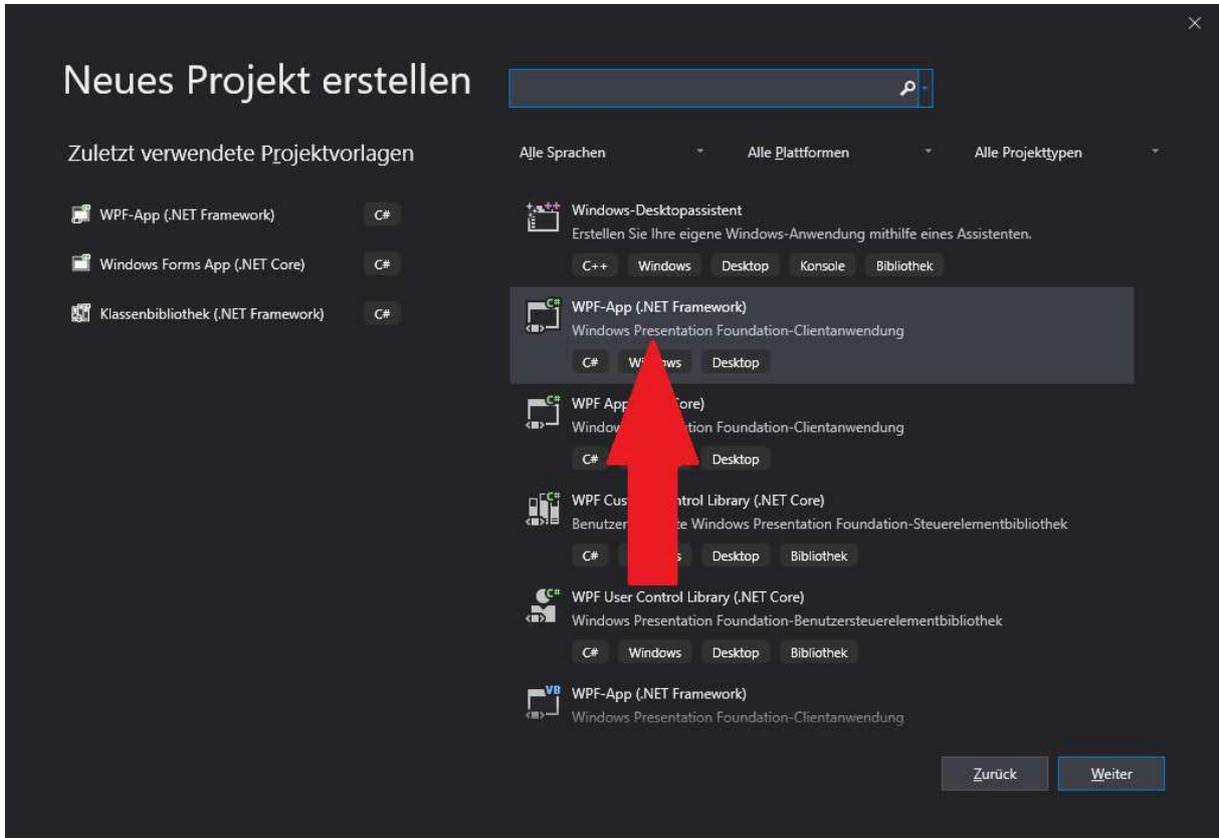
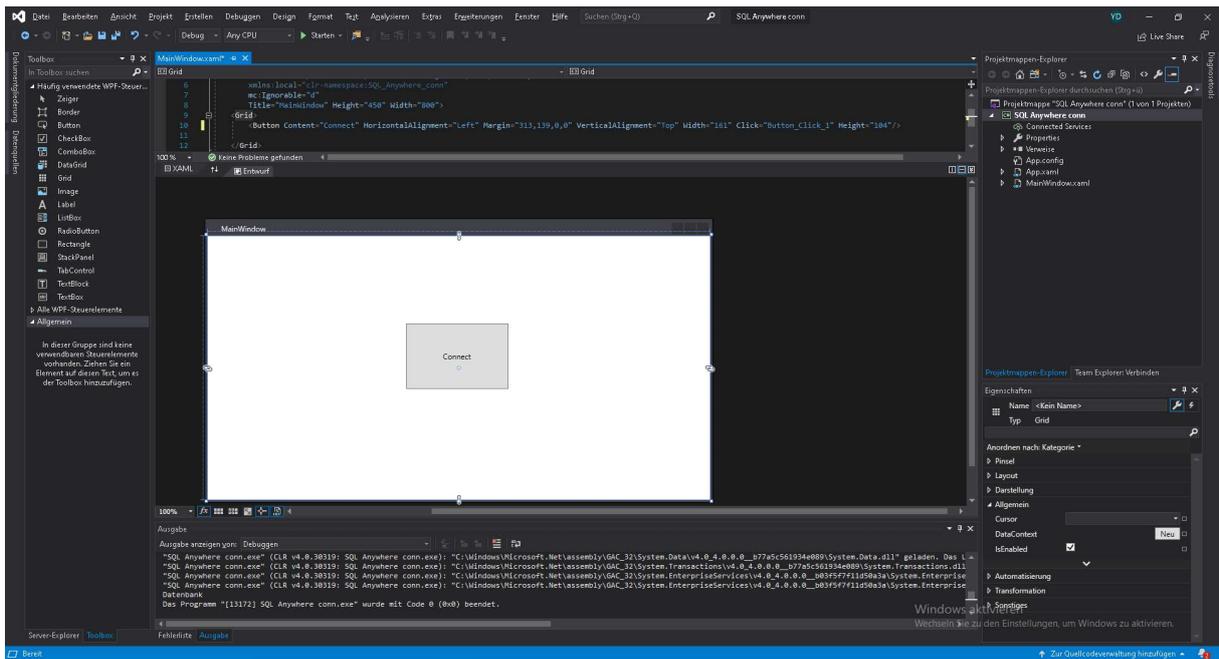


