

Bazy danych

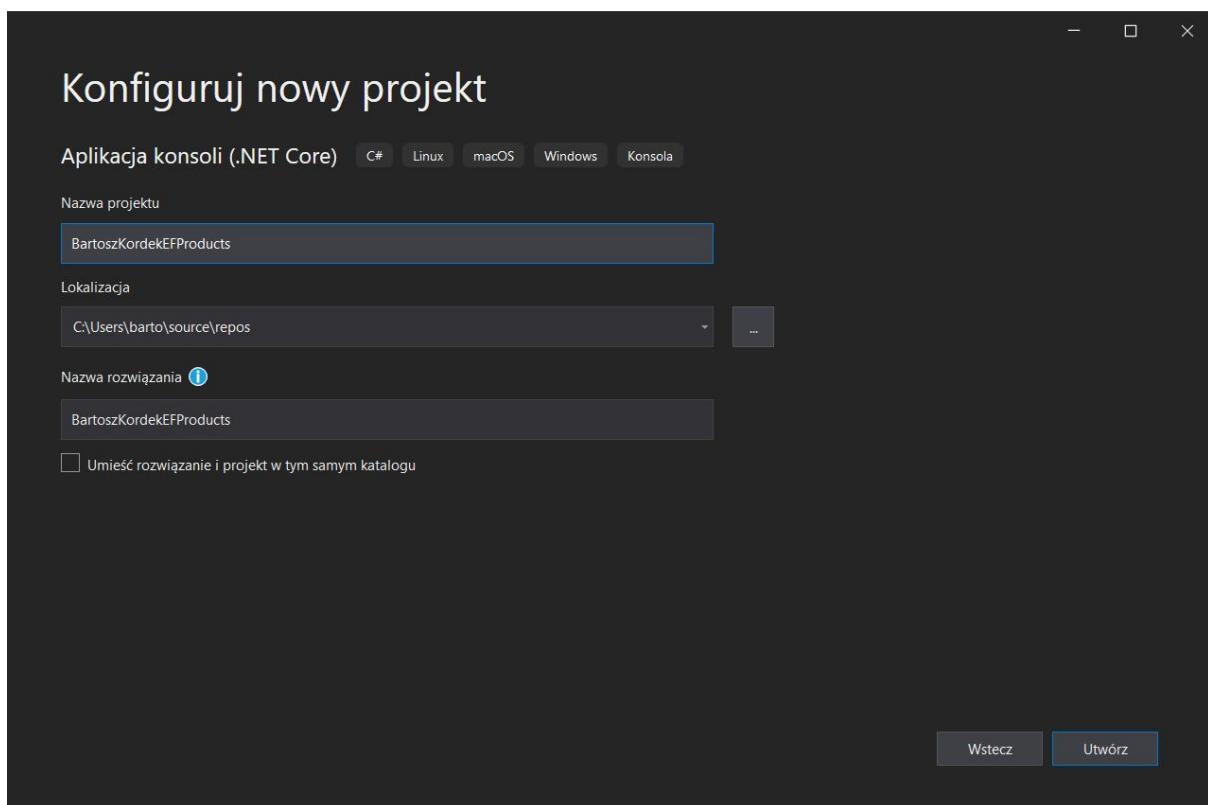
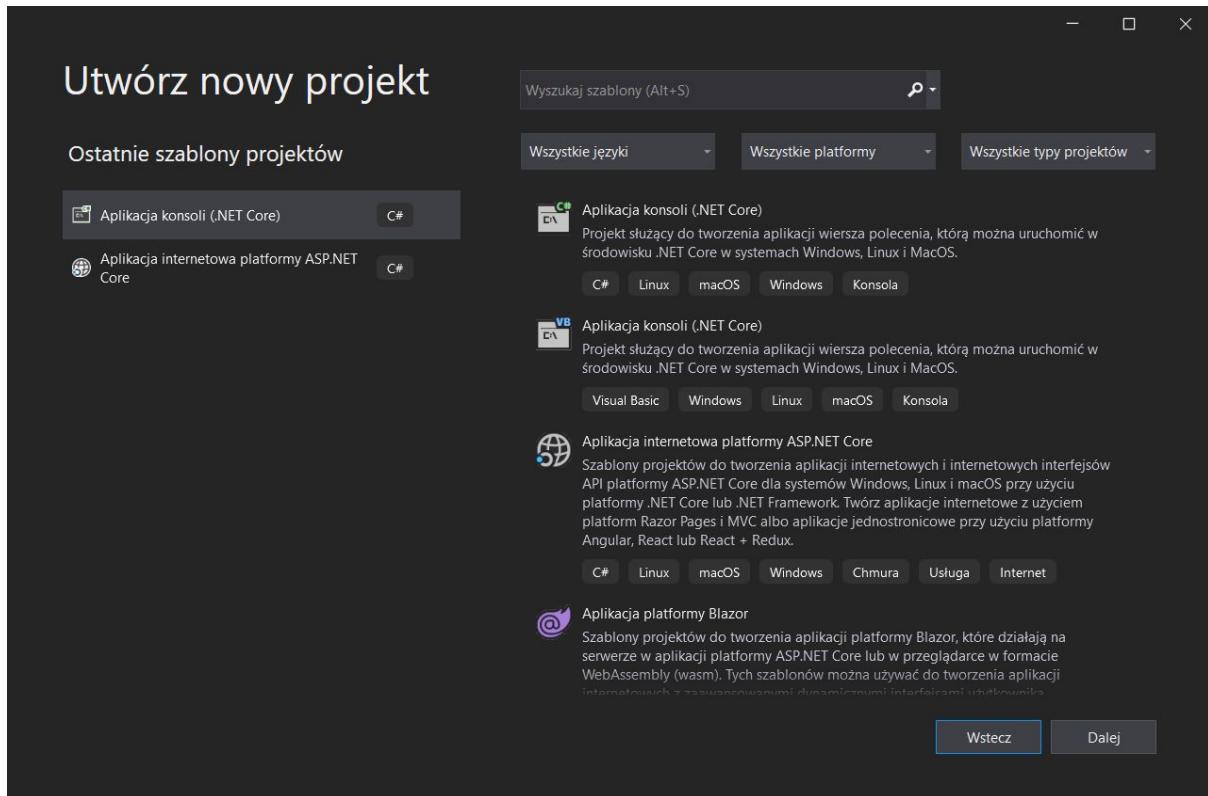
Wprowadzenie do EntityFramework.Core

Bartosz Kordek

Informatyka, semestr VII, studia niestacjorarne

I. HelloWorld

Utworzenie nowego projektu



Uruchomiony HelloWorld

The screenshot shows the Microsoft Visual Studio interface with the title bar "BartoszKordekEFProducts". The main code editor window displays a C# file named "Program.cs" with the following code:

```
1  using System;
2
3  namespace BartoszKordekEFProducts
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10         }
11     }
12 }
```

The output window titled "Konsola debugowania programu Microsoft Visual Studio" (Debug Console) shows the output "Hello World!". Below the code editor, the status bar indicates "100 %" and "Nie znaleziono żadnych problemów" (No problems found).

Klasa Product

The screenshot shows the Microsoft Visual Studio interface with the title bar "BartoszKordekEFProducts". The main code editor window displays a C# file named "Product.cs" with the following code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordekEFProducts
6  {
7      public class Product
8      {
9          public int ProductID { get; set; }
10         public string ProductName { get; set; }
11         public int UnitsOnStock { get; set; }
12     }
13 }
```

Klasa ProductContext

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordekEFProducts
6  {
7      public class ProductContext : DbContext
8      {
9      }
10 }
11
```

Dodanie pakietu *Microsoft.EntityFrameworkCore* do projektu

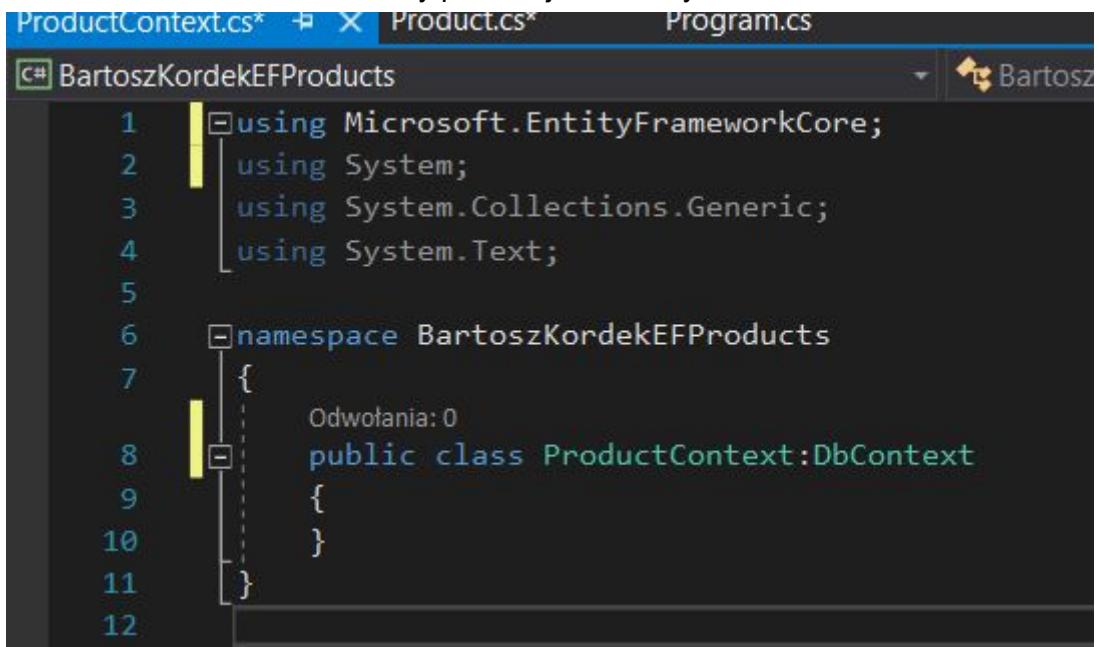
```
Wiersz polecenia
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\barto>cd C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet add package Microsoft.EntityFrameworkCore
Trwa określanie projektów do przywrócenia...
Writing C:\Users\barto\AppData\Local\Temp\tmp75F3.tmp
info : Dodawanie elementu PackageReference dla pakietu "Microsoft.EntityFrameworkCore" do projektu "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/index.json 561 ms
info : Trwa przywracanie pakietów dla elementu C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj...
info : Pakiet "Microsoft.EntityFrameworkCore" jest zgodny ze wszystkimi strukturami określonymi w projekcie "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Element PackageReference dla pakietu "Microsoft.EntityFrameworkCore" w wersji "5.0.0" został dodany do pliku "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Trwa zatwierdzanie przywracania...
info : Zapisywanie pliku zasobów na dysk. Ścieżka: C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\project.assets.json
log : Przywrócono element C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj (w 176 ms).

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Podświetlenie na czerwono klasy po której dziedziczy *ProductContext* - *DbContext* zniknęło

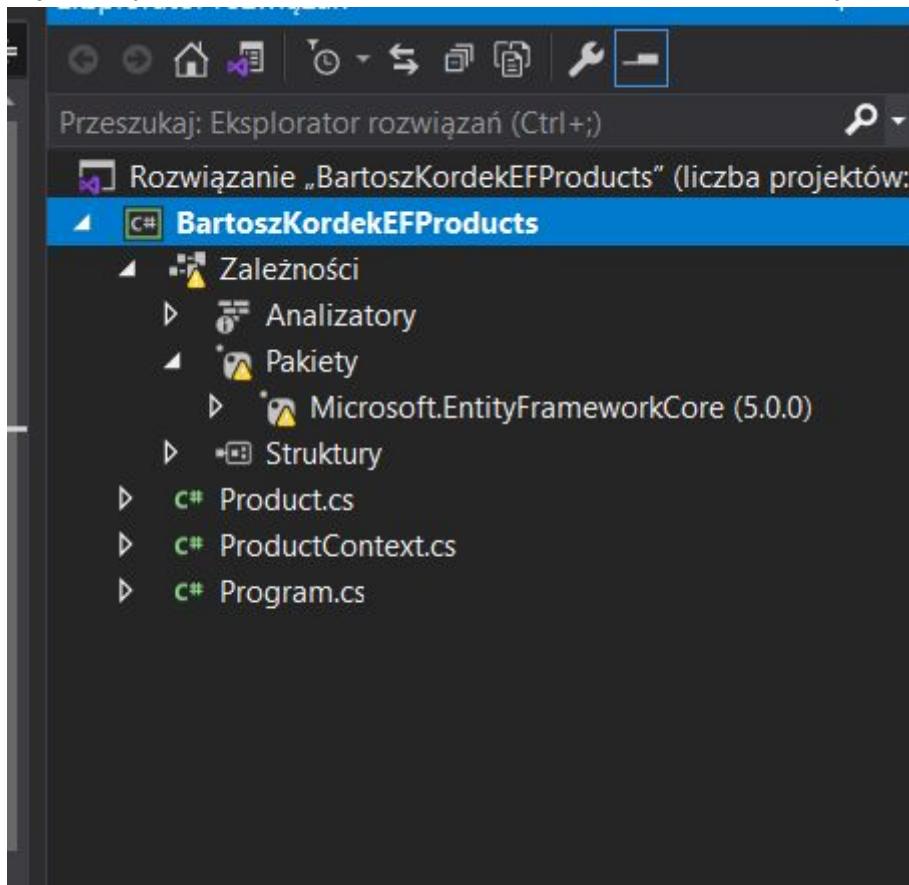


The screenshot shows a code editor window with three tabs at the top: "ProductContext.cs*", "Product.cs*", and "Program.cs". The "ProductContext.cs*" tab is active. The code is as follows:

```
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class ProductContext : DbContext
9      {
10  }
11 }
12
```

The word "DbContext" is highlighted in red, indicating a potential issue or warning.

Pojawił się pakiet *Microsoft.EntityFrameworkCore* w drzewie projektu



Dodanie kolekcji *Products* do klasy *ProductContext*

```
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class ProductContext : DbContext
9      {
10         public DbSet<Product> Products { get; set; }
11     }
12 }
```

Próba utworzenia kodu odpowiedzialnego za migrację modelu

```
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet add package Microsoft.EntityFrameworkCore.Core
Trwa określanie projektów do przywrócenia...
Writing C:\Users\barto\AppData\Local\Temp\tmp75F3.tmp
info : Dodawanie elementu PackageReference dla pakietu "Microsoft.EntityFrameworkCore" do projektu "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/index.json 561 ms
info : Trwa przywracanie pakietów dla elementu C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj...
info : Pakiet "Microsoft.EntityFrameworkCore" jest zgodny ze wszystkimi strukturami określonymi w projekcie "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Element PackageReference dla pakietu "Microsoft.EntityFrameworkCore" w wersji "5.0.0" został dodany do pliku "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Trwa zatwierdzanie przywracania...
info : Zapisywanie pliku zasobów na dysku. Ścieżka: C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\project.assets.json
log : Przywrócono C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj (w 176 ms).

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add InitProductDatabase
Build started...
Build succeeded.
Your startup project 'BartoszKordekEFProducts' doesn't reference Microsoft.EntityFrameworkCore.Design. This package is required for the Entity Framework Core Tools to work. Ensure your startup project is correct, install the package, and try again.
```

Jak widać powyżej, brakuje pakietu *Microsoft.EntityFrameworkCore.Design*

Dodanie pakietu *Microsoft.EntityFrameworkCore.Design* do projektu

```
Wiersz polecenia

Build started...
Build succeeded.
Your startup project 'BartoszKordekEFProducts' doesn't reference Microsoft.EntityFrameworkCore.Design. This package is required for the Entity Framework Core Tools to work. Ensure your startup project is correct, install the package, and try again.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet add package Microsoft.EntityFrameworkCore.Design
Trwa określanie projektów do przywrócenia...
Writing C:\Users\barto\AppData\Local\Temp\tmpE786.tmp
info : Dodawanie elementu PackageReference dla pakietu "Microsoft.EntityFrameworkCore.Design" do projektu "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/index.json
info :   OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/index.json 678 ms
info : Trwa przywracanie pakietów dla elementu C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj...
info : Pakiet "Microsoft.EntityFrameworkCore.Design" jest zgodny ze wszystkimi strukturami określonymi w projekcie "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Element PackageReference dla pakietu "Microsoft.EntityFrameworkCore.Design" w wersji "5.0.0" został dodany do pliku "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Trwa zatwierdzanie przywracania...
info : Generowanie pliku programu MSBuild C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\BartoszKordekEFProducts.csproj.nuget.g.props.
info : Zapisywanie pliku zasobów na dysk. Ścieżka: C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\project.assets.json
log  : Przywrócono element C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj (w 185 ms).

