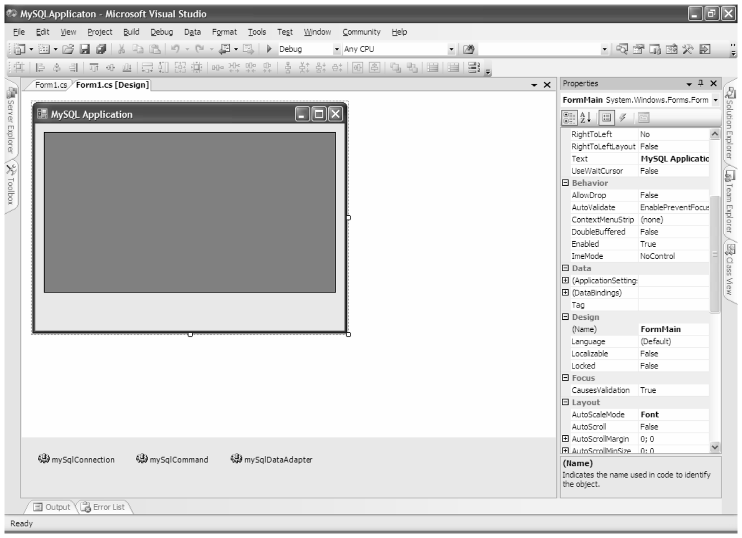
```
namespace MySQLConnection {
    public partial class FormMain : Form {
        public FormMain () {
            InitializeComponent ();
        }

        private void FormMain_Load (object sender, EventArgs e) {

        }

        private void buttonConnect_Click (object sender, EventArgs e) {
            if (mySqlConnection.State == ConnectionState.Closed) {
                richTextBoxLog.Clear ();
                mySqlConnection.Open ();
                buttonConnect.Text = "Disconnect";
            } else {
                richTextBoxLog.Clear ();
                mySqlConnection.Close ();
                buttonConnect.Text = "Connect";
            }
            labelState.Text = "Current state: " + mySqlConnection.State.ToString ();
        }

        private void mySqlConnection_StateChange (object sender, StateChangeEventArgs e
            richTextBoxLog.AppendText (
                "State changed from " + e.OriginalState +
                " to " + e.CurrentState + "\n");
        }
    }
}
```

E: MySQLConnection



```
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
State changed from Open to Fetching
State changed from Fetching to Open
```

E: MySQLApplication

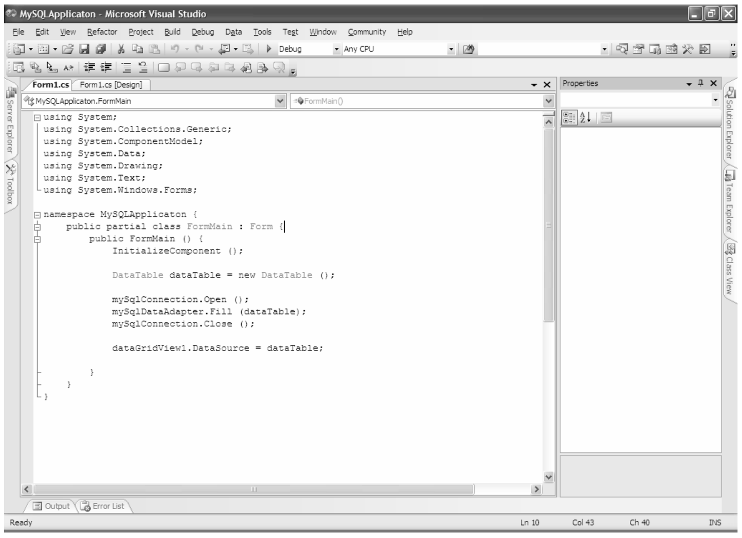


FormMain System.Windows.Forms.Form

RightToLeft	No
RightToLeftLayout	False
Text	**MySQL Applicatic**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocus
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	
(ApplicationSetting:	
(DataBindings)	
Tag	
Design	
(Name)	**FormMain**
Language	(Default)
Localizable	False
Locked	False
Focus	
CausesValidation	True
Layout	
AutoScaleMode	**Font**
AutoScroll	False
AutoScrollMargin	0; 0
AutoScrollMinSize	0; 0

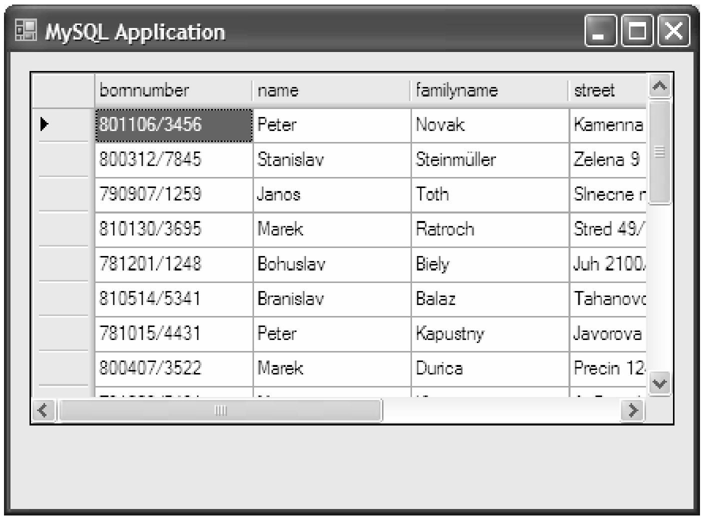
(Name)
Indicates the name used in code to identify the object.