# SQL Anywhere in C#

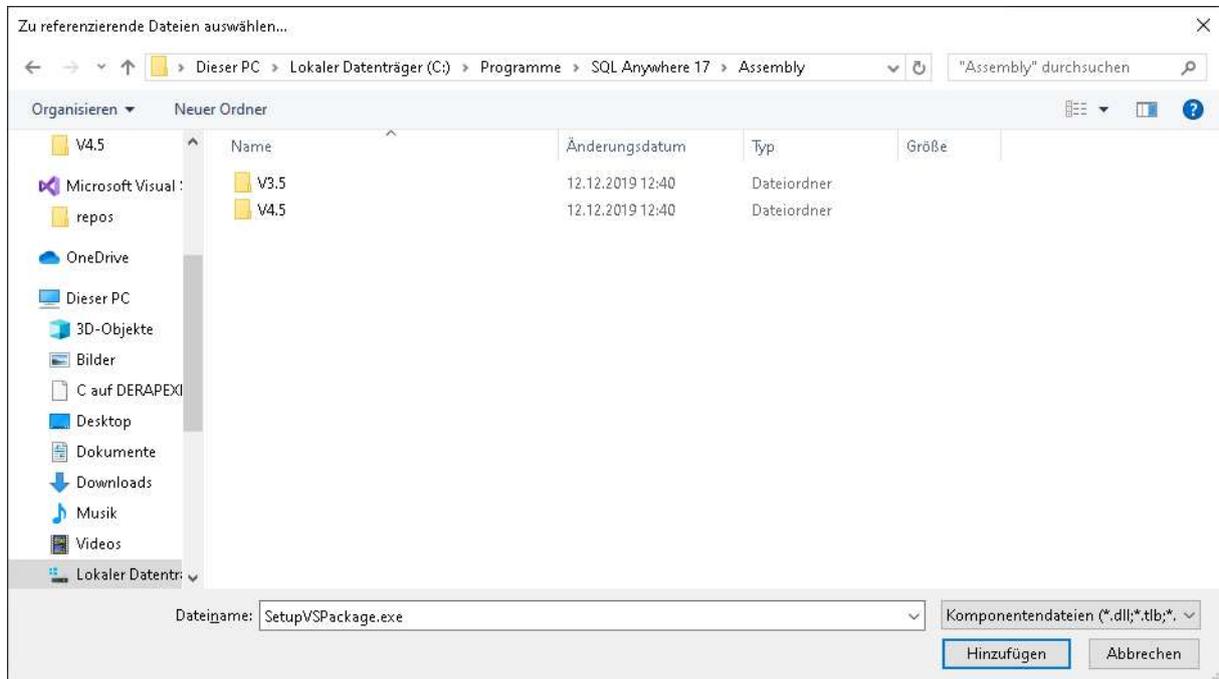
Zuerst sollten Sie Visual Studio öffnen und erstellen ein neues Projekt. Das Projekt, das Sie erstellen, sollte eine WPF-App sein, die ein .Net-Framework verwendet.