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Ponowna próba utworzenia kodu odpowiedzialnego za migrację modelu.

```
Wiersz polecenia

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet add package Microsoft.EntityFrameworkCore.Design
Trwa określanie projektów do przywrócenia...
Writing C:\Users\barto\AppData\Local\Temp\tmpE786.tmp
info : Dodawanie elementu PackageReference dla pakietu "Microsoft.EntityFrameworkCore.Design" do projektu "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/index.json
info :   OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/index.json 678 ms
info : Trwa przywracanie pakietów dla elementu C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj...
info : Pakiet "Microsoft.EntityFrameworkCore.Design" jest zgodny ze wszystkimi strukturami określonymi w projekcie "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Element PackageReference dla pakietu "Microsoft.EntityFrameworkCore.Design" w wersji "5.0.0" został dodany do pliku "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Trwa zatwierdzanie przywracania...
info : Generowanie pliku programu MSBuild C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\BartoszKordekEFProducts.csproj.nuget.g.props.
info : Zapisywanie pliku zasobów na dysk. Ścieżka: C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\project.assets.json
log  : Przywrócono element C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj (w 185 ms).

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add InitProductDatabase
Build started...
Build succeeded.
No DbContext was found in assembly 'BartoszKordekEFProducts'. Ensure that you're using the correct assembly and that the type is neither abstract nor generic.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Konieczne było uruchomienie aplikacji

```
Wiersz polecenia

Build started...
Build succeeded.
System.InvalidOperationException: No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.
   at Microsoft.EntityFrameworkCore.Internal.DbContextServices.Initialize(IServiceProvider scopedProvider, IDbContextOptions contextOptions, DbContext context)
   at Microsoft.EntityFrameworkCore.DbContext.get_InternalServiceProvider()
   at Microsoft.EntityFrameworkCore.DbContext.Microsoft.EntityFrameworkCore.Infrastructure.IInfrastructure<System.IServiceProvider>.get_Instance()
   at Microsoft.EntityFrameworkCore.Infrastructure.Internal.InfrastructureExtensions.GetService[TService](IInfrastructure`1 accessor)
   at Microsoft.EntityFrameworkCore.Infrastructure.AccessorExtensions.GetService[TService](IInfrastructure`1 accessor)
   at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(Func`1 factory)
   at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(String contextType)
   at Microsoft.EntityFrameworkCore.Design.Internal.MigrationsOperations.AddMigration(String name, String outputDir, String contextType, String namespace)
   at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigrationImpl(String name, String outputDir, String contextType, String namespace)
   at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigration.<>c__DisplayClass0_0.<>c__DisplayClass0_0.<>b__0()
   at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.<>c__DisplayClass3_0`1.<>c__DisplayClass3_0_1.<>b__0()
   at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.Execute(Action action)
No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.
```

```
BartoszKordekEFProducts
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class ProductContext:DbContext
9      {
10         public DbSet<Product> Products { get; set; }
11
12         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
13         {
14             base.OnConfiguring(optionsBuilder);
15             optionsBuilder.UseSqlite("Datasource=ProductsDatabase");
16         }
17     }
18 }
```

Dodanie pakietu *Microsoft.EntityFrameworkCore.Sqlite*

```
Wiersz polecenia
extType, String namespace)
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigration.<>c__DisplayClass0_0.<.ctor>b__0()
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.<>c__DisplayClass3_0`1.<Execute>b__0()
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.Execute(Action action)
No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet add package Microsoft.EntityFrameworkCore.Sqlite
Trwa określanie projektów do przywrócenia...
Writing C:\Users\barto\AppData\Local\Temp\tmp72FA.tmp
info : Dodawanie elementu PackageReference dla pakietu "Microsoft.EntityFrameworkCore.Sqlite" do projektu "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/index.json 907 ms
info : Trwa przywracanie pakietów dla elementu C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj...
info : Pakiet "Microsoft.EntityFrameworkCore.Sqlite" jest zgodny ze wszystkimi strukturami określonymi w projekcie "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Element PackageReference dla pakietu "Microsoft.EntityFrameworkCore.Sqlite" w wersji "5.0.0" został dodany do pliku "C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj".
info : Trwa zatwierdzanie przywracania...
info : Zapisywanie pliku zasobów na dysk. Ścieżka: C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\obj\project.assets.json
log : Przywrócono element C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj (w 192 ms).

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

```
Wiersz polecenia
Build started...
Build succeeded.
System.InvalidOperationException: No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.
    at Microsoft.EntityFrameworkCore.Internal.DbContextServices.Initialize(IServiceProvider scopedProvider, IDbContextOptions contextOptions, DbContext context)
    at Microsoft.EntityFrameworkCore.DbContext.get_InternalServiceProvider()
    at Microsoft.EntityFrameworkCore.DbContext.Microsoft.EntityFrameworkCore.Infrastructure.IInfrastructure<System.IServiceProvider>.get_Instance()
    at Microsoft.EntityFrameworkCore.Infrastructure.Internal.InfrastructureExtensions.GetService[TService](IInfrastructure`1 accessor)
    at Microsoft.EntityFrameworkCore.Infrastructure.AccessorExtensions.GetService[TService](IInfrastructure`1 accessor)
    at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(Func`1 factory)
    at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(String contextType)
    at Microsoft.EntityFrameworkCore.Design.Internal.MigrationsOperations.AddMigration(String name, String outputDir, String contextType, String namespace)
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigrationImpl(String name, String outputDir, String contextType, String namespace)
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigration.<>c__DisplayClass0_0.<.ctor>b__0()
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.<>c__DisplayClass3_0`1.<Execute>b__0()
    at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.Execute(Action action)
No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Konieczne było ponowne odpalenie aplikacji, która nic nie wykonuje (ewentualnie wyświetla Hello World).

Ponowna próba stworzenia kodu odpowiedzialnego za migrację modelu, tym razem pomyślna.

```
Wiersz polecenia
onstructor and passes it to the base constructor for DbContext.
  at Microsoft.EntityFrameworkCore.Internal.DbContextServices.Initialize(IServiceProvider scopedProvider, IDbContextOptions contextOptions, DbContext context)
  at Microsoft.EntityFrameworkCore.DbContext.get_InternalServiceProvider()
  at Microsoft.EntityFrameworkCore.DbContext.Microsoft.EntityFrameworkCore.Infrastructure.IInfrastructure<System.IServiceProvider>.get_Instance()
  at Microsoft.EntityFrameworkCore.Infrastructure.Internal.InfrastructureExtensions.GetService[TService](IInfrastructure`1 accessor)
  at Microsoft.EntityFrameworkCore.Infrastructure.AccessorExtensions.GetService[TService](IInfrastructure`1 accessor)
  at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(Func`1 factory)
  at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(String contextType)
  at Microsoft.EntityFrameworkCore.Design.Internal.MigrationsOperations.AddMigration(String name, String outputDir, String contextType, String namespace)
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigrationImpl(String name, String outputDir, String contextType, String namespace)
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigration.<>c__DisplayClass0_0.<.ctor>b__0()
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.<>c__DisplayClass3_0`1.<Execute>b__0()
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.Execute(Action action)
No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add InitProductDatabase

Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Aktualizacja bazy dancych

```
Wiersz polecenia
  at Microsoft.EntityFrameworkCore.Infrastructure.Internal.InfrastructureExtensions.GetService[TService](IInfrastructure`1 accessor)
  at Microsoft.EntityFrameworkCore.Infrastructure.AccessorExtensions.GetService[TService](IInfrastructure`1 accessor)
  at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(Func`1 factory)
  at Microsoft.EntityFrameworkCore.Design.Internal.DbContextOperations.CreateContext(String contextType)
  at Microsoft.EntityFrameworkCore.Design.Internal.MigrationsOperations.AddMigration(String name, String outputDir, String contextType, String namespace)
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigrationImpl(String name, String outputDir, String contextType, String namespace)
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.AddMigration.<>c__DisplayClass0_0.<.ctor>b__0()
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.<>c__DisplayClass3_0`1.<Execute>b__0()
  at Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.Execute(Action action)
No database provider has been configured for this DbContext. A provider can be configured by overriding the 'DbContext.OnConfiguring' method or by using 'AddDbContext' on the application service provider. If 'AddDbContext' is used, then also ensure that your DbContext type accepts a DbContextOptions<TContext> object in its constructor and passes it to the base constructor for DbContext.

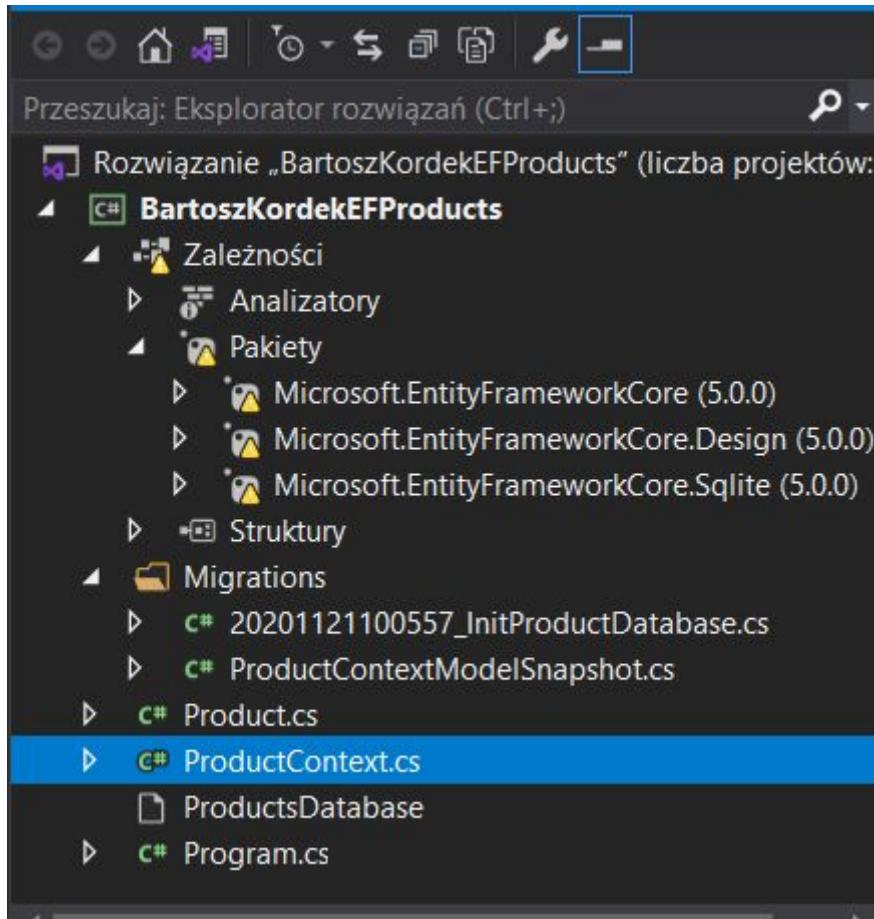
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add InitProductDatabase

Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121100557_InitProductDatabase'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Pojawiła się migracja *InitProductDatabase*



Modyfikacja klasy sterującej Program i próba odpalenia aplikacji.

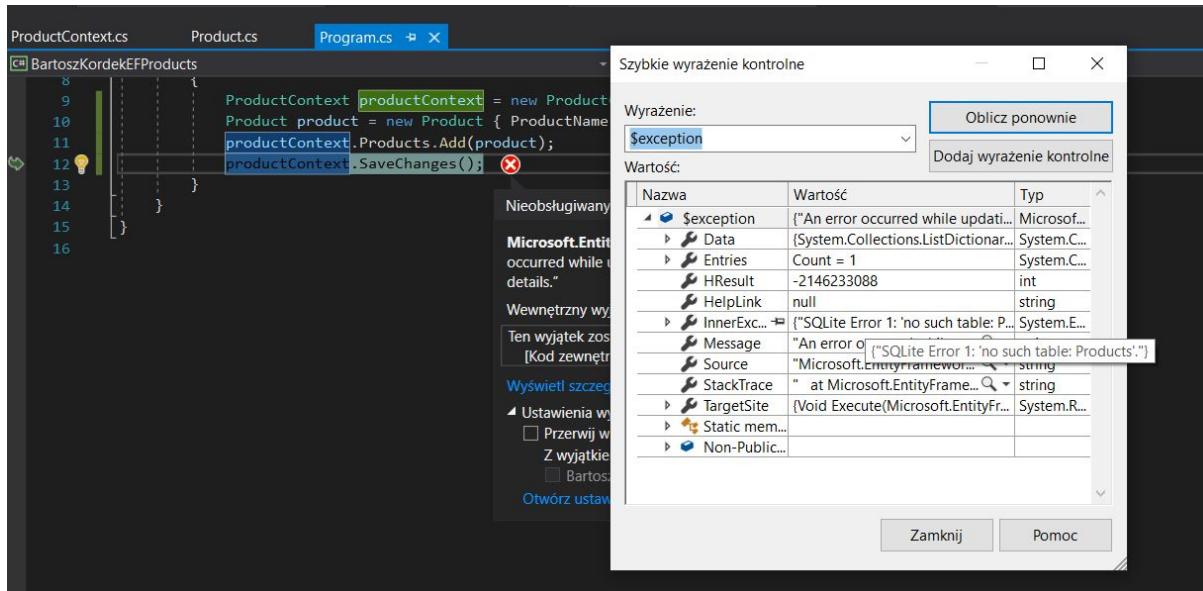
The screenshot shows the Visual Studio code editor with the 'Program.cs' file open. The code is as follows:

