DATABASES APPLICATION DEVELOPED IN THE VISUAL STUDIO PROGRAMMING ENVIRONMENT

Gîlcă Gheorghe, Lecturer Phd, *"Constantin Brâncuși" University from Târgu Jiu, ROMANIA*
Ionescu Marian, Phd Student, *"Constantin Brâncuși" University from Târgu Jiu, ROMANIA*

ABSTRACT: *The article aims to develop a visual application with databases in visual studio .net. The paper shows the following important aspects such as: creating a simple interface in visual studio, how to connect a database to the interface, how to define the database in visual studio, how the bindingnavigator can be used, performing some basic operations on the created database.*

KEY WORDS: database, connection, data source, table designer, bindingnavigator

1. INTRODUCTION

In the Visual Studio .NET development system you can create visually (by selecting from toolbars) both the communication objects with the database (connections, commands, adapter, data set), as well as the graphical controls for displaying the data: display lists (ListView), display boards (DataGrid) and connections between them [1].
Visual Studio and .NET together provide extensive API and tooling support for connecting to databases, modeling data in memory, and displaying the data in the user interface. The .NET classes that provide data-access functionality are known as ADO.NET. ADO.NET, along with the data tooling in Visual Studio, was designed primarily to support relational databases and XML. These days, many NoSQL database vendors, or third parties, offer ADO.NET providers [2].
.NET Core supports ADO.NET, except for datasets and their related types. If you're targeting .NET Core and require an object-relational mapping (ORM) layer, use Entity Framework Core.
The following diagram shows a simplified view of the basic architecture [3]:

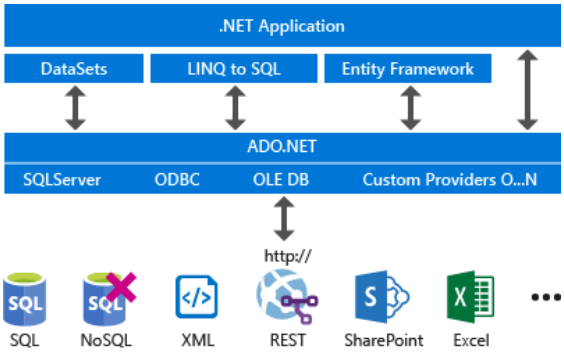


Fig. 1. The diagram of the .net core architecture [3]

2. CREATE A PROJECT AND A LOCAL DATABASE

In order to attach a blank database to a visual studio project, we must go through the following steps:

1. Create a new Windows Forms App project and name it.
2. On the menu bar, select Project > Add New Item.
3. In the list of item templates, scroll down and select Service-based Database.
4. Name it the database, and then click Add.

Figure 2 shows the primary interface to form Form1 file created in Visual Studio Windows Forms.

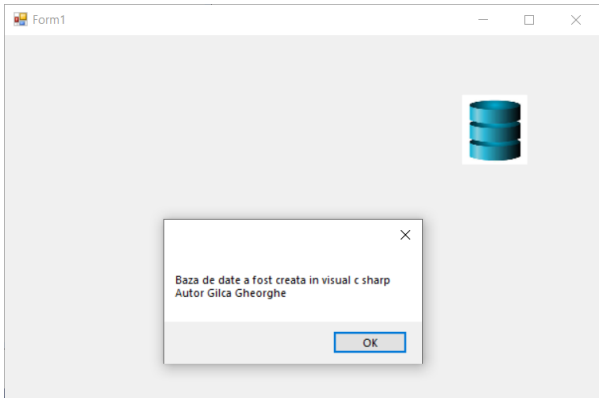


Fig. 2. Creating the project interface in windows forms

A. Add a data source to our project:

1. If the Data Sources window isn't open, open it by pressing Shift+Alt+D or selecting View > Other Windows > Data Sources on the menu bar.
2. In the Data Sources window, select Add New Data Source. The Data Source Configuration Wizard opens.
3. On the Choose a Data Source Type page, choose Database and then choose Next.
4. On the Choose a Database Model page, choose Next to accept the default (Dataset).
5. On the Choose Your Data Connection page, select the Contacte.mdf file in the drop-down list, and then choose Next.
6. On the Save the Connection String to the Application Configuration File page, choose Next.
7. On the Choose your Database Objects page, you'll see a message that says the database doesn't contain any objects. Choose Finish.