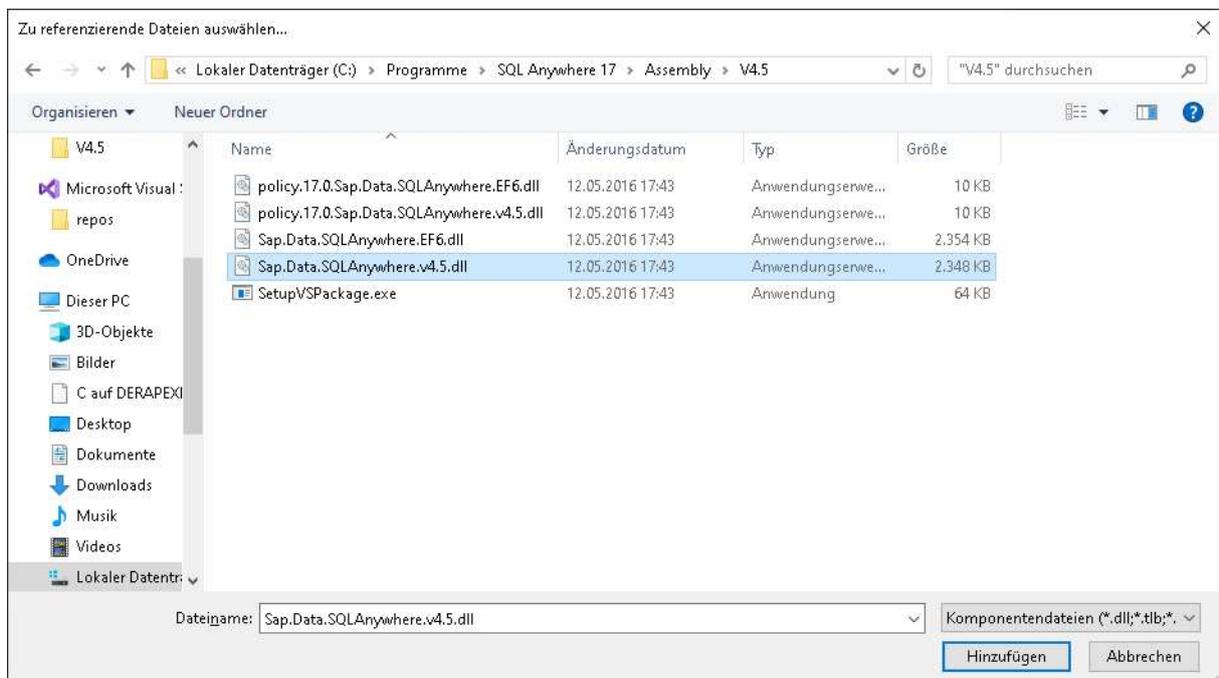
Nachdem Sie das Projekt erstellt haben, fügen Sie im MainWindow einen Button ein.