```
1  using System;
2
3  namespace BartoszKordekEFProducts
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              ProductContext productContext = new ProductContext();
10             Product product = new Product { ProductName = "Flamaster" };
11             productContext.Products.Add(product);
12             productContext.SaveChanges(); // Breakpoint here
13         }
14     }
15 }
```

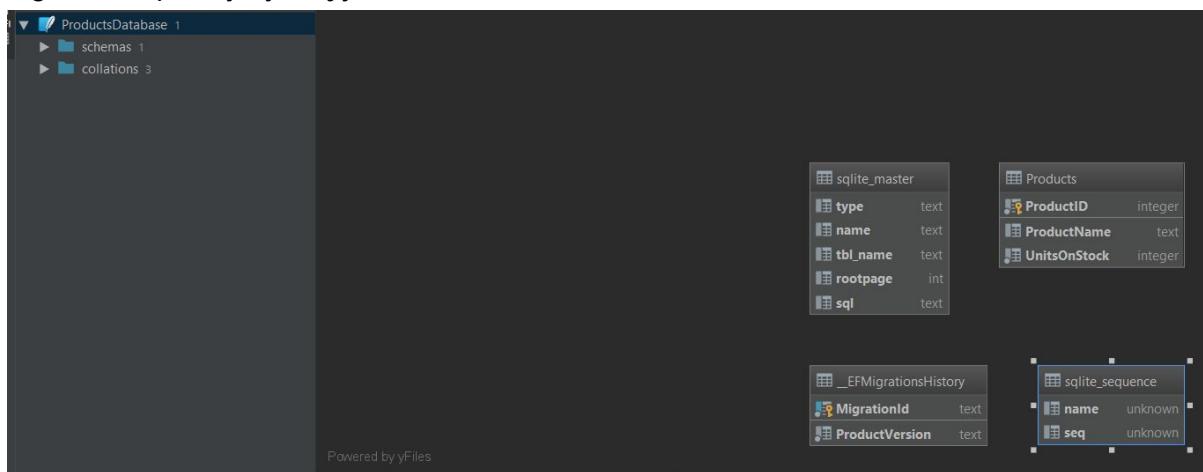
A yellow lightbulb icon is shown on line 12, indicating a potential issue. An exception dialog is open at the bottom right:

- Nieobsługiwany wyjątek
- Microsoft.EntityFrameworkCore.DbUpdateException:** "An error occurred while updating the entries. See the inner exception for details."
- Wewnętrzny wyjątek
- Ten wyjątek został pierwotnie zgłoszony w tym stosie wywołań:
[Kod zewnętrzny]
- [Wyświetl szczegóły](#) | [Kopiuj szczegóły](#)
- [Ustawienia wyjątków](#)

Sprawdzenie wyjątku.



Wg DataGrip znajduje się jednak tabela *Products*.



sqlite_master	
type	text
name	text
tbl_name	text
rootpage	int
sql	text

Products	
ProductID	integer
ProductName	text
UnitsOnStock	integer

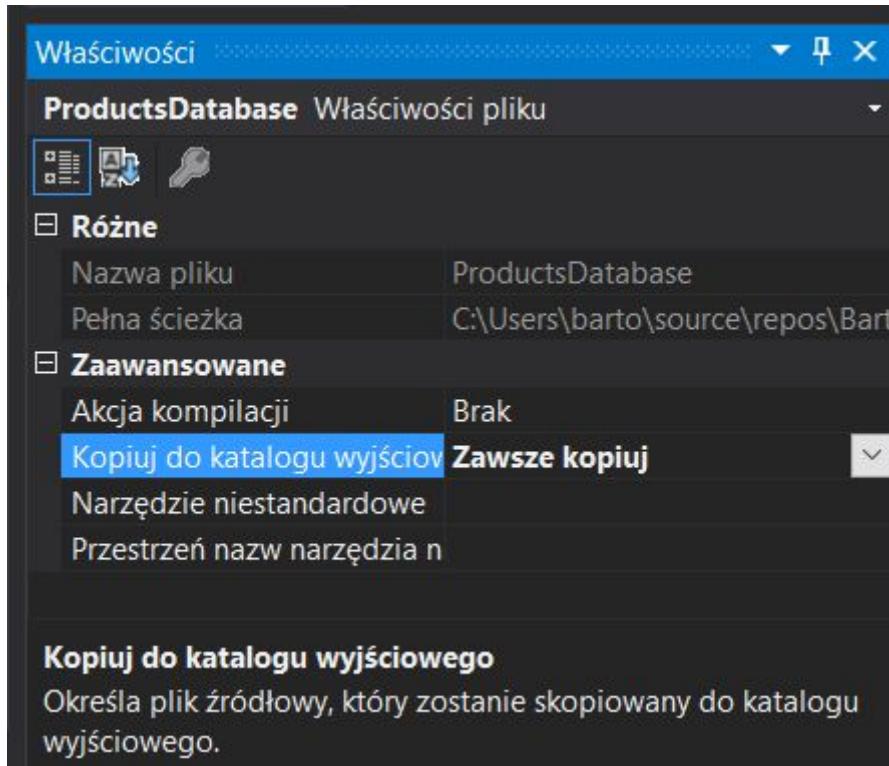
_EFMigrationsHistory	
MigrationId	text
ProductVersion	text

sqlite_sequence	
name	unknown
seq	unknown

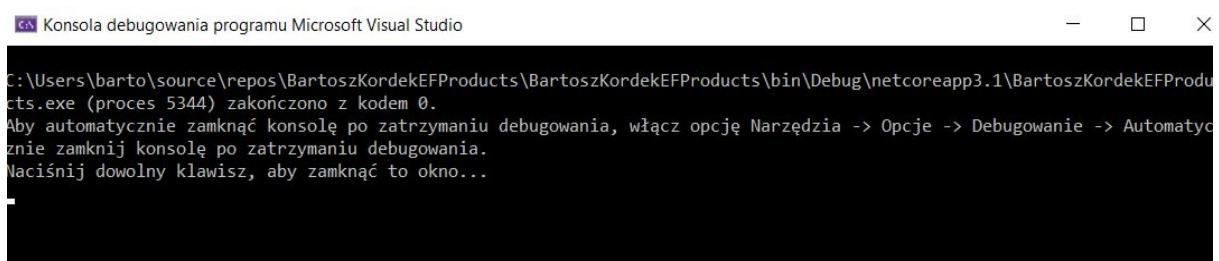
Nie ma jej jednak w folderze: *bin/Debug/netcoreapp3.1/*

sqlite_master	
type	text
name	text
tbl_name	text
rootpage	int
sql	text

Zmiana właściwości pliku *ProductDatabase* na Zawsze kopiuj do katalogu wyjściowego.



Ponowne odpalenie aplikacji, tym razem pomyślne.



Widok bazy w DataGrip, folder: *bin/Debug/netcoreapp3.1/*

The screenshot shows the DataGrip interface displaying the schema of a SQLite database. There are four tables visible:

- sqlite_master**: Contains columns: type (text), name (text), tbl_name (text), rootpage (int), and sql (text).
- Products**: Contains columns: ProductID (integer, primary key), ProductName (text), and UnitsOnStock (integer).
- __EFMigrationsHistory**: Contains columns: MigrationId (text, primary key), ProductVersion (text), and seq (text).
- sqlite_sequence**: Contains columns: name (unknown) and seq (unknown).

W bazie danych znajduje się ponadto tabela z migracjami.

The screenshot shows the __EFMigrationsHistory table with one entry:

MigrationId	ProductVersion
1 20201121100557_InitProductDatabase	5.0.0

W DataGrip można zobaczyć, że produkt *Flamaster* został dodany do tabeli *Products*.

The screenshot shows the results of a `select * from main.Products` query and the structure of the Products table:

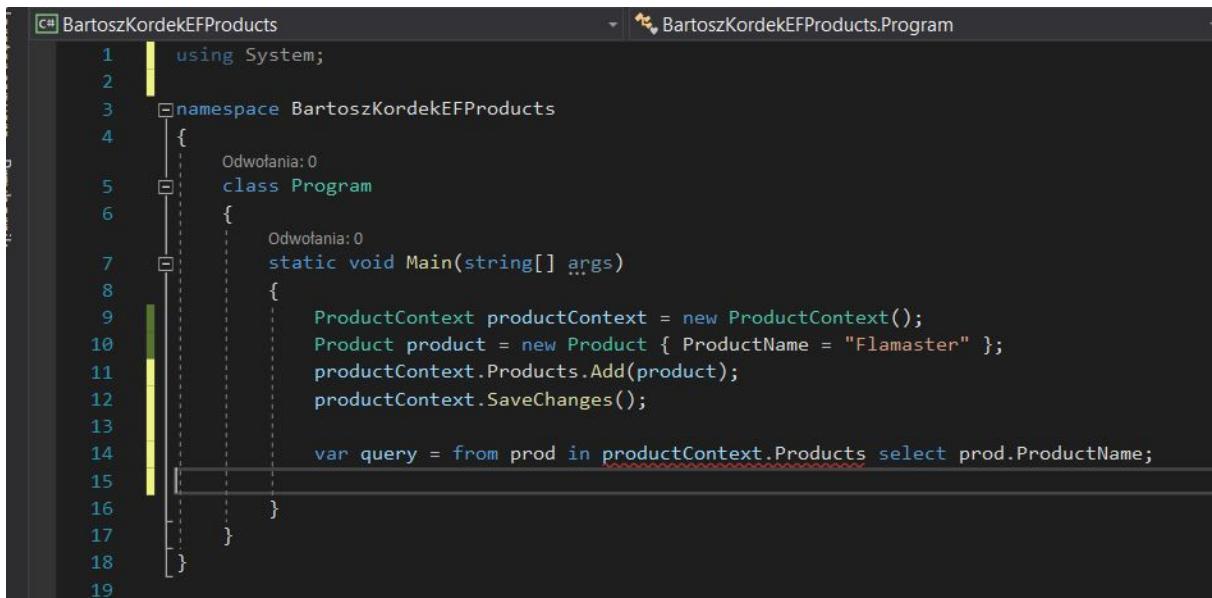
Query results:

ProductID	ProductName	UnitsOnStock
1	Flamaster	0

Table structure:

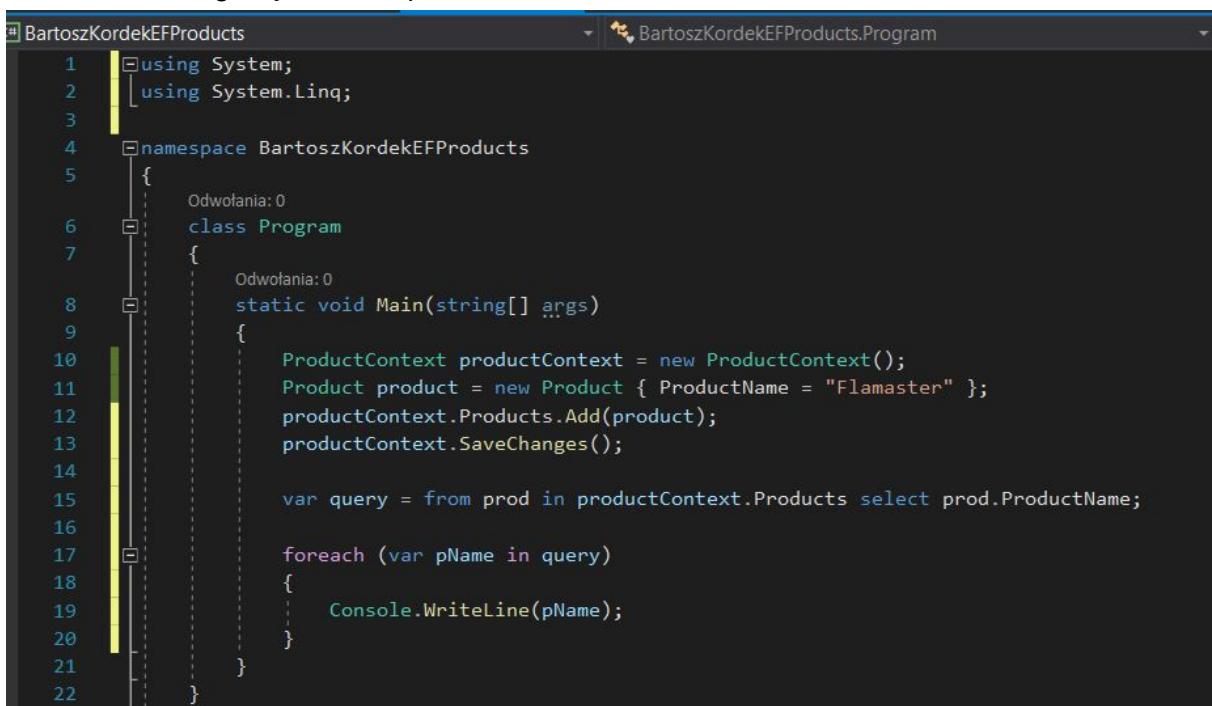
ProductID	ProductName	UnitsOnStock
1	Flamaster	0

Modyfikacja klasy sterującej *Program*. Widać podświetlenie *productContext* w przypadku użycia obiektu do napisania zapytania do bazy danych.



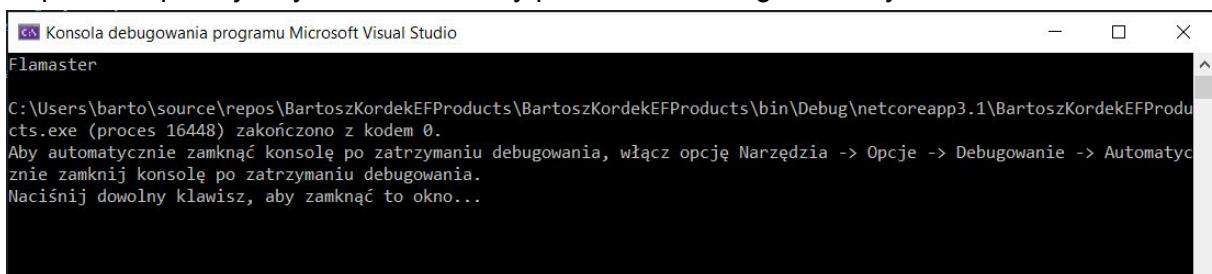
```
1  using System;
2
3  namespace BartoszKordekEFProducts
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              ProductContext productContext = new ProductContext();
10             Product product = new Product { ProductName = "Flamaster" };
11             productContext.Products.Add(product);
12             productContext.SaveChanges();
13
14             var query = from prod in productContext.Products select prod.ProductName;
15
16         }
17     }
18 }
19
```

Wstawienie usinga *System.Linq*



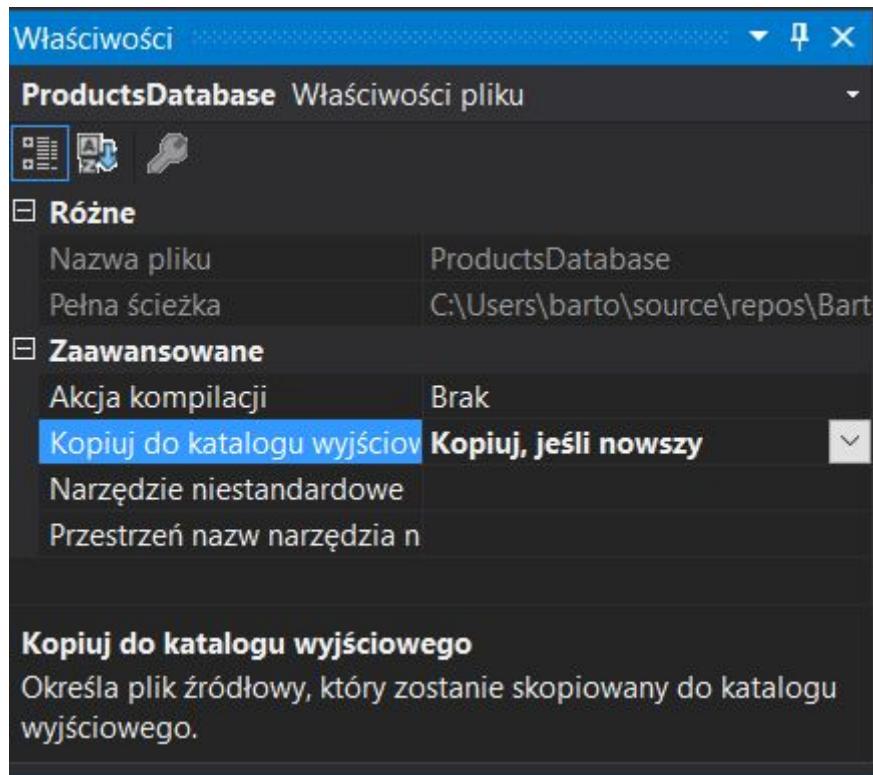
```
1  using System;
2  using System.Linq;
3
4  namespace BartoszKordekEFProducts
5  {
6      class Program
7      {
8          static void Main(string[] args)
9          {
10             ProductContext productContext = new ProductContext();
11             Product product = new Product { ProductName = "Flamaster" };
12             productContext.Products.Add(product);
13             productContext.SaveChanges();
14
15             var query = from prod in productContext.Products select prod.ProductName;
16
17             foreach (var pName in query)
18             {
19                 Console.WriteLine(pName);
20             }
21         }
22     }
23 }
```

Odpalenie aplikacji i wyświetlenie nazwy produktu dodanego do bazy.

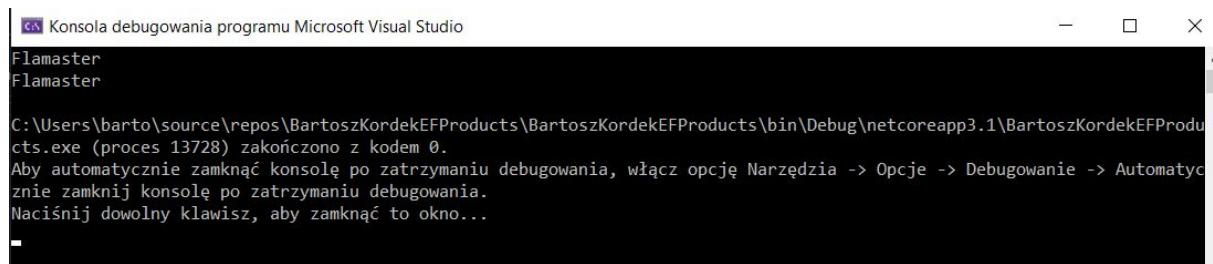


```
Konsola debugowania programu Microsoft Visual Studio
Flamaster
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\bin\Debug\netcoreapp3.1\BartoszKordekEFProducts.exe (proces 16448) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Zaznaczenie opcji "Kopiuj jeśli nowszy", aby zachować istniejące obiekty w bazie (chyba że nastąpi zmiana schematu bazy danych źródłowych).



Ponowne odpalenie aplikacji.

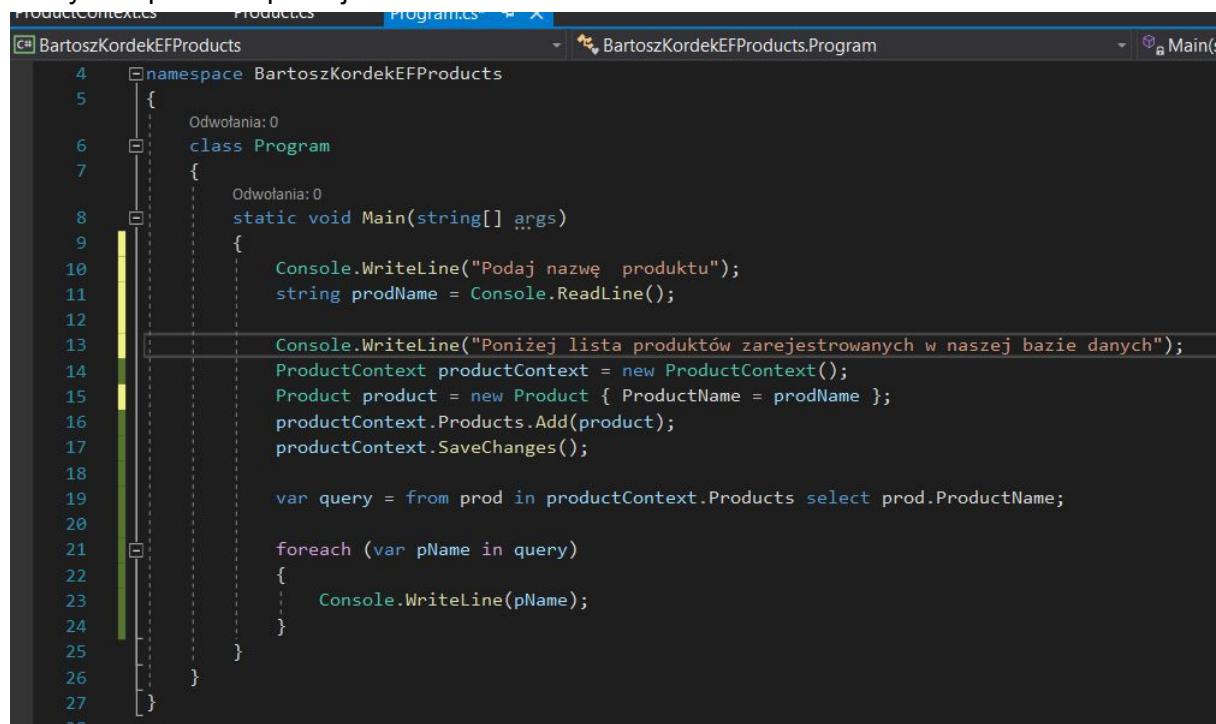


Jak widać, do bazy został dodany kolejny *Flamaster*.

The screenshot shows a grid view with three columns: ProductID, ProductName, and UnitsOnStock. There are two rows, both labeled 'Flamaster'.

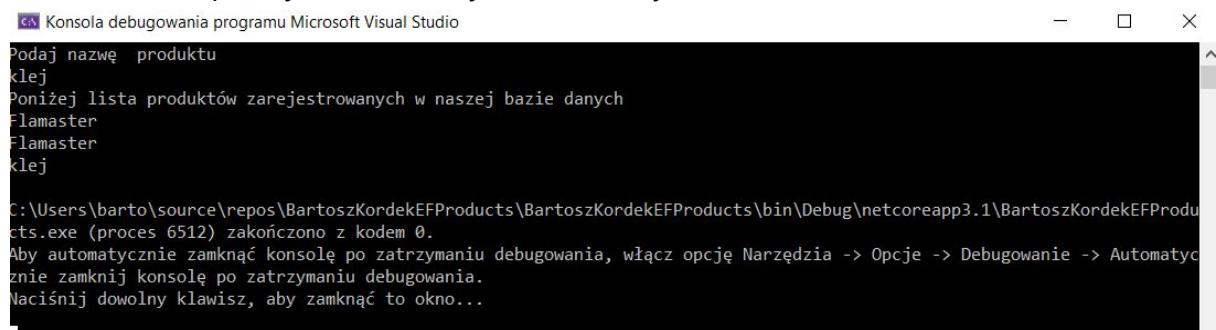
ProductID	ProductName	UnitsOnStock
1	Flamaster	0
2	Flamaster	0

Modyfikacja klasy Program, tak żeby można było dodawać nazwę produktu z klawiatury przy każdym odpaleniu aplikacji.



```
4  namespace BartoszKordekEFProducts
5  {
6      class Program
7      {
8          static void Main(string[] args)
9          {
10             Console.WriteLine("Podaj nazwę produktu");
11             string prodName = Console.ReadLine();
12
13             Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
14             ProductContext productContext = new ProductContext();
15             Product product = new Product { ProductName = prodName };
16             productContext.Products.Add(product);
17             productContext.SaveChanges();
18
19             var query = from prod in productContext.Products select prod.ProductName;
20
21             foreach (var pName in query)
22             {
23                 Console.WriteLine(pName);
24             }
25         }
26     }
27 }
```

Uruchomienie aplikacji. Produkt **Klej** został dodany do tabeli *Products*.



