

# Source libre

[HOME](#)[MUSIC BLOG](#)[PROFESSIONAL COMMITMENTS](#)

## CONNECTING A MYSQL TABLE TO A DATAGRIDVIEW CONTROL IN C#

5:02 AM

NISANSA DILUSHAN

[18 COMMENTS](#)

### Introduction

Since I started coding in grade 7 or so, I have worked with many programming languages and so far C# remains my favourite. (Java coming behind as a close second.) So when a small program is to be written, I prefer to choose C#. But if that program involves working with databases we are in a bit of a trouble. C# is naturally built to work with Microsoft SQL. But what if we do not want to sacrifice the freedom that we have with MySQL?

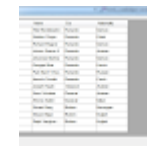
Connecting MySQL to .NET is a tricky thing. And it is a precondition to what I am going to write here. I would suggest you to go through [this example](#) and get some experience in connecting MySQL to .NET. I am going to build on it.

### Data Grid View

[DataGridView](#) is a cool windows forms component that can give you a grid view of your database tables similar to what you see in MS Access.

[Popular](#)[Tags](#)[Blog Archives](#)[Plus](#)[Crée](#)

### POPULAR POSTS



#### Connecting a MySQL table to a DataGridView control in C#

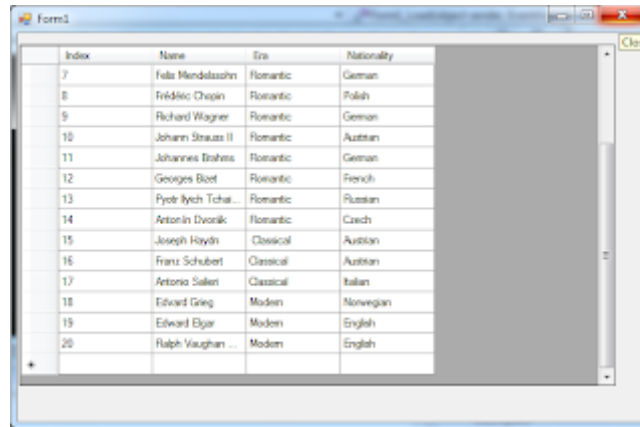
Introduction Since I started coding in grade 7 or so, I have worked with many programming languages and so far C# remains my favourite. (...)



#### Source libre - Introduction

Since long, people have asked me why I do not write a tech blog. The answer was; "Because I do not have time". The same reason is beh...

[Followers](#)[Profile](#)[Page Views](#)



## Connecting with MS SQL

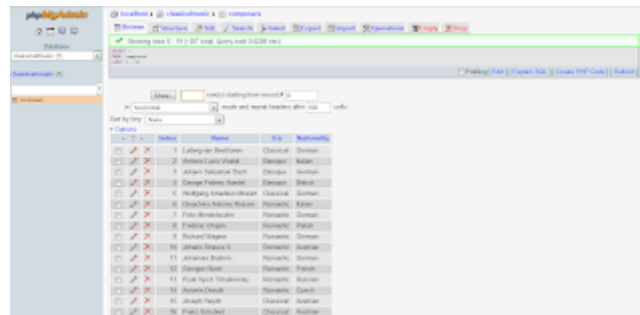
Connecting the DataGridView to MS SQL is a straight forward activity. It is called “Binding Data” to the control.

[This MSDN article](#) gives a good explanation on that.

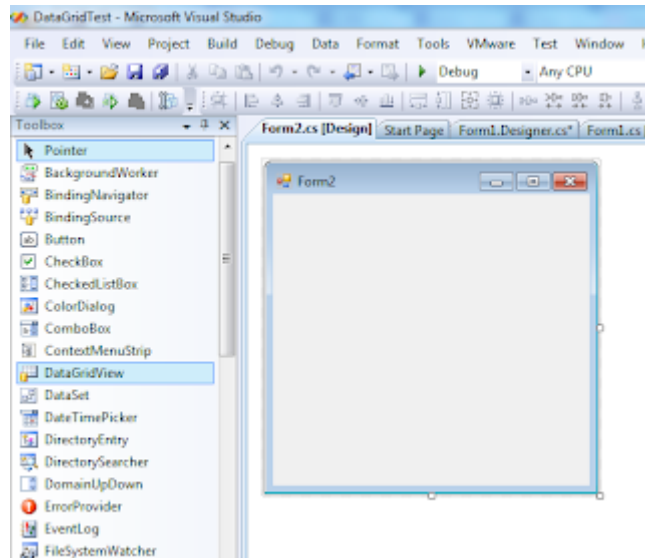
## Connecting with MySQL

Let’s get down to the business now, shall we? First of all let’s create the database and a table in MySQL. Here