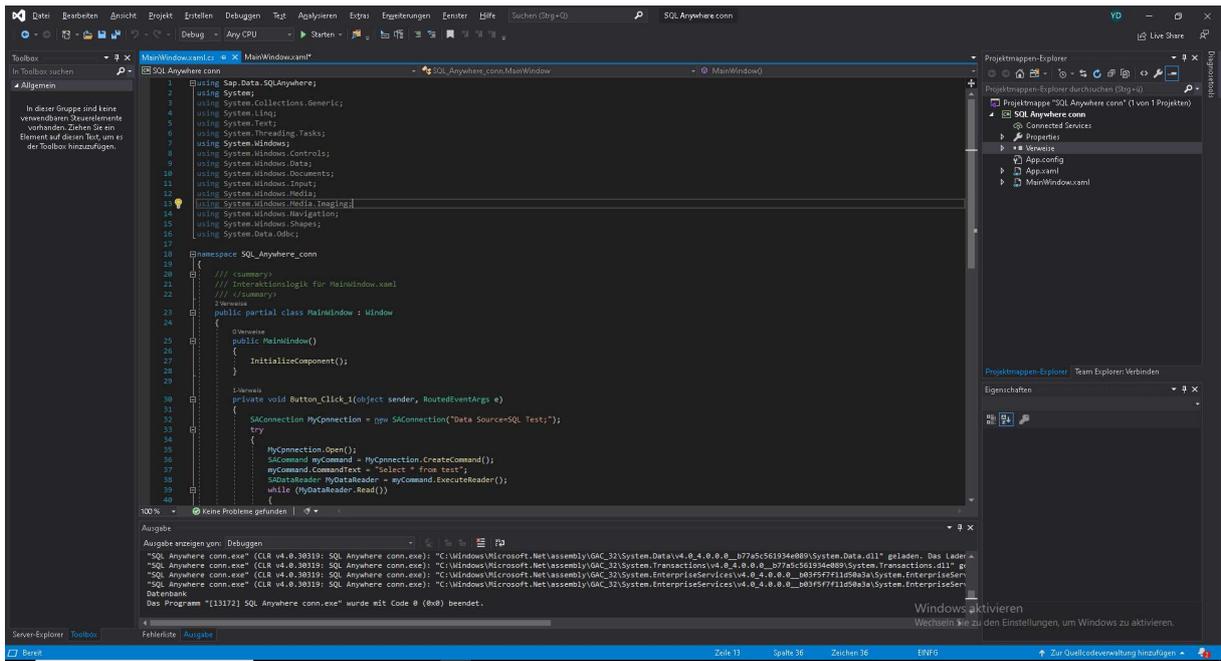
Anschließend müssen Sie einen neuen Verweis hinzufügen. Sie können diesen in der oberen Leiste bei "Projekt" finden oder machen im Projektmappen-Explorer bei Verweis einen Rechtsklick und klicken daraufhin auf "Verweis hinzufügen...". Die Assembly Datei, die Sie benötigen, finden Sie im Ordner "Assembly" der sich im Ordner von SQL Anywhere befindet. Sie