E: MySQLApplication



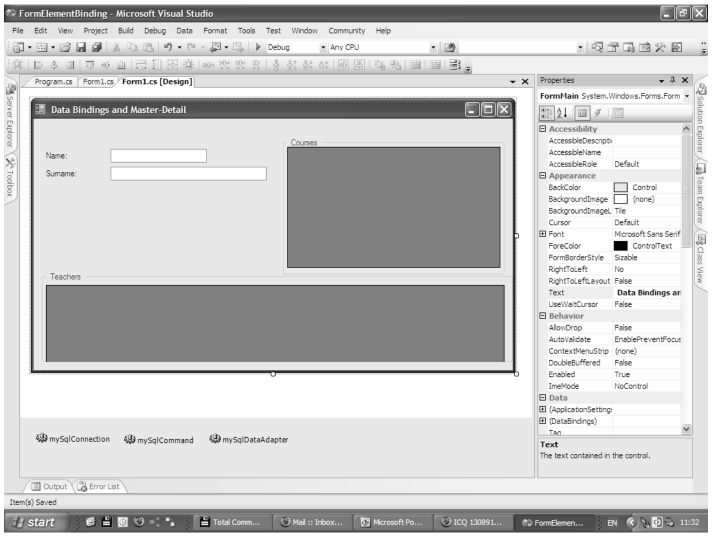
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace MySQLApplicaton {
    public partial class FormMain : Form {
        public FormMain () {
            InitializeComponent ();

            DataTable dataTable = new DataTable ();

            mySqlConnection.Open ();
            mySqlDataAdapter.Fill (dataTable);
            mySqlConnection.Close ();

            dataGridView1.DataSource = dataTable;

        }
    }
}
```

E: MySQLApplication



bornnumber	name	familyname	street
801106/3456	Peter	Novak	Kamenna
800312/7845	Stanislav	Steinmüller	Zelena 9
790907/1259	Janos	Toth	Slnecne r
810130/3695	Marek	Ratroch	Stred 49/
781201/1248	Bohuslav	Biely	Juh 2100.
810514/5341	Branislav	Balaz	Tahanovc
781015/4431	Peter	Kapustny	Javorova
800407/3522	Marek	Durica	Precin 12

E: FormElementBinding



FormElementBinding - Microsoft Visual Studio

Data Bindings and Master-Detail

Name:

Surname:

Courses

Teachers

mySqlConnection mySqlCommand mySqlDataAdapter

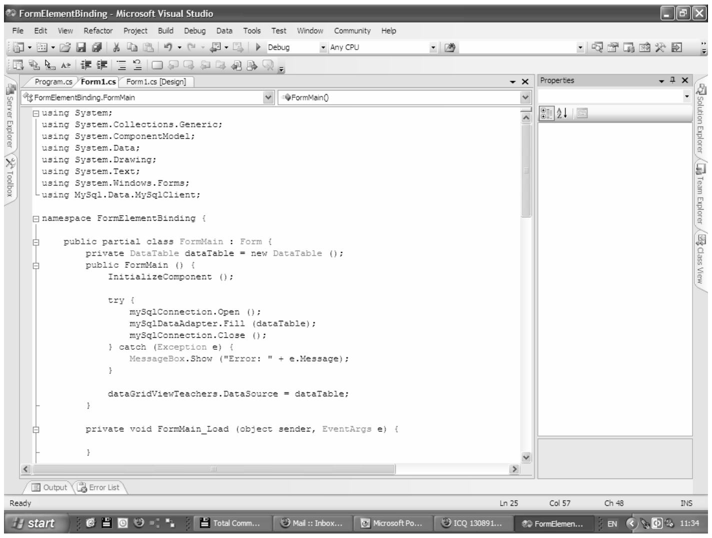
Properties: FormMain System.Windows.Forms.Form

Accessibility	
AccessibleDescripti	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImageL	Tile
Cursor	Default
Font	Microsoft Sans Serif
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Data Bindings an**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocus
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	
(ApplicationSetting	
(DataBindings)	
Tag	

Text
The text contained in the control.

Item(s) Saved

E: FormElementBinding



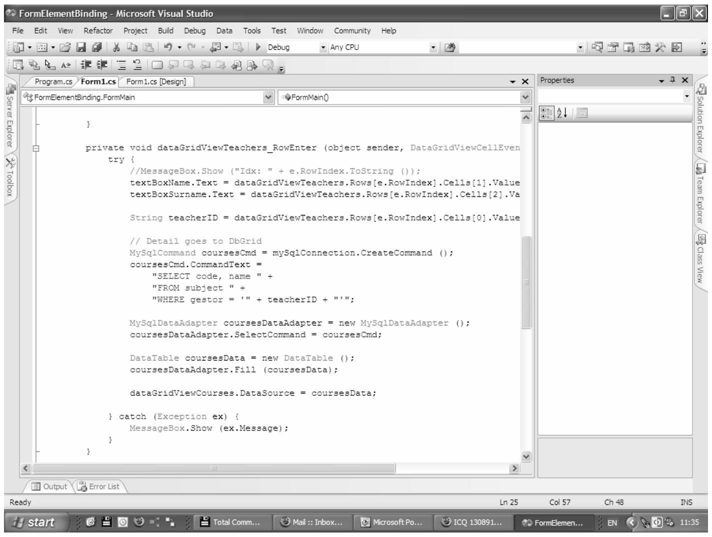
FormElementBinding - Microsoft Visual Studio

FormElementBinding.FormMain — FormMain()

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

namespace FormElementBinding {

    public partial class FormMain : Form {
        private DataTable dataTable = new DataTable ();
        public FormMain () {
            InitializeComponent ();

            try {
                mySqlConnection.Open ();
                mySqlDataAdapter.Fill (dataTable);
                mySqlConnection.Close ();
            } catch (Exception e) {
                MessageBox.Show ("Error: " + e.Message);
            }

            dataGridViewTeachers.DataSource = dataTable;
        }

        private void FormMain_Load (object sender, EventArgs e) {

        }
```