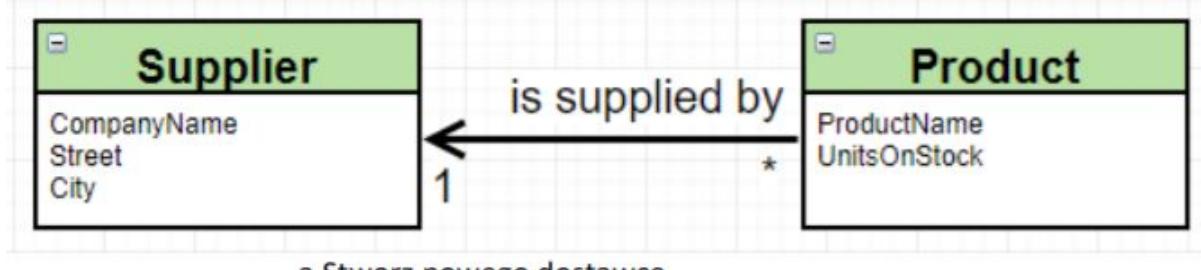
```
Podaj nazwę produktu
klej
Poniżej lista produktów zarejestrowanych w naszej bazie danych
Flamaster
Flamaster
klej

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\bin\Debug\netcoreapp3.1\BartoszKordekEFProducts.exe (proces 6512) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

II. One to Many

Zmodyfikuj model wprowadzając pojęcie Dostawcy

- Many to one



- Tworzę klasę *Supplier*

The screenshot shows the Visual Studio IDE with the Supplier.cs file open. The code defines a class named Supplier with properties for SupplierID, CompanyName, Street, and City. The SupplierID property is annotated with [Key]. The code is part of the BartoszKordekEFProducts project.

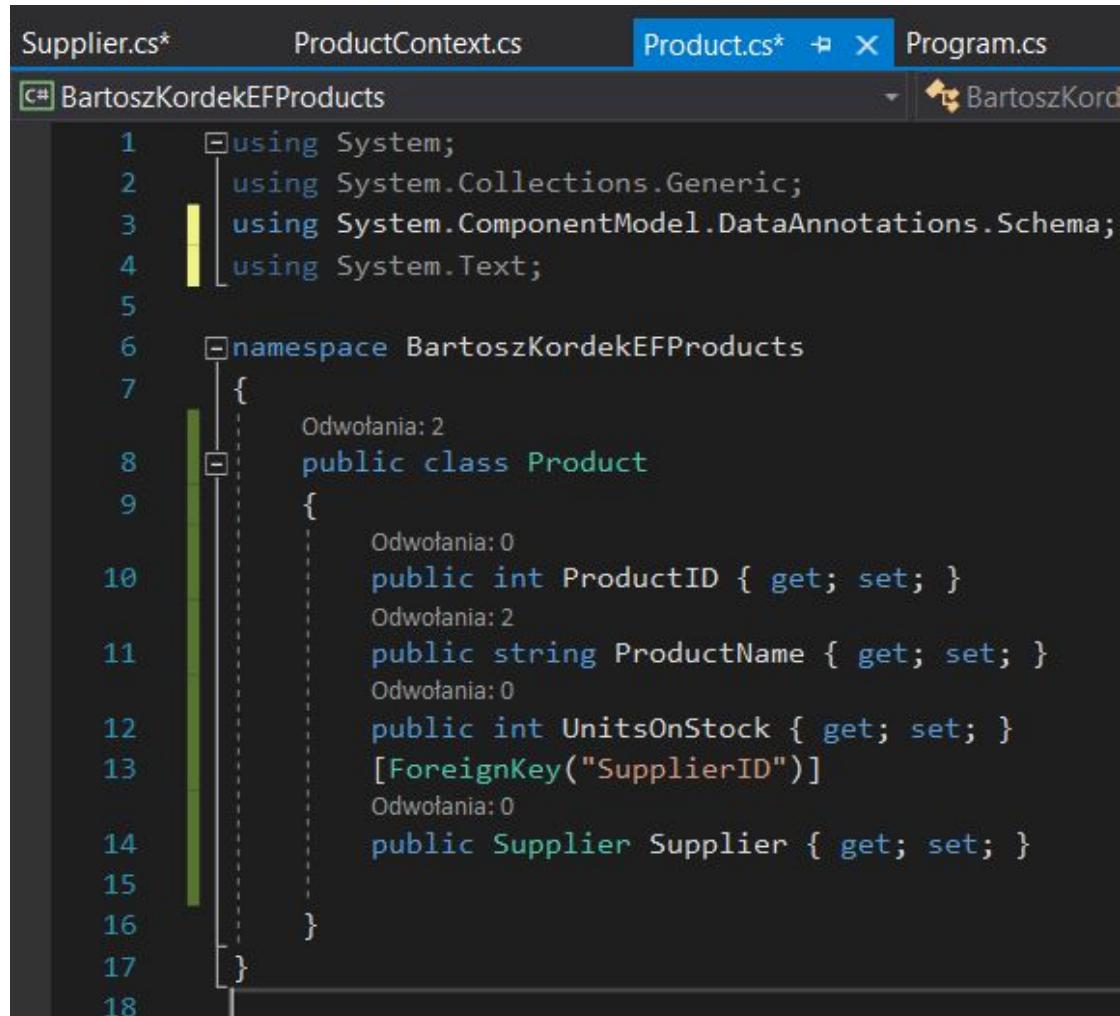
```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordekEFProducts
6  {
7      public class Supplier
8      {
9          public int SupplierID { get; set; }
10         public string CompanyName { get; set; }
11         public string Street { get; set; }
12         public string City { get; set; }
13     }
14 }
15
```

- dodaję klucz obcy *SupplierID* w klasie *Product*

The screenshot shows the Visual Studio IDE with the Product.cs file open. The code defines a class named Product with properties for ProductID, ProductName, UnitsOnStock, and Supplier. The UnitsOnStock property is annotated with [Required]. The Supplier property is annotated with [ForeignKey("SupplierID")]. The code is part of the BartoszKordekEFProducts project.

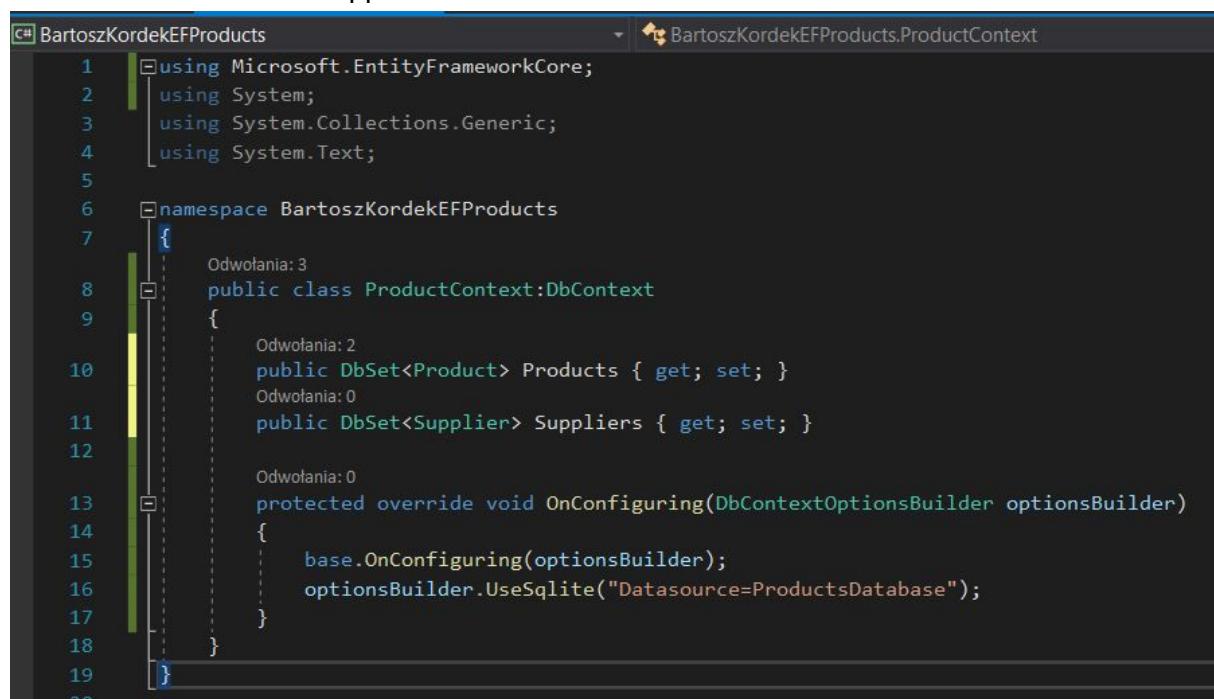
```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordekEFProducts
6  {
7      public class Product
8      {
9          public int ProductID { get; set; }
10         public string ProductName { get; set; }
11         public int UnitsOnStock { get; set; }
12         [ForeignKey("SupplierID")]
13         public Supplier Supplier { get; set; }
14     }
15 }
```

Konieczne dodanie usingu związanego z Adnotacjami



```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations.Schema;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class Product
9      {
10         public int ProductID { get; set; }
11         public string ProductName { get; set; }
12         public int UnitsOnStock { get; set; }
13         [ForeignKey("SupplierID")]
14         public Supplier Supplier { get; set; }
15     }
16 }
17
18
```

- dodałem DbSet Suppliers w klasie ProductContext



```
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class ProductContext : DbContext
9      {
10         public DbSet<Product> Products { get; set; }
11         public DbSet<Supplier> Suppliers { get; set; }
12
13         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
14         {
15             base.OnConfiguring(optionsBuilder);
16             optionsBuilder.UseSqlite("Datasource=ProductsDatabase");
17         }
18     }
19 }
20
```

- migracja modelu

```
Wiersz polecenia
pos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKeyAttribute" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKey" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]

Komplikacja NIE POWIÓDŁA SIĘ.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(13,25): error CS0053: Niespójność
    dostępności: typ właściwości "Supplier" jest mniej dostępny niż właściwość "Product.Supplier" [C:\Users\barto\source\re
pos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKeyAttribute" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKey" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
    Ostrzeżenia: 0
    Liczba błędów: 3

Czas, który upływał: 00:00:01.06

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplier
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

Upřednio należało uruchomić aplikacjē z prostym HelloWorld lub taką, która nie ma odnošników do Bazy Danych. Ja wykonałem taką która nic nie robi (pusta metoda Main). Wcześniesjsze zapisanie zmian, niestety nie pomogło.

- aktualizacja bazy danych

```
Wiersz polecenia
rce\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]

Komplikacja NIE POWIÓDŁA SIĘ.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(13,25): error CS0053: Niespójność
    dostępności: typ właściwości "Supplier" jest mniej dostępny niż właściwość "Product.Supplier" [C:\Users\barto\source\re
pos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKeyAttribute" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKey" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
    Ostrzeżenia: 0
    Liczba błędów: 3

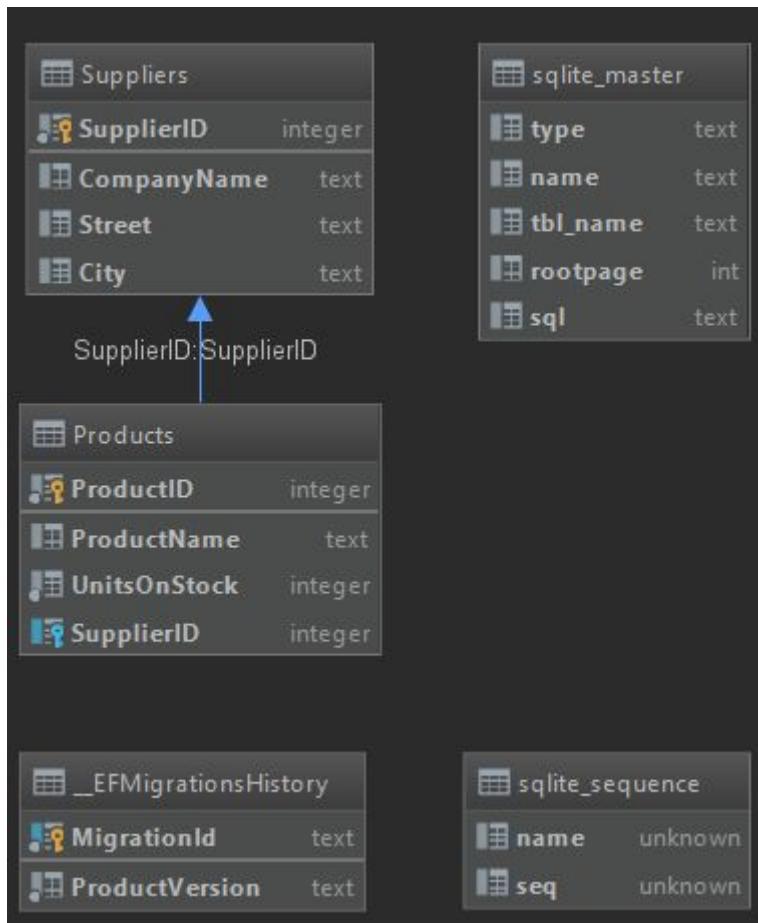
Czas, który upływał: 00:00:01.06

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplier
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

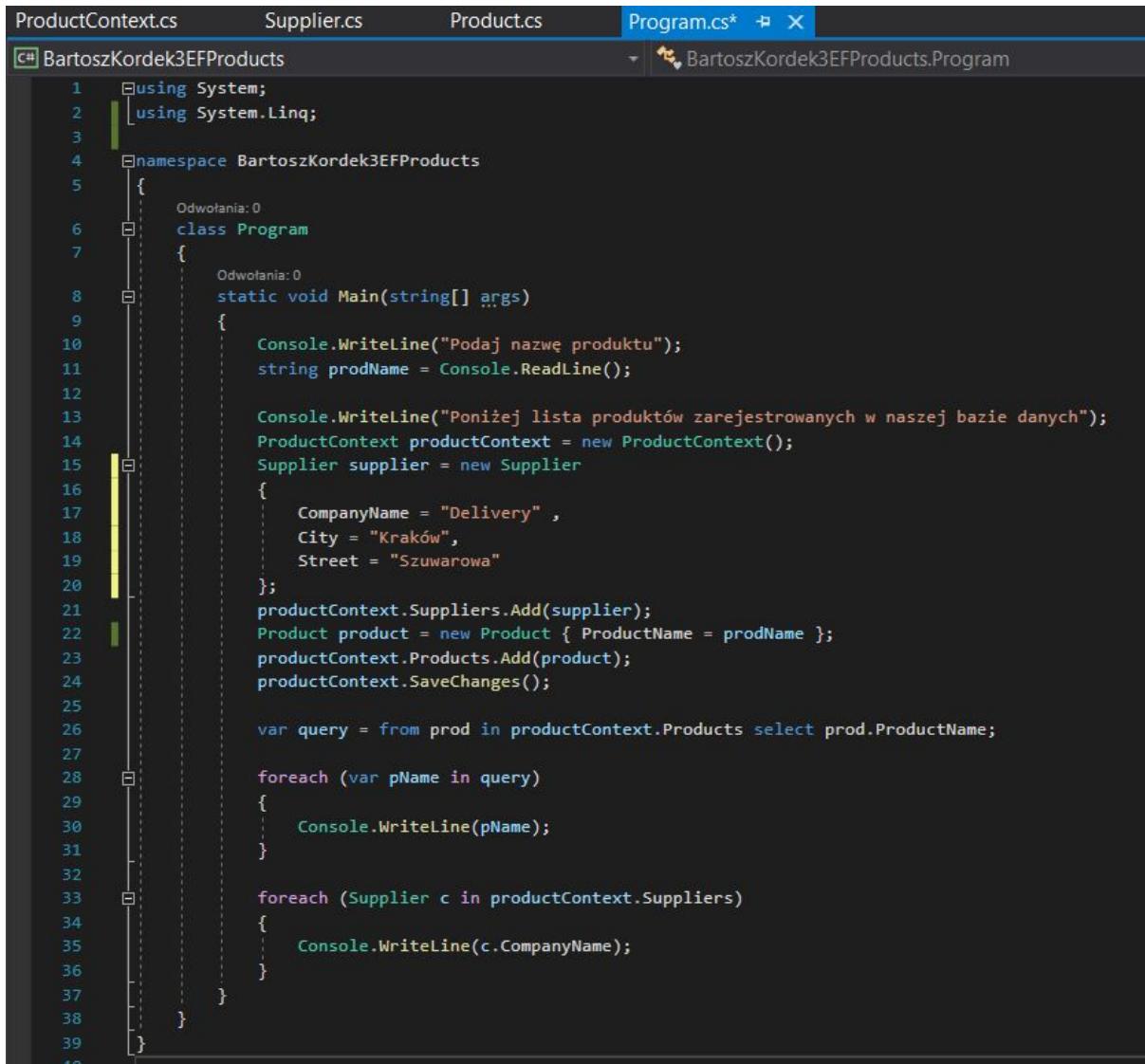
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121104638_AddedSupplier'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

- schemat wygląda następująco



- dodanie Produktu i Dostawcy do bazy bez przypisania dostawcy do produktu
Aktualizacja klasy Program (Tę część wykonałem na końcu, więc różni się
namespace, pierwotny projekt w późniejszym etapie uległ poważnemu uszkodzeniu i
nie było możliwości łatwego przywrócenia do działania. Łatwiej było odtworzyć część
projektu od początku)

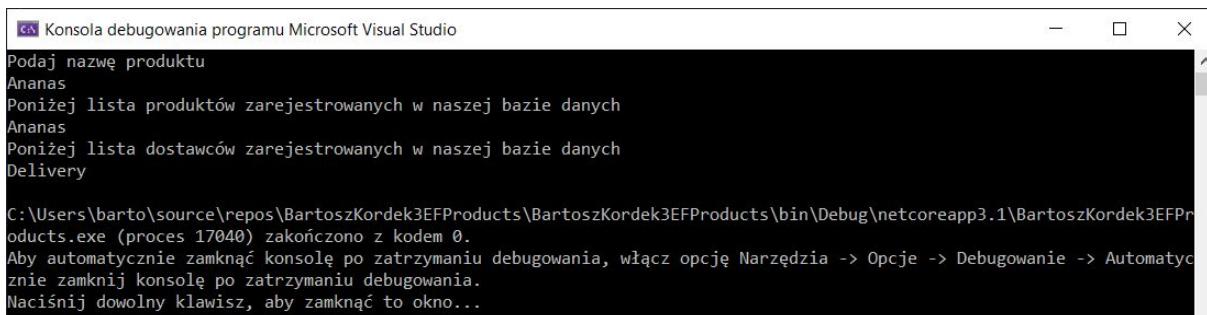