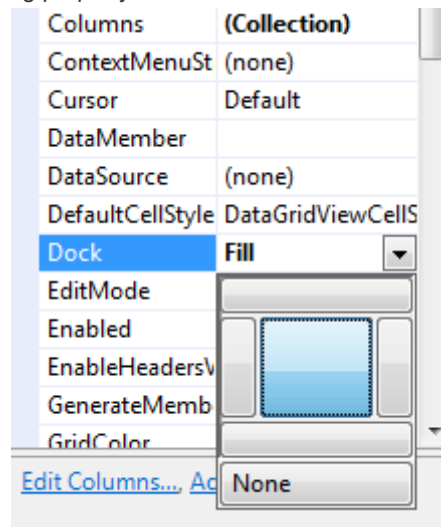
I created a database called “**classicalmusic**” and inside the said database I created a table called “**composers**”.



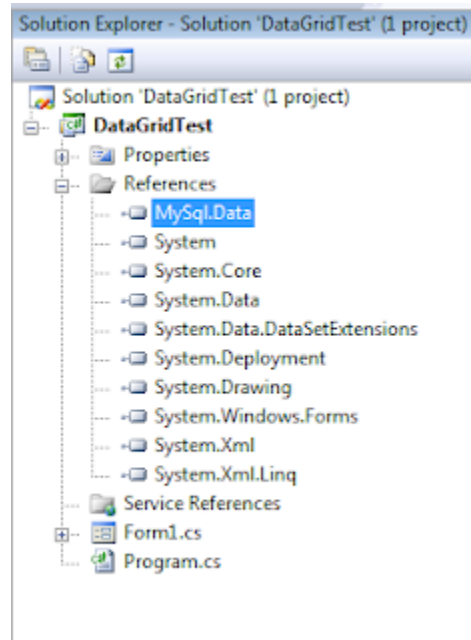
Next create a new “WindowsForms” project and add the DataGridView control to the form.



To make it look nicer, set the docking property to "Fill".



Okay. That is all that we'd do with the GUI. Let's go do some coding. But before that, make sure that you have referred the **"MySQL.Data"** component. If you haven't, go back to the [Codeproject example](http://solibnis.blogspot.com/2013/02/connecting-mysql-table-to-datagridview.html) and see how it is done.



Import the MySQLClient at the top of the class.

```
1 | using MySql.Data.MySqlClient;
```

Now let's create the global variables to create a connection between the C# project and the Database.

```
1 | private string server;
2 | private string database;
3 | private string uid;
4 | private string password;
5 | private MySqlConnection connection;
```

Since this is only a test application we will initialize the connection in the `Form_Load` method. And do not forget to create a user named "username" with the password "password" at the "Privileges" tab of your database.

```
1 | private void Form1_Load(object sender, EventArgs e)
2 | {
3 |     server = "localhost";
4 |     database = "ClassicalMusic";
5 |     uid = "username";
6 |     password = "password";
7 |     string connectionString;
8 |     connectionString = "SERVER=" + server + ";" + "DATABASE=" + database + ";" + "
9 |
10 |     connection = new MySqlConnection(connectionString);
11 | }
```

```
12 | }
```

With the connection in place, we need a way to open it. For that, we will use the following method. (From the Codeproject example)

```
1  //open connection to database
2  private bool OpenConnection()
3  {
4      try
5      {
6          connection.Open();
7          return true;
8      }
9      catch (MySqlException ex)
10     {
11         //When handling errors, you can your application's response based on the e
12         //The two most common error numbers when connecting are as follows:
13         //0: Cannot connect to server.
14         //1045: Invalid user name and/or password.
15         switch (ex.Number)
16         {
17             case 0:
18                 MessageBox.Show("Cannot connect to server. Contact administrator")
19                 break;
20             case 1045:
21                 MessageBox.Show("Invalid username/password, please try again");
22                 break;
23             default:
24                 MessageBox.Show(ex.Message);
25                 break;
26         }
27         return false;
28     }
29 }
```