werden zwei Ordner in "Assembly" vorfinden. Ermitteln Sie vorher welches .Net-Framework Sie verwenden.



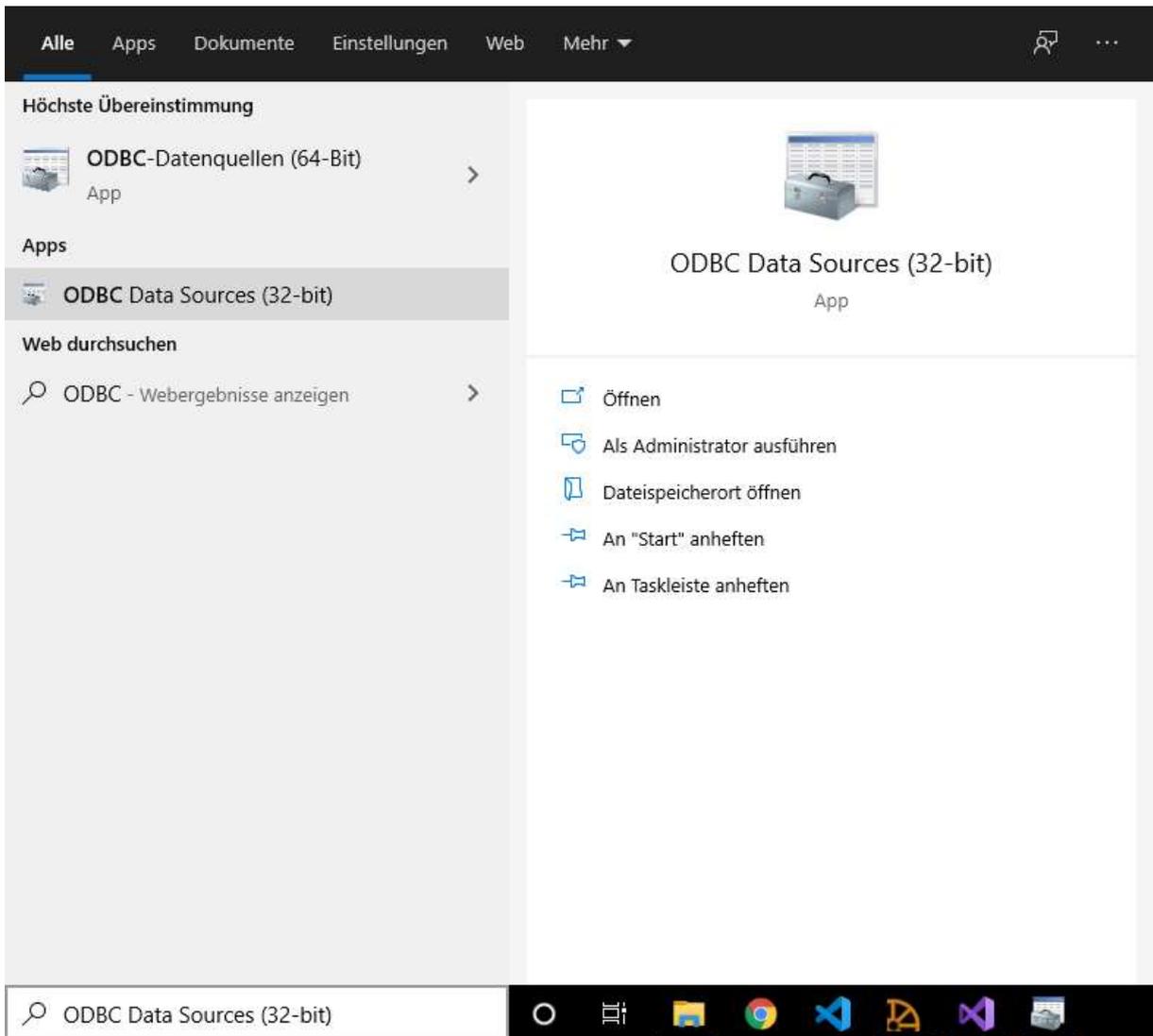
Navigieren Sie in den Ordner, je nach dem, welche .Net-Framework Version Sie verwenden. In dem Ordner benötigen Sie dazu folgende Assembly: "SAP.Data.SQLAnywhere.vX.X.dll". Die Xe sind Zahlen, die von den Versionen vom .Net-Framework abhängig sind.



