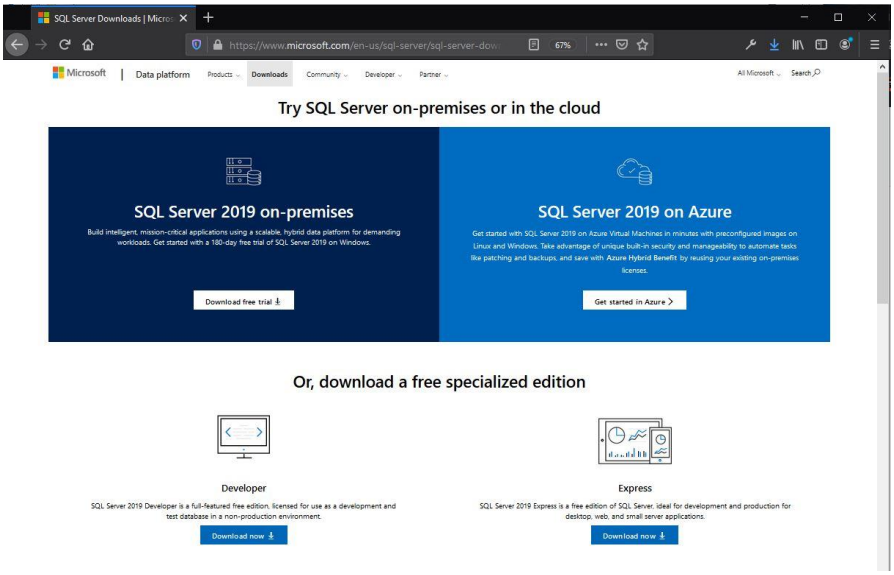
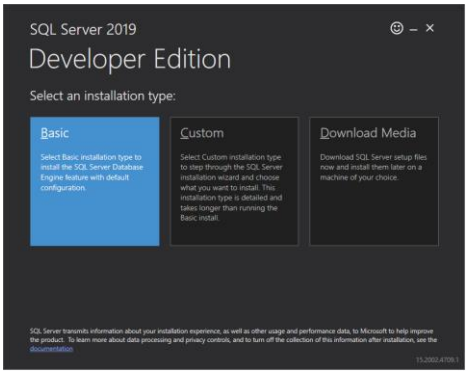
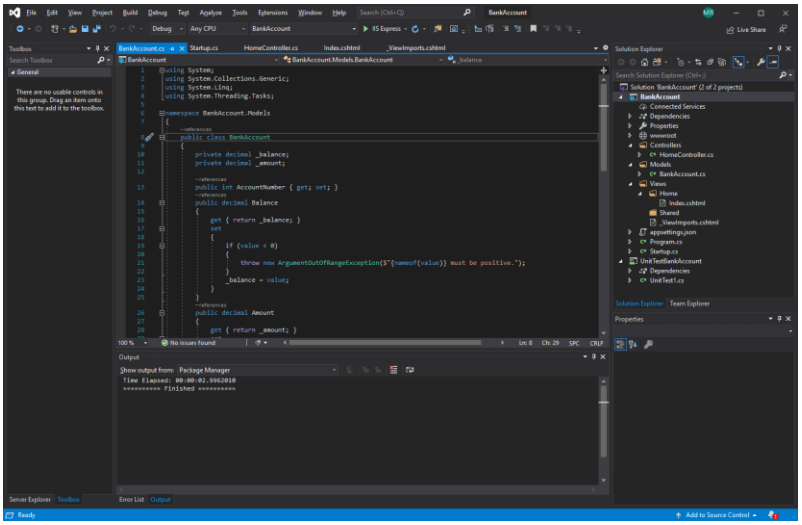
Installation

First [SQL Server Developer Edition,](#) Basic.



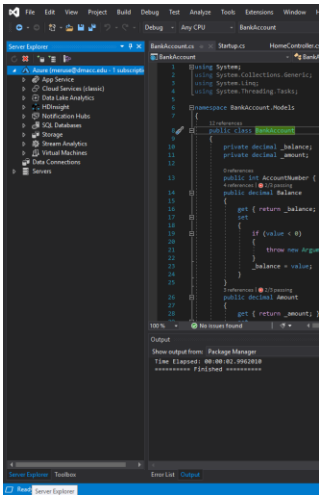


Open Visual Studio 2019, your BankAccount Solution.