Similarly, we need a way to close the database connection. To do that, we will use the following method. (From the Codeproject example)

```
1  //Close connection
2  private bool CloseConnection()
3  {
4      try
5      {
6          connection.Close();
7          return true;
8      }
9      catch (MySqlException ex)
10     {
11         MessageBox.Show(ex.Message);
```

```

12         return false;
13     }
14 }

```

## Loading the MySQL table to the DataGridView control

Let's load the data from the `composers` table to the `DataGridView` now. Create a `mySqlDataAdapter` variable to load the data adapter.

```

1 private MySqlDataAdapter mySqlDataAdapter; ?

```

We will use the `Form_Load` method to do the data loading as well. Add the following code chunk under the previous bit of code that we added to the `Form_Load` method to open the connection and close it after usage.

```

1 if (this.OpenConnection() == true) ?
2 {
3     //Business logic
4
5     //close connection
6     this.CloseConnection();
7 }

```

First we have to initialize the `mySqlDataAdapter`. We will initialize it with a "Select" query to select all from the **Composers** table.

```

1 mySqlDataAdapter = new MySqlDataAdapter("select * from composers", connection); ?

```

Next we will create a `DataSet` to load the table which is held by the `mySqlDataAdapter`.

```

1 DataSet DS = new DataSet(); ?

```

Now "fill" the `DataSet` with table connected to the `mySqlDataAdapter` with the following line of code.

```

1 mySqlDataAdapter.Fill(DS); ?

```

Only thing left now is to assign the prepared data source to the `DaraGridView` control.

```

1 dataGridView1.DataSource = DS.Tables[0]; ?

```

If you did everything correct, your `Form_Load` method. Should look like this.

```

1 private void Form1_Load(object sender, EventArgs e) ?
2 {
3     server = "localhost";
4     database = "ClassicalMusic";
5     uid = "username";
6     password = "password";
7     string connectionString;

```

```

8   connectionString = "SERVER=" + server + ";" + "DATABASE=" + database + ";" + "
9
10  connection = new MySqlConnection(connectionString);
11
12  if (this.OpenConnection() == true)
13  {
14      mySqlDataAdapter = new MySqlDataAdapter("select * from composers", connect
15      DataSet DS = new DataSet();
16      mySqlDataAdapter.Fill(DS);
17      dataGridView1.DataSource = DS.Tables[0];
18
19      //close connection
20      this.CloseConnection();
21  }
22 }

```

Run your project and see how the MySQL table is displayed in the DataGridView.



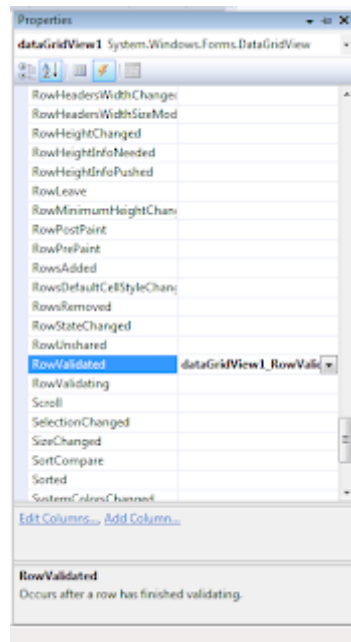
Index	Name	Era	Nationality
7	Felix Mendelssohn	Romantic	German
8	Frédéric Chopin	Romantic	Polish
9	Richard Wagner	Romantic	German
10	Johann Strauss II	Romantic	Austrian
11	Johannes Brahms	Romantic	German
12	Georges Bizet	Romantic	French
13	Pyotr Ilyich Tchaikovsky	Romantic	Russian
14	Antonín Dvořák	Romantic	Czech
15	Joseph Haydn	Classical	Austrian
16	Franz Schubert	Classical	Austrian
17	Antonio Salieri	Classical	Italian
18	Edvard Grieg	Modern	Norwegian
19	Edvard Elgar	Modern	English
20	Ralph Vaughan Williams	Modern	English

## Updating the MySQL table via the DataGridView control

You will notice that you can just type in the DataGridView and it will dynamically add rows, delete rows, alter cell contents, etc. But alas when you go and look at the MySQL table or reload your program, you will see that the changes are not reflected in the database. We need to fix this issue.