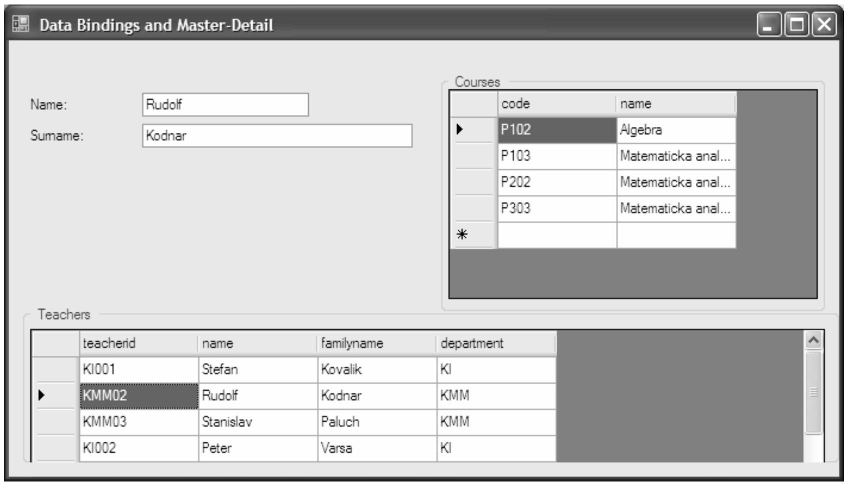
Ready Ln 25 Col 57 Ch 48 INS

E: FormElementBinding



```
        }

        private void dataGridViewTeachers_RowEnter (object sender, DataGridViewCellEven
            try {
                //MessageBox.Show ("Idx: " + e.RowIndex.ToString ());
                textBoxName.Text = dataGridViewTeachers.Rows[e.RowIndex].Cells[1].Value
                textBoxSurname.Text = dataGridViewTeachers.Rows[e.RowIndex].Cells[2].Va

                String teacherID = dataGridViewTeachers.Rows[e.RowIndex].Cells[0].Value

                // Detail goes to DbGrid
                MySqlCommand coursesCmd = mySqlConnection.CreateCommand ();
                coursesCmd.CommandText =
                    "SELECT code, name " +
                    "FROM subject " +
                    "WHERE gestor = '" + teacherID + "'";

                MySqlDataAdapter coursesDataAdapter = new MySqlDataAdapter ();
                coursesDataAdapter.SelectCommand = coursesCmd;

                DataTable coursesData = new DataTable ();
                coursesDataAdapter.Fill (coursesData);

                dataGridViewCourses.DataSource = coursesData;

            } catch (Exception ex) {
                MessageBox.Show (ex.Message);
            }
        }
```

E: FormElementBinding



Data Bindings and Master-Detail

Name: Rudolf

Surname: Kodnar

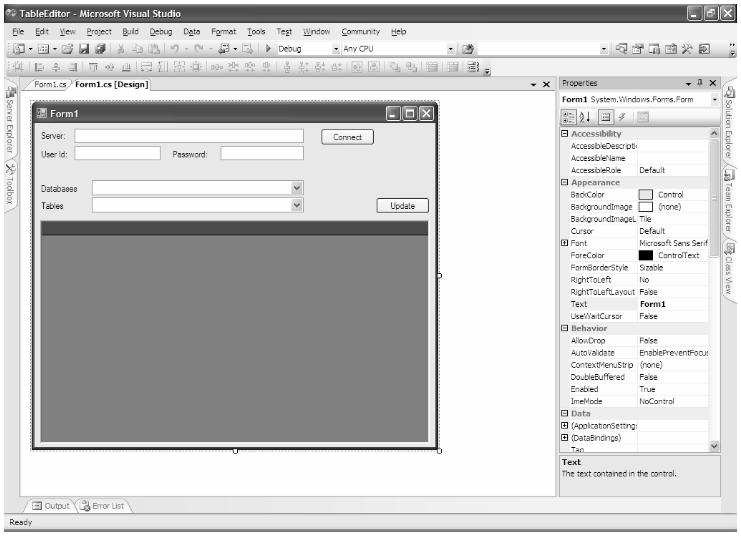
Courses

	code	name
▶	P102	Algebra
	P103	Matematicka anal...
	P202	Matematicka anal...
	P303	Matematicka anal...
*		

Teachers

	teacherid	name	familyname	department
	KI001	Stefan	Kovalik	KI
▶	KMM02	Rudolf	Kodnar	KMM
	KMM03	Stanislav	Paluch	KMM
	KI002	Peter	Varsa	KI

E: MySQLTableEditor


TableEditor - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Community Help
Debug
Any CPU
Form1.cs
Form1.cs [Design]
Form1
Server:
Connect
User Id:
Password:
Databases
Tables
Update
Output
Error List
Ready

