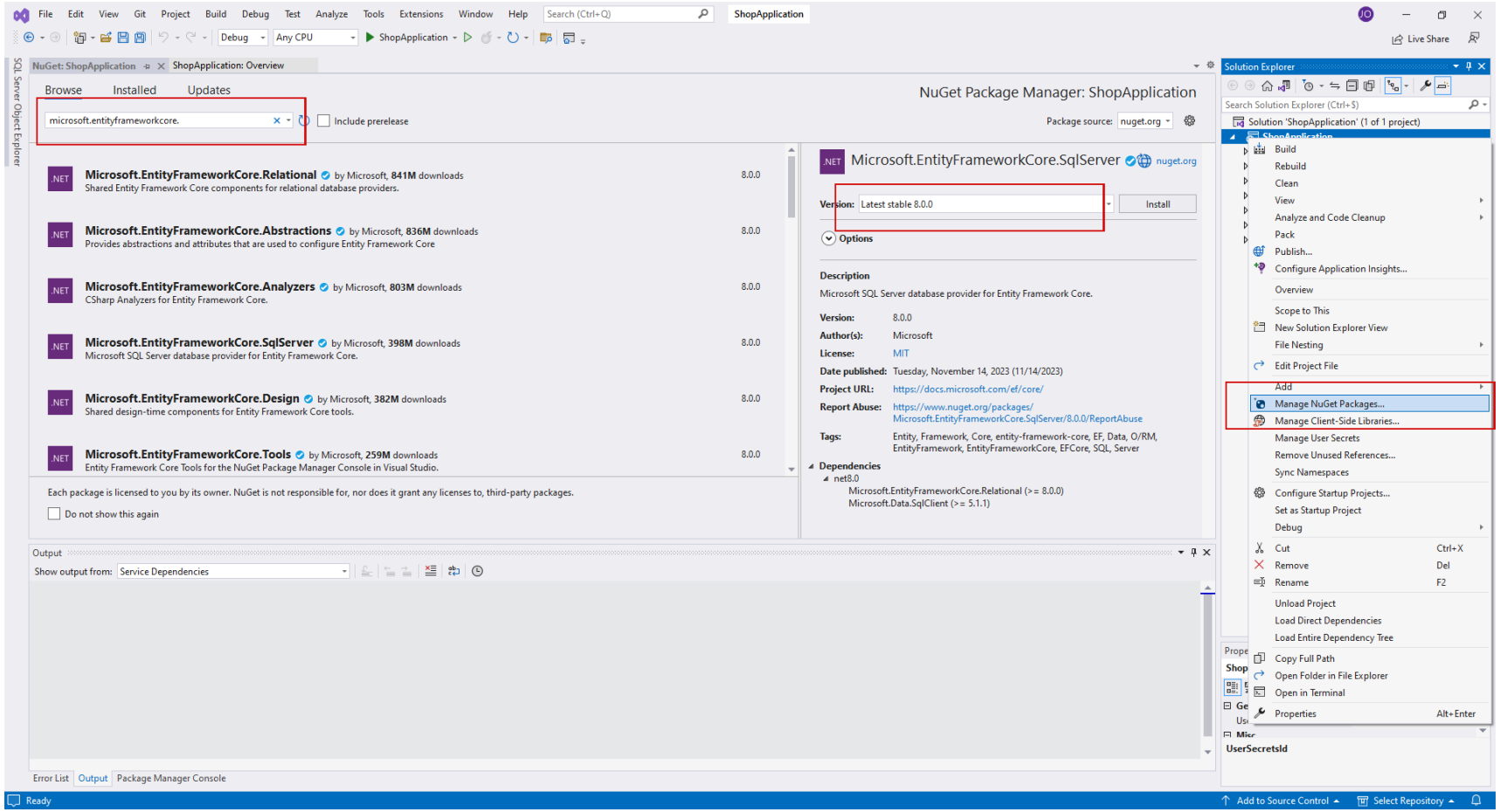
ADD A DATABASE to your .NET application

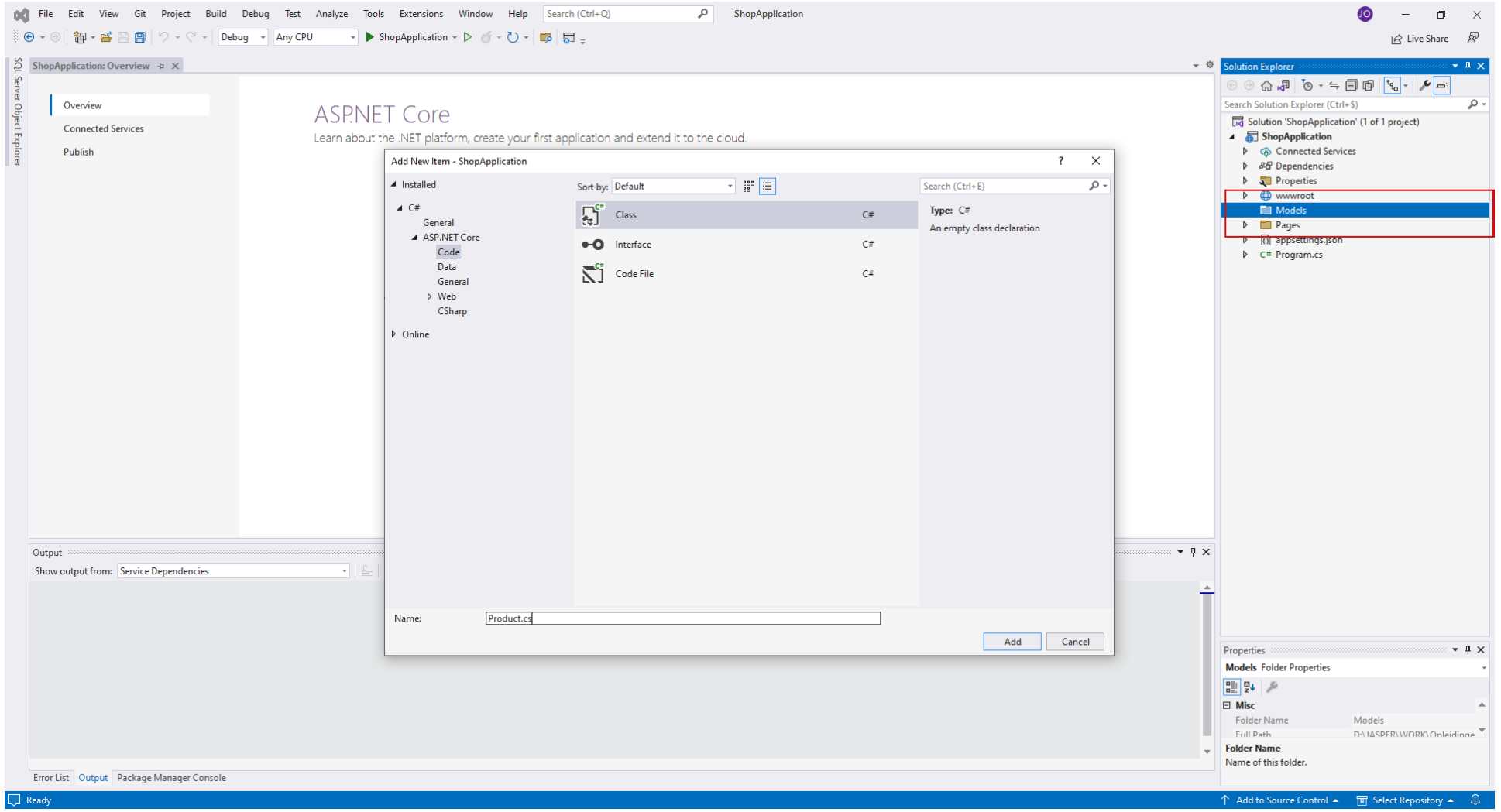
Adding Seed Data to your application

Nugget packages

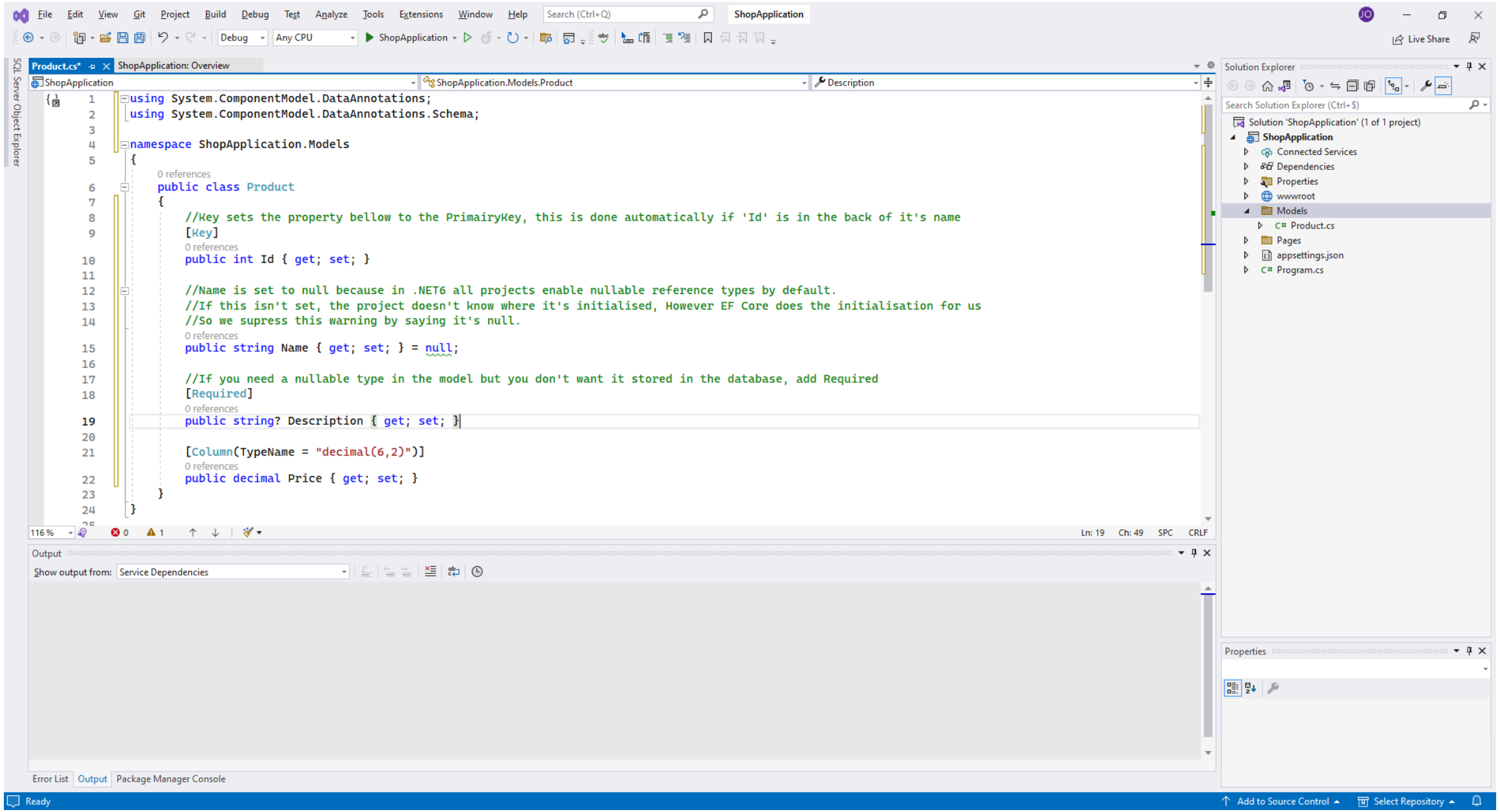
- Add Microsoft.EntityFrameworkCore.SqlServer
- Add Microsoft.EntityFrameworkCore.Design