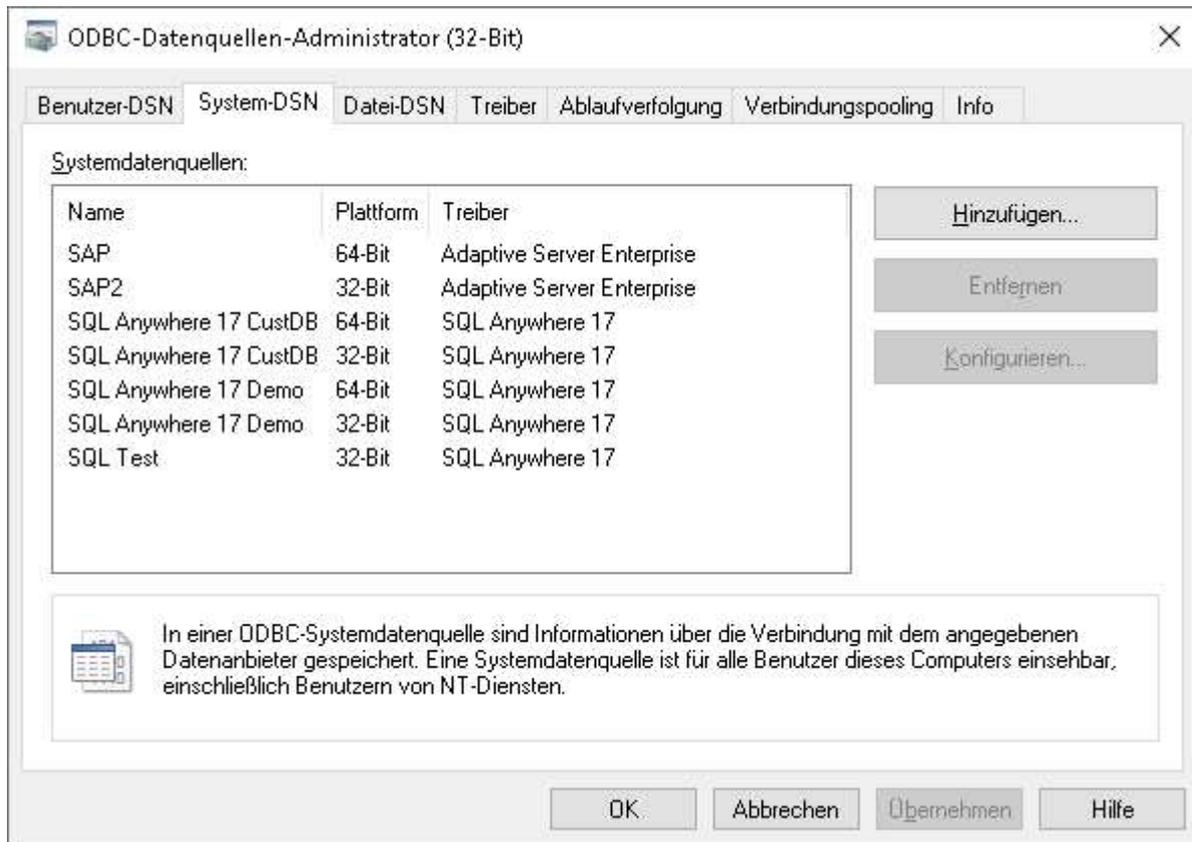
Fügen Sie die Assembly Datei nun dem Verweis hinzu. Darauffolgend sollte es möglich sein, mit den Components der Assembly Datei zu einem SQL Anywhere Server, eine Verbindung aufzubauen. Binden Sie die Assembly in das Programm mit: "using Sap.Data.SQLAnywhere" ein.