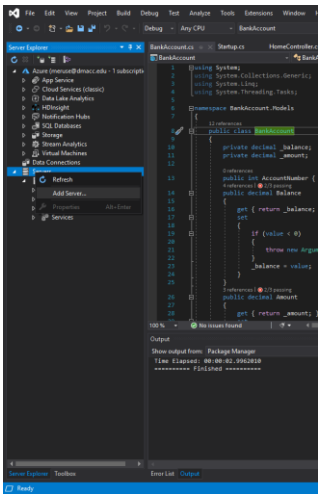


Open SQL Server Object Explorer

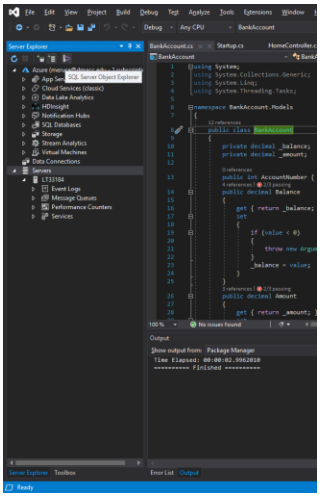
Open the Server Explorer



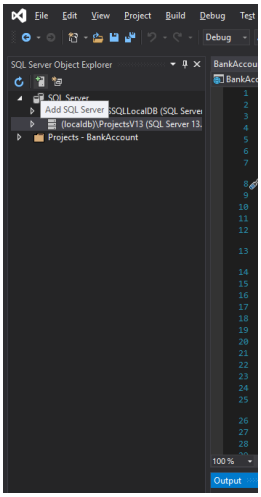
Click Servers and Add



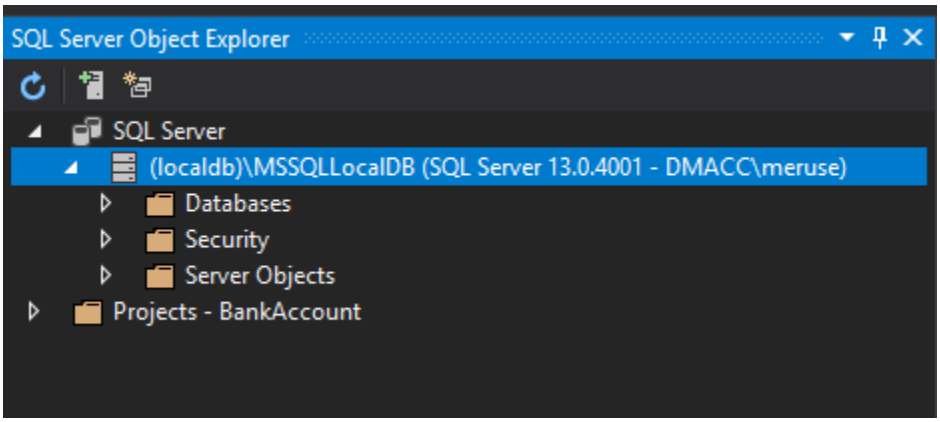
Open SQL Server Object Explorer



Click Add SQL Server

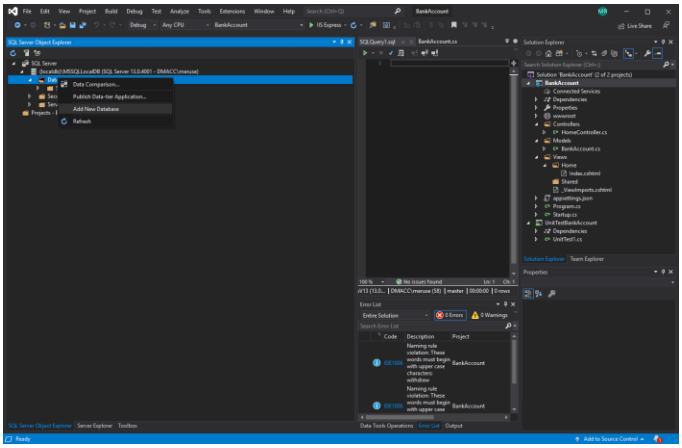


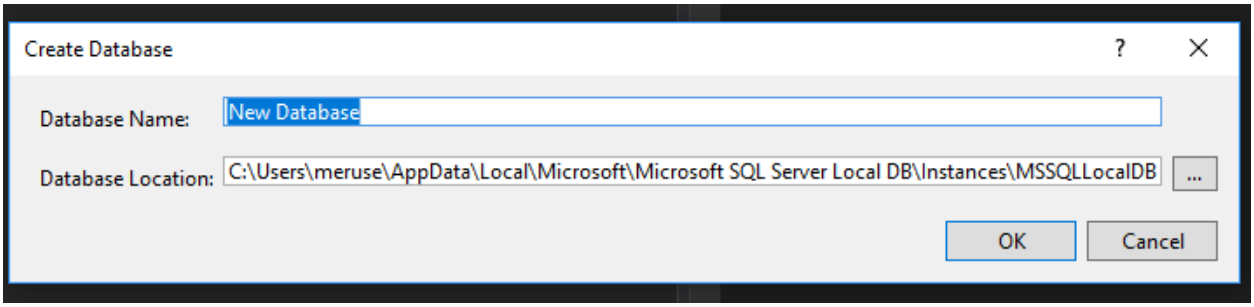
Choose to add a local database, and put in your connection strings. Refer to your install screen shot, if necessary. If you have used this before, it was likely already there.