- Add Microsoft.EntityFrameworkCore.Tools



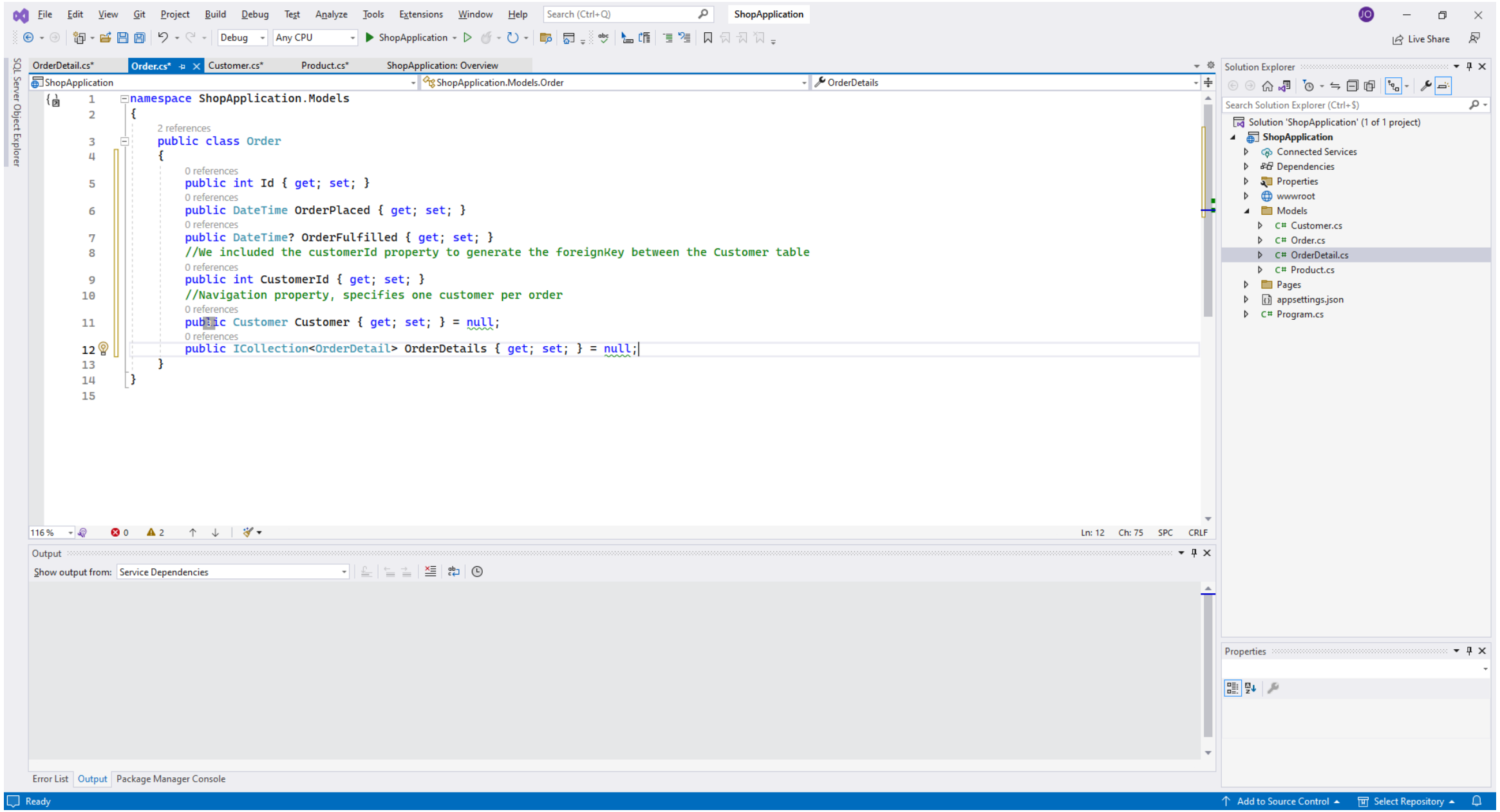
- Objects that are used within the project are refured to as Models. These models represent how your database looks like.
- Models are placed in Solution -> Project -> Models (add this folder)