```

1  using System;
2  using System.Linq;
3
4  namespace BartoszKordek3EFProducts
5  {
6      class Program
7      {
8          static void Main(string[] args)
9          {
10             Console.WriteLine("Podaj nazwę produktu");
11             string prodName = Console.ReadLine();
12
13             Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
14             ProductContext productContext = new ProductContext();
15             Supplier supplier = new Supplier
16             {
17                 CompanyName = "Delivery" ,
18                 City = "Kraków",
19                 Street = "Szuwarowa"
20             };
21             productContext.Suppliers.Add(supplier);
22             Product product = new Product { ProductName = prodName };
23             productContext.Products.Add(product);
24             productContext.SaveChanges();
25
26             var query = from prod in productContext.Products select prod.ProductName;
27
28             foreach (var pName in query)
29             {
30                 Console.WriteLine(pName);
31             }
32
33             foreach (Supplier c in productContext.Suppliers)
34             {
35                 Console.WriteLine(c.CompanyName);
36             }
37         }
38     }
39 }

```

- uruchomienie programu



```

Konsola debugowania programu Microsoft Visual Studio
Podaj nazwę produktu
Ananas
Poniżej lista produktów zarejestrowanych w naszej bazie danych
Ananas
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
Delivery

C:\Users\barto\source\repos\BartoszKordek3EFProducts\BartoszKordek3EFProducts\bin\Debug\netcoreapp3.1\BartoszKordek3EFProducts.exe (proces 17040) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

Widac, że produkt "Ananas" i dostawca "Delivery" został dodany do bazy.

- widok w DataGrip

The screenshot shows two separate DataGrip sessions. The top session runs the query `select * from main.Suppliers` and returns one row of data:

	SupplierID	CompanyName	Street	City
1	1	Delivery	Szuwarowa	Kraków

The bottom session runs the query `select * from main.Products` and also returns one row of data:

	ProductID	ProductName	UnitsOnStock	SupplierID
1	1	Ananas	0	<null>

- przypisanie ostatnio dodanego (w tym przypadku pierwszego ze względu że tylko jeden jest w bazie) dostawcy do produktu

- Aktualizacja klasy *Program*

The screenshot shows the `Program.cs` file in a code editor. The code defines a `Program` class with a `Main` method. The `Main` method creates a `ProductContext` object, adds a new `Supplier` named "Delivery" from Kraków to the `Suppliers` collection, adds a new `Product` named "Ananas" to the `Products` collection, and then saves the changes.

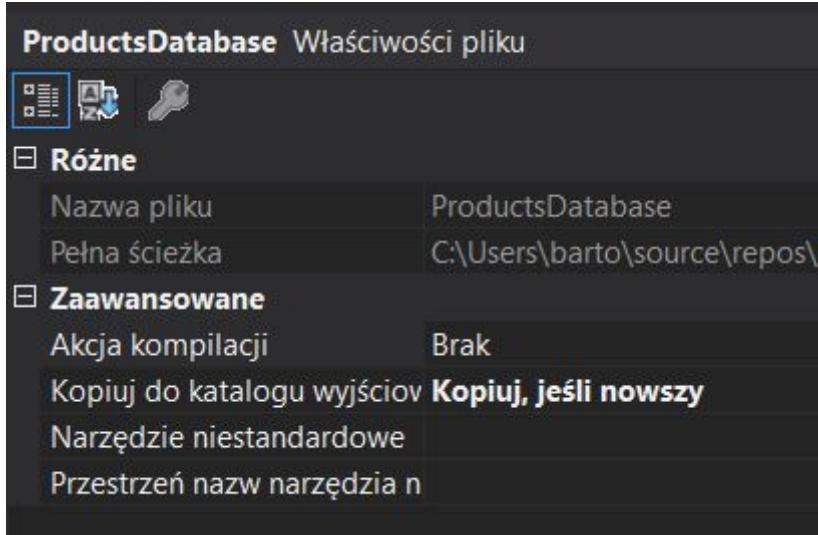
```

1  using System;
2  using System.Linq;
3
4  namespace BartoszKordek3EFProducts
5  {
6      class Program
7      {
8          static void Main(string[] args)
9          {
10             /*
11             Console.WriteLine("Podaj nazwę produktu");
12             string prodName = Console.ReadLine();
13         */
14             ProductContext productContext = new ProductContext();
15             /*
16             Supplier supplier = new Supplier
17             {
18                 CompanyName = "Delivery" ,
19                 City = "Kraków",
20                 Street = "Szuwarowa"
21             };
22             productContext.Suppliers.Add(supplier);
23             Product product = new Product { ProductName = prodName };
24             productContext.Products.Add(product);
25             productContext.SaveChanges();
26         */
27             var selectedProducts = from prod in productContext.Products orderby prod.ProductID descending select prod;
28             var selectedSuppliers = from sup in productContext.Suppliers orderby sup.SupplierID descending select sup;
29
30             Product selectedFirstProduct = selectedProducts.First();
31             Supplier selectedFirstSupplier = selectedSuppliers.First();

```

```
28
29
30     Product selectedFirstProduct = selectedProducts.First();
31     Supplier selectedFirstSupplier = selectedSuppliers.First();
32
33     selectedFirstProduct.Supplier = selectedFirstSupplier;
34     productContext.Products.Update(selectedFirstProduct);
35     productContext.SaveChanges();
36
37     Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
38     var query = from prod in productContext.Products select prod.ProductName;
39
40     foreach (var pName in query)
41     {
42         Console.WriteLine(pName);
43     }
44
45     Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych");
46     foreach (Supplier c in productContext.Suppliers)
47     {
48         Console.WriteLine(c.CompanyName);
49     }
50 }
```

- ustawiam opcję "Kopiuj jeśli nowszy"



- wyniki

```
Konsola debugowania programu Microsoft Visual Studio
Poniżej lista produktów zarejestrowanych w naszej bazie danych
Ananas
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
Delivery

C:\Users\barto\source\repos\BartoszKordek3EFProducts\BartoszKordek3EFProducts\bin\Debug\netcoreapp3.1\BartoszKordek3EFProducts.exe (proces 17188) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Jak widać ilość produktów i dostawców w bazie nie uległa zmianie pomimo ustawienia opcji "Kopiuj jeśli nowszy" w opcjach bazy danych *ProductDatabase*.

- wyniki w DataGrip

The screenshot shows two separate DataGrip sessions. The top session runs the query `select * from main.Suppliers` and returns one row of data:

SupplierID	CompanyName	Street	City
1	Delivery	Szuwarowa	Kraków

The bottom session runs the query `select * from main.Products` and also returns one row of data:

ProductID	ProductName	UnitsOnStock	SupplierID
1	Ananas	0	1

Widać powyżej, że został przypisany dostawca do produktu

- Stworzenie dostawców i od razu przypisanie ich do odpowiednich produktów.
 - aktualizacja klasy Program

```

C# BartoszKordekEFProducts
Odwołania: 0
BartoszKordekEFProducts.Program
Main(string[] args)
{
    Console.WriteLine("Podaj nazwę pierwszego produktu");
    string prodName1 = Console.ReadLine();
    Console.WriteLine("Podaj nazwę drugiego produktu");
    string prodName2 = Console.ReadLine();

    ProductContext productContext = new ProductContext();
    Supplier supplier = new Supplier
    {
        CompanyName = "ABud",
        City = "Kraków",
        Street = "Długa"
    };
    Product product1 = new Product { ProductName = prodName1, UnitsOnStock = 2, Supplier = supplier };
    Product product2 = new Product { ProductName = prodName2, UnitsOnStock = 3, Supplier = supplier };

    productContext.Suppliers.Add(supplier);
    productContext.Products.Add(product1);
    productContext.Products.Add(product2);
    productContext.SaveChanges();

    Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
    var query = from prod in productContext.Products select prod.ProductName;
    foreach (var pName in query)
    {
        Console.WriteLine(pName);
    }

    Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych");
    foreach (Supplier c in productContext.Suppliers)
    {
        Console.WriteLine(c.CompanyName);
    }
}

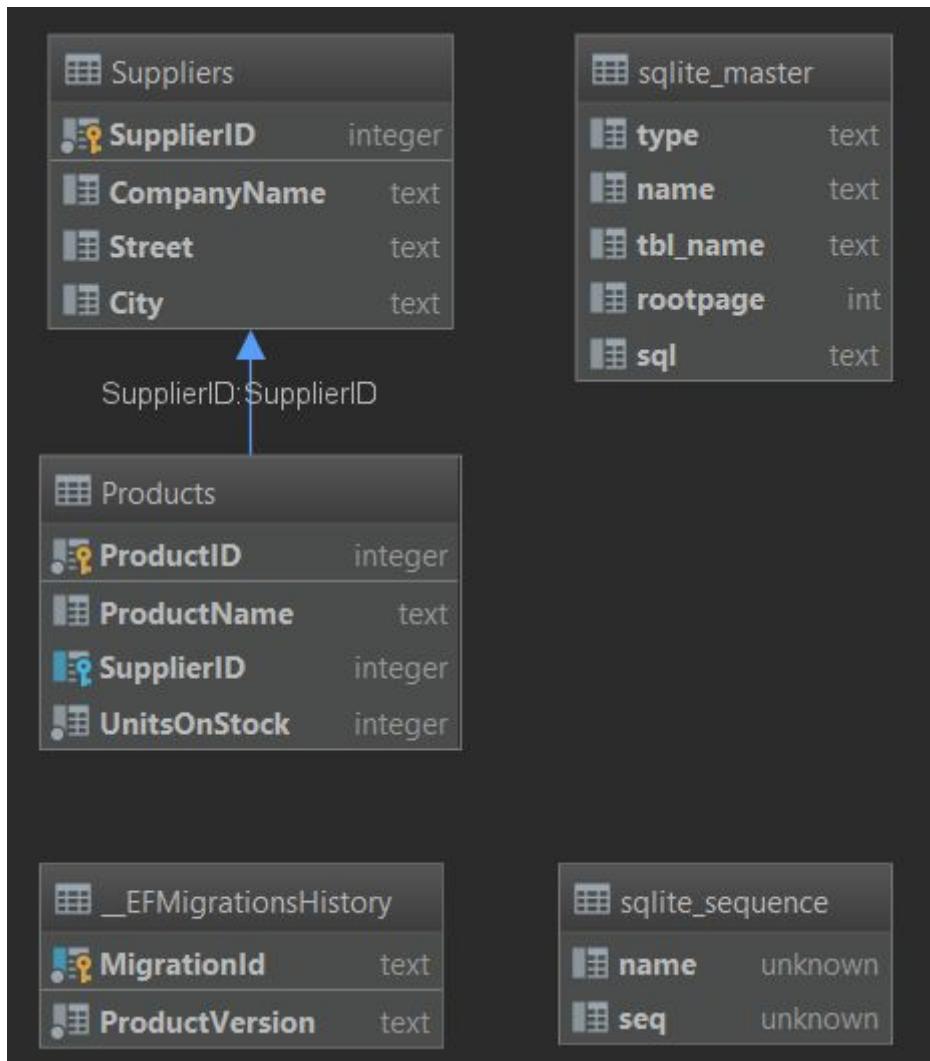
```

```
ProductContext.Products.Add(product);
Konsola debugowania programu Microsoft Visual Studio
Podaj nazwę pierwszego produktu
młotek
Podaj nazwę drugiego produktu
cegła
Poniżej lista produktów zarejestrowanych w naszej bazie danych
młotek
cegła
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
ABud

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\bin\Debug\netcoreapp3.1\BartoszKordekEFProducts.exe (proces 14076) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Do poprawnego działania konieczna była zmiana w pliku *ProductsDatabase* na Zawsze kopiuj w innym razie zwracał błąd, że nie ma tabeli Suppliers.

- schemat w DataGrip



```
select * from main.Products
```

Output main.Products

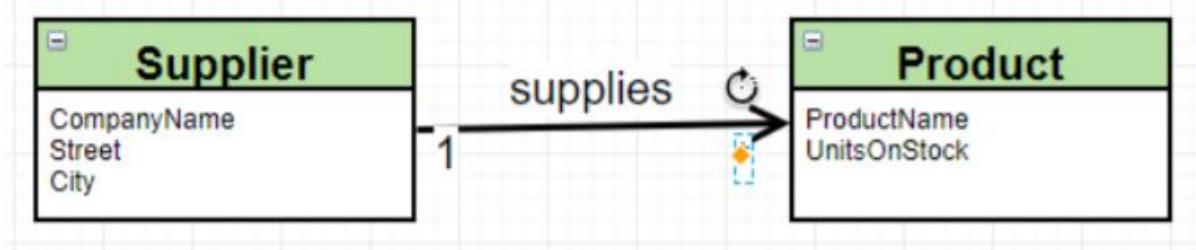
	ProductID	ProductName	SupplierID	UnitsOnStock
1	1	młotek	1	2
2	2	cegła	1	3

select * from main.Suppliers

	SupplierID	CompanyName	Street	City
1	1	ABud	Długa	Kraków

III. Many to one - odwrócenie relacji

Odwróć relacje zgodnie z poniższym schematem



- modyfikacja klasy *Supplier* - dodałem nowe pole public *ICollection<Product> Products*

```
# BartoszKordekEFProducts          # BartoszKordekEFProducts.S
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordekEFProducts
6  {
7      public class Supplier
8      {
9          public int SupplierID { get; set; }
10         public string CompanyName { get; set; }
11         public string Street { get; set; }
12         public string City { get; set; }
13
14         public ICollection<Product> Products { get; set; }
15     }
16 }
17
```

- modyfikacja klasy *Product* - zahashowałem pole *Supplier*

The screenshot shows a code editor window with the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations.Schema;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class Product
9      {
10         public int ProductID { get; set; }
11         public string ProductName { get; set; }
12         public int UnitsOnStock { get; set; }
13         /*
14         [ForeignKey("SupplierID")]
15         public Supplier Supplier { get; set; }
16         */
17     }
18 }
19
```

The code defines a *Product* class within the *BartoszKordekEFProducts* namespace. The *Supplier* property is annotated with *[ForeignKey("SupplierID")]*. The code editor interface includes tabs for "BartoszKordekEFProducts.cs" and "BartoszKordek", and a status bar at the bottom.

- modyfikacja klasy *Program*

```
bartosz@KODERELF:~/Products> dotnet run
Odwotania: 0
Bartosz@KODERELF:~/Products> Main(String[])
static void Main(string[] args)
{
    Console.WriteLine("Podaj nazwę pierwszego produktu");
    string prodName1 = Console.ReadLine();
    Console.WriteLine("Podaj nazwę drugiego produktu");
    string prodName2 = Console.ReadLine();

    ProductContext productContext = new ProductContext();

    Product product1 = new Product { ProductName = prodName1, UnitsOnStock = 2 };
    Product product2 = new Product { ProductName = prodName2, UnitsOnStock = 3 };
    List<Product> products = new List<Product>();
    products.Add(product1);
    products.Add(product2);
    Supplier supplier = new Supplier
    {
        CompanyName = "ABud",
        City = "Kraków",
        Street = "Długa",
        Products = products
    };

    productContext.Suppliers.Add(supplier);
    productContext.Products.Add(product1);
    productContext.Products.Add(product2);
    productContext.SaveChanges();

    Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
    var query = from prod in productContext.Products select prod.ProductName;
    foreach (var pName in query)
    {
        Console.WriteLine(pName);
    }

    Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych");
}
```

- migracja

```
Wiersz polecenia
pos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKeyAttribute" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\Product.cs(12,10): error CS0246: Nie można z
naleźć nazwy typu lub przestrzeni nazw "ForeignKey" (brak dyrektywy using lub odwołania do zestawu?) [C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
Ostrzeżenia: 0
Liczba błędów: 3

Czas, który upłynął: 00:00:01.06

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplier
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121104638_AddedSupplier'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierDivers
eRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierDivers
eRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121111512_AddedSupplierDiverseRelation'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

