



Copilot

To connect a C# GUI application to a SQL Server database, you have a few options. Let's explore a couple of common approaches:

1. Using LINQ to SQL:

- **Step 1:** Add a connection to your **Server Explorer** in Visual Studio. Once added, it will appear under the **Server Explorer** tab on the left side of your screen.
- **Step 2:** Add a **Linq to SQL file (.dbml)** to your project.
- **Step 3:** Open the `.dbml` file from the file explorer on the right-hand side of Visual Studio.
- **Step 4:** Open up your connection via the drop-down arrows on your server browser to see the tables you want to use in your console app.
- **Step 5:** Drag the desired tables onto the design area.
- **Step 6:** Go back to your `.cs` page for your console application, instantiate the data context, and start working with your data [1](#).

2. Using System.Data.SqlClient Namespace:

- **Step 1:** Install **Visual Studio 2019 Community** and create an empty C# project.
- **Step 2:** Add the `System.Data.SqlClient` namespace to your C# project.
- **Step 3:** Set up a **SQL Server connection string**.
- **Step 4:** Perform **SELECT**, **UPDATE**, and **INSERT** operations from your C# program using SQL commands [2](#).

3. Creating a Simple Windows Form:

- Use the following code snippet to create a simple Windows form that accesses a SQL Server database:

C#



```
SqlConnection con = new  
SqlConnection(@"server=.SQLEXPRESS;AttachDbFilename=path_of_database_file;In  
tegrated Security=True;User Instance=True");
```

AI-generated code. Review and use carefully. [More info on FAQ.](#)

Replace `path_of_database_file` with the actual path to your SQL Server database file [3](#).

Remember to adjust the steps based on your specific requirements and project setup. Good luck with

Ask me anything...



0/2000