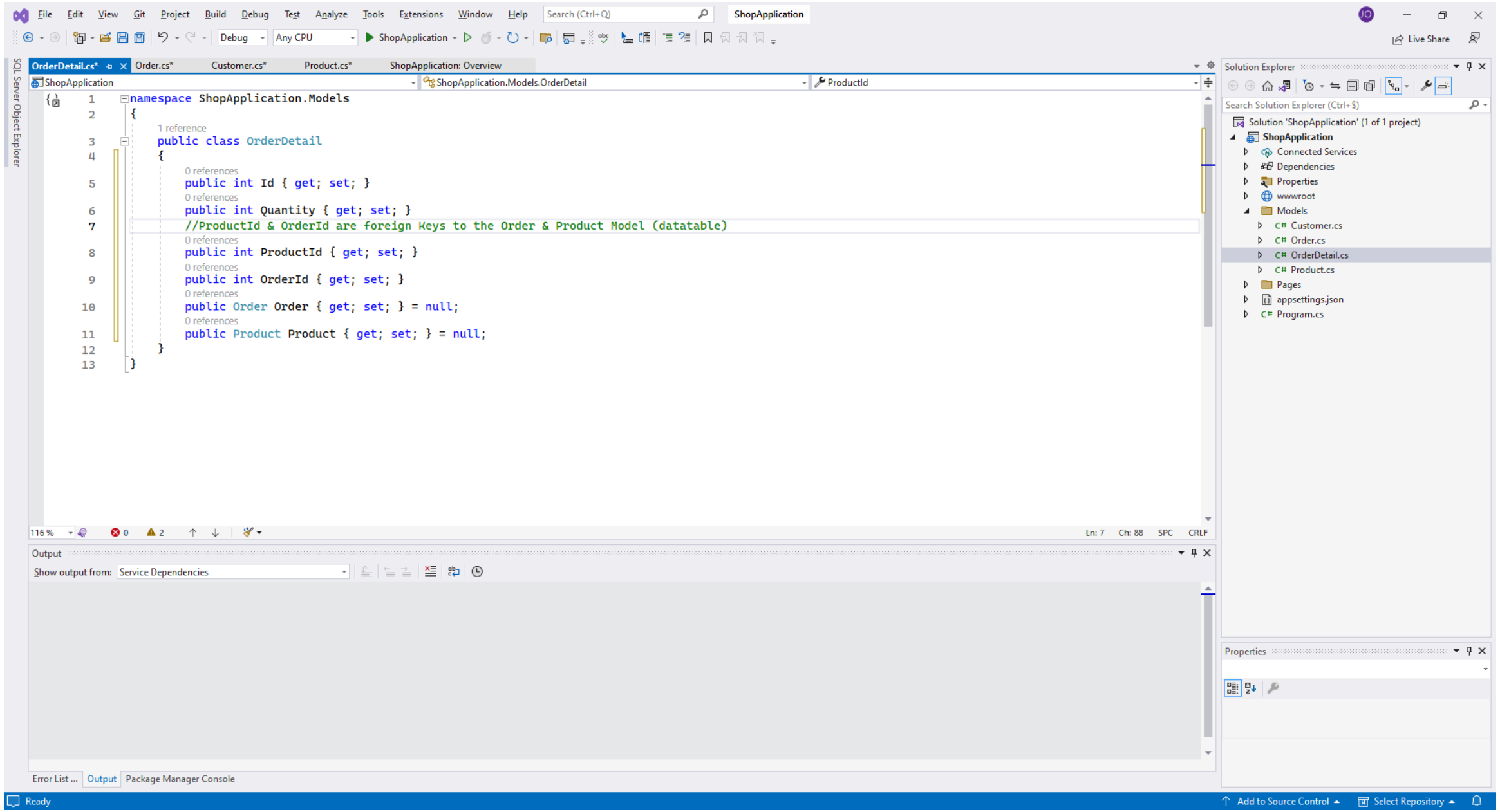
- Models are standard a normal Class



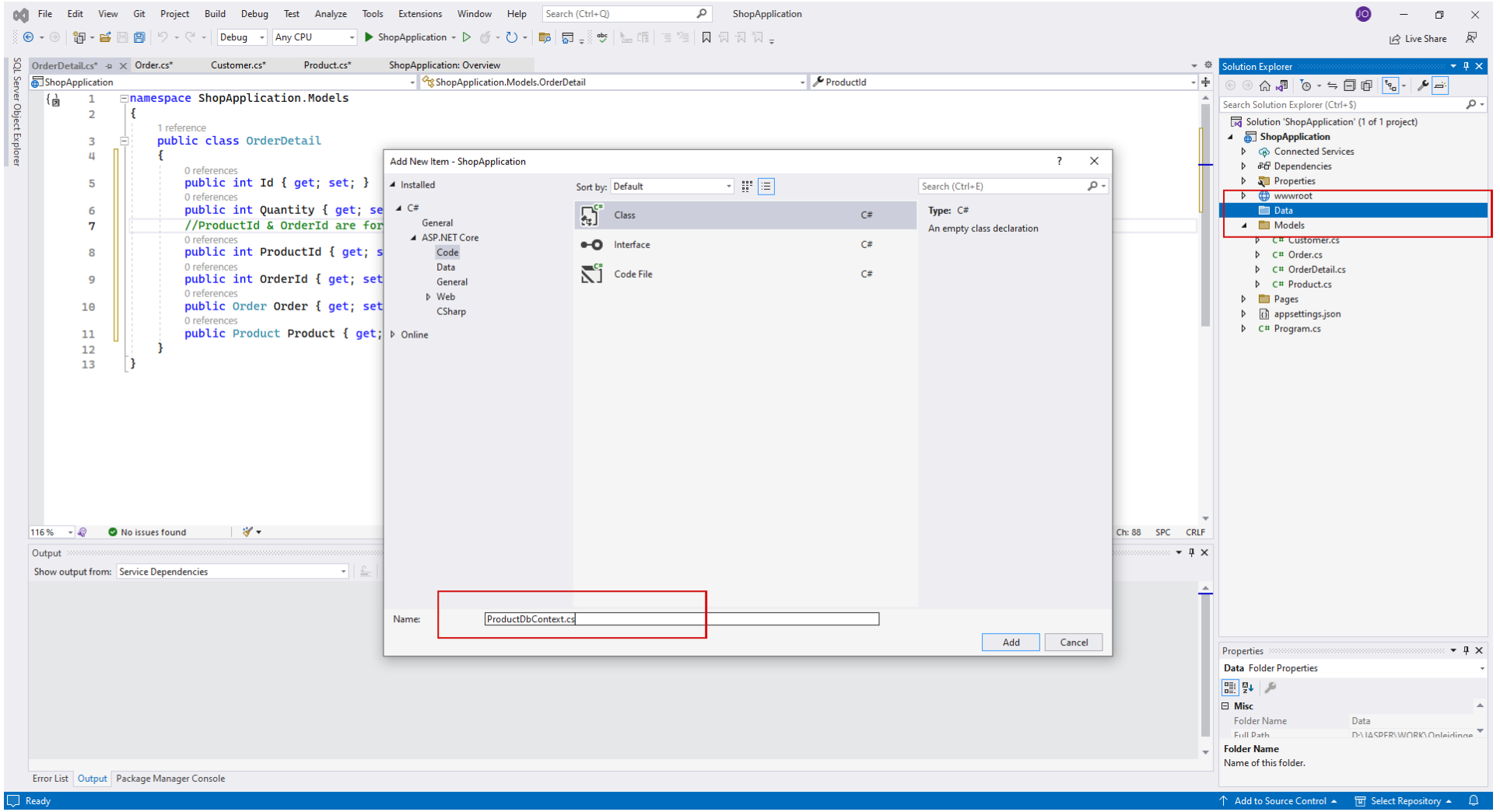
- Add additional models/tables and add the foreignKey's



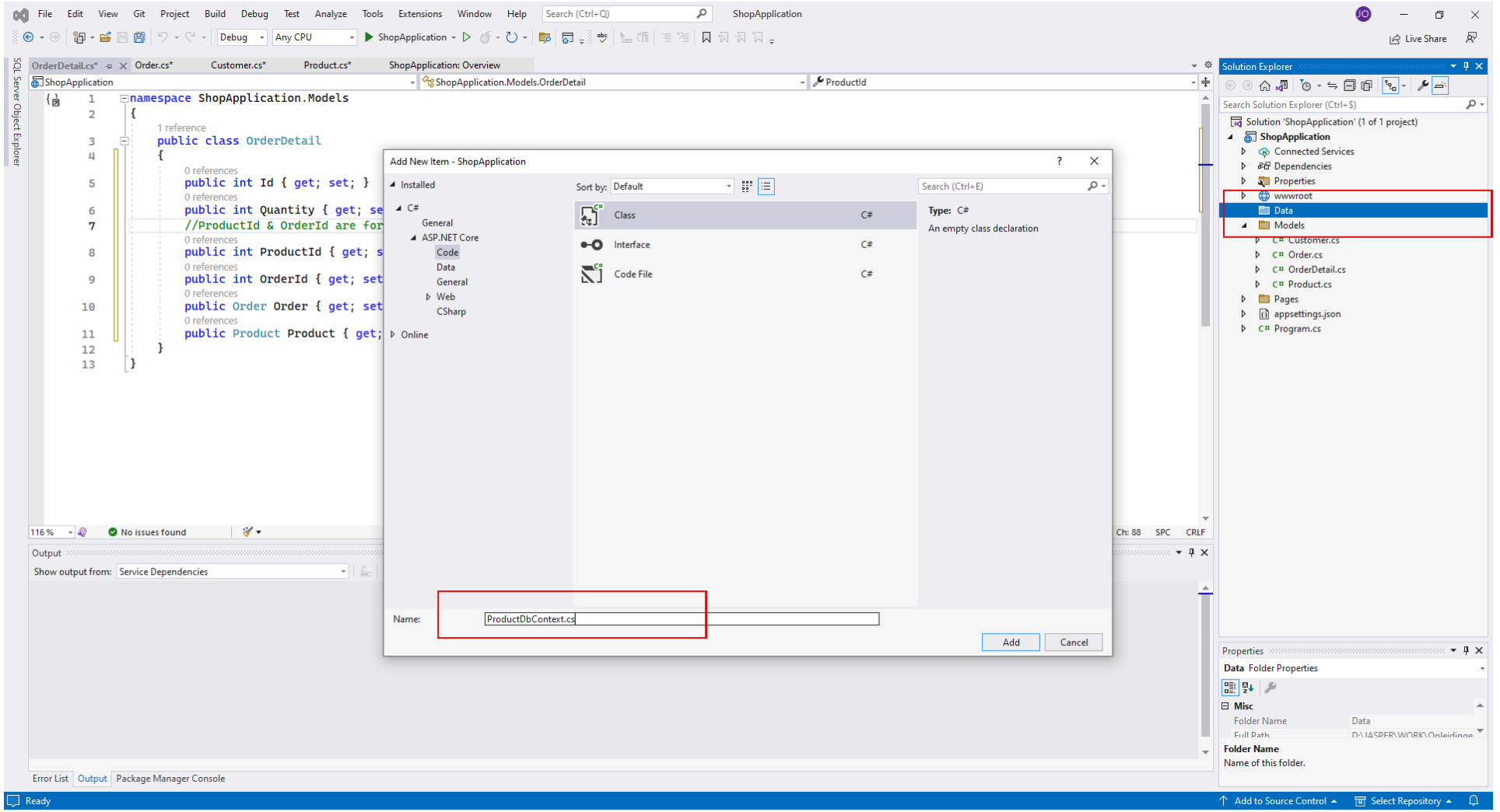
- Add additional models/tables and add the foreignKey's