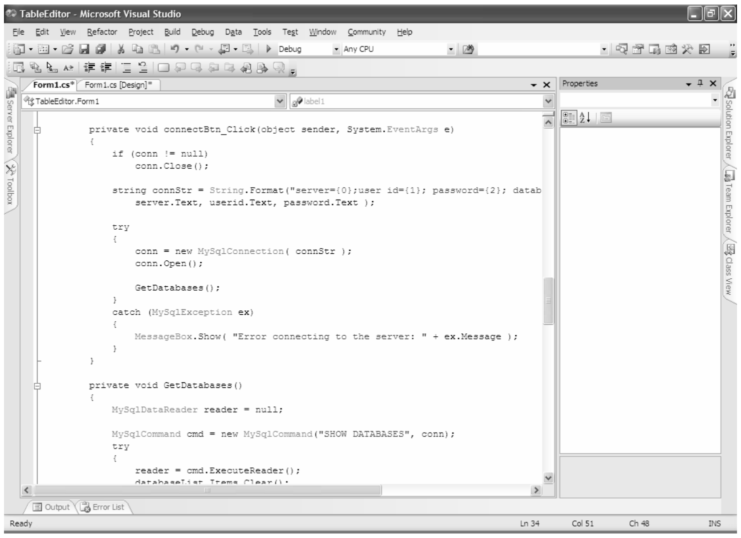
Properties

Form1 System.Windows.Forms.Form

Accessibility	
AccessibleDescripti	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImageL	Tile
Cursor	Default
Font	Microsoft Sans Serif
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Form1**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocus
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	
(ApplicationSetting	
(DataBindings)	
Tag	

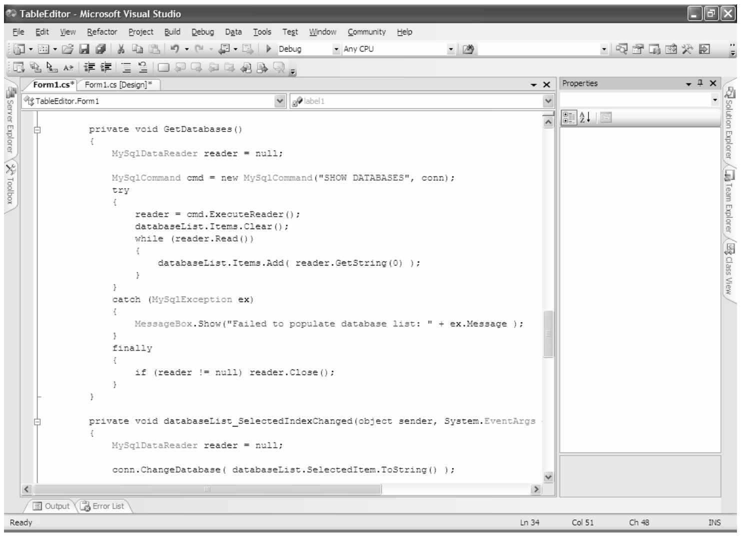
Text
The text contained in the control.

E: MySQLTableEditor


TableEditor - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Window Community Help
Debug
Any CPU
Form1.cs*
Form1.cs [Design]*
TableEditor.Form1
label1
Properties
Output
Error List
Ready
Ln 34
Col 51
Ch 48
INS

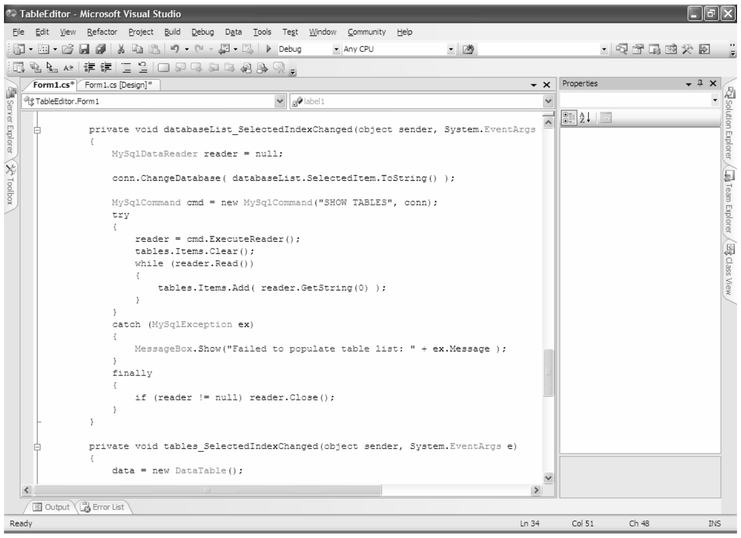
```
        private void connectBtn_Click(object sender, System.EventArgs e)
        {
            if (conn != null)
                conn.Close();

            string connStr = String.Format("server={0};user id={1}; password={2}; datab
                server.Text, userid.Text, password.Text );

            try
            {
                conn = new MySqlConnection( connStr );
                conn.Open();

                GetDatabases();
            }
            catch (MySqlException ex)
            {
                MessageBox.Show( "Error connecting to the server: " + ex.Message );
            }
        }

        private void GetDatabases()
        {
            MySqlDataReader reader = null;

            MySqlCommand cmd = new MySqlCommand("SHOW DATABASES", conn);
            try
            {
                reader = cmd.ExecuteReader();
                databaseList.Items.Clear();
```