- aktualizacja schematu bazy danych

```
Wiersz polecenia
pos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts.csproj]
Ostrzeżenia: 0
Liczba błędów: 3

Czas, który upłynął: 00:00:01.06

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplier
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121104638_AddedSupplier'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierDivers
eRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121111512_AddedSupplierDiverseRelation'.
Done.

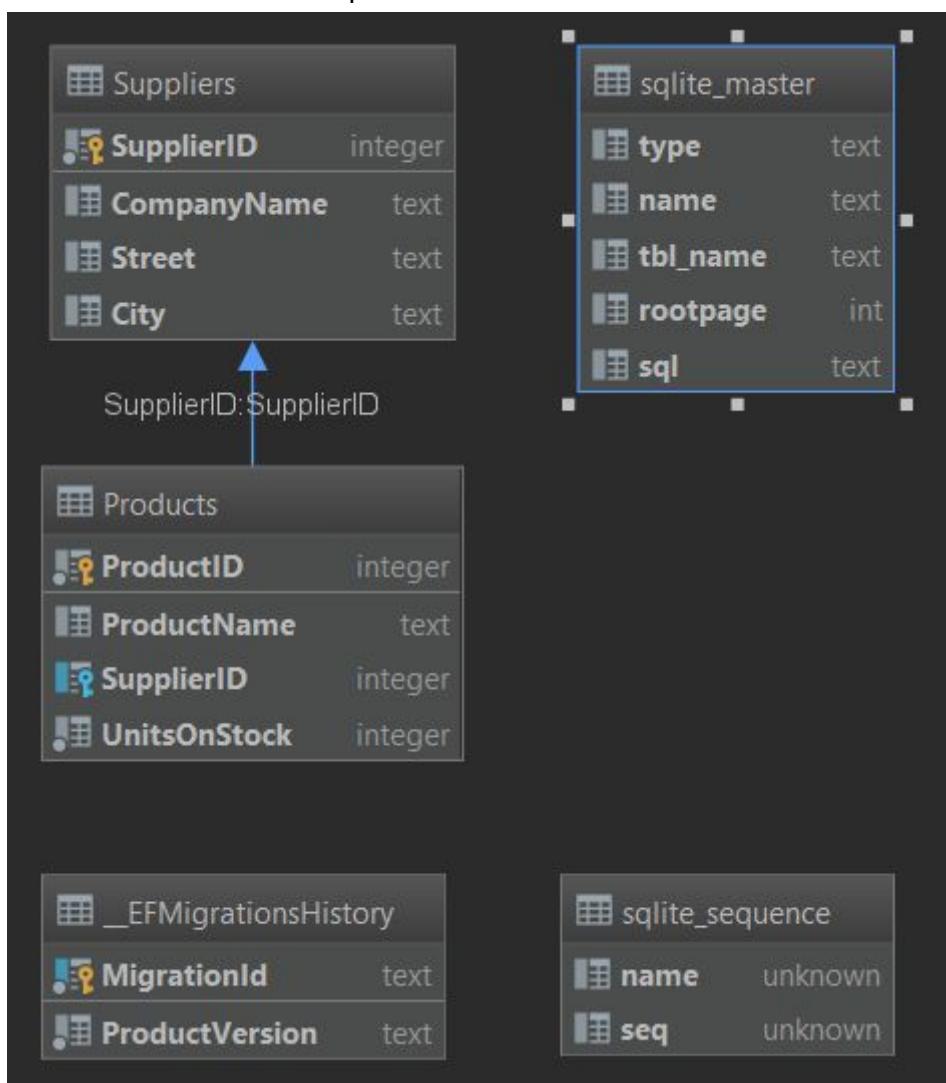
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

- odpalenie aplikacji

```
Konsola debugowania programu Microsoft Visual Studio
Podaj nazwę pierwszego produktu

```

- schemat w DataGrip



The screenshot shows a database management interface with two query panes and their corresponding results.

Top Query:

```
1 | select * from main.Products
```

Result:

	ProductID	ProductName	SupplierID	UnitsOnStock
1	1	pustak	1	2
2	2	cegla	1	3

Bottom Query:

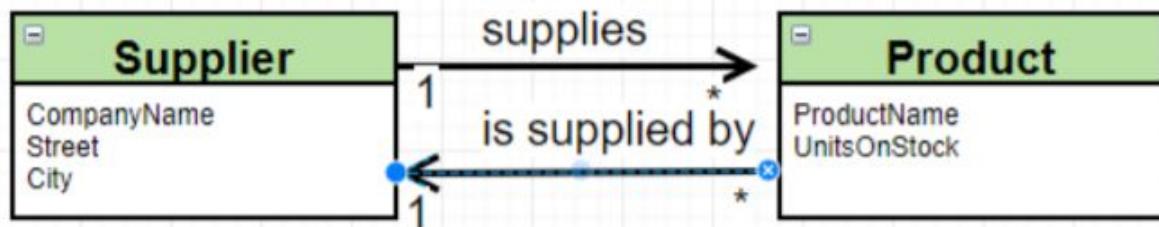
```
1 | select * from main.Suppliers
```

Result:

	SupplierID	CompanyName	Street	City
1	ABud	Długa	Kraków	

IV. Relacja dwustronna

Zamodeluj relacje dwustronną jak poniżej:



- modyfikacja klasy Products - odhashowanie pola Supplier

A screenshot of a code editor showing the `Product.cs` file. The code defines a class `Product` with properties `ProductID`, `ProductName`, and `UnitsOnStock`. It also contains a navigation property `Supplier` which is annotated with `[ForeignKey("SupplierID")]`. The code editor interface shows tabs for `Supplier.cs`, `ProductContext.cs`, `Product.cs*`, and `Program.cs`. The `Product.cs*` tab is active.

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations.Schema;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class Product
9      {
10         public int ProductID { get; set; }
11         public string ProductName { get; set; }
12         public int UnitsOnStock { get; set; }
13
14         [ForeignKey("SupplierID")]
15         public Supplier Supplier { get; set; }
16     }
17 }
```

- migracja

```
Wiersz polecenia
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplier
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121104638_AddedSupplier'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierDiverseRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121111512_AddedSupplierDiverseRelation'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierMutualRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

- aktualizacja bazy danych

```
Wiersz polecenia
Build started...
Build succeeded.
Applying migration '20201121104638_AddedSupplier'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierDiverseRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121111512_AddedSupplierDiverseRelation'.
Done.

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedSupplierMutualRelation
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201121112724_AddedSupplierMutualRelation'.
Done.

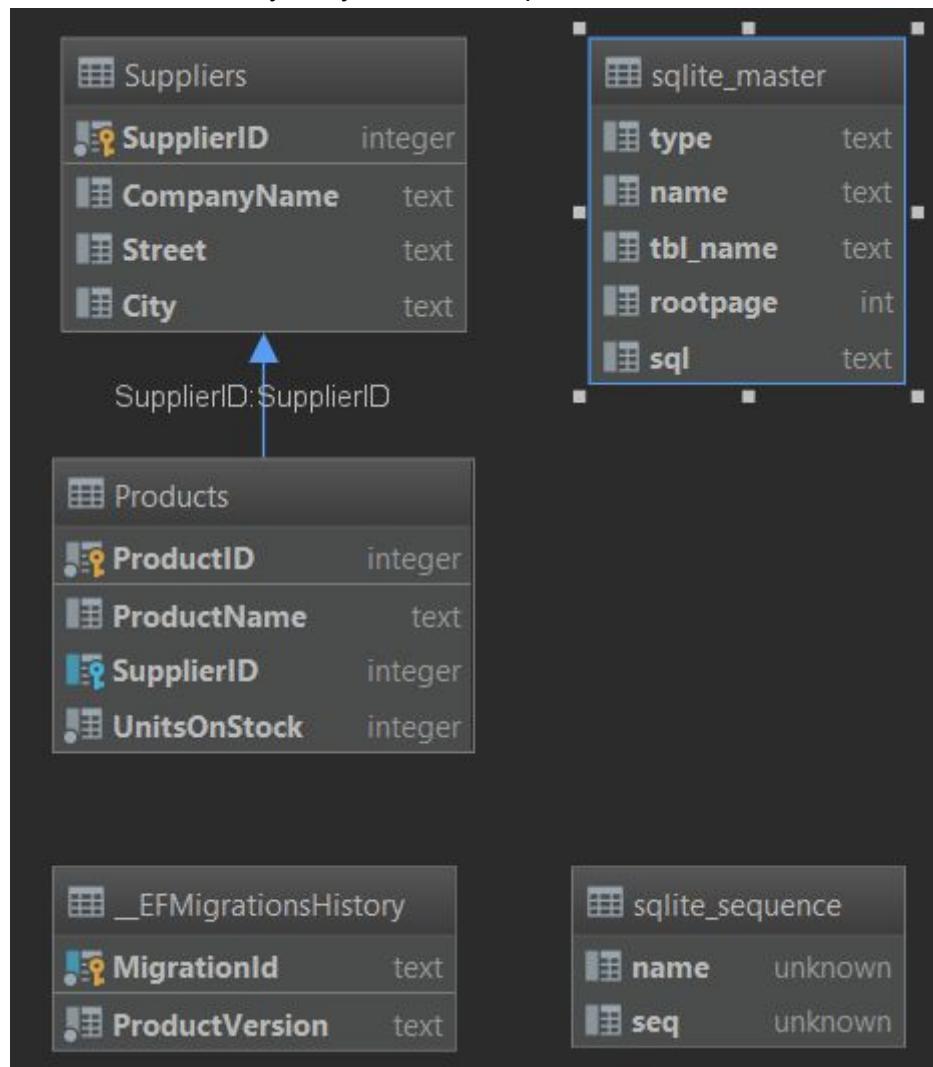
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>
```

- odpalenie aplikacji

```
Konsola debugowania programu Microsoft Visual Studio
Podaj nazwę pierwszego produktu
drabina
Podaj nazwę drugiego produktu
deskta tarasowa
Poniżej lista produktów zarejestrowanych w naszej bazie danych
drabina
deskta tarasowa
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
ABud

C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\bin\Debug\netcoreapp3.1\BartoszKordekEFProducts.exe (proces 16232) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

- schemat bazy danych w DataGrip



1 | select * from main.Products

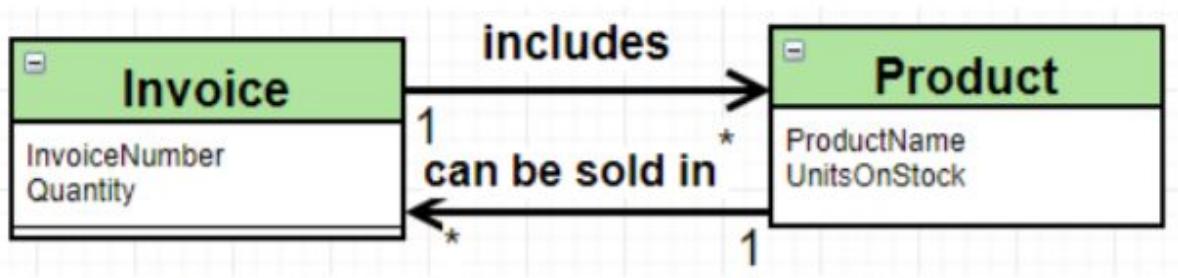
	ProductID	ProductName	SupplierID	UnitsOnStock
1	1	drabina	1	2
2	2	desk tarasowa	1	3

1 | select * from main.Suppliers

	SupplierID	CompanyName	Street	City
1	1	ABud	Długa	Kraków

V. Relacja wiele do wielu (Many 2 Many)

Zamodeluj relacje wiele-do-wielu, jak poniżej:

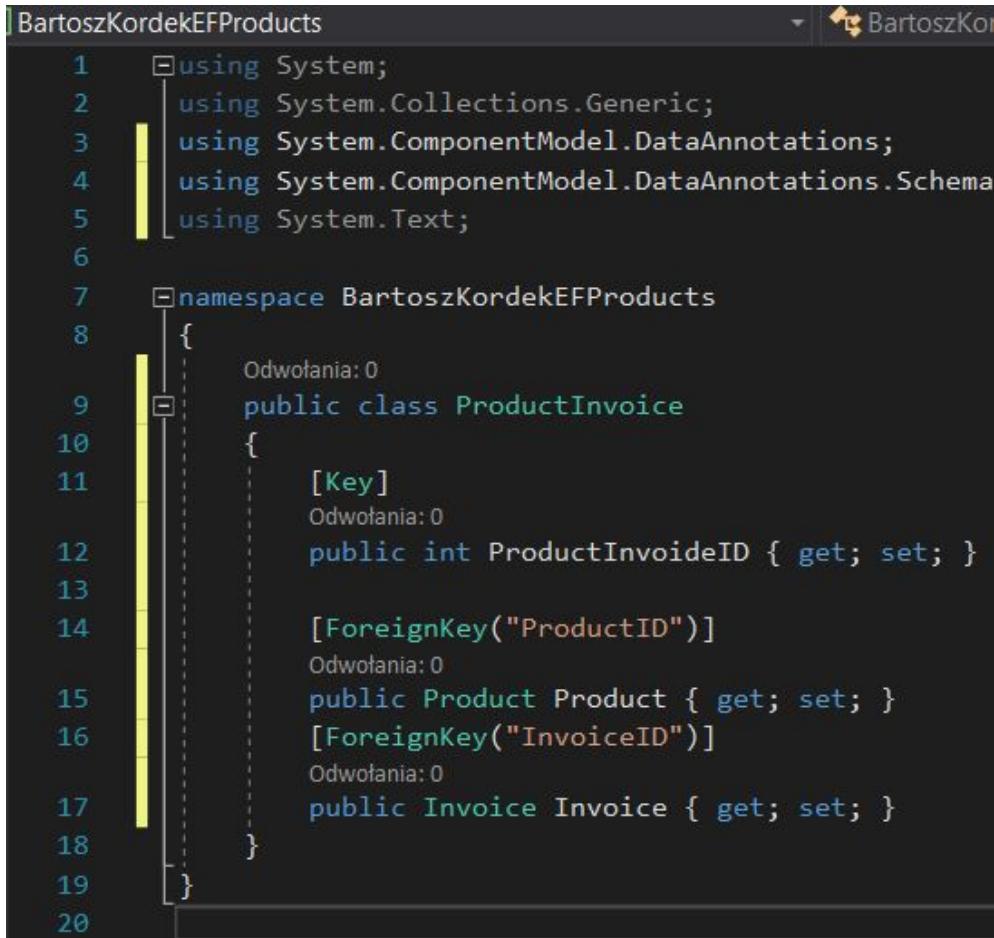


- stworzenie klasy *Invoice*

```

# BartoszKordekEFProducts
4
5     namespace BartoszKordekEFProducts
6     {
7         public class Invoice
8         {
9             public int InvoiceID { get; set; }
10            public int Quantity { get; set; }
11        }
12    }
13
14
  
```

- Pierwszy sposób (tylko zacząłem, bo drugi jest lepszy)
 - tworzę klasę *ProductInvoice* reprezentującą produkty i przypisane do nich faktury (faktury i przypisane do nich produkty)

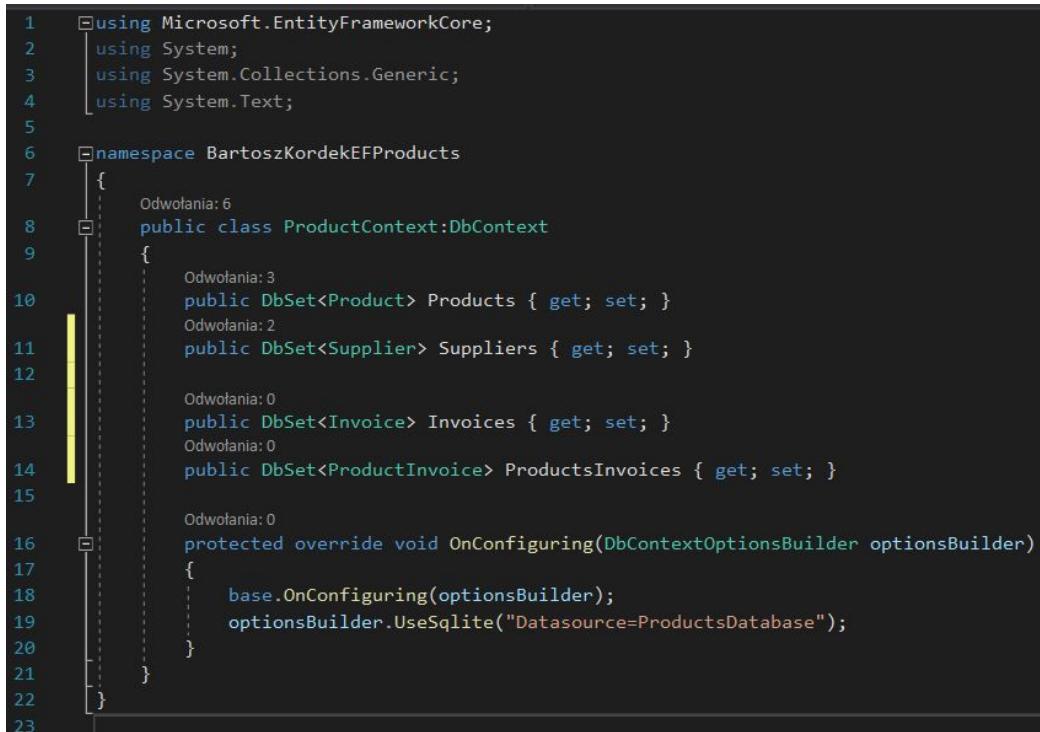


```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.ComponentModel.DataAnnotations.Schema;
5  using System.Text;
6
7  namespace BartoszKordekEFProducts
8  {
9      public class ProductInvoice
10     {
11         [Key]
12         public int ProductInvoiceID { get; set; }
13
14         [ForeignKey("ProductID")]
15         public Product Product { get; set; }
16         [ForeignKey("InvoiceID")]
17         public Invoice Invoice { get; set; }
18     }
19 }
20