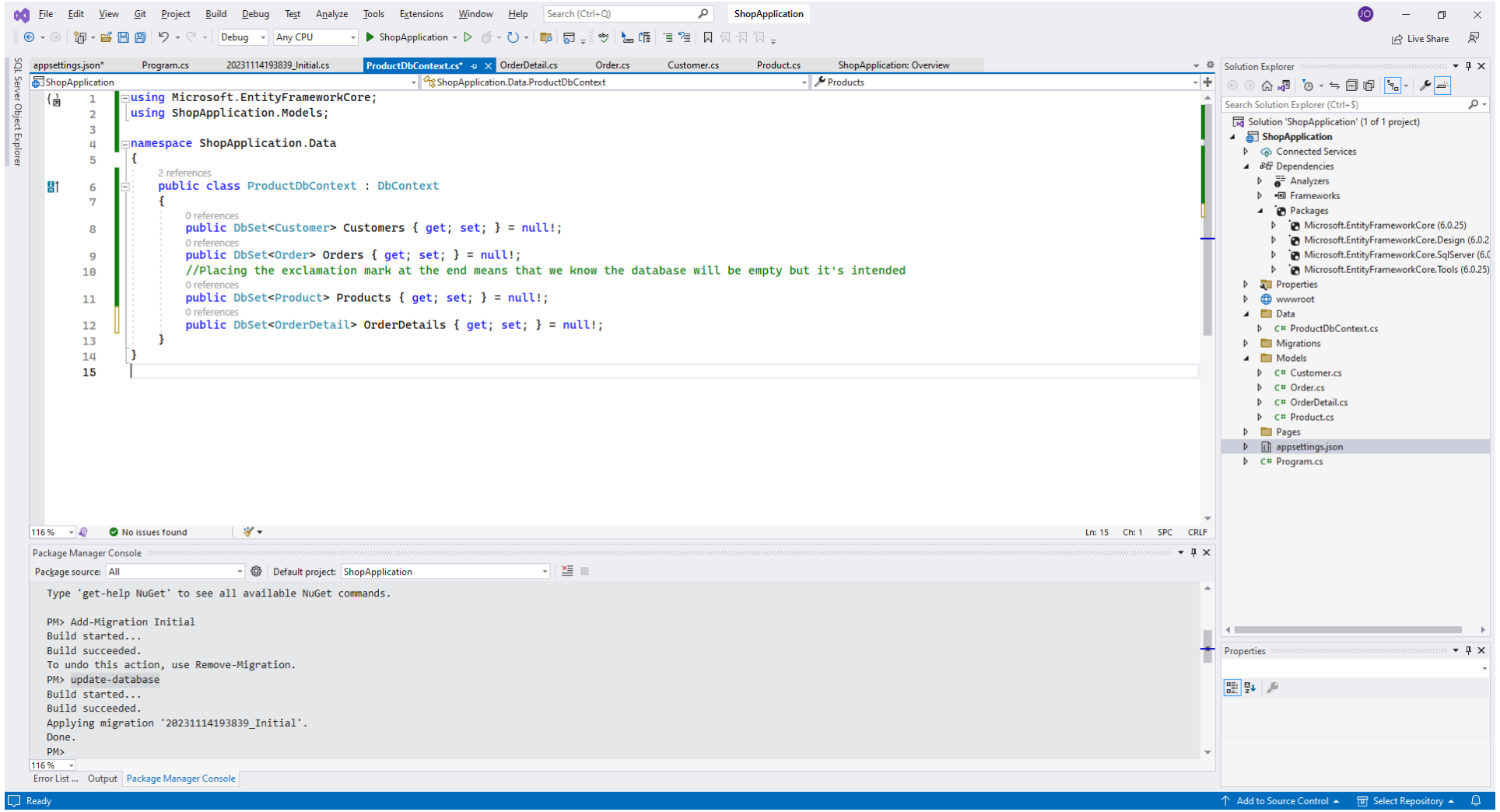
- Add the Data folder in Solution -> Project -> Data (create this folder)
- Add the class NAMEDbContext.cs (ProductDbContext.cs for example)



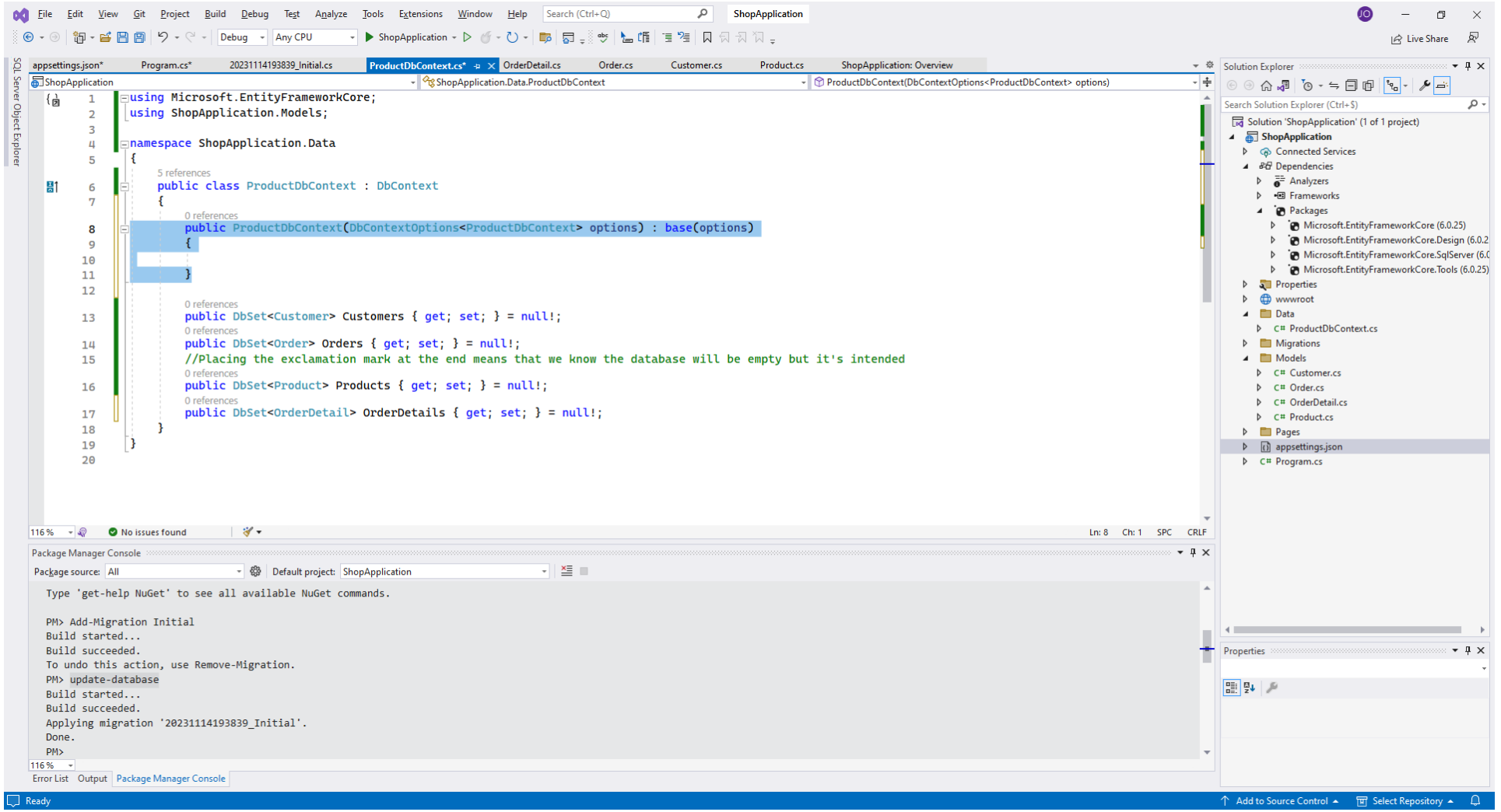
- Add the Data folder in Solution -> Project -> Data (create this folder)
- Add the class NAMEDbContext.cs (ProductDbContext.cs for example)



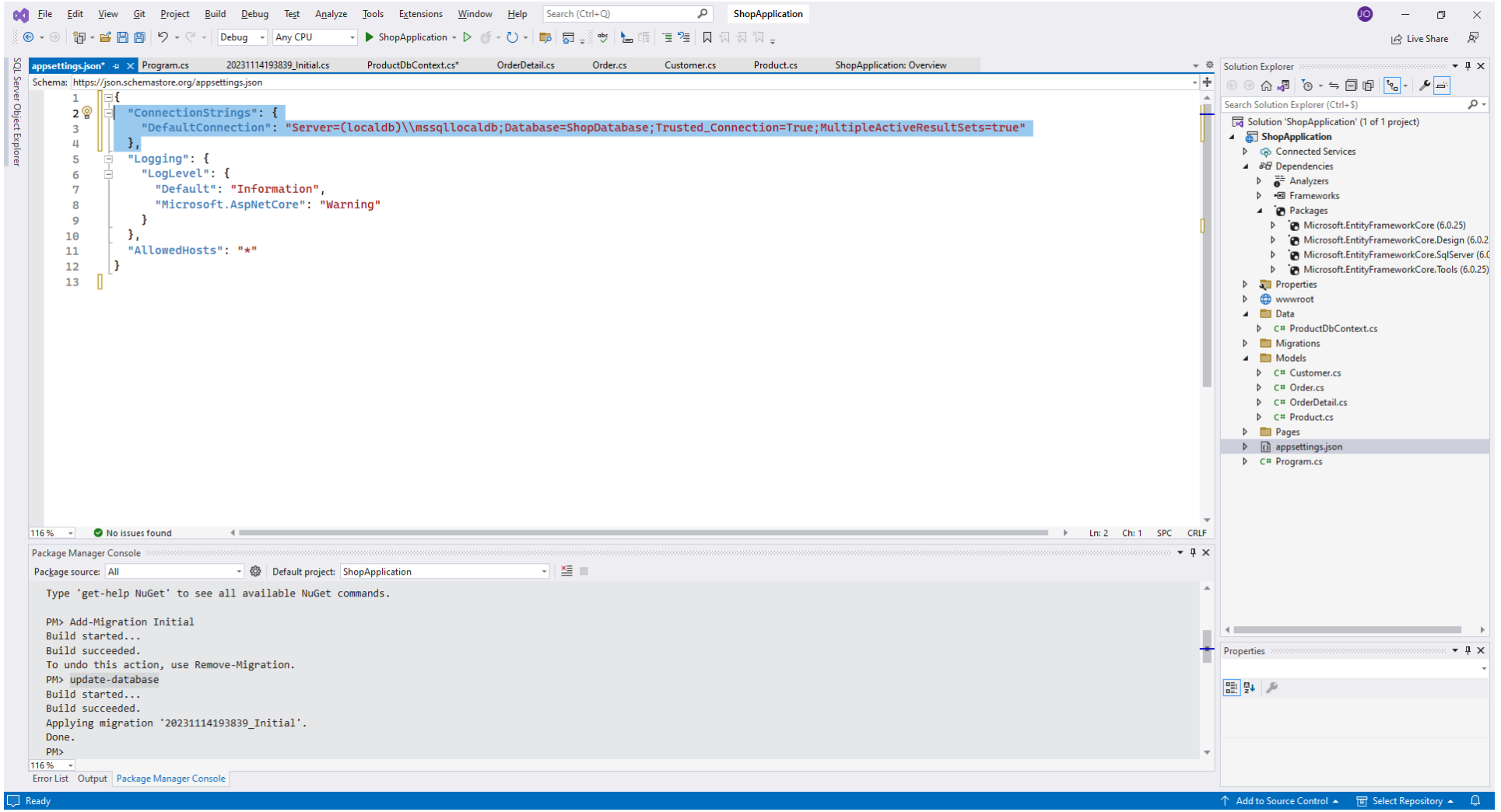
- ProductDbContext inherits from DbContext
- Each DbSet matches a table that will be created in the database