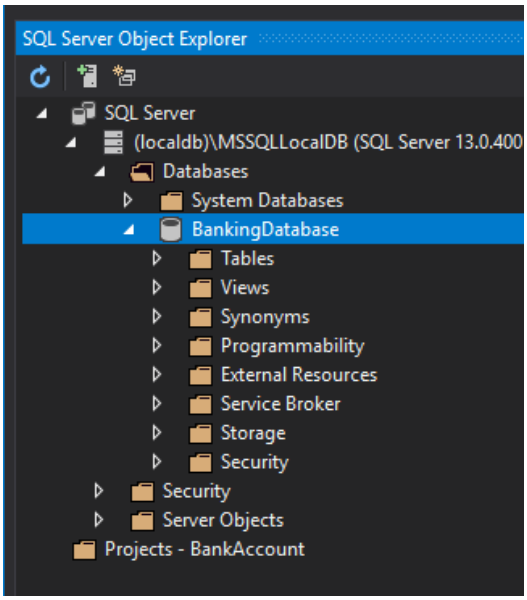
Create a Database

Right click on the Databases in your local server, Add New Database





Name it BankingDatabase



Adding Tables

Using Table Create View

Right click on Tables, Add New Table

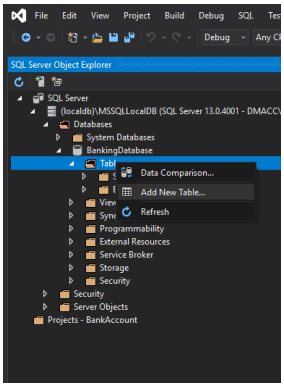
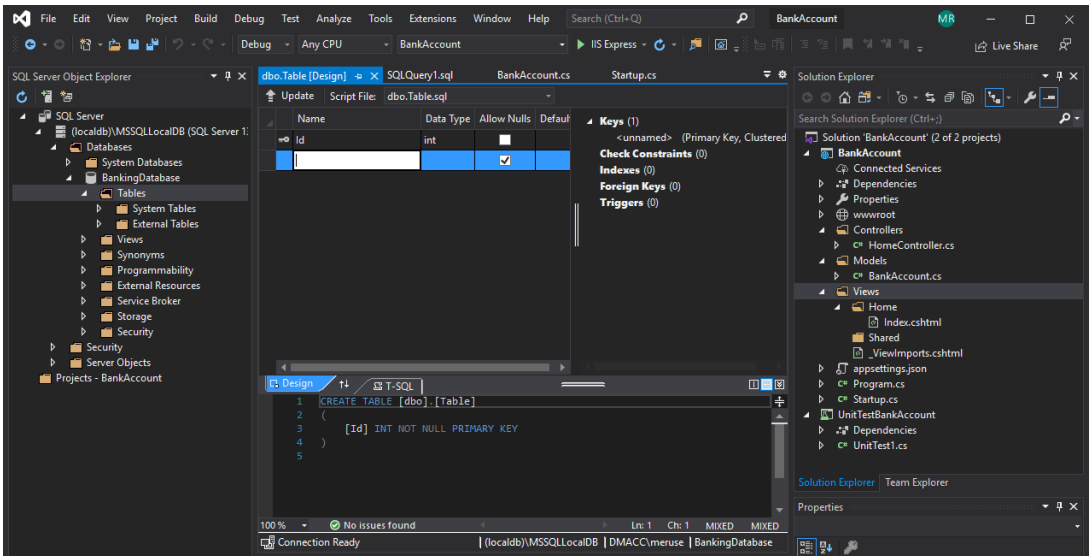


Table Create view:



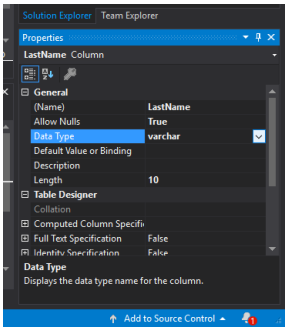
Type LastName under the Id property.

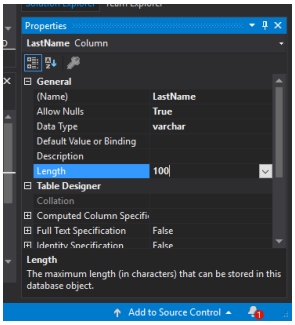
In the Properties box

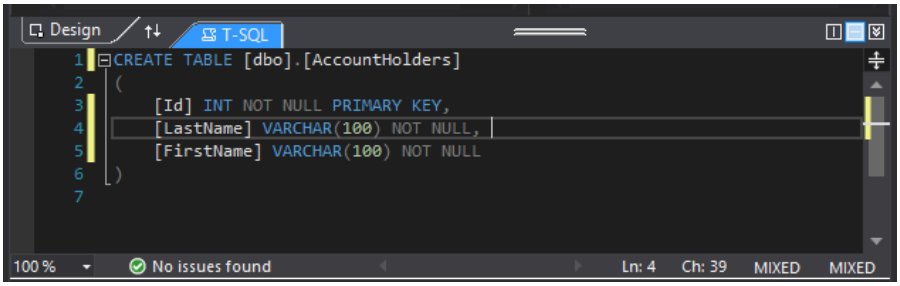
Change the data type to varchar

Change length to 100

Repeat for FirstName, uncheck Allow Nulls.

