

- Northwind DB, tables Customers and Orders in 1:n
- Visual Studio -> *Create a new project -> Windows Forms App (.NET Framework)*
- specify a Project name, choose a Location, Create
- *Toolbox* -> drag and drop 2 *DataGridView* controls and a *Button* onto the form
- rename *DataGridViews* to *dgvCustomers* and *dgvOrders* (select *DataGridView* on form, change (*Name*) value in the *Properties* window)
- rename *Button* to *btnUpdateDB*, change its text to *Update DB* ((*Name*) and *Text* in the *Properties* window)
- right click on *Form1.cs* in *Solution Explorer* -> *View Code*

```
//namespaces
using System.Data;
using System.Data.SqlClient;
...

//required objects
SqlConnection dbConn;
SqlDataAdapter daCustomers, daOrders;
DataSet ds;
SqlCommandBuilder cb;
BindingSource bsCustomers, bsOrders;
...

//get data from the db, e.g., in the handler for the form Load's event,
//which can be created by double clicking on the form
dbConn = new SqlConnection(@"Data Source = SQLSRVEXPRESS; Initial Catalog =
Northwind; Integrated Security = true");
ds = new DataSet();
daCustomers = new SqlDataAdapter("SELECT * FROM Customers", dbConn);
daOrders = new SqlDataAdapter("SELECT * FROM Orders", dbConn);
cb = new SqlCommandBuilder(daOrders);
daCustomers.Fill(ds, "Customers");
daOrders.Fill(ds, "Orders");
DataRelation dr = new DataRelation("FK_Orders_Customers",
ds.Tables["Customers"].Columns["CustomerID"],
ds.Tables["Orders"].Columns["CustomerID"]);
ds.Relations.Add(dr);
//data binding
bsCustomers = new BindingSource();
bsCustomers.DataSource = ds;
bsCustomers.DataMember = "Customers";
bsOrders = new BindingSource();
bsOrders.DataSource = bsCustomers;
bsOrders.DataMember = "FK_Orders_Customers";
dgvCustomers.DataSource = bsCustomers;
dgvOrders.DataSource = bsOrders;
```

- run the app (F5)

- when a different customer is selected in the customers DataGridView, the orders DataGridView displays the selected customer's orders

...

//add code to send changes back to the db (the handler for the click event of the *update db* button, which can be created by double clicking the button on the form)
daOrders.Update(ds, "Orders");

- run the app
- add / modify / remove orders through the orders DataGridView
- update the DB by clicking the button
- check the changes in the DB