We have to select the best possible moment to update the MySQL table making that we do not submit half cooked data at the same time ensuring that all the valid changes are reflected in the database without any loss.

The best place to do this is the `RowValidated` event of the `DataGridView` control. Go to the Events section of the properties window for the `DataGridView` control and generate the code chunk for the `RowValidated` event.



We only have to commit the changes to the MySQL table. Committing all the data including the unchanged data is a waste of resources. So we use the following nifty bit of code to extract only the changes that we did.

```
1 | DataTable changes = ((DataTable)dataGridView1.DataSource).GetChanges();
```

A `RowValidated` event can occur even when there were no changes. In that case the above variable will be null. We have to take precautions to avoid the `NullReferenceException`.

```
1 | if (changes != null)
2 | {
3 |
4 | }
```

Remember, when we were loading the table to the `DataGridView` control we wrote the MySQL query to load the table to memory? One great thing about `mySqlDataAdapter` is the fact that it can generate the **Update** or **Delete** commands based on the **Select** command we used to create it with. To do this we need an instance of the `MySqlCommandBuilder` class created with our `mySqlDataAdapter` as a parameter.

```
1 | MySqlCommandBuilder mcb = new MySqlCommandBuilder(mySqlDataAdapter);
```



Now we can use the instance of the [MySQLCommandBuilder](#) to create an **Update** command and assign it to our `mySqlDataAdapter`.

```
1 | mySqlDataAdapter.UpdateCommand = mcb.GetUpdateCommand();
```

Using the said update command and the *changes* DataSet we created earlier, we can update our MySQL database.

```
1 | mySqlDataAdapter.Update(changes);
```

Are we done? No! Why? Because we have to tell the DataGridView control that we have committed the changes it had. Otherwise next time when we ask for changes we will be in trouble. This is done by the following line of code.

```
1 | ((DataTable)dataGridView1.DataSource).AcceptChanges();
```

If you did everything correct, your `RowValidated` event handler. Should look like this.

```
1 | private void dataGridView1_RowValidated(object sender, DataGridViewCellEventArgs e)
2 | {
3 |     DataTable changes = ((DataTable)dataGridView1.DataSource).GetChanges();
4 |
5 |     if (changes != null)
6 |     {
7 |         MySqlCommandBuilder mcb = new MySqlCommandBuilder(mySqlDataAdapter);
8 |         mySqlDataAdapter.UpdateCommand = mcb.GetUpdateCommand();
9 |         mySqlDataAdapter.Update(changes);
10 |        ((DataTable)dataGridView1.DataSource).AcceptChanges();
11 |    }
12 | }
```

Done and done! Run your code and see how the MySQL table gets updated.

And here is the line you scrolled down for; here are the links to download the [sample code](#) and the [MySQLdatabase](#).

Posted in: [.NET](#), [C#](#), [code](#), [DataBase](#), [DataGridView](#), [DataSource](#), [DataTable](#), [events](#), [Microsoft SQL](#), [MySQL](#), [MySQLCommandBuilder](#), [mySqlDataAdapter](#), [open source](#)

[Home](#)

[Older Post](#)

**18 comments:****NISHAL DARSHANA** says:

February 3, 2013 at 9:27 PM

nice article. congratz ...!!!

Reply

**Nisansa Dilushan** says:

February 3, 2013 at 9:43 PM

Thank you! :-D

Reply

**kechong nguyen** says:

July 2, 2013 at 7:50 AM

I keep getting "Concurrency violation: the UpdateCommand affected 0 of the expected 1 records" everytime I make a change and it calls mySqlDataAdapter.Update .... any ideas?

Reply

**David Woosley** says:

December 29, 2013 at 5:02 PM

Nice article. Unfortunately, I can't use it because I have no idea why some website named 4shared.com is wanting to download an executable to my computer. Silly. Why put the code in a large text box or provide a link to a text file or ZIP file or anything simple?

Reply

**Simo Venäläinen** says:

May 25, 2014 at 11:59 AM

How do I hide the first column, that contains table primary key? I don't want to show it in the dgw.

[Reply](#)**lingmaaki** says:

October 8, 2014 at 8:12 AM

Find here a complete...[C# Datagridview](#) Tutorial.