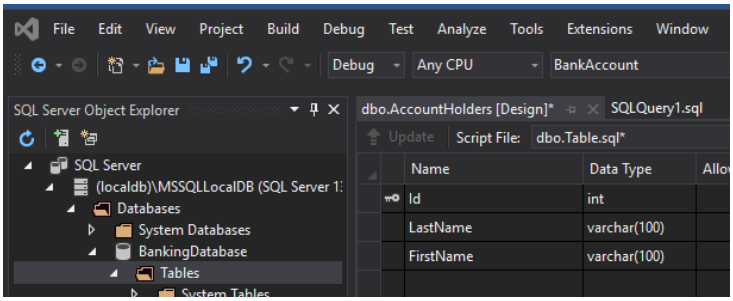


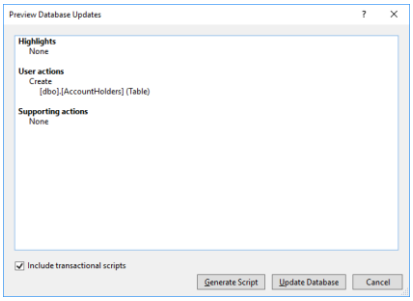




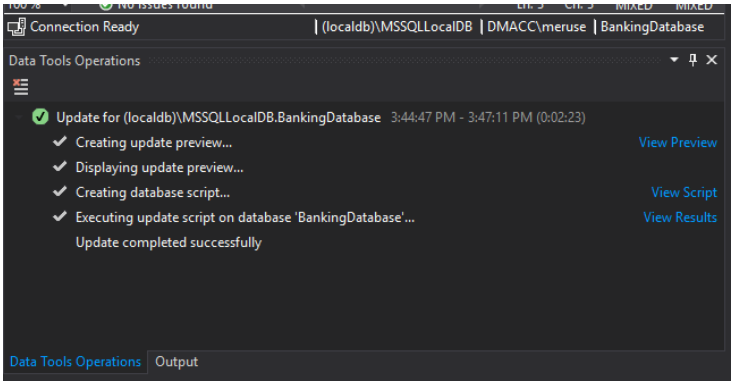
In the Design T-SQL box change Table to AccountHolders.

The table is not yet created, the tools allow the design of the database and tables. Verify your SQL code is the same as above, Update the database with the Update button:





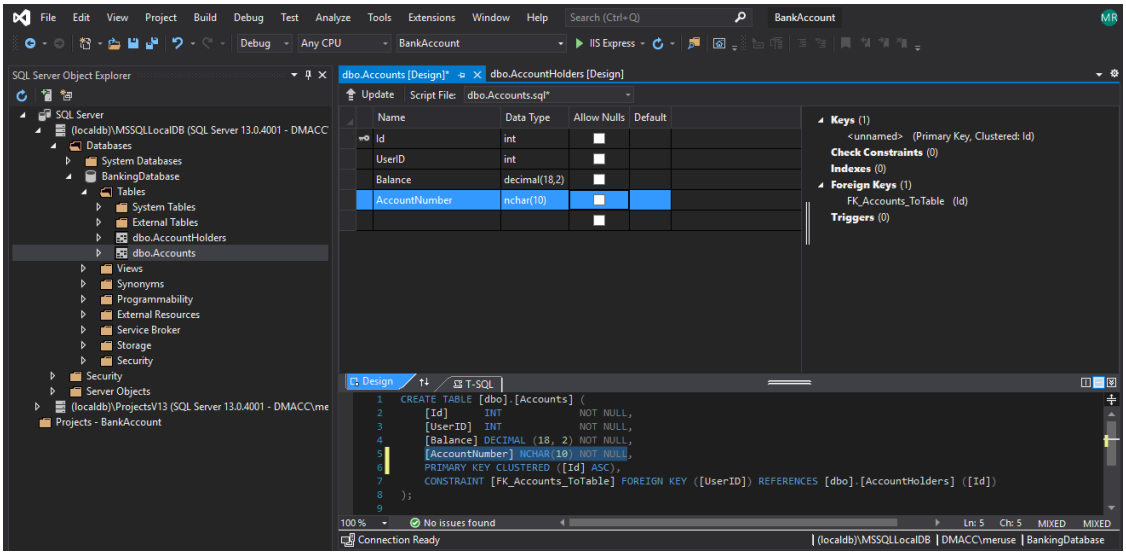
The output will show if successful:



Using a Generated Script

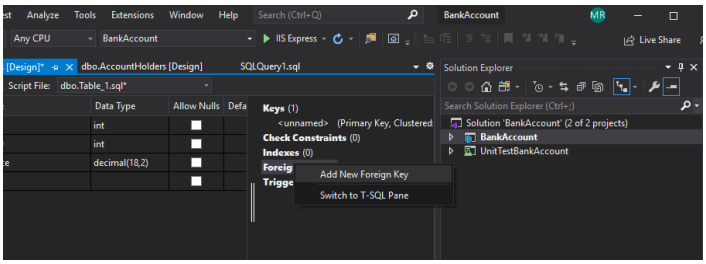
Repeat above for the Table Accounts. This will have a Foreign Key, so do not Update until you add the Foreign Key in the instructions that follow.

Accounts has UserId of type int, AccountNumber of type nchar(10), and Balance of type decimal 18, 2. Use property box to set data type, Precision to 18 and Scale to 2. Uncheck Allow Nulls. Continue to add the Foreign Key.