E: MySQLTableEditor



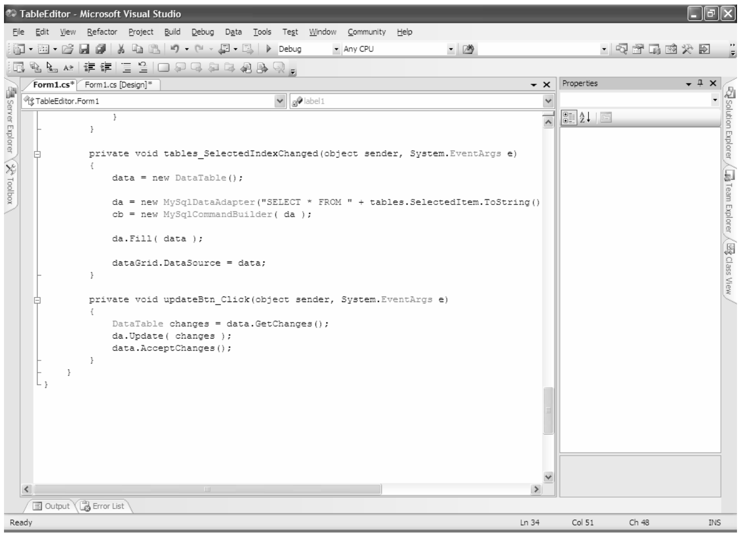
```
        private void GetDatabases()
        {
            MySqlDataReader reader = null;

            MySqlCommand cmd = new MySqlCommand("SHOW DATABASES", conn);
            try
            {
                reader = cmd.ExecuteReader();
                databaseList.Items.Clear();
                while (reader.Read())
                {
                    databaseList.Items.Add( reader.GetString(0) );
                }
            }
            catch (MySqlException ex)
            {
                MessageBox.Show("Failed to populate database list: " + ex.Message );
            }
            finally
            {
                if (reader != null) reader.Close();
            }
        }

        private void databaseList_SelectedIndexChanged(object sender, System.EventArgs
        {
            MySqlDataReader reader = null;

            conn.ChangeDatabase( databaseList.SelectedItem.ToString() );
```

E: MySQLTableEditor



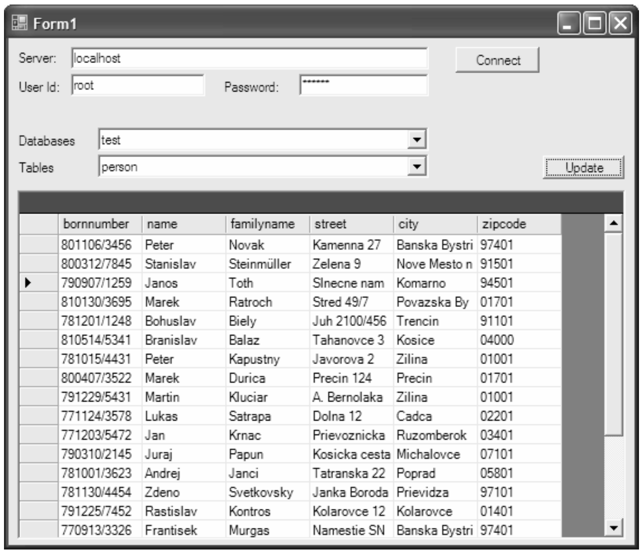
```
        private void databaseList_SelectedIndexChanged(object sender, System.EventArgs
        {
            MySqlDataReader reader = null;

            conn.ChangeDatabase( databaseList.SelectedItem.ToString() );

            MySqlCommand cmd = new MySqlCommand("SHOW TABLES", conn);
            try
            {
                reader = cmd.ExecuteReader();
                tables.Items.Clear();
                while (reader.Read())
                {
                    tables.Items.Add( reader.GetString(0) );
                }
            }
            catch (MySqlException ex)
            {
                MessageBox.Show("Failed to populate table list: " + ex.Message );
            }
            finally
            {
                if (reader != null) reader.Close();
            }
        }

        private void tables_SelectedIndexChanged(object sender, System.EventArgs e)
        {
            data = new DataTable();
```

E: MySQLTableEditor



```
            }
        }

        private void tables_SelectedIndexChanged(object sender, System.EventArgs e)
        {
            data = new DataTable();

            da = new MySqlDataAdapter("SELECT * FROM " + tables.SelectedItem.ToString()
            cb = new MySqlCommandBuilder( da );

            da.Fill( data );

            dataGrid.DataSource = data;
        }

        private void updateBtn_Click(object sender, System.EventArgs e)
        {
            DataTable changes = data.GetChanges();
            da.Update( changes );
            data.AcceptChanges();
        }
    }
}
```

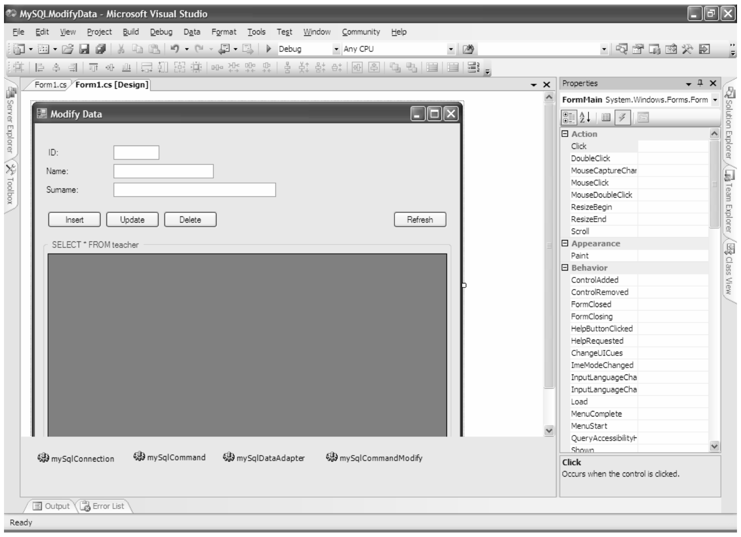
E: MySQLTableEditor



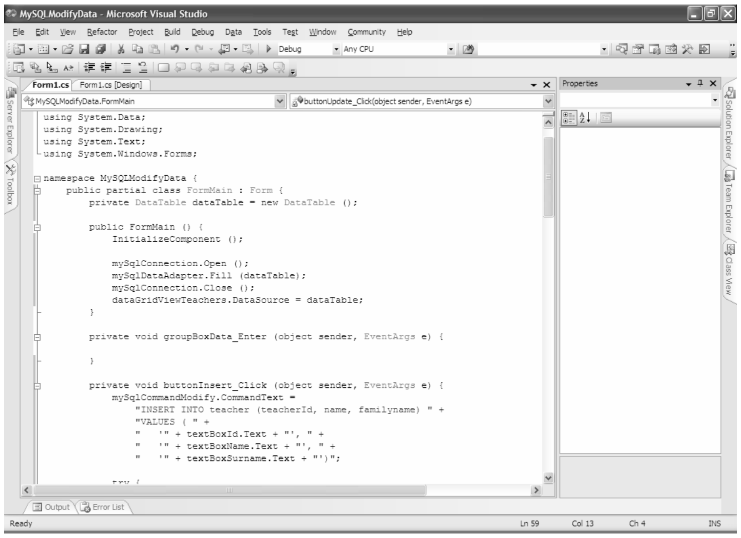
Server: localhost
User Id: root
Password: ******
Databases: test
Tables: person