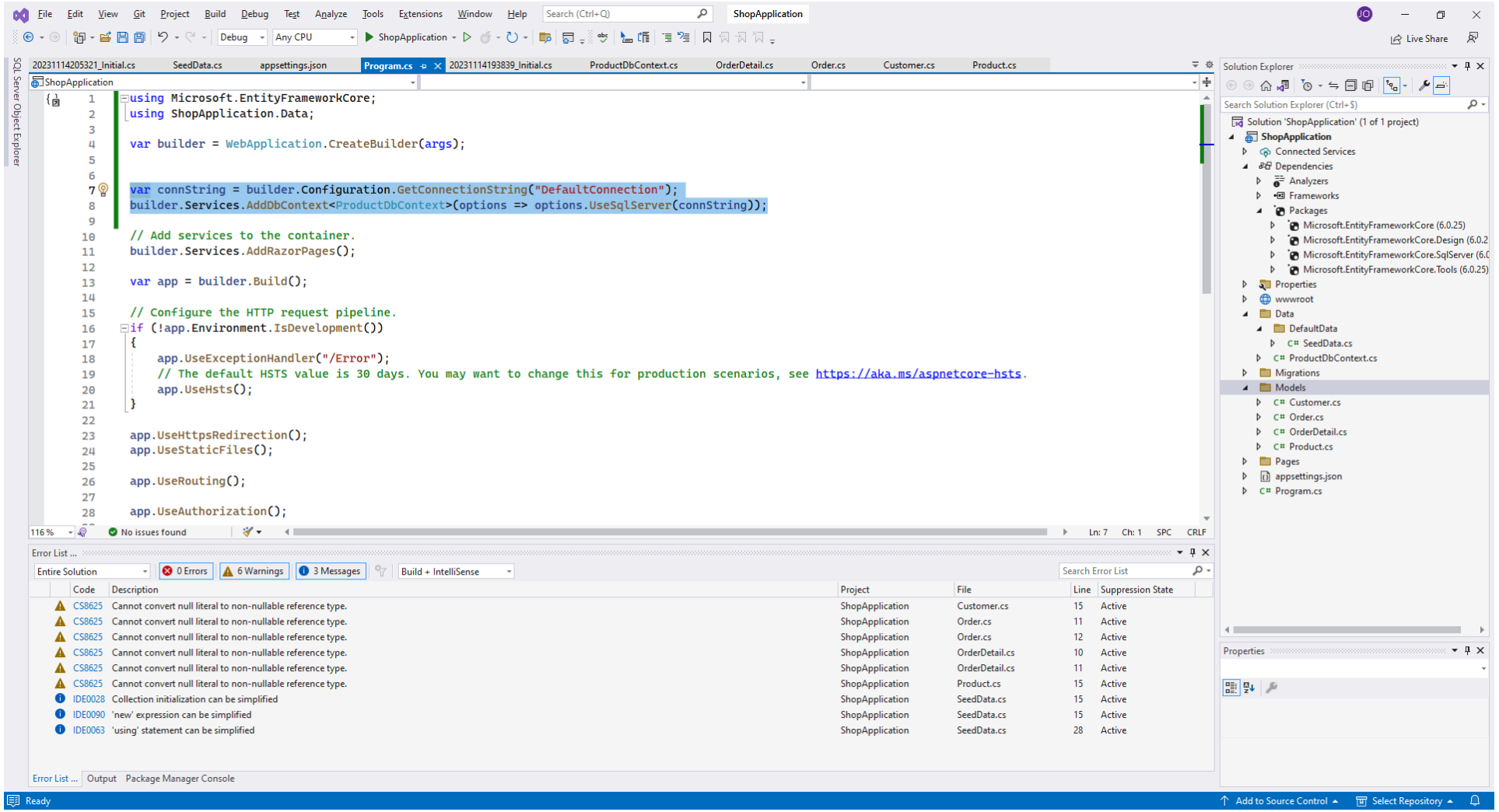
- We add the constructor that takes ‘DbContextOptions<ProductDbContext>’ as a parameter this is how we can modify the details of the DB in program.cs



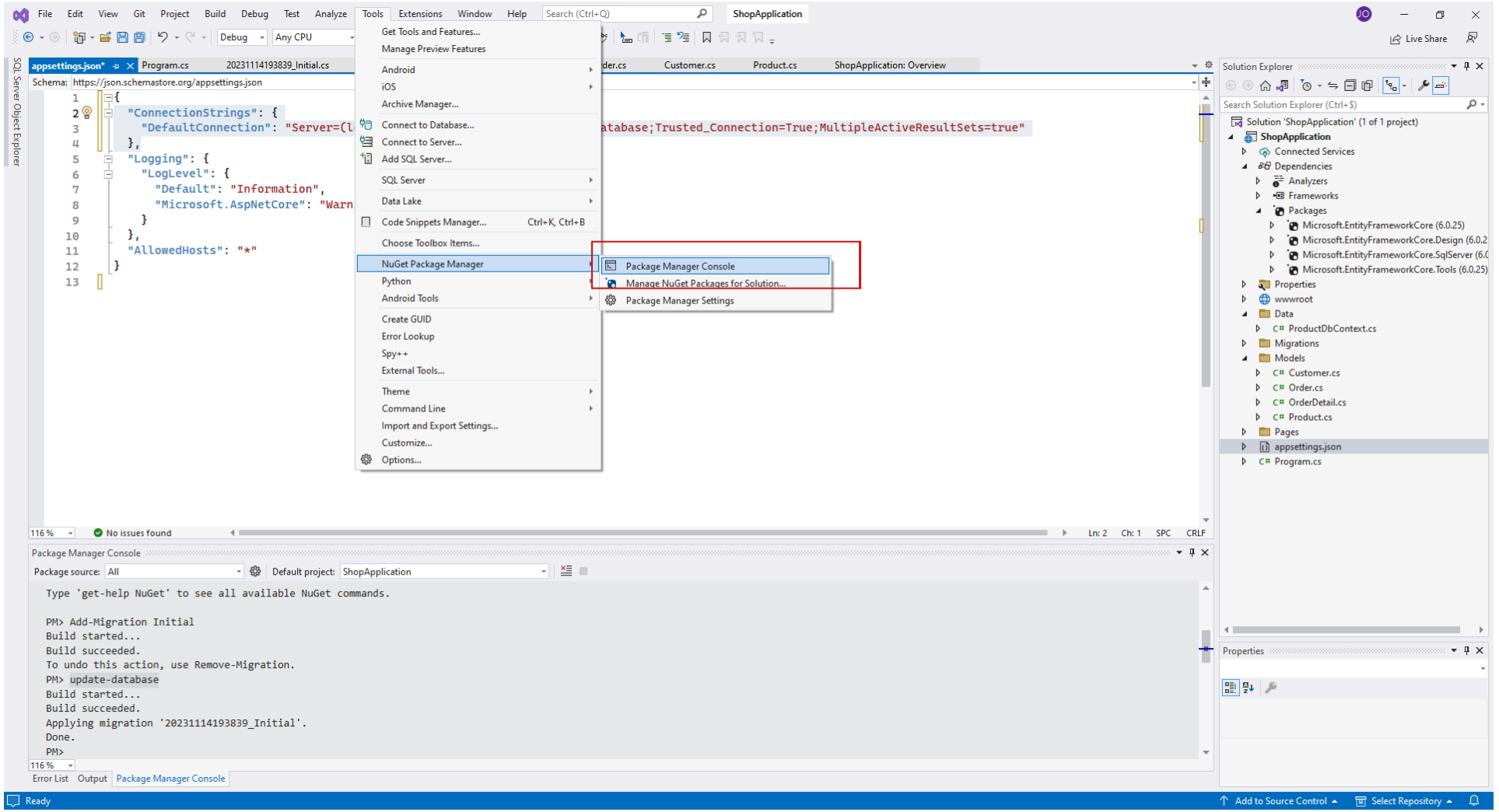
- We open appsettings.json and add a connectionstring
- The connectionstring here is named DefaultConnection



- Below ‘var builder’ we add this line, it says to ge the connection from ProductDbContext, we add this as a service. Make sure that the string in GetConnectionString is equal to the name you gave it.