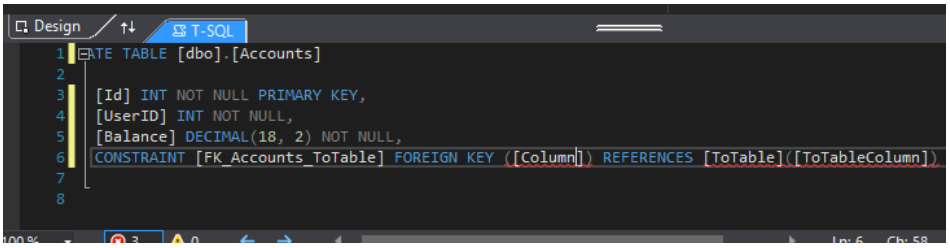


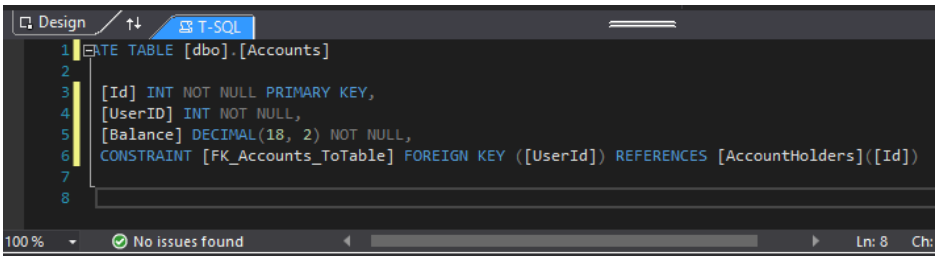
Add Foreign Key

To the right of the table fields, click on Foreign Keys, Add New Foreign Key

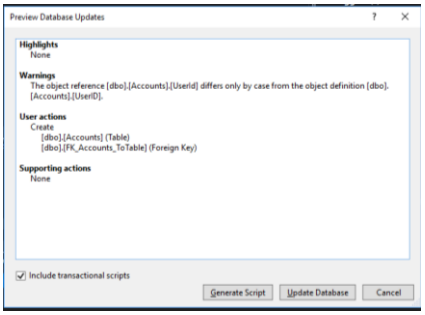


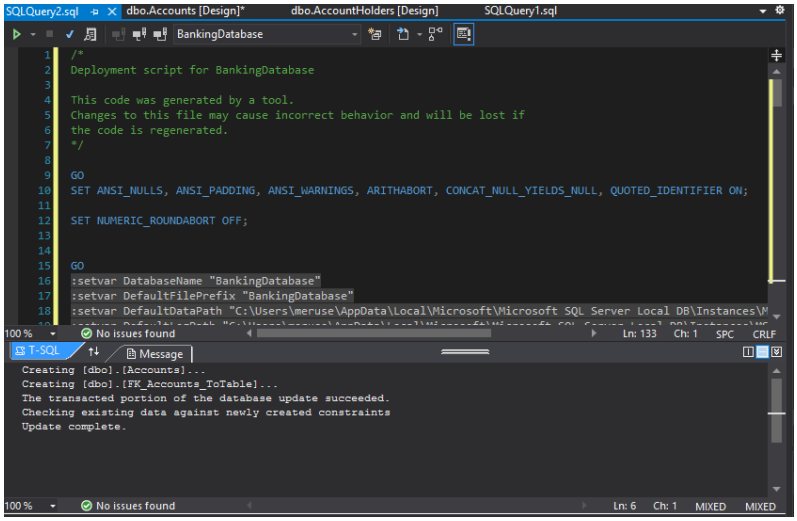
Notice the T-SQL had red lines, you change those to refer to UserID REFERENCES AccountHoders Id





Ready to update! Click Update, and this time, click Generate Script.