Ling

[Reply](#)**Mary Joey Intas** says:

March 3, 2015 at 9:21 AM

This error occurred highlighting this line: `mySqlDataAdapter.UpdateCommand = mcb.GetUpdateCommand();`

ERROR:

An exception of type 'System.InvalidOperationException' occurred in System.Data.dll but was not handled in user code

Additional information: The DataAdapter.SelectCommand property needs to be initialized.

[Reply](#)**วิสิทธิ์ แพ้วกระโทก** says:

November 14, 2015 at 9:53 PM

Nice tutorial , Thank you so much.

[Reply](#)

**Stathis Votsis** says:

December 10, 2015 at 5:51 AM

You save my life!!!!Thanks

Reply

**Anonymous** says:

December 12, 2015 at 7:10 AM

Thanks man

Reply

**Bardnet** says:

March 6, 2016 at 8:36 AM

How would I parameterize the query and update the GridView when I change an input?

Reply

**cha** says:

July 17, 2016 at 11:26 PM

*the author.*

Reply

*This comment  
has been  
removed by***CJ** says:

July 17, 2016 at 11:32 PM

An Answer for:

This error occurred highlighting this line: `mySqlDataAdapter.UpdateCommand = mcb.GetUpdateCommand();`

ERROR:

An exception of type 'System.InvalidOperationException' occurred in System.Data.dll but was not handled in user code

Additional information: The DataAdapter.SelectCommand property needs to be initialized.

---

SET PRIMARY KEY FIELD FOR TABLE

Reply



**danile yoo** says:

February 18, 2017 at 4:57 AM

Thank you.

But my code occur concurrency exception like above people's comments.  
I checked my table in database, the table has field include primary key.  
how to solve this problem?

my table property is below

```
CREATE TABLE tblData
(
  ScheduleName VARCHAR(255),
  StepNo INT(11) unsigned NOT NULL AUTO_INCREMENT PRIMARY KEY,
  Type VARCHAR(32) NOT NULL ,
  Volt FLOAT(11) unsigned NOT NULL ,
  Current FLOAT(11) unsigned NOT NULL ,
  cutoff_time TIME(3 ),
  modified_time TIMESTAMP(2)
);
```

some data inserted by below query  
insert into tbldata values ("my\_schedule", NULL, "charge" , 1.23, 45.8, "11:22:33.456" , NOW ( ) );

Reply



**Cengiz CGL** says:

December 20, 2017 at 6:05 AM

Really helpfull!

Reply

**Unknown** says:

May 9, 2018 at 7:26 AM

*the author.**This comment  
has been  
removed by*

Reply

**Cognegic** says:

May 22, 2018 at 4:40 AM

Instructions to Solve C# MySQL Connection Problem through MySQL Backup Database

At Cognegic we will help you to take care of your C# MySQL association issue and furthermore enable you to recognize the root to cause behind your specialized issues. Here we give MySQL Enterprise Backup support and MySQL Remote Support to recoup from any blackouts. We have quite a while of experience and devoted specialized specialists who resolve to tackle your issue and make you blunder free. Along these lines, don't sit tight for some other help organization essentially contact to our MySQL Remote Service and get the best help.

For More Info: <https://cognegicsystems.com/>

Contact Number: 1-800-450-8670

Email Address- [info@cognegicsystems.com](mailto:info@cognegicsystems.com)

Company's Address- 507 Copper Square Drive Bethel Connecticut (USA) 06801

Reply

**Cognegic** says:

July 23, 2018 at 1:37 AM

Get in Touch with MySQL Technical Support for Common Errors When Using MySQL Programs | Cognegic  
At whatever point we are working with MySQL there might be odds of confronting some kind specialized blunders or glitches. These sorts of issues forestall you to work with your MySQL. Possibly you don't know that you will discover numerous specialized issues some of them are: "Secret key fizzled when entered intuitively", "Correspondence blunders and prematurely ended associations", "Table defilement issues" and some more. Ensure; in the event that you communicate with these issues than without quite a bit of stretch rapidly connect with Cognegic's MySQL Remote Support and MySQL Remote Service.

For More Info: <https://cognegicsystems.com/>

Contact Number: 1-800-450-8670

Email Address- [info@cognegicsystems.com](mailto:info@cognegicsystems.com)

Company's Address- 507 Copper Square Drive Bethel Connecticut (USA) 06801

Reply

## Post a Comment

Enter your comment...



Comment as:

nopya2@gmail ▼

Sign out

Publish

Preview

☐ Notify me