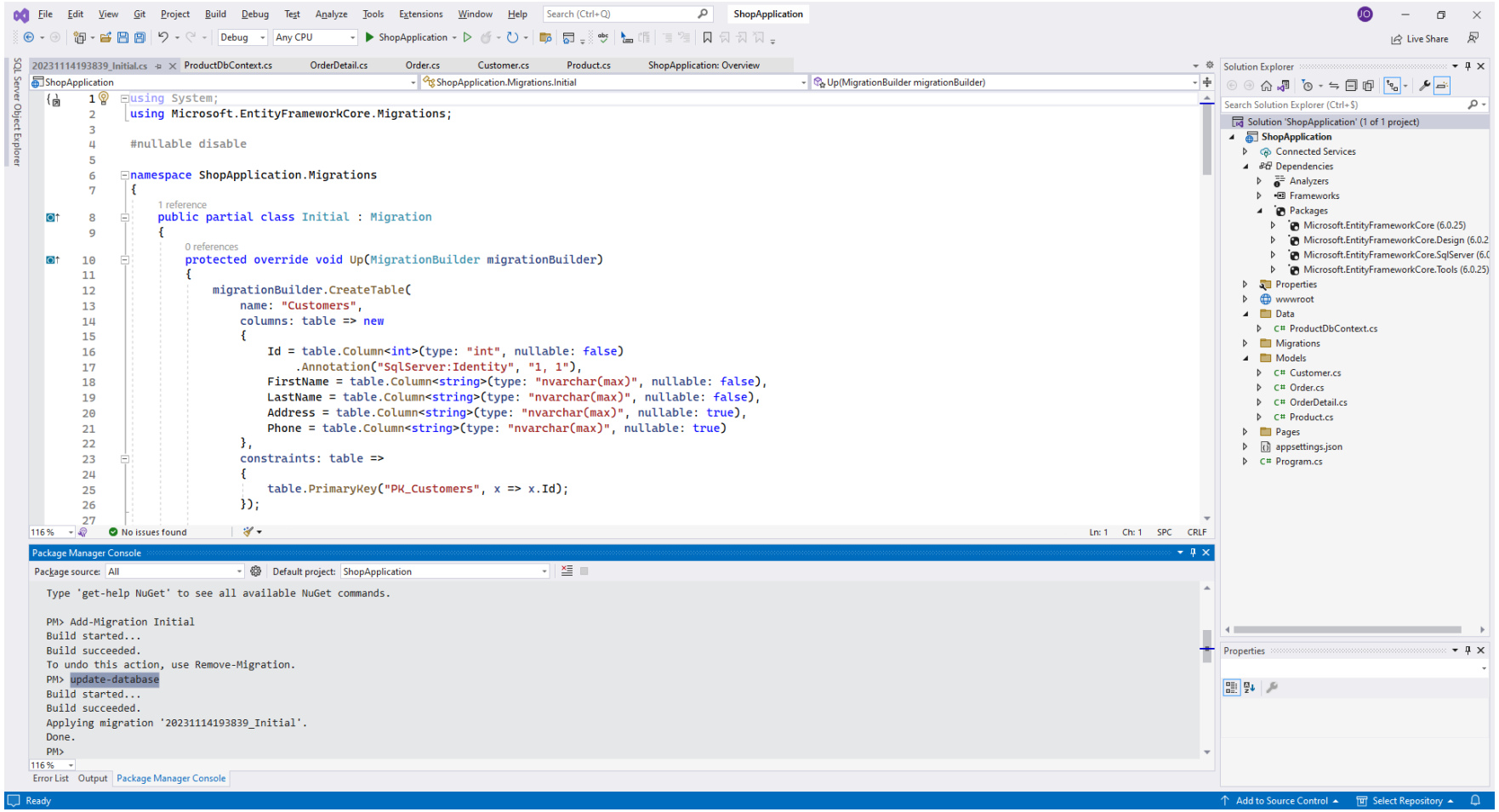


- Now we migrate our database by opening Package Manger Console
- Tools -> NuGet Package Manager -> Package Manager Console
- Typ -> “ Add-Migration Initial “

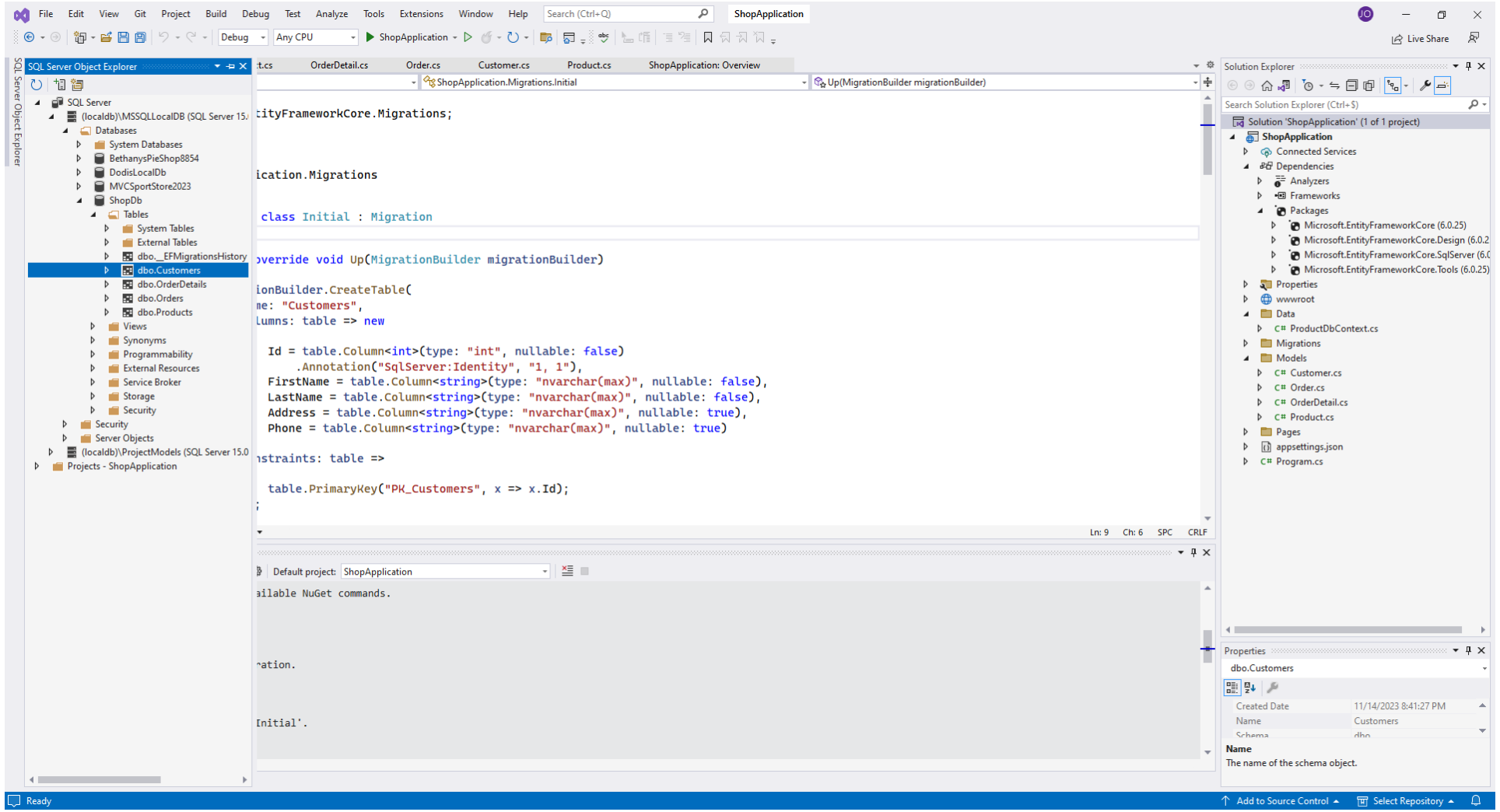


- If by any chance the migration has errors look if you already have a Migrations folder in your solution and delete it, then try again
- If this doesn’t work try using Remove-Migration
- If this still doesn’t work rename your database you created in slide 9