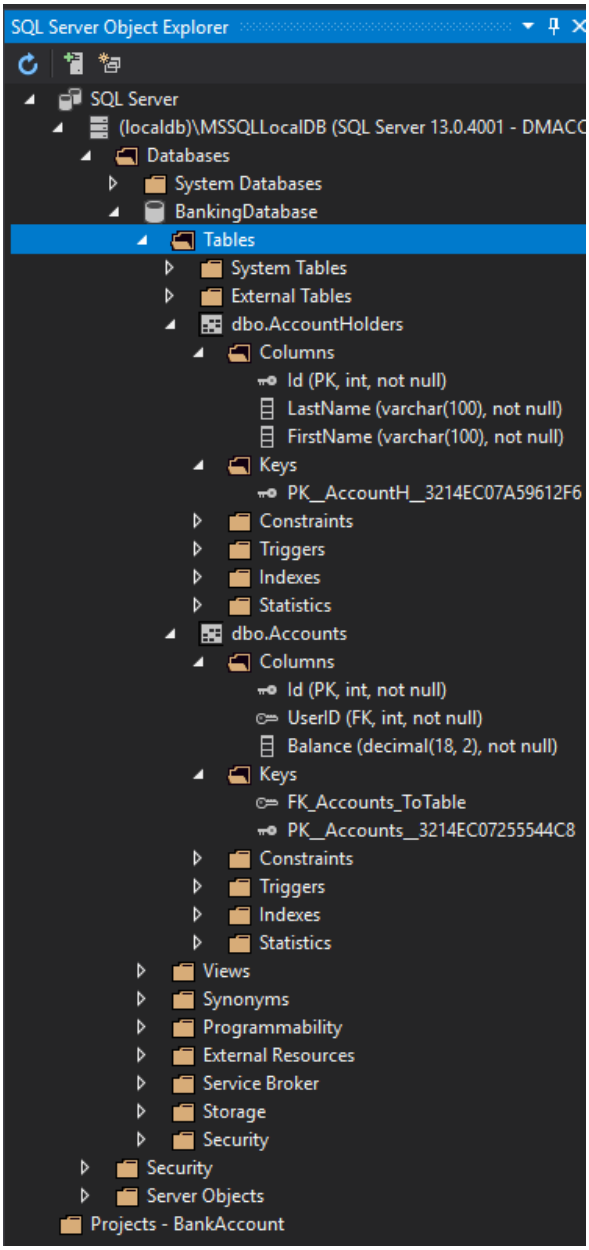


Once the script is generated, click the green Run icon to create the table.

You may be asking why the account numbers is Accounts. One Account Holder could have many accounts. Also, what is nchar? Try a quick search to find out what nchar is.

If you updated and did not generate script, you still have the new table! This was just to show you the tools work in several ways. If you like writing SQL code, you can use scripts to create a database table as well.

Expand in your SQL Object Server for each new Table in your database Columns and Keys you just created.



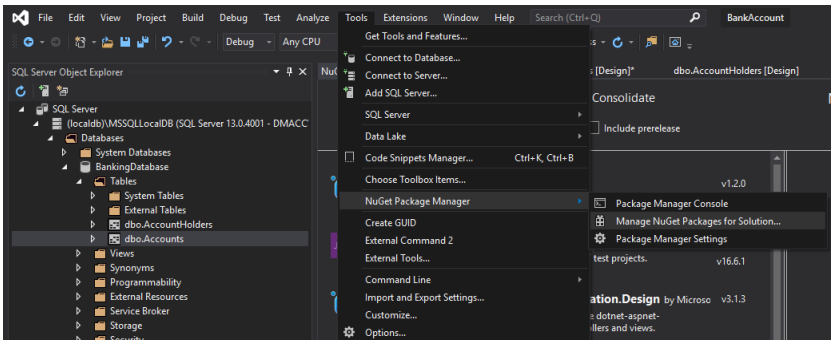
Take a screen shot, name it **CreateDatabase.jpg**

Continue to the next section to use the database.

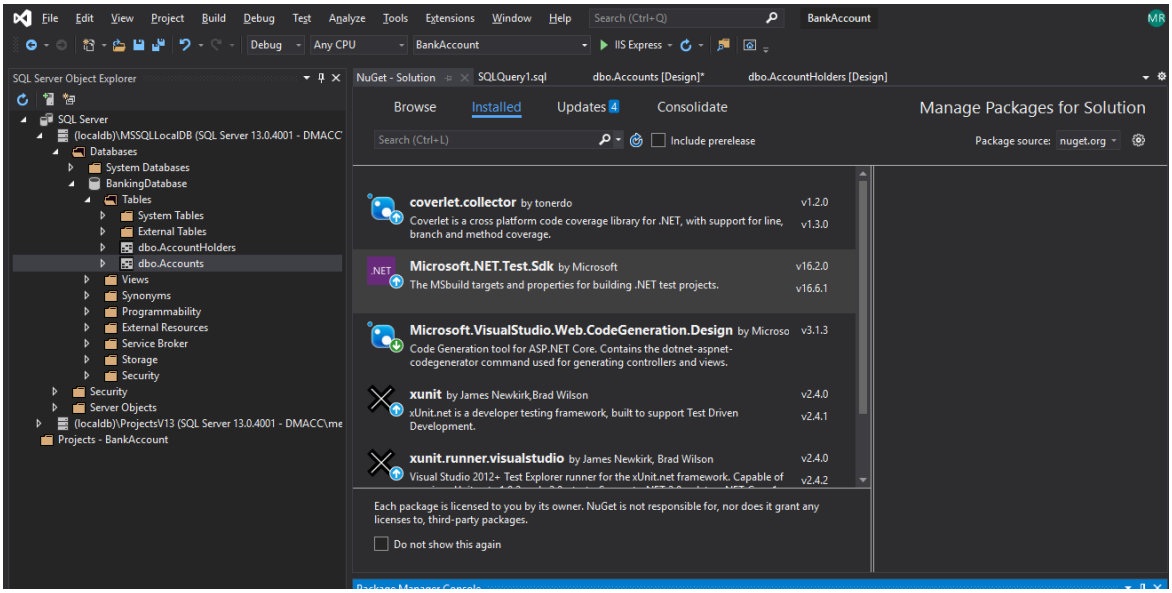
Use a Database

Install NuGet Packages

First in your Web App, Install NuGet packages by first going to Tools->NuGet Package Manager->Manage NuGet Packages for Solution…