bornnumber	name	familyname	street	city	zipcode
801106/3456	Peter	Novak	Kamenna 27	Banska Bystri	97401
800312/7845	Stanislav	Steinmüller	Zelena 9	Nove Mesto n	91501
790907/1259	Janos	Toth	Slnecne nam	Komarno	94501
810130/3695	Marek	Ratroch	Stred 49/7	Povazska By	01701
781201/1248	Bohuslav	Biely	Juh 2100/456	Trencin	91101
810514/5341	Branislav	Balaz	Tahanovce 3	Kosice	04000
781015/4431	Peter	Kapustny	Javorova 2	Zilina	01001
800407/3522	Marek	Durica	Precin 124	Precin	01701
791229/5431	Martin	Kluciar	A. Bernolaka	Zilina	01001
771124/3578	Lukas	Satrapa	Dolna 12	Cadca	02201
771203/5472	Jan	Krnac	Prievoznicka	Ruzomberok	03401
790310/2145	Juraj	Papun	Kosicka cesta	Michalovce	07101
781001/3623	Andrej	Janci	Tatranska 22	Poprad	05801
781130/4454	Zdeno	Svetkovsky	Janka Boroda	Prievidza	97101
791225/7452	Rastislav	Kontros	Kolarovce 12	Kolarovce	01401
770913/3326	Frantisek	Murgas	Namestie SN	Banska Bystri	97401

E: MySQLModifyData

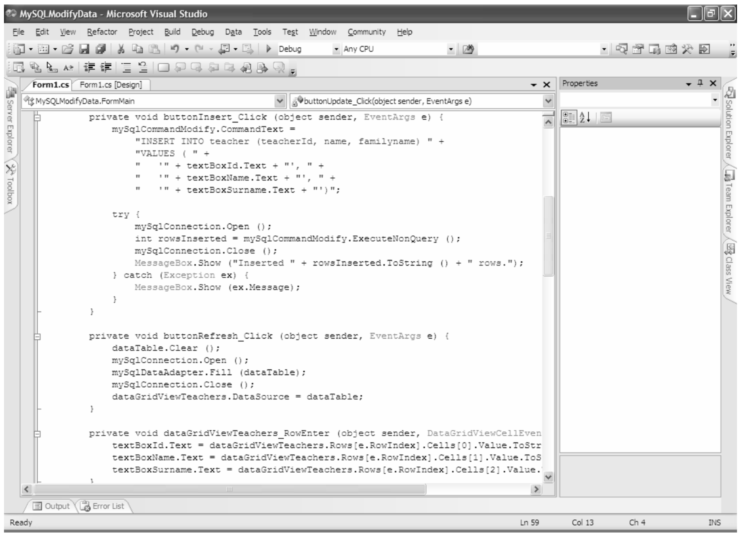


E: MySQLModifyData



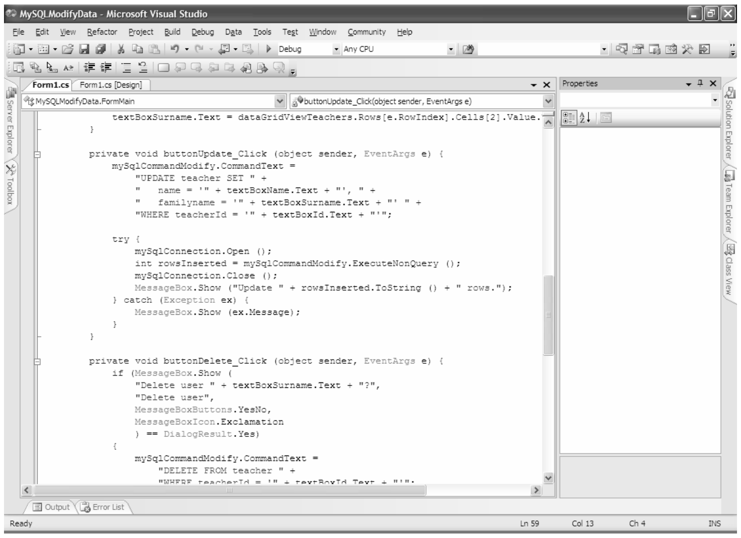
```
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace MySQLModifyData {
    public partial class FormMain : Form {
        private DataTable dataTable = new DataTable ();

        public FormMain () {
            InitializeComponent ();

            mySqlConnection.Open ();
            mySqlDataAdapter.Fill (dataTable);
            mySqlConnection.Close ();
            dataGridViewTeachers.DataSource = dataTable;
        }

        private void groupBoxData_Enter (object sender, EventArgs e) {

        }

        private void buttonInsert_Click (object sender, EventArgs e) {
            mySqlCommandModify.CommandText =
                "INSERT INTO teacher (teacherId, name, familyname) " +
                "VALUES ( " +
                "   '" + textBoxId.Text + "', " +
                "   '" + textBoxName.Text + "', " +
                "   '" + textBoxSurname.Text + "')";

            try {
```