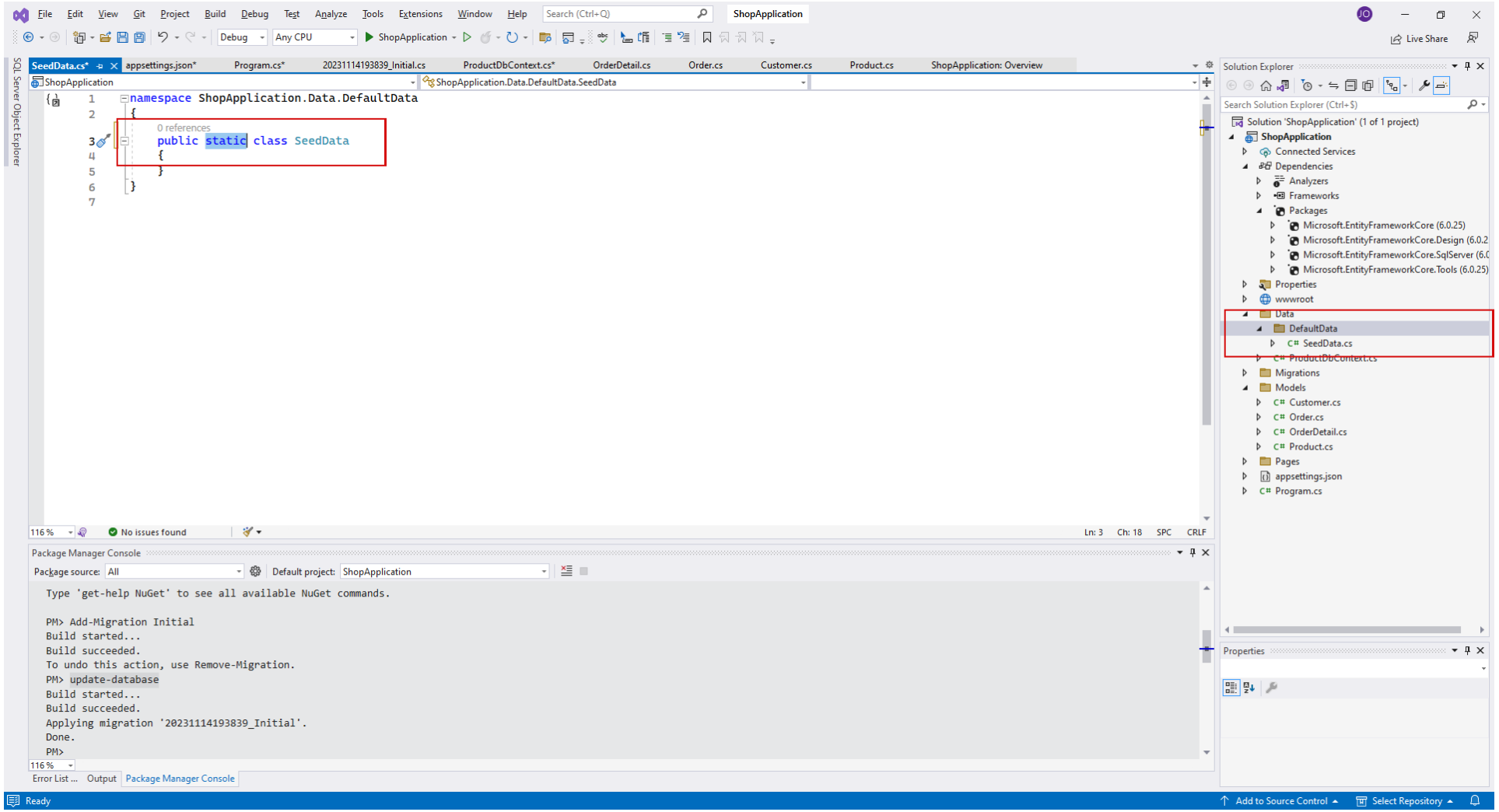
- Next we type in the command “ Update-Database “
- If this doesn’t work try using “ Drop-Database “
- Note: if you modify a model type “Add-Migration (type what changed)” and also “ Update-Database “



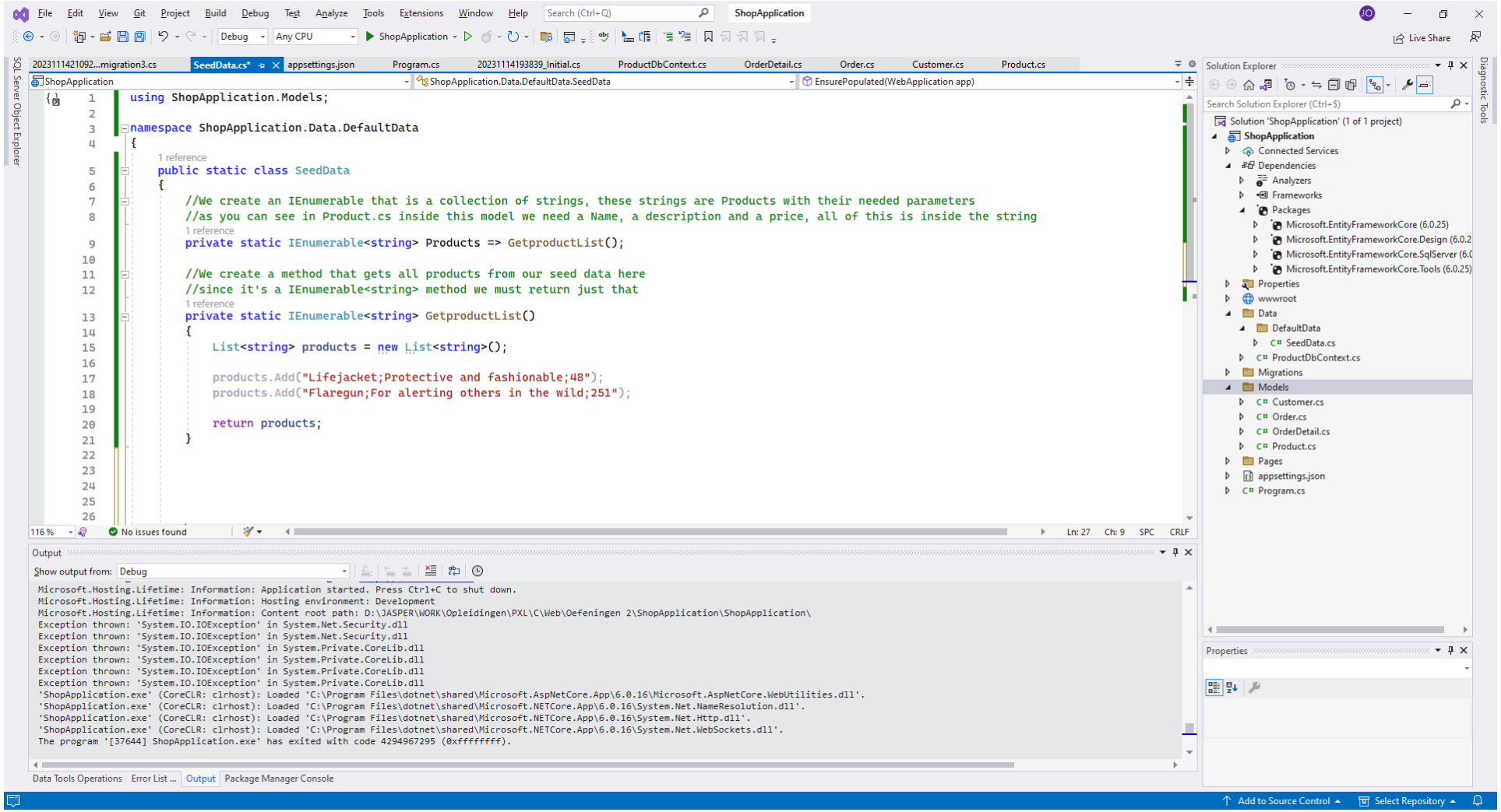
- To view what's in the database go to View -> SQL Server Object Explorer.



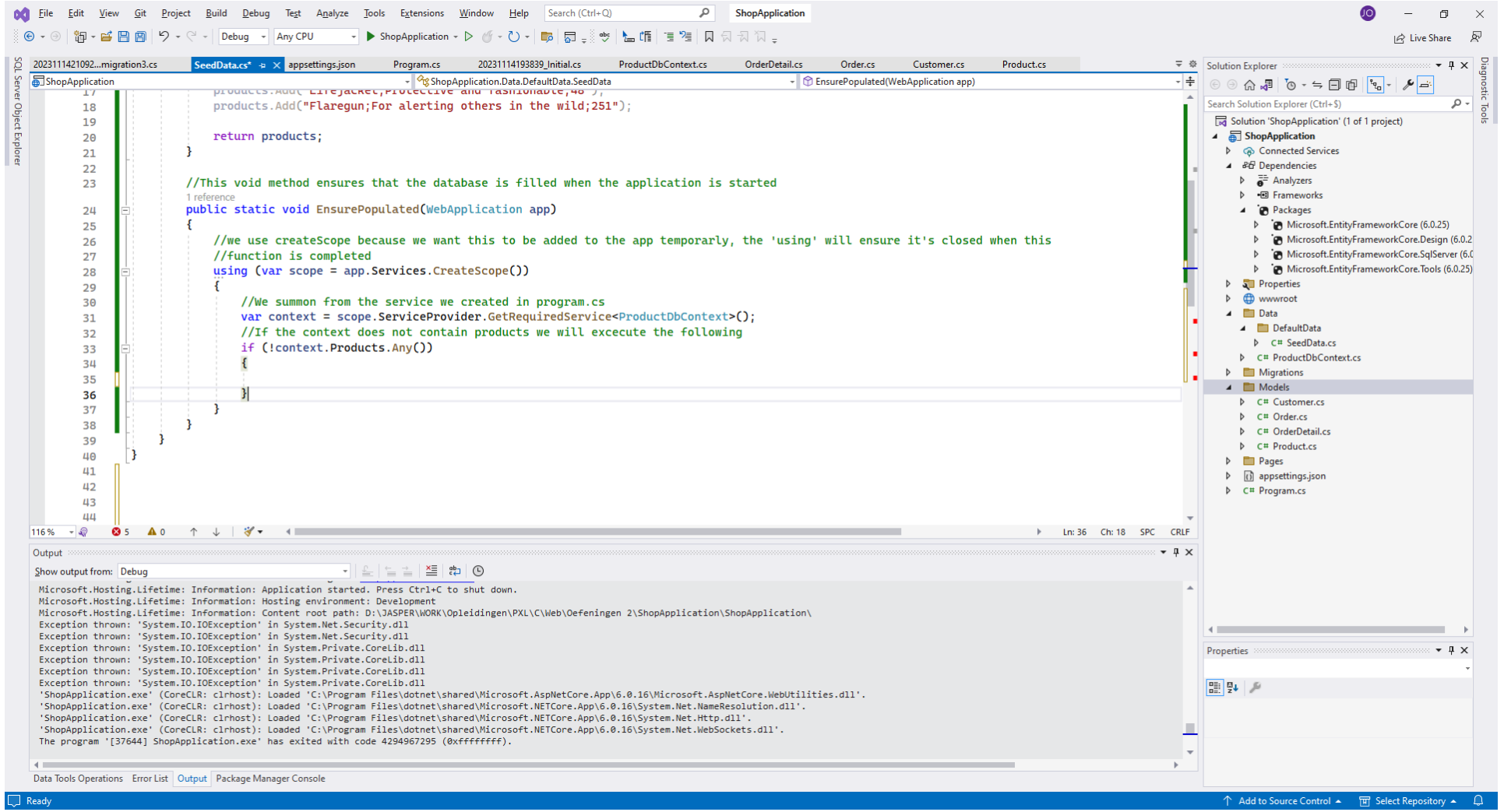
- In our Data folder create a new folder named “DefaultData” and create a class “SeedData” this is a public STATIC class.
- It is static so we can get properties from here without a constructor