We can view the connection string for the Contacte.*mdf* file by opening the Properties window of the data connection:

- Select View > SQL Server Object Explorer to open the SQL Server Object Explorer window. Expand (localdb)\MSSQLLocalDB > Databases, and then right-click on Contacte.*mdf* and select Properties.
- Alternatively, we can select View > Server Explorer, if that window isn't already open. Open the Properties window by expanding the Data Connections node, right-clicking on Contacte.*mdf*, and then selecting Properties.

B. Create tables and keys by using Table Designer

In this section, we'll create a table with ten columns, a primary key in table, and a few rows of sample data.

Create the Contacte table

1. In Server Explorer, expand the Data Connections node, and then expand the Contacte.mdf node. If you can't expand the Data Connections node, or the Contacte.mdf connection is not listed, select the Connect to Database button in the Server Explorer toolbar. In the Add Connection dialog box, make sure that Microsoft SQL Server Database File is selected under Data source, and then browse to and select theSampleDatabase.mdf file. Finish adding the connection by selecting OK.

2. Right-click on Tables and select Add New Table. The Table Designer opens and shows a grid with one default row, which represents a single column in the table that you're creating. By adding rows to the grid, you'll add columns in the table.
3. In the grid, add a row for each of the entries.
4. Right-click on the ContactID row, and then select Set Primary Key.
5. Right-click on the default row (Id), and then select Delete.
6. Name the Contacte table by updating the first line in the script pane to match the following sample.
7. In the upper-left corner of Table Designer, select Update.
8. In the Preview Database Updates dialog box, select Update Database. The Contacte table is created in the local database file.

Figure 3 shows the table construction to configure the database. For each field, the type of data contained is selected and can be checked if it allows nullity (except making the primary key which cannot be null). Figure 4 shows the database, containing 10 lines and 10 columns.