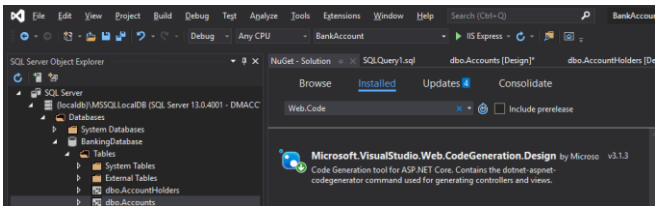


You will see

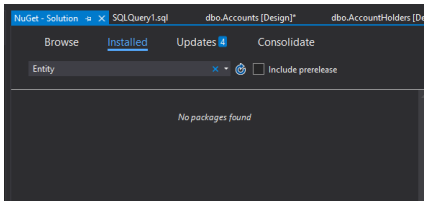


Search for Microsoft.EntityFrameworkCore.Tools, Microsoft.VisualStudio.Web.CodeGeneration.Design and Microsoft.EntityFrameworkCore.SqlServer.

Found

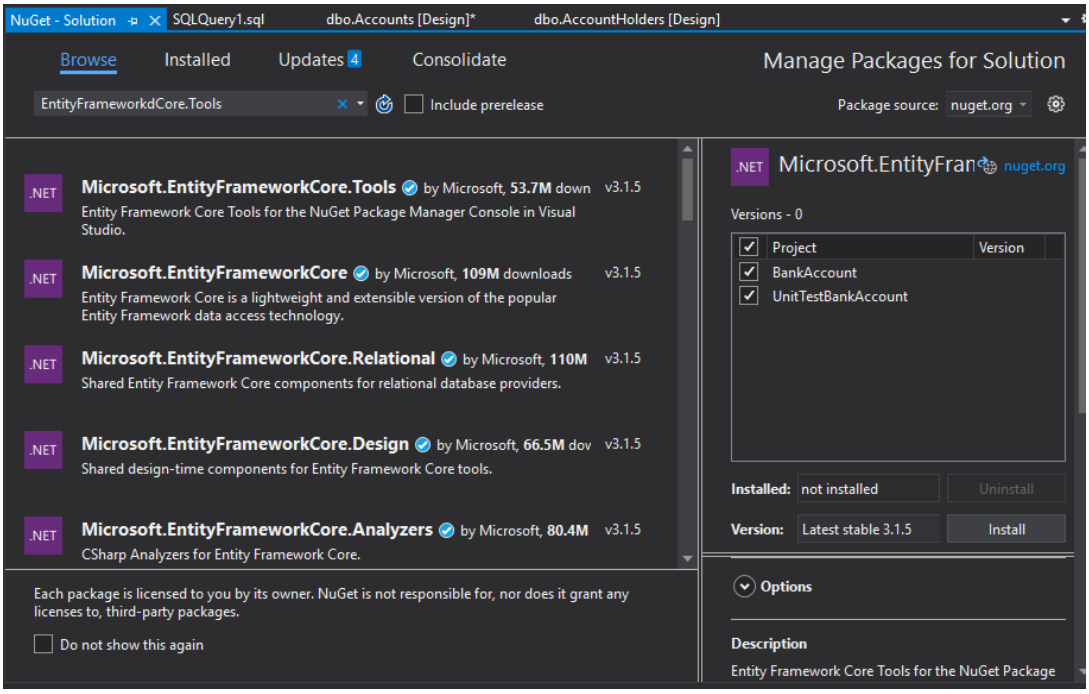


Not Found

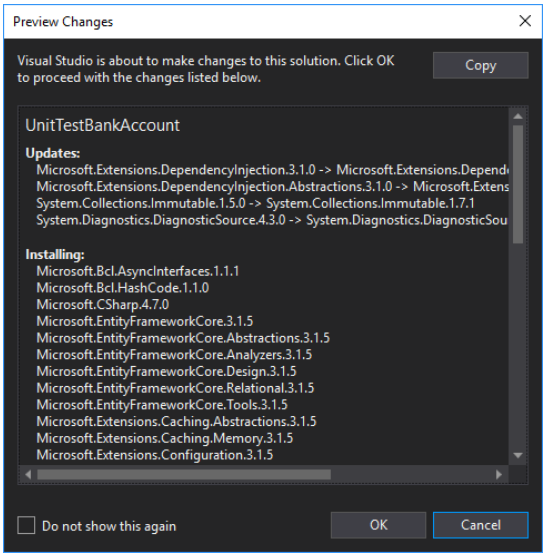


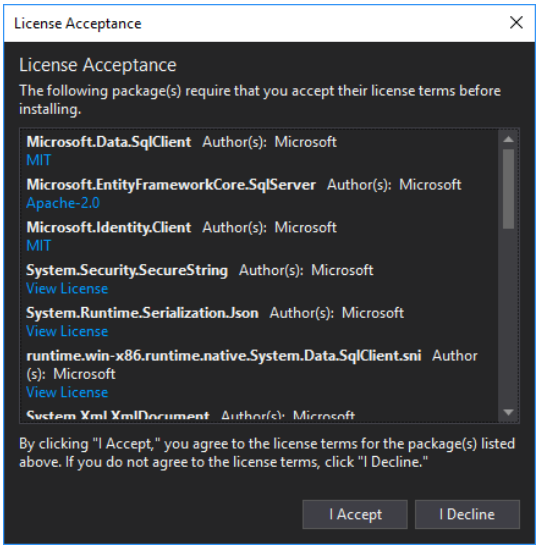
Go to Browse tab to install any missing packages.