```

- modyfikacja klasy *ProductContext*

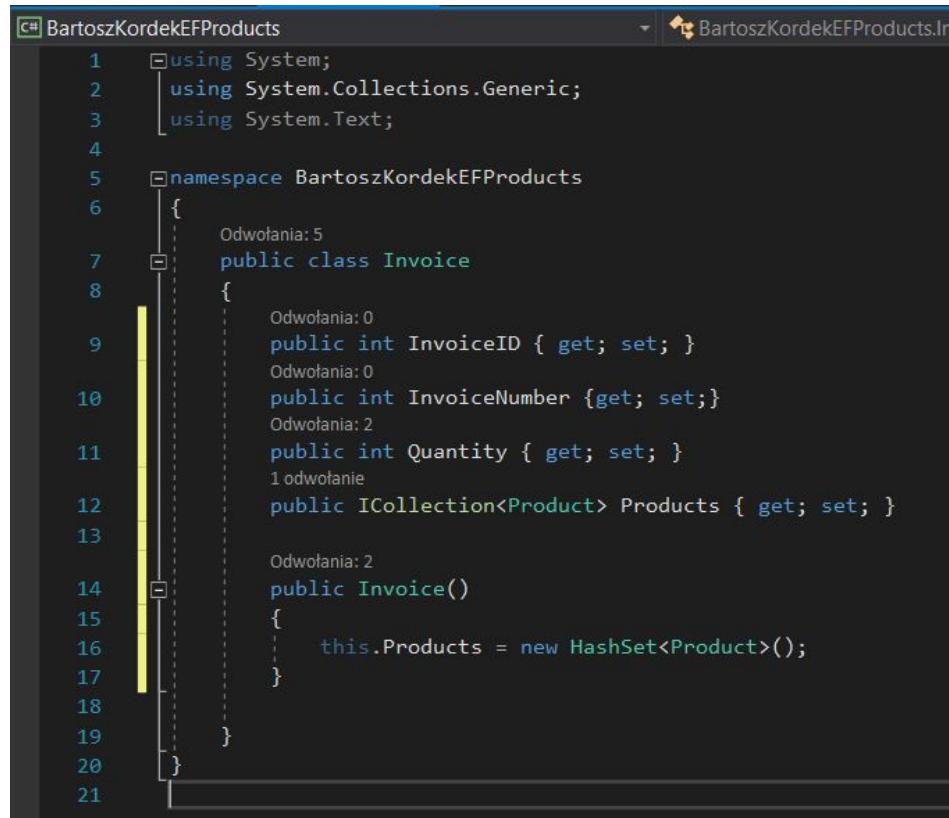


```

1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class ProductContext : DbContext
9      {
10         public DbSet<Product> Products { get; set; }
11         public DbSet<Supplier> Suppliers { get; set; }
12
13         public DbSet<Invoice> Invoices { get; set; }
14         public DbSet<ProductInvoice> ProductsInvoices { get; set; }
15
16         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
17         {
18             base.OnConfiguring(optionsBuilder);
19             optionsBuilder.UseSqlite("Datasource=ProductsDatabase");
20         }
21     }
22 }
23

```

- Drugi (lepszy sposób)
 - stworzenie klasy *Invoice*

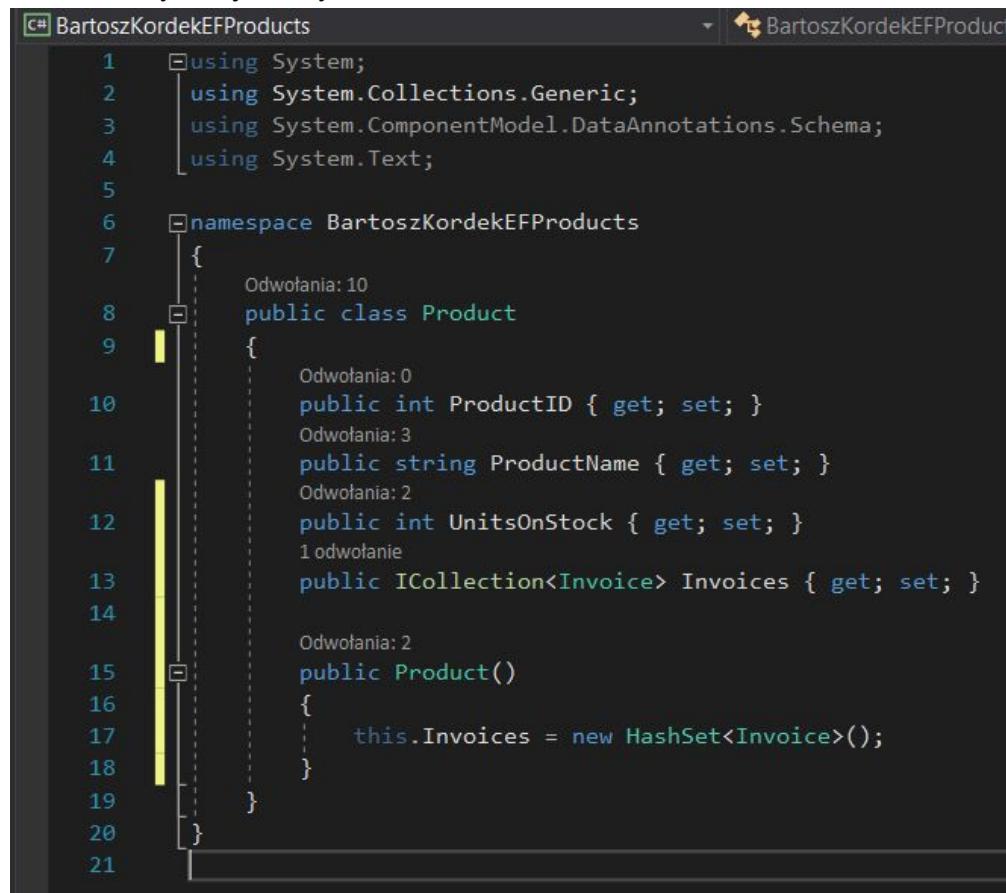


```

C# BartoszKordekEFProducts
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordekEFProducts
6  {
7      public class Invoice
8      {
9          public int InvoiceID { get; set; }
10         public int InvoiceNumber { get; set; }
11         public int Quantity { get; set; }
12         public ICollection<Product> Products { get; set; }
13
14         public Invoice()
15         {
16             this.Products = new HashSet<Product>();
17         }
18     }
19 }
20
21
  
```

The screenshot shows a code editor with a dark theme. It displays the `Invoice` class within the `BartoszKordekEFProducts` namespace. The class has four properties: `InvoiceID`, `InvoiceNumber`, `Quantity`, and `Products`. The `Products` property is of type `ICollection<Product>`. A constructor is also provided, initializing the `Products` collection to a new `HashSet<Product>`.

- modyfikacja klasy *Products*



```

C# BartoszKordekEFProducts
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations.Schema;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class Product
9      {
10         public int ProductID { get; set; }
11         public string ProductName { get; set; }
12         public int UnitsOnStock { get; set; }
13         public ICollection<Invoice> Invoices { get; set; }
14
15         public Product()
16         {
17             this.Invoices = new HashSet<Invoice>();
18         }
19     }
20 }
21
  
```

The screenshot shows a code editor with a dark theme. It displays the `Product` class within the `BartoszKordekEFProducts` namespace. The class has three properties: `ProductID`, `ProductName`, and `UnitsOnStock`. A many-to-many relationship is established through the `Invoices` property, which is of type `ICollection<Invoice>`. A constructor initializes the `Invoices` collection to a new `HashSet<Invoice>`.

- modyfikacja klasy *ProductContext*

```

1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordekEFProducts
7  {
8      public class ProductContext:DbContext
9      {
10         public DbSet<Product> Products { get; set; }
11         public DbSet<Supplier> Suppliers { get; set; }
12
13         public DbSet<Invoice> Invoices { get; set; }
14
15         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
16         {
17             base.OnConfiguring(optionsBuilder);
18             optionsBuilder.UseSqlite("Datasource=ProductsDatabase");
19         }
20     }
21 }

```

- modyfikacja klasy sterującej

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4
5  namespace BartoszKordekEFProducts
6  {
7      class Program
8      {
9          static void Main(string[] args)
10         {
11             ProductContext productContext = new ProductContext();
12
13             Product product1 = new Product { ProductName = "Cegła", UnitsOnStock = 2 };
14             Product product2 = new Product { ProductName = "Pustak", UnitsOnStock = 3 };
15             Product product3 = new Product { ProductName = "Deska dębową dł. 2m", UnitsOnStock = 5 };
16             Product product4 = new Product { ProductName = "Biała emulsja", UnitsOnStock = 1 };
17
18             List<Product> productsForSupplier1 = new List<Product>();
19             productsForSupplier1.Add(product1);
20             productsForSupplier1.Add(product2);
21             Supplier supplier1 = new Supplier
22             {
23                 CompanyName = "ABud",
24                 City = "Kraków",
25                 Street = "Długa",
26                 Products = productsForSupplier1
27             };
28
29         }
29     }

```

```

29         List<Product> productsForSupplier2 = new List<Product>();
30         productsForSupplier2.Add(product3);
31         productsForSupplier2.Add(product4);
32         Supplier supplier2 = new Supplier
33         {
34             CompanyName = "Majster",
35             City = "Wrocław",
36             Street = "Szeroka",
37             Products = productsForSupplier2
38         };
39
40         List<Product> productsForInvoice1 = new List<Product>();
41         productsForInvoice1.Add(product1);
42         productsForInvoice1.Add(product2);
43         productsForInvoice1.Add(product3);
44         Invoice invoice1 = new Invoice {InvoiceNumber = 56664, Quantity = 1, Products = productsForInvoice1 };
45
46         List<Product> productsForInvoice2 = new List<Product>();
47         productsForInvoice2.Add(product2);
48         productsForInvoice2.Add(product3);
49         productsForInvoice2.Add(product4);
50         Invoice invoice2 = new Invoice { InvoiceNumber = 57885, Quantity = 2 , Products = productsForInvoice2 };
51
52
53
54         productContext.Suppliers.Add(supplier1);
55         productContext.Suppliers.Add(supplier2);
56         productContext.Products.Add(product1);
57         productContext.Products.Add(product2);
58         productContext.Invoices.Add(invoice1);
59         productContext.Invoices.Add(invoice2);
60         productContext.SaveChanges();
61
62
63         Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
64         var query = from prod in productContext.Products select prod.ProductName;
65         foreach (var pName in query)
66         {
67             Console.WriteLine(pName);
68         }
69
70         Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych");
71         foreach (Supplier c in productContext.Suppliers)
72         {
73             Console.WriteLine(c.CompanyName);
74         }
75
76         Console.WriteLine("Poniżej lista faktur w naszej bazie danych");
77         foreach (Invoice c in productContext.Invoices)
78         {
79             Console.WriteLine(c.InvoiceNumber);
80         }
81     }
82 }
83
84

```

- utworzenie migracji

```
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef migrations add AddedInvoiceMany2Many
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
```

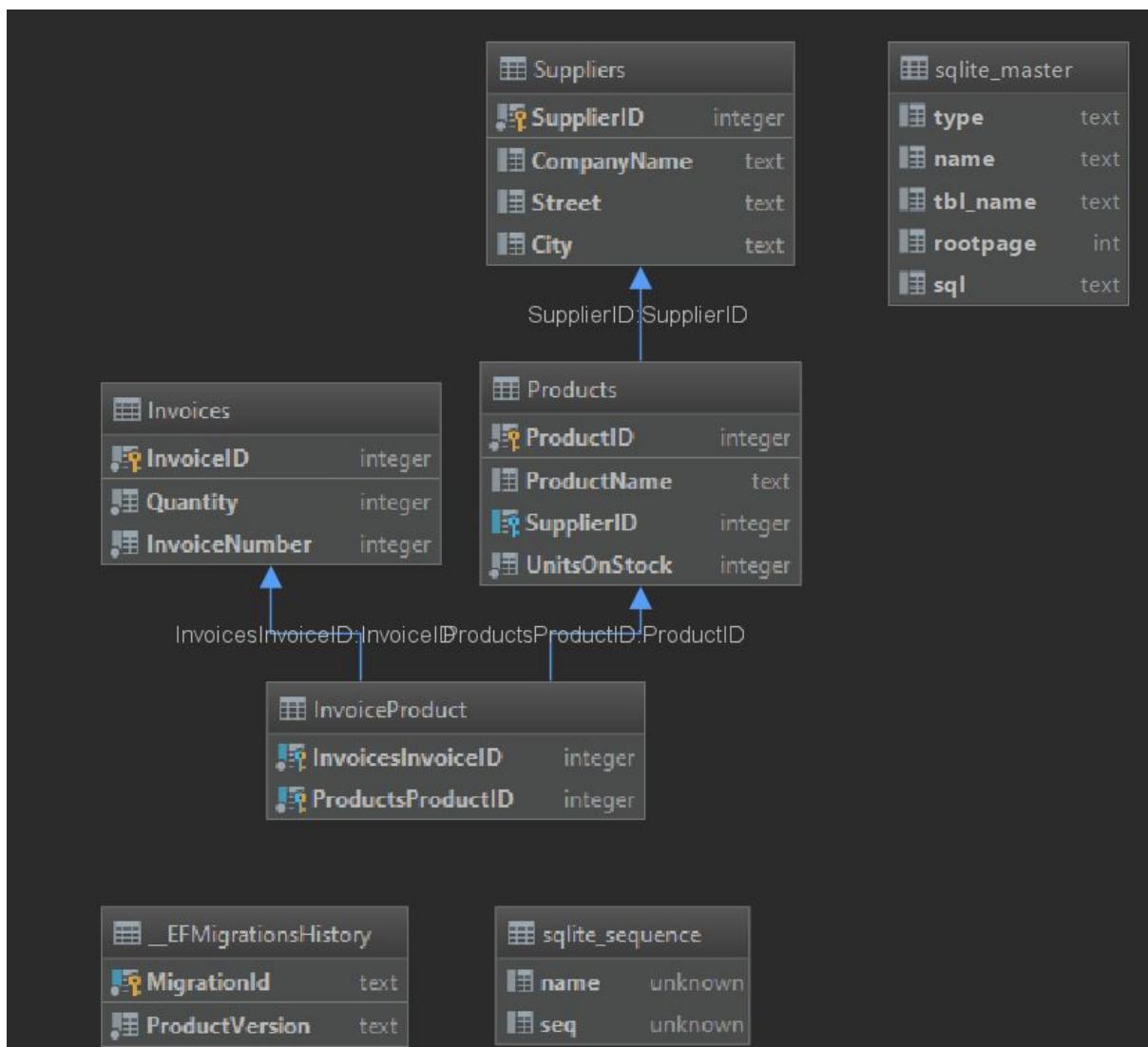
- zaktualizowanie bazy danych

```
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201130230235_AddedInvoiceMany2Many'.
Done.
```

- odpalenie aplikacji

Konsola debugowania programu Microsoft Visual Studio
Poniżej lista produktów zarejestrowanych w naszej bazie danych
Cegl
Pustak
Deska dębową dł. 2m
Biała emulsja
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
ABud
Majster
Poniżej lista faktur w naszej bazie danych
56664
57885
C:\Users\barto\source\repos\BartoszKordekEFProducts\BartoszKordekEFProducts\bin\Debug\netcoreapp3.1\BartoszKordekEFProducts.exe (proces 3724) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

- schemat bazy danych w DataGrip



ProductsDatabase [console] × main ×

Tx: Auto ✓ ↻

```
1 | select * from main.Products
```

Output main.Products ×

ProductID ProductName UnitsOnStock SupplierID

ProductID	ProductName	UnitsOnStock	SupplierID
1	Cegła	2	1
2	Pustak	3	1
3	Deska dębową dł. 2m	5	2
4	Biała emulsja	1	2

ProductsDatabase [console] × main ×

Tx: Auto ✓ ↻

```
1 | select * from main.Suppliers
```

Database Console: ProductsDatabase [console] ×

Output main.Suppliers ×

SupplierID CompanyName Street City

SupplierID	CompanyName	Street	City
1	ABud	Długa	Kraków
2	Majster	Szeroka	Wrocław

ProductsDatabase [console] × main ×

Tx: Auto ✓ ↻

```
1 | select * from main.Invoices
```

Output main.Invoices ×

InvoicID InvoiceNumber Quantity

InvoicID	InvoiceNumber	Quantity
1	56664	1
2	57885	2

The screenshot shows a database interface with two panes. The top pane contains a SQL query:

```
1 | select * from main.InvoiceProduct
```

The bottom pane displays the results of the query as a table:

	InvoicesInvoiceID	ProductsProductID
1	1	1
2	1	2
3	2	2
4	1	3
5	2	3
6	2	4

- modyfikacja kodu w klasie sterującej - *Program*, żeby można było wypisać Faktury i powiązane z nią Produkty oraz Produkty i powiązane z nimi faktury

```
int chosenInvoiceId = 1;

var chosenInvoice = productContext.Invoices
    .Include(i => i.Products)
    .Where(i => i.InvoiceID == chosenInvoiceId);

Console.WriteLine("Poniżej lista produktów sprzedanych w ramach faktury o ID: " + chosenInvoiceId);

foreach (var c in chosenInvoice)
{
    Console.WriteLine("Numer faktury: " + c.InvoiceNumber);
    foreach (Product p in c.Products)
    {
        Console.WriteLine(" Product ID: " + p.ProductID + " Product Name: " + p.ProductName);
    }
}

Console.WriteLine("Poniżej lista faktur i produktów sprzedanych wraz z nimi");
foreach (Invoice i in productContext.Invoices)
{
    Console.Write("Invoice ID: " + i.InvoiceID + " Invoice Number: " + i.InvoiceNumber + " ");
    foreach (Product p in i.Products)
    {
        Console.Write(" Product ID: " + p.ProductID + " Product Name: " + p.ProductName);
    }
    Console.WriteLine("");
}
```

```

int chosenProductId = 1;

var chosenProduct = productContext.Products
    .Include(i => i.Invoices)
    .Where(i => i.ProductID == chosenProductId);