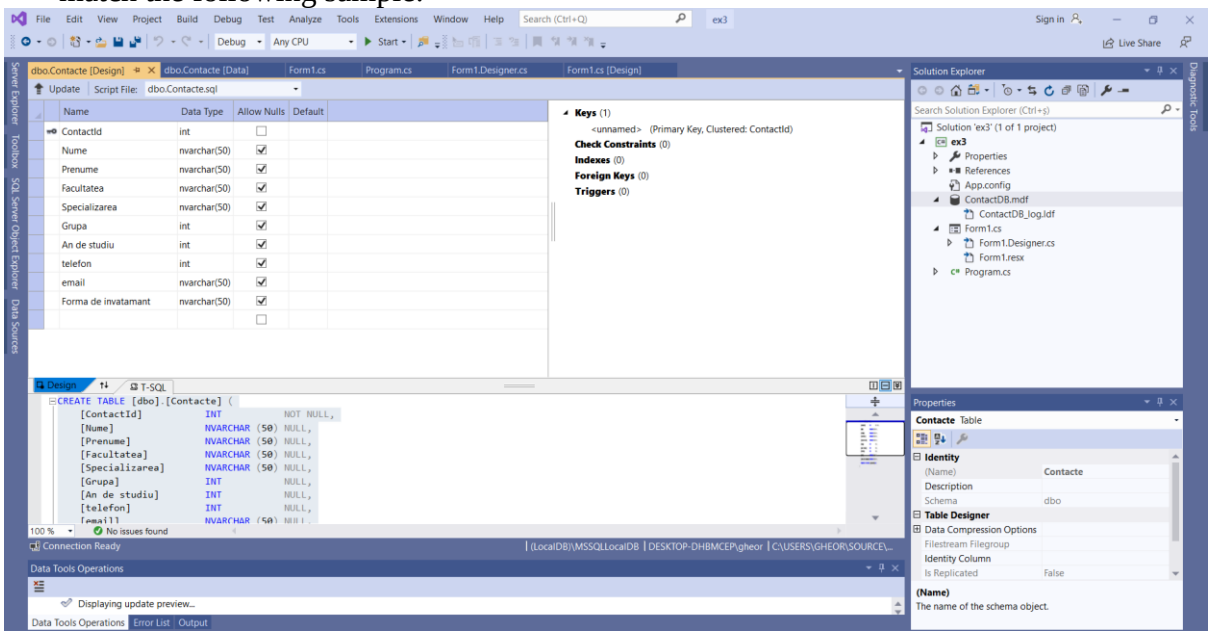


Fig. 3. Define the table for the database

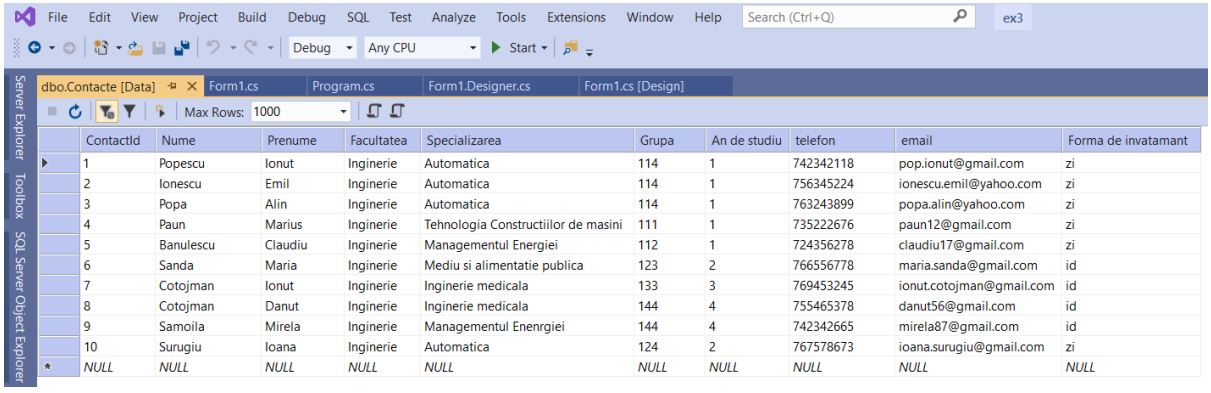
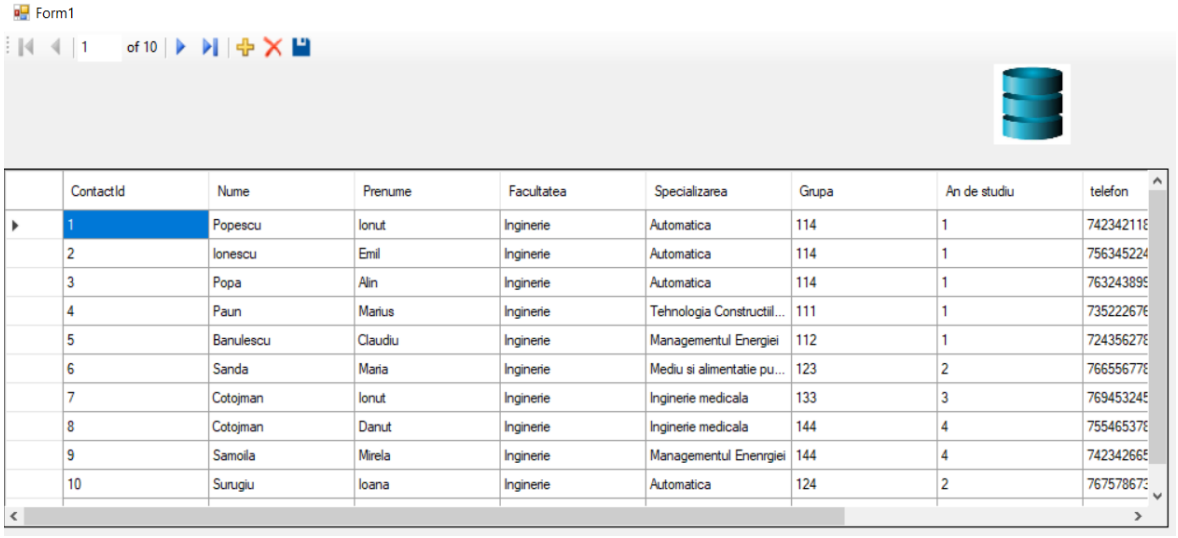


Fig. 4. The display the database created in visual studio

3. BINDINGNAVIGATOR CONTROL OVERVIEW (WINDOWS FORMS)

We can use the BindingNavigator control to create a standardized means for users to search and change data on a Windows Form. You frequently use BindingNavigator with the BindingSource component to enable users to move through data records on a form and interact with the records.

The BindingNavigator control is composed of a ToolStrip with a series of ToolStripItem objects for most of the common data-related actions: adding data, deleting data, and navigating through data [4]. By default, the BindingNavigator control contains these standard buttons. The following screenshot shows the BindingNavigator control on a form created.