E: MySQLModifyData



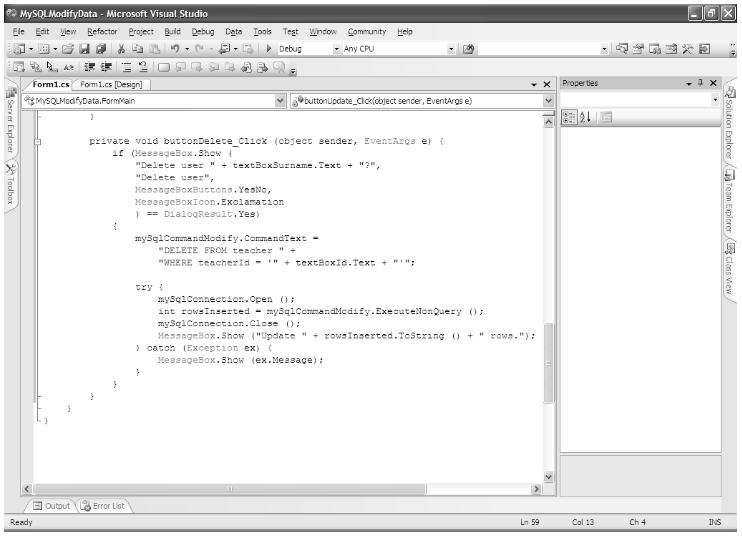
```
    private void buttonInsert_Click (object sender, EventArgs e) {
        mySqlCommandModify.CommandText =
            "INSERT INTO teacher (teacherId, name, familyname) " +
            "VALUES ( " +
            "   '" + textBoxId.Text + "', " +
            "   '" + textBoxName.Text + "', " +
            "   '" + textBoxSurname.Text + "')";

        try {
            mySqlConnection.Open ();
            int rowsInserted = mySqlCommandModify.ExecuteNonQuery ();
            mySqlConnection.Close ();
            MessageBox.Show ("Inserted " + rowsInserted.ToString () + " rows.");
        } catch (Exception ex) {
            MessageBox.Show (ex.Message);
        }
    }

    private void buttonRefresh_Click (object sender, EventArgs e) {
        dataTable.Clear ();
        mySqlConnection.Open ();
        mySqlDataAdapter.Fill (dataTable);
        mySqlConnection.Close ();
        dataGridViewTeachers.DataSource = dataTable;
    }

    private void dataGridViewTeachers_RowEnter (object sender, DataGridViewCellEven
        textBoxId.Text = dataGridViewTeachers.Rows[e.RowIndex].Cells[0].Value.ToStr
        textBoxName.Text = dataGridViewTeachers.Rows[e.RowIndex].Cells[1].Value.ToS
        textBoxSurname.Text = dataGridViewTeachers.Rows[e.RowIndex].Cells[2].Value.
```

E: MySQLModifyData



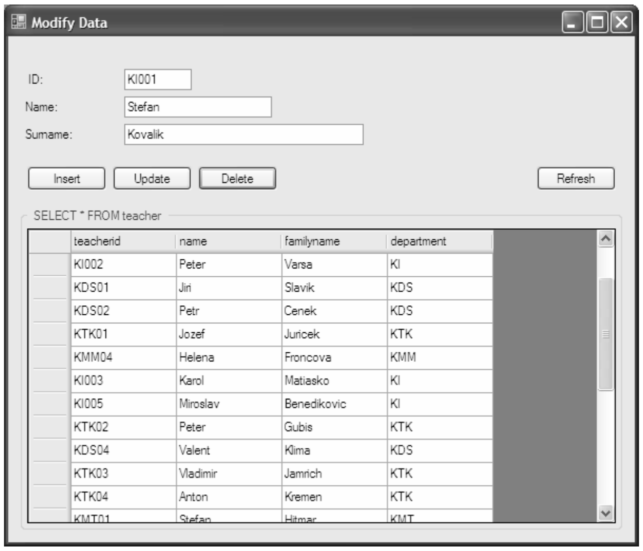
```
        textBoxSurname.Text = dataGridViewTeachers.Rows[e.RowIndex].Cells[2].Value.
    }

    private void buttonUpdate_Click (object sender, EventArgs e) {
        mySqlCommandModify.CommandText =
            "UPDATE teacher SET " +
            "   name = '" + textBoxName.Text + "', " +
            "   familyname = '" + textBoxSurname.Text + "' " +
            "WHERE teacherId = '" + textBoxId.Text + "'";

        try {
            mySqlConnection.Open ();
            int rowsInserted = mySqlCommandModify.ExecuteNonQuery ();
            mySqlConnection.Close ();
            MessageBox.Show ("Update " + rowsInserted.ToString () + " rows.");
        } catch (Exception ex) {
            MessageBox.Show (ex.Message);
        }
    }

    private void buttonDelete_Click (object sender, EventArgs e) {
        if (MessageBox.Show (
            "Delete user " + textBoxSurname.Text + "?",
            "Delete user",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Exclamation
            ) == DialogResult.Yes)
        {
            mySqlCommandModify.CommandText =
                "DELETE FROM teacher " +
                "WHERE teacherId = '" + textBoxId.Text + "'";
```