Select the package, check the Projects, click install



You may be prompted to preview the changes, you can check 'Do not show this again' for future installs if you wish. There may be a license accept, message. These may take some time.



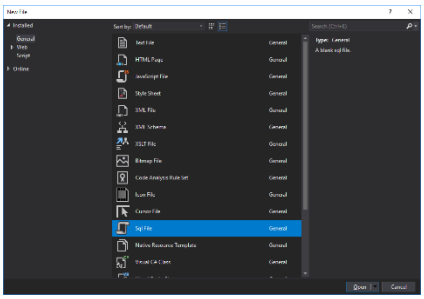


Repeat for all uninstalled packages.

Add to a Database

The instructions that follow will add the following entries to the database table
In Account Holders
(1, Martinez, Mateo)
(2, Nguyen, Tran)
(3, YourLastName, YourFirstName)

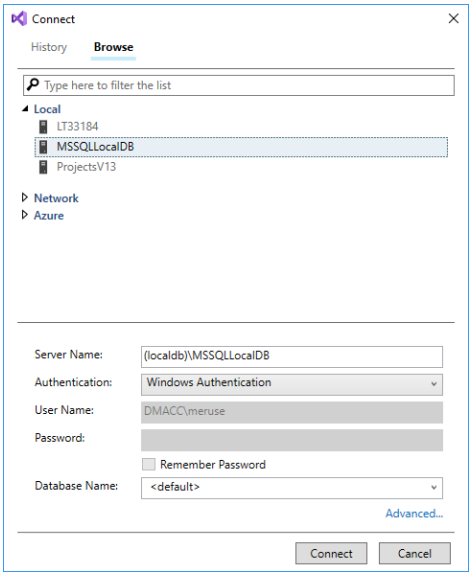
In Accounts
(1, 3, 100.23, 1234567891)
(2, 1, 101.34, 1234567892)
(3, 2, 104.35, 1234567893)



First add a file, a SQL script to run SQL commands.

In the file, add the following SQL code

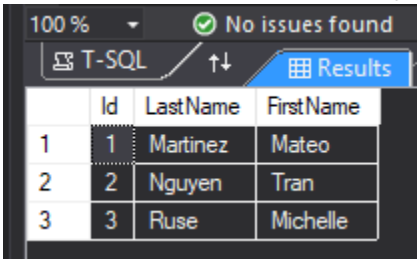
```
INSERT dbo.AccountHolders (Id, LastName, FirstName)
    VALUES (1, 'Martinez', 'Mateo');
INSERT dbo.AccountHolders (Id, LastName, FirstName)
    VALUES (2, 'Nguyen', 'Tran');
Add one for your name
```



Run the script with the green Run icon in the corner of the file. If a Connect window that pops up, make sure your database is highlighted and set Database Name: from <default> BankingDatabase.

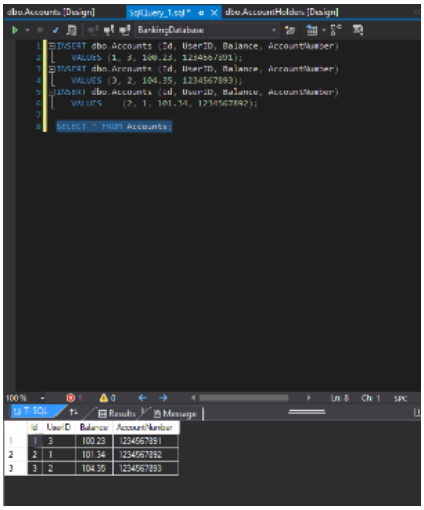
Check with a new SQL script & screen shot **AccountHolders.jpg**

```
SELECT * FROM AccountHolders;
```



Next add to the Accounts Database

```
INSERT dbo.Accounts (Id, UserID, Balance, AccountNumber)
   VALUES (1, 3, 100.23, 1234567891);
INSERT dbo.Accounts (Id, UserID, Balance, AccountNumber)
   VALUES (3, 2, 104.35, 1234567893);
INSERT dbo.Accounts (Id, UserID, Balance, AccountNumber)
   VALUES (2, 1, 101.34, 1234567892);
```



- How would check on the Accounts table? Write the necessary statement. Take a screen shot of the query results name it **Accounts.jpg**.
- Write the SQL statement to find the accounts of Mateo Martinez (id is 1). Add to the Select statement after the table name `WHERE UserID = 1`. Take a screen shot name it **Query.jpg**.

In the Sql file, you can highlight one line run with green run icon.

Submit **Database.jpg**, **Accounts.jpg**, **AccountHolders.jpg**, and **Query.jpg**