Bevor Sie Code schreiben können, sollten Sie eine ODBC-Verbindung erstellen. Dafür öffnen Sie die "ODBC-Datenquelle 32-Bit".



Je nach Gebrauch, des ODBC's erstellen Sie eine System-DSN oder Benutzer-DSN. Achten Sie darauf, dass der Treiber für den SQL Anywhere 17 eingestellt ist.



Nachdem Sie den Treiber ausgewählt haben, taucht ein Fenster auf in dem Sie folgende Angaben machen:

ODBC-Konfiguration für SQL Anywhere

ODBC Login Netzwerk Sicherheit Erweitert

Datenguellename:

Beschreibung:

Isolationsstufe:

Microsoft-Anwendung (Schlüssel in SQLStatistics)

Delphi-Anwendungen

Fetch-Warnungen unterdrücken

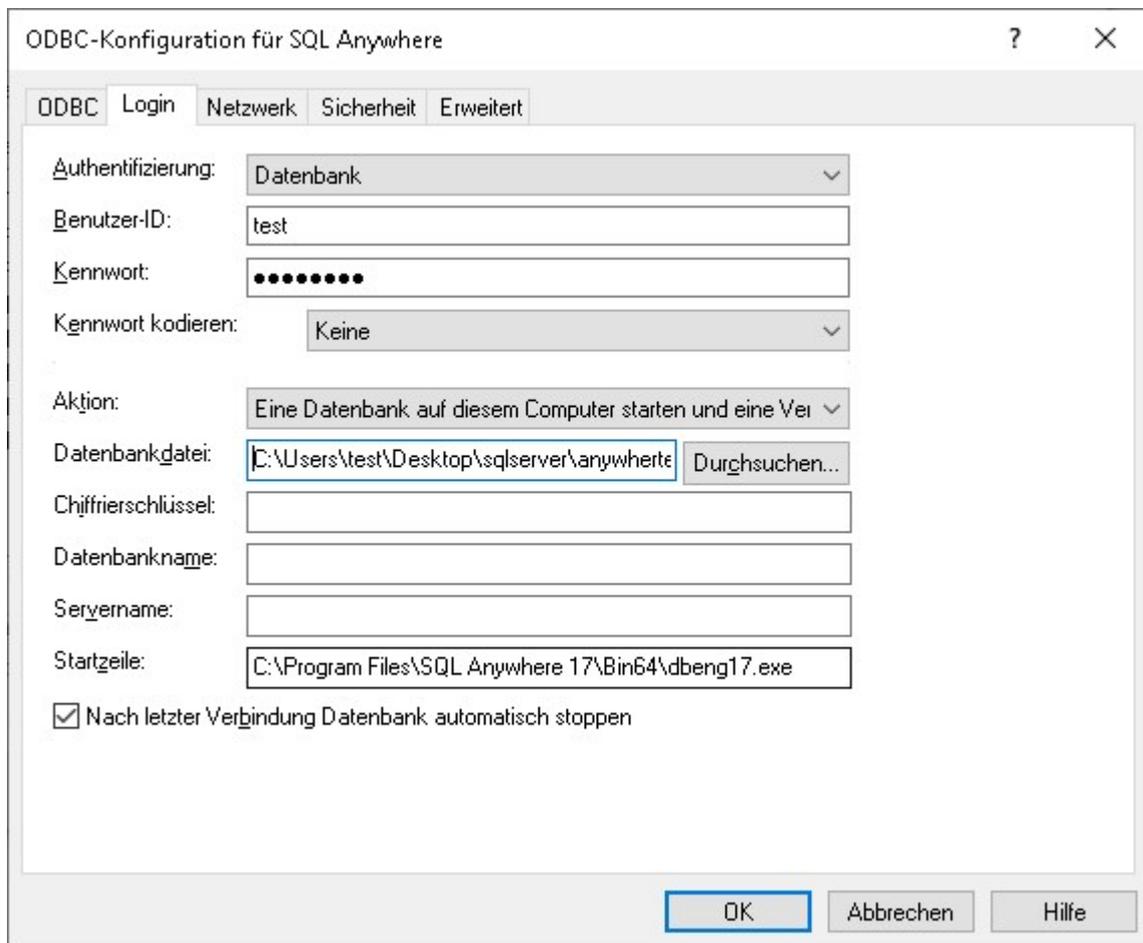
Treiberfunktionsfehler ausblenden

AutoCommit nach Anweisungsende

Cursor neu beschreiben

Nie  Wenn nötig  Immer

Nach der Benennung des DSN, wechseln Sie das Fenster auf Login. Sie können dort optional die Benutzer-ID, sowie das Passwort eingeben. Wählen Sie Folgendes aus: "Mit einer laufenden Datenbank auf diesen Computer verbinden" oder "Eine Datenbank auf diesem Computer starten und eine Verbindung herstellen". In diesem Beispiel wird die zweite Option verwendet.



Nach der Erstellung der ODBC-DSN können Sie folgenden Code verwenden, um sich mithilfe des ODBC-Treibers zum SQL Anywhere zu verbinden:

```

SAConnection MyConnection = new SAConnection("Data Source=<DSN-Name>");

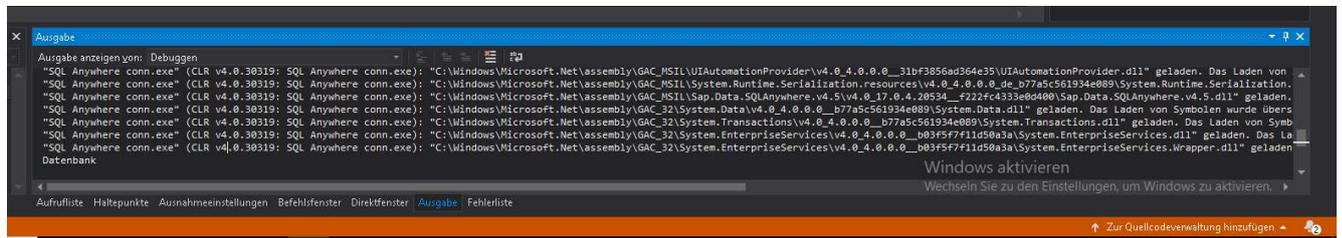
try
{
    MyConnection.Open();
    SACommand myCommand = MyConnection.CreateCommand();
    myCommand.CommandText = "Select * from <Name der Tabelle>";
    SADataReader MyDataReader = myCommand.ExecuteReader();
    while (MyDataReader.Read())
    {
        Console.WriteLine("{0}", MyDataReader["Spalte von der Tabelle"]);
    }
    MyConnection.Close();
}
catch (Exception sql)
{
    Console.WriteLine(sql);
}

```

}

Wenn Sie im ODBC im Login keinen Benutzernamen und Passwort angegeben haben, ergänzen Sie diese im Connection String. Dieser sähe wie folgt aus: " SAConnection MyConnection = new SAConnection("Data Source=<DSN-Name>;UID=<Username>;PWD=<Passwort>");"

Wenn alles klappt, sollte Folgendes passieren, wenn Sie auf den Button drücken:



Bei Fragen:

[y.duong@tobax.de](mailto:y.duong@tobax.de)

Tel. 02171 3606 19

[www.tobax.de](http://www.tobax.de)

Yao Zu Duong