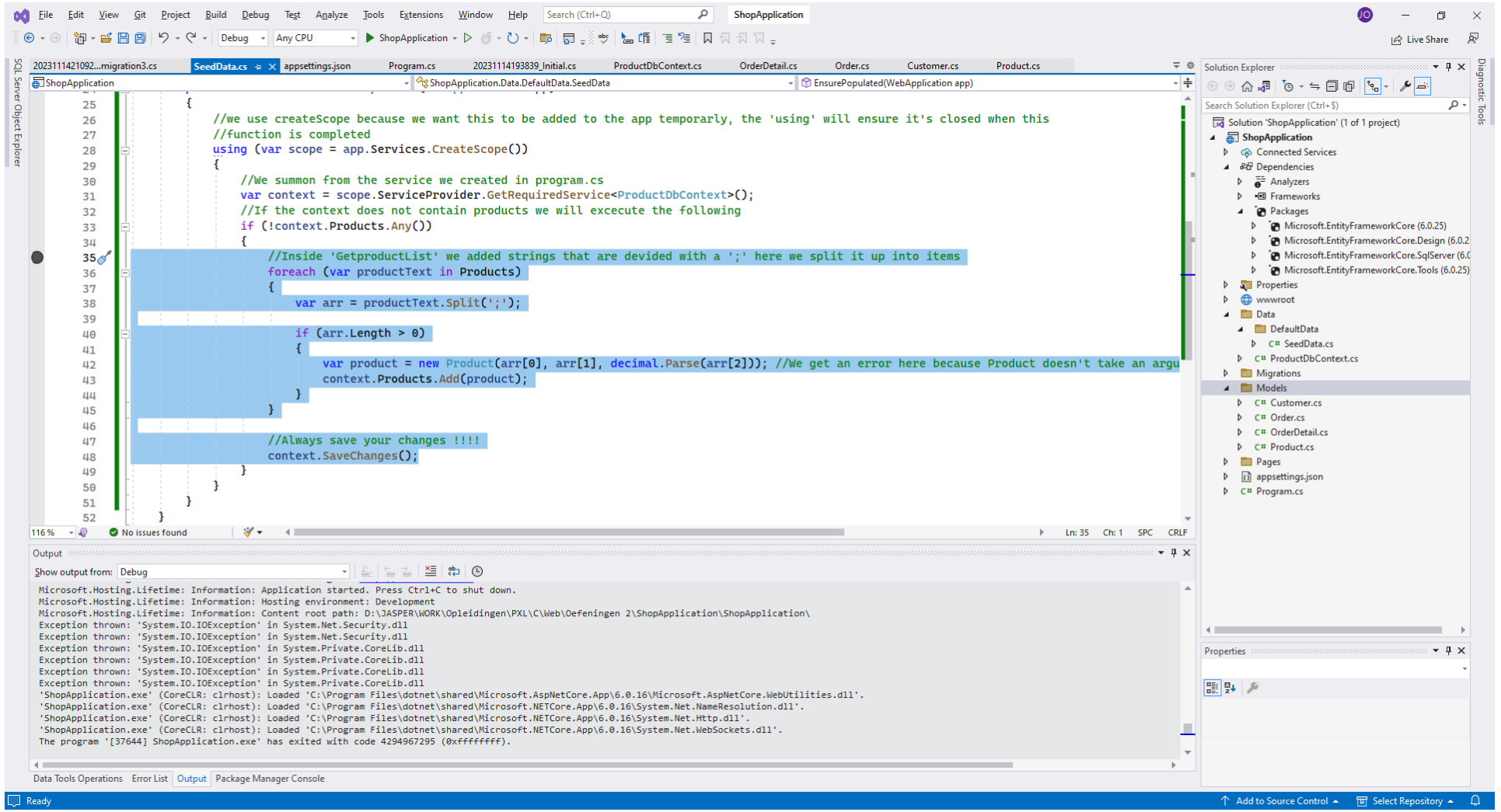
- Inside our SeedData we create an Ienumerable field and a Ienumerable method that uses the Ienumerable field



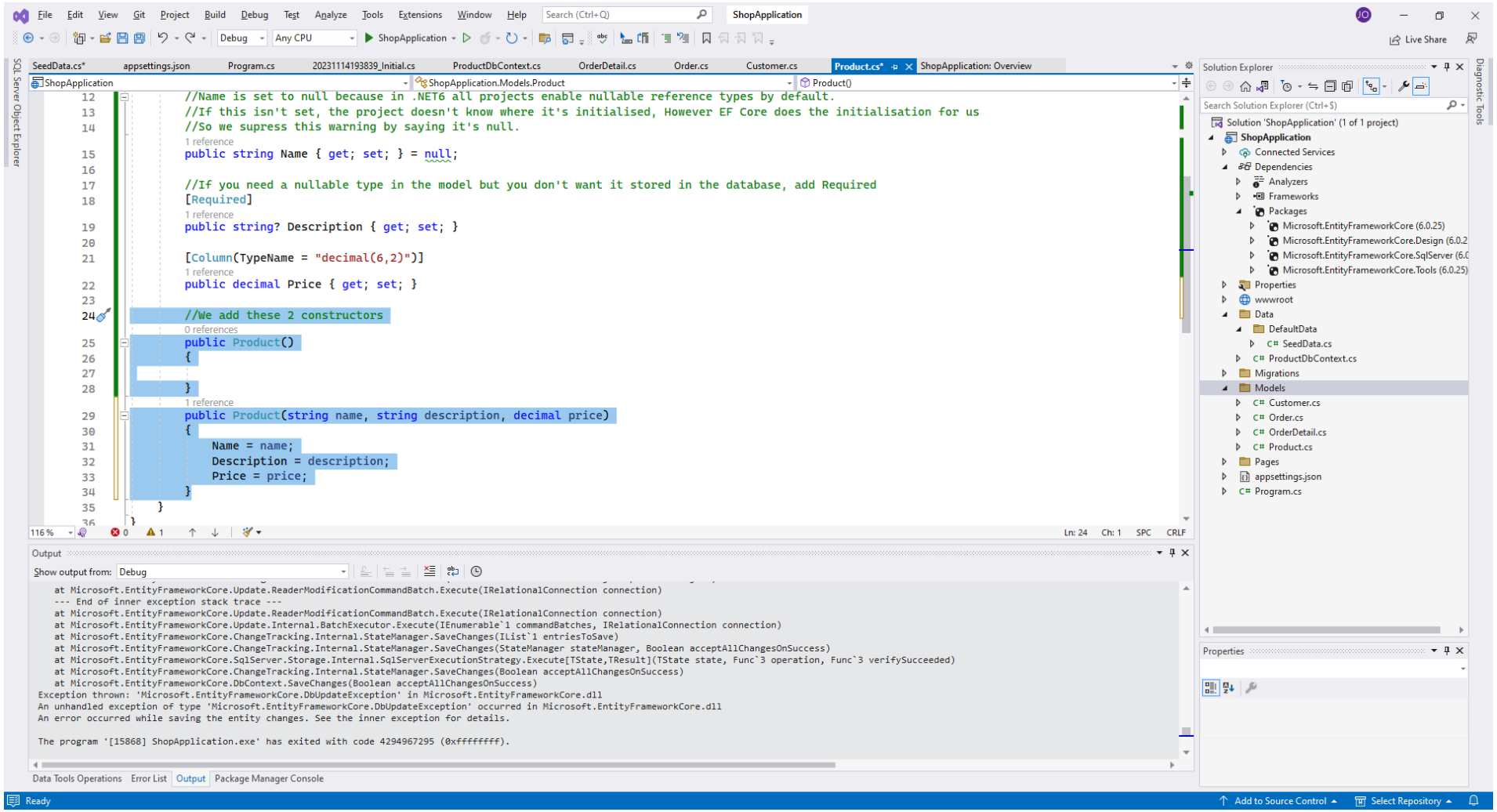
- We are still in SeedData



- We are still in SeedData



- We go back to our Models -> Products and add these 2 constructors
- The constructor that takes the arguments is the one we asked for inside our SeedData



- In Program.cs We make sure the method EnsurePopulate is triggered right before the app launches