E: MySQLModifyData


MySQLModifyData - Microsoft Visual Studio
Form1.cs
Form1.cs [Design]
MySQLModifyData.FormMain
buttonUpdate_Click(object sender, EventArgs e)
Properties
Ready
Ln 59
Col 13
Ch 4
INS

```
        }

        private void buttonDelete_Click (object sender, EventArgs e) {
            if (MessageBox.Show (
                "Delete user " + textBoxSurname.Text + "?",
                "Delete user",
                MessageBoxButtons.YesNo,
                MessageBoxIcon.Exclamation
                ) == DialogResult.Yes)
            {
                mySqlCommandModify.CommandText =
                    "DELETE FROM teacher " +
                    "WHERE teacherId = '" + textBoxId.Text + "'";

                try {
                    mySqlConnection.Open ();
                    int rowsInserted = mySqlCommandModify.ExecuteNonQuery ();
                    mySqlConnection.Close ();
                    MessageBox.Show ("Update " + rowsInserted.ToString () + " rows.");
                } catch (Exception ex) {
                    MessageBox.Show (ex.Message);
                }
            }
        }
    }
}
```

E: MySQLModifyData



ID:	KI001
Name:	Stefan
Surname:	Kovalik

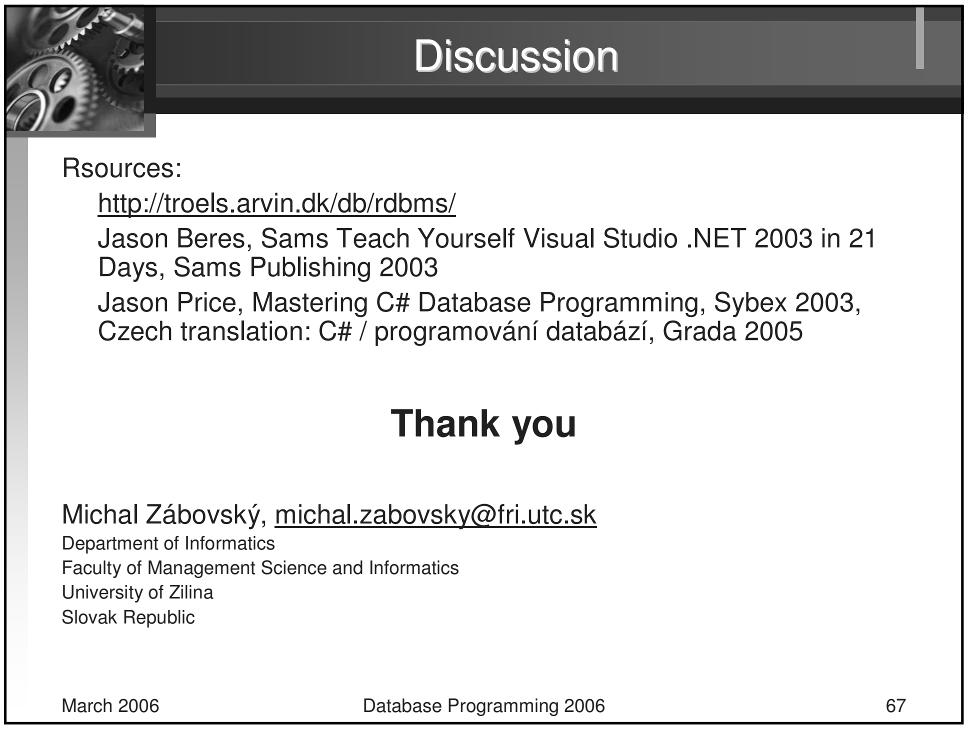
teacherid	name	familyname	department
KI002	Peter	Varsa	KI
KDS01	Jiri	Slavik	KDS
KDS02	Petr	Cenek	KDS
KTK01	Jozef	Juricek	KTK
KMM04	Helena	Froncova	KMM
KI003	Karol	Matiasko	KI
KI005	Miroslav	Benedikovic	KI
KTK02	Peter	Gubis	KTK
KDS04	Valent	Klima	KDS
KTK03	Vladimir	Jamrich	KTK
KTK04	Anton	Kremen	KTK
KMT01	Stefan	Hitmar	KMT

Transactions

```
try {
      mySqlConnection.Open ();
      MySqlTransaction myTransaction = mySqlConnection.BeginTransaction ();
      MySqlCommand commandModify = mySqlConnection.CreateCommand ();
      commandModify.Transaction = myTransaction;
      try {
                commandModify.CommandText =
                          "DELETE FROM teacher " +
                          "WHERE teacherId = '" + textBoxId.Text + "'";
                int rowsInserted = commandModify.ExecuteNonQuery ();

                commandModify.CommandText =
                "UPDATE subject SET gestor = 'KI001' " +
                "WHERE gestor = '" + textBoxId.Text + "'";
                commandModify.ExecuteNonQuery ();

                myTransaction.Commit ();
                MessageBox.Show ("Delete " + rowsInserted.ToString () + " rows.");
      } catch (Exception ex) {
                myTransaction.Rollback ();
                MessageBox.Show (ex.Message);
      }
} catch (Exception ee) {
      MessageBox.Show (ee.Message);
} finally {
      mySqlConnection.Close ();
}
```

Command parameters

```
...
commandModify.CommandText =
      "DELETE FROM teacher " +
      "WHERE teacherId = @TeacherId";
commandModify.Parameters.Add ("@TeacherId", MySqlDbType.VarChar, 5);

commandModify.Parameters["@TeacherId"].Value = teacherId;
int rowsInserted = commandModify.ExecuteNonQuery ();
...
```



Discussion

Rsources:

- http://troels.arvin.dk/db/rdbms/
- Jason Beres, Sams Teach Yourself Visual Studio .NET 2003 in 21 Days, Sams Publishing 2003
- Jason Price, Mastering C# Database Programming, Sybex 2003, Czech translation: C# / programování databází, Grada 2005

Thank you

Michal Zábovský, michal.zabovsky@fri.utc.sk
Department of Informatics
Faculty of Management Science and Informatics
University of Zilina
Slovak Republic