ContactId	Nume	Prenume	Facultatea	Specializarea	Grupa	An de studiu	telefon
1	Popescu	Ionut	Inginerie	Automatica	114	1	742342118
2	Ionescu	Emil	Inginerie	Automatica	114	1	756345224
3	Popa	Alin	Inginerie	Automatica	114	1	763243895
4	Paun	Marius	Inginerie	Tehnologia Constructiil...	111	1	735222676
5	Banulescu	Claudiu	Inginerie	Managementul Energiei	112	1	724356278
6	Sanda	Maria	Inginerie	Mediu si alimentatie pu...	123	2	766556778
7	Cotojman	Ionut	Inginerie	Inginerie medicala	133	3	769453245
8	Cotojman	Danut	Inginerie	Inginerie medicala	144	4	755465378
9	Samoila	Mirela	Inginerie	Managementul Enerrgiei	144	4	742342665
10	Surugiu	Ioana	Inginerie	Automatica	124	2	767578673

Fig. 5. The display the database using binding browser

4. THE BASIC OPERATIONS ON THE CREATED DATABASE

In the created database we can execute basic commands that are applied to most databases: selection, insertion, deletion.

A. Selection from the database

The selection operation is performed to select only certain data from the database. Figure 6 presents the requirement for selection from the database of students who are in the specialization Automatic and group 114.

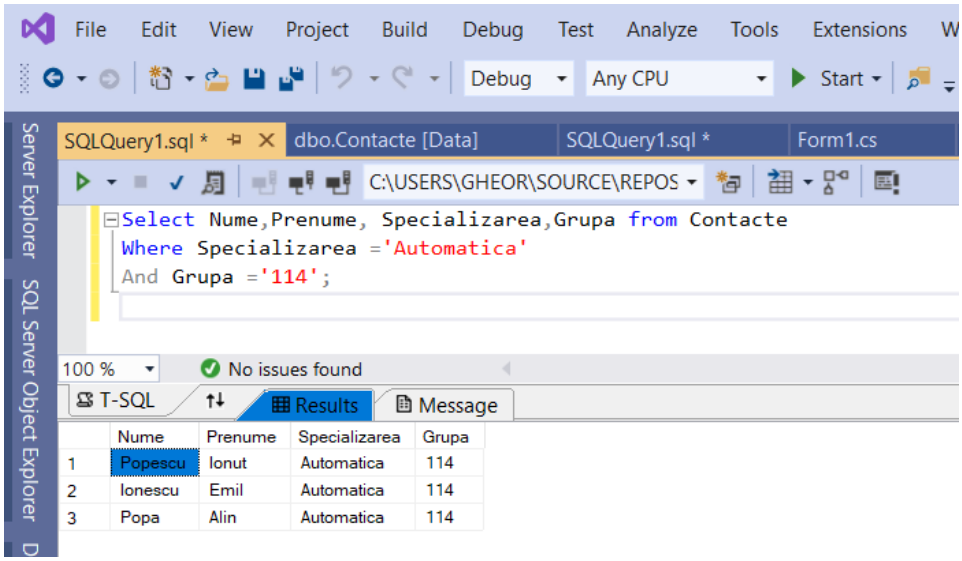


Fig. 6. The interrogation of the database for the selection and display of the result

B. The insertion into a database

This operation is important because we almost always have to enter data after the database has been created. Figure 7 shows the database that was completed with the newly inserted row, this being the row with the number 11.

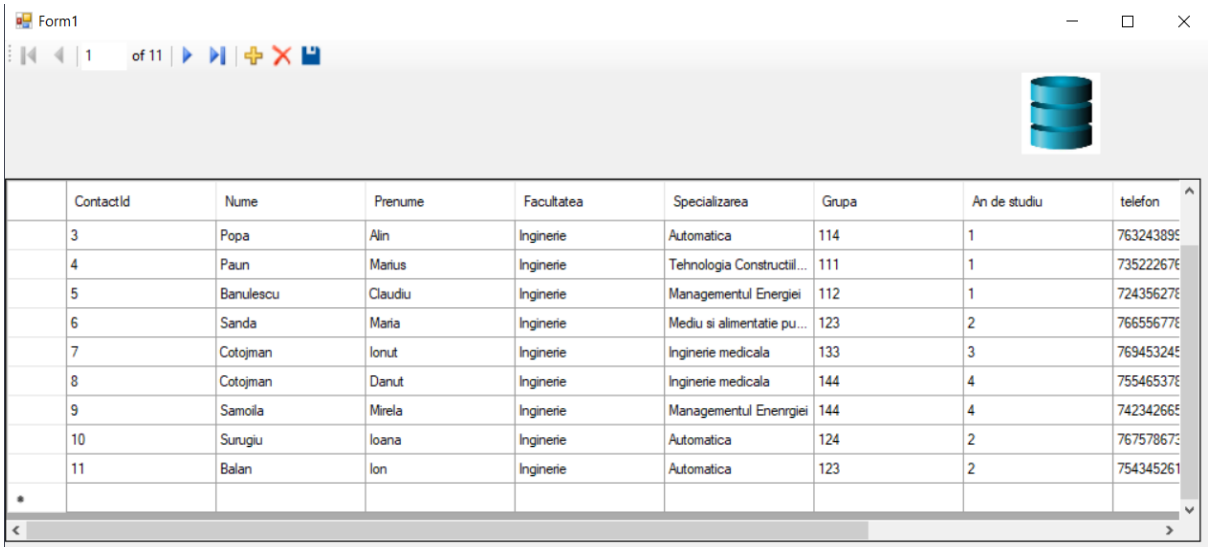


Fig. 7. The insertion of a new row in the database

C. Deleting from the database

The deleting operation is performed to remove data from the database.
Figure 8 shows the result of erasing the student with id 5.

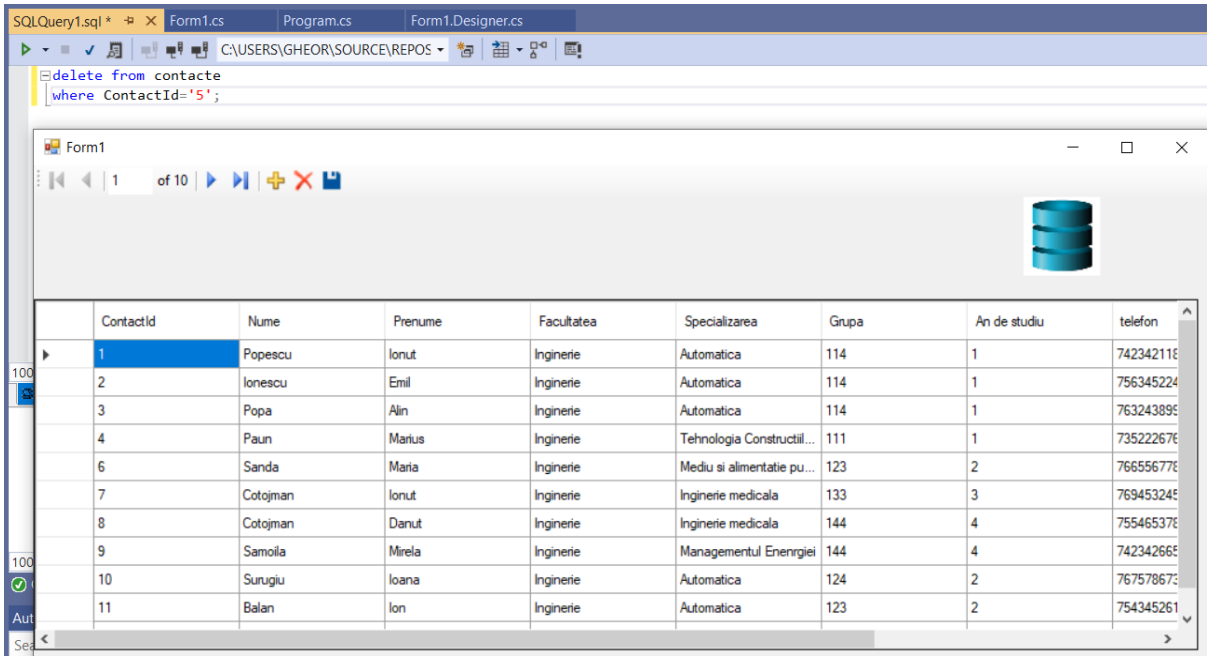


Fig. 8. The deleting of row with id 5 from the database

5. CONCLUSIONS

The Visual Studio .net development environment has powerful and suggestive tools for using databases in applications. Every institution or private company currently uses at least one database with information, and the simple way of creating and interrogating the database in visual studio would improve the widespread use of these types of applications. Conceptually, behind a window where we work with data retrieved from one or more tables of a database are the objects in the categories Connection, Command, DataAdapter and DataSet. What can be seen there are controls of the type DataGridView, or TableGridView, BindingNavigator, etc.

REFERENCES

[1] Gilca N., Gilca G., *Create a timed math quiz in visual c#*, Annals of Constantin Brâncuşi University of Târgu-Jiu - Engineering Series, Issue 1, 2019, pp 26-32.

[2] Herbert Schildt, *C#: A Beginner's Guide*, 2001.

[3] https://docs.docker.com/assemble/dot-net/.

[4] https://docs.microsoft.com