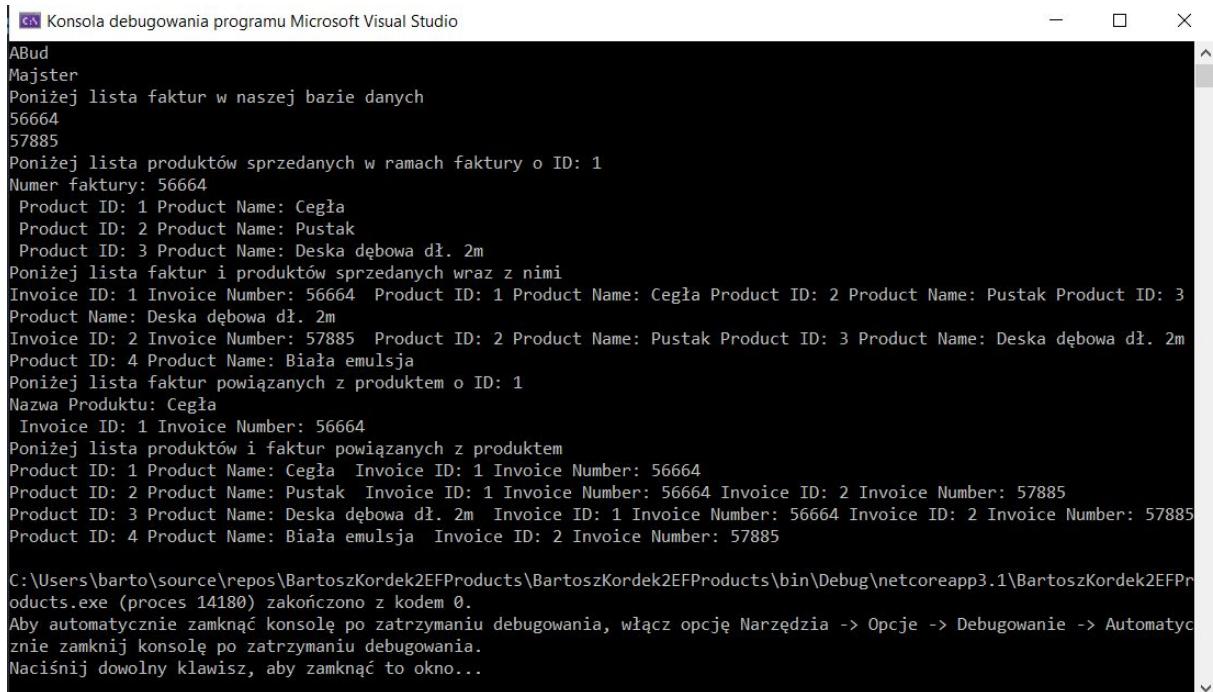
Console.WriteLine("Poniżej lista faktur powiązanych z produktem o ID: " + chosenInvoiceId);

foreach (var c in chosenProduct)
{
    Console.WriteLine("Nazwa Produktu: " + c.ProductName);
    foreach (Invoice i in c.Invoices)
    {
        Console.WriteLine(" Invoice ID: " + i.InvoiceID + " Invoice Number: " + i.InvoiceNumber);
    }
}

Console.WriteLine("Poniżej lista produktów i faktur powiązanych z produktem");
foreach (Product p in productContext.Products)
{
    Console.Write("Product ID: " + p.ProductID + " Product Name: " + p.ProductName + " ");
    foreach (Invoice i in p.Invoices)
    {
        Console.Write(" Invoice ID: " + i.InvoiceID + " Invoice Number: " + i.InvoiceNumber);
    }
    Console.WriteLine("");
}

```

- odpalenie aplikacji



The screenshot shows the Microsoft Visual Studio Command Prompt window. The title bar reads "Konsola debugowania programu Microsoft Visual Studio". The window displays the following text output:

```

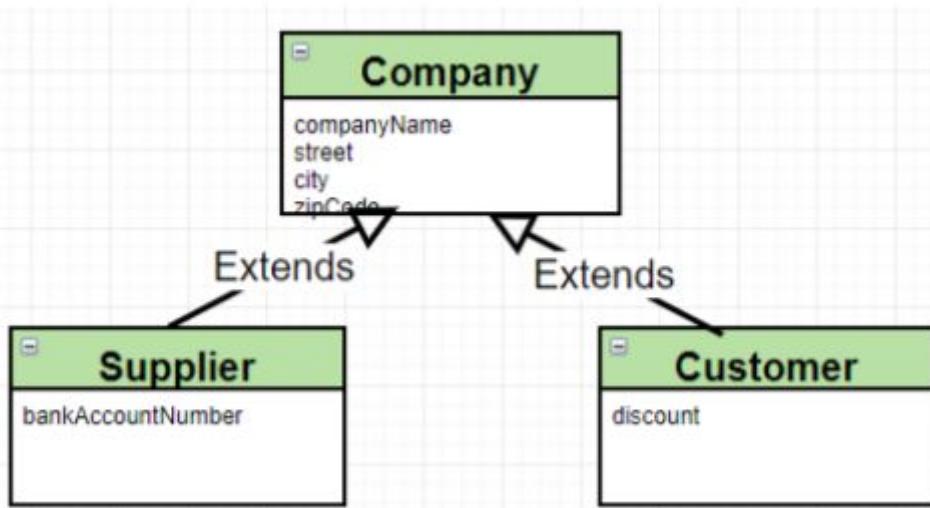
ABud
Majster
Poniżej lista faktur w naszej bazie danych
56664
57885
Poniżej lista produktów sprzedanych w ramach faktury o ID: 1
Numer faktury: 56664
Product ID: 1 Product Name: Cegła
Product ID: 2 Product Name: Pustak
Product ID: 3 Product Name: Deska dębową dł. 2m
Poniżej lista faktur i produktów sprzedanych wraz z nimi
Invoice ID: 1 Invoice Number: 56664 Product ID: 1 Product Name: Cegła Product ID: 2 Product Name: Pustak Product ID: 3
Product Name: Deska dębową dł. 2m
Invoice ID: 2 Invoice Number: 57885 Product ID: 2 Product Name: Pustak Product ID: 3 Product Name: Deska dębową dł. 2m
Product ID: 4 Product Name: Biała emulsja
Poniżej lista faktur powiązanych z produktem o ID: 1
Nazwa Produktu: Cegła
Invoice ID: 1 Invoice Number: 56664
Poniżej lista produktów i faktur powiązanych z produktem
Product ID: 1 Product Name: Cegła Invoice ID: 1 Invoice Number: 56664
Product ID: 2 Product Name: Pustak Invoice ID: 1 Invoice Number: 56664 Invoice ID: 2 Invoice Number: 57885
Product ID: 3 Product Name: Deska dębową dł. 2m Invoice ID: 1 Invoice Number: 56664 Invoice ID: 2 Invoice Number: 57885
Product ID: 4 Product Name: Biała emulsja Invoice ID: 2 Invoice Number: 57885

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts\bin\Debug\netcoreapp3.1\BartoszKordek2EFProducts.exe (proces 14180) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

VI. Dziedziczenie - Table per Hierarchy

Wprowadź do modelu poniższą hierarchię dziedziczenia używając strategii Table-Per-Hierarchy:



- utworzenie klasy *Company*

```
Company.cs  ✎ ✖ Invoice.cs          Product.cs          ProductContext.cs
c# BartoszKordek2EFProducts
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordek2EFProducts
6  {
7      public class Company
8      {
9          public int CompanyID { get; set; }
10         public string CompanyName { get; set; }
11         public string Street { get; set; }
12         public string City { get; set; }
13         public string ZipCode { get; set; }
14     }
15 }
```

- utworzenie klasy *Customer* dziedziczącej po klasie *Company*

Customer.cs* Company.cs

BartoszKordek2EFProducts

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordek2EFProducts
6  {
7      class Customer:Company
8      {
9          public int CustomerID { get; set; }
10         public double Discount { get; set; }
11     }
12 }
13

```

- modyfikacja klasy *Supplier*

Program.cs ProductContext.cs Supplier.cs* Customer.cs Company.cs

BartoszKordek2EFProducts

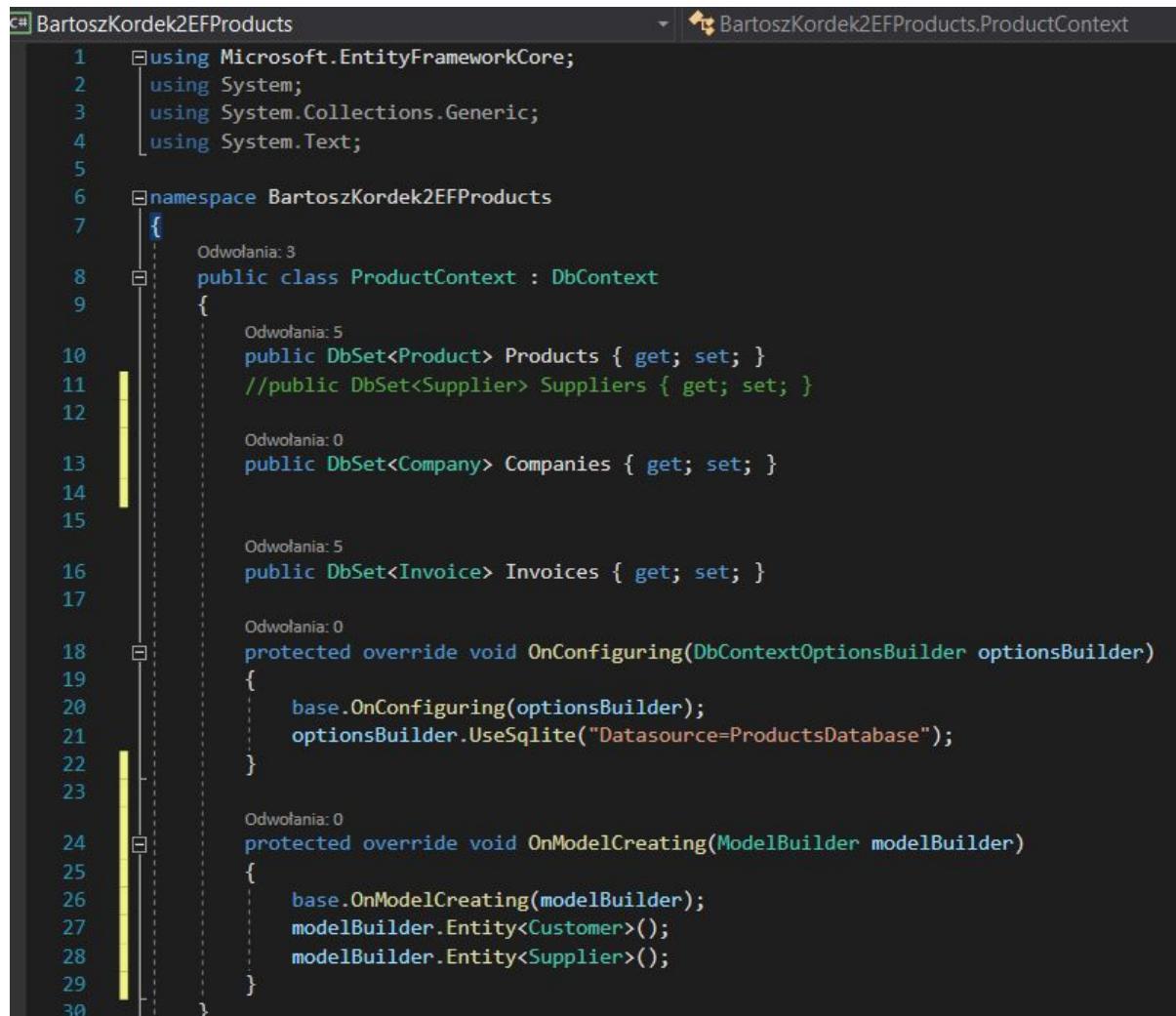
```

4
5  namespace BartoszKordek2EFProducts
6  {
7      public class Supplier:Company
8      {
9          public int SupplierID { get; set; }
10         public String BankAccountNumber { get; set; }
11         public string CompanyName { get; set; }
12         public string Street { get; set; }
13         public string City { get; set; }
14         public ICollection<Product> Products { get; set; }
15     }
16 }
17
18
19

```

- modyfikacja klasy *ProductContext*

Nadpisanie metody *OnModelCreating*, dodanie zbioru danych *Companies*
zahashowanie zbioru *Suppliers*.



```
C# BartoszKordek2EFProducts BartoszKordek2EFProducts.ProductContext
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5
6  namespace BartoszKordek2EFProducts
7  {
8      public class ProductContext : DbContext
9      {
10         public DbSet<Product> Products { get; set; }
11         //public DbSet<Supplier> Suppliers { get; set; }
12
13         public DbSet<Company> Companies { get; set; }
14
15
16         public DbSet<Invoice> Invoices { get; set; }
17
18         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
19         {
20             base.OnConfiguring(optionsBuilder);
21             optionsBuilder.UseSqlite("Datasource=ProductsDatabase");
22         }
23
24         protected override void OnModelCreating(ModelBuilder modelBuilder)
25         {
26             base.OnModelCreating(modelBuilder);
27             modelBuilder.Entity<Customer>();
28             modelBuilder.Entity<Supplier>();
29         }
30     }
}
```

- modyfikacja klasy sterującej *Program*

The screenshot shows a code editor window with the following details:

- Title Bar:** BartoszKordek2EFPProducts
- Tab Bar:** BartoszKordek2EFPProducts.Program
- Code Area:** Main(string[] args)
- Code Content:**

```
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5
6  namespace BartoszKordek2EFPProducts
7  {
8      class Program
9      {
10         static void Main(string[] args)
11         {
12             ProductContext productContext = new ProductContext();
13
14             Product product1 = new Product { ProductName = "Cegła", UnitsOnStock = 2 };
15             Product product2 = new Product { ProductName = "Pustak", UnitsOnStock = 3 };
16             Product product3 = new Product { ProductName = "Deska dębową dł. 2m", UnitsOnStock = 5 };
17             Product product4 = new Product { ProductName = "Biała emulsja", UnitsOnStock = 1 };
18
19             List<Product> productsForSupplier1 = new List<Product>();
20             productsForSupplier1.Add(product1);
21             productsForSupplier1.Add(product2);
22             Supplier supplier1 = new Supplier
23             {
24                 CompanyName = "BudB",
25                 Street = "Wodna",
26                 City = "Warszawa",
27                 ZipCode = "00-223",
28                 BankAccountNumber = "1231231230555444",
29                 Products = productsForSupplier1
30             };
31 }
```

```
C# BartoszKordek2EFProducts
  31
  32
  33     List<Product> productsForSupplier2 = new List<Product>();
  34     productsForSupplier2.Add(product3);
  35     productsForSupplier2.Add(product4);
  36     Supplier supplier2 = new Supplier
  37     {
  38         CompanyName = "Mrówka",
  39         Street = "Cienista",
  40         City = "Rzeszów",
  41         ZipCode = "36-778",
  42         BankAccountNumber = "3211231230555444",
  43         Products = productsForSupplier2
  44     };
  45
  46     Customer customer1 = new Customer
  47     {
  48         CompanyName = "Januszex",
  49         Street = "Rajska",
  50         City = "Kraków",
  51         ZipCode = "30-201",
  52         Discount = 20
  53     };
  54
  55     Customer customer2 = new Customer
  56     {
  57         CompanyName = "DomBud",
  58         Street = "Kręta",
  59         City = "Warszawa",
  60         ZipCode = "00-051",
  61         Discount = 10
  62     };
  63
```

```
 63
 64     List<Product> productsForInvoice1 = new List<Product>();
 65     productsForInvoice1.Add(product1);
 66     productsForInvoice1.Add(product2);
 67     productsForInvoice1.Add(product3);
 68     Invoice invoice1 = new Invoice { InvoiceNumber = 56664, Quantity = 1, Products = productsForInvoice1 };

 69
 70     List<Product> productsForInvoice2 = new List<Product>();
 71     productsForInvoice2.Add(product2);
 72     productsForInvoice2.Add(product3);
 73     productsForInvoice2.Add(product4);
 74     Invoice invoice2 = new Invoice { InvoiceNumber = 57885, Quantity = 2, Products = productsForInvoice2 };

 75
 76     /*
 77     productContext.Suppliers.Add(supplier1);
 78     productContext.Suppliers.Add(supplier2);
 79     */
 80
 81     productContext.Companies.Add(supplier1);
 82     productContext.Companies.Add(supplier2);
 83     productContext.Companies.Add(customer1);
 84     productContext.Companies.Add(customer2);
 85     productContext.Products.Add(product1);
 86     productContext.Products.Add(product2);
 87     productContext.Invoices.Add(invoice1);
 88     productContext.Invoices.Add(invoice2);
 89     productContext.SaveChanges();
 90
```

```

91
92         Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
93         var query = from prod in productContext.Products select prod.ProductName;
94         foreach (var pName in query)
95         {
96             Console.WriteLine(pName);
97         }
98
99         Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych");
100        foreach (Supplier c in productContext.Companies.OfType<Supplier>().Select(s => s))
101        {
102            Console.WriteLine(c.CompanyName + " " + c.BankAccountNumber);
103        }
104
105        Console.WriteLine("Poniżej lista klientów zarejestrowanych w naszej bazie danych");
106        foreach (Customer c in productContext.Companies.OfType<Customer>().Select(s => s))
107        {
108            Console.WriteLine(c.CompanyName + " " + c.Discount);
109        }

```

- utworzenie kodu migracji i zaktualizowanie bazy danych

```

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts>dotnet ef migrations add InheritanceTablePerHierarchyVer3
Build started...
Build succeeded.
An operation was scaffolded that may result in the loss of data. Please review the migration for accuracy.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201201125748_InheritanceTablePerHierarchyVer3'.
Done.

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts>

```

- Wynik

Konsola debugowania programu Microsoft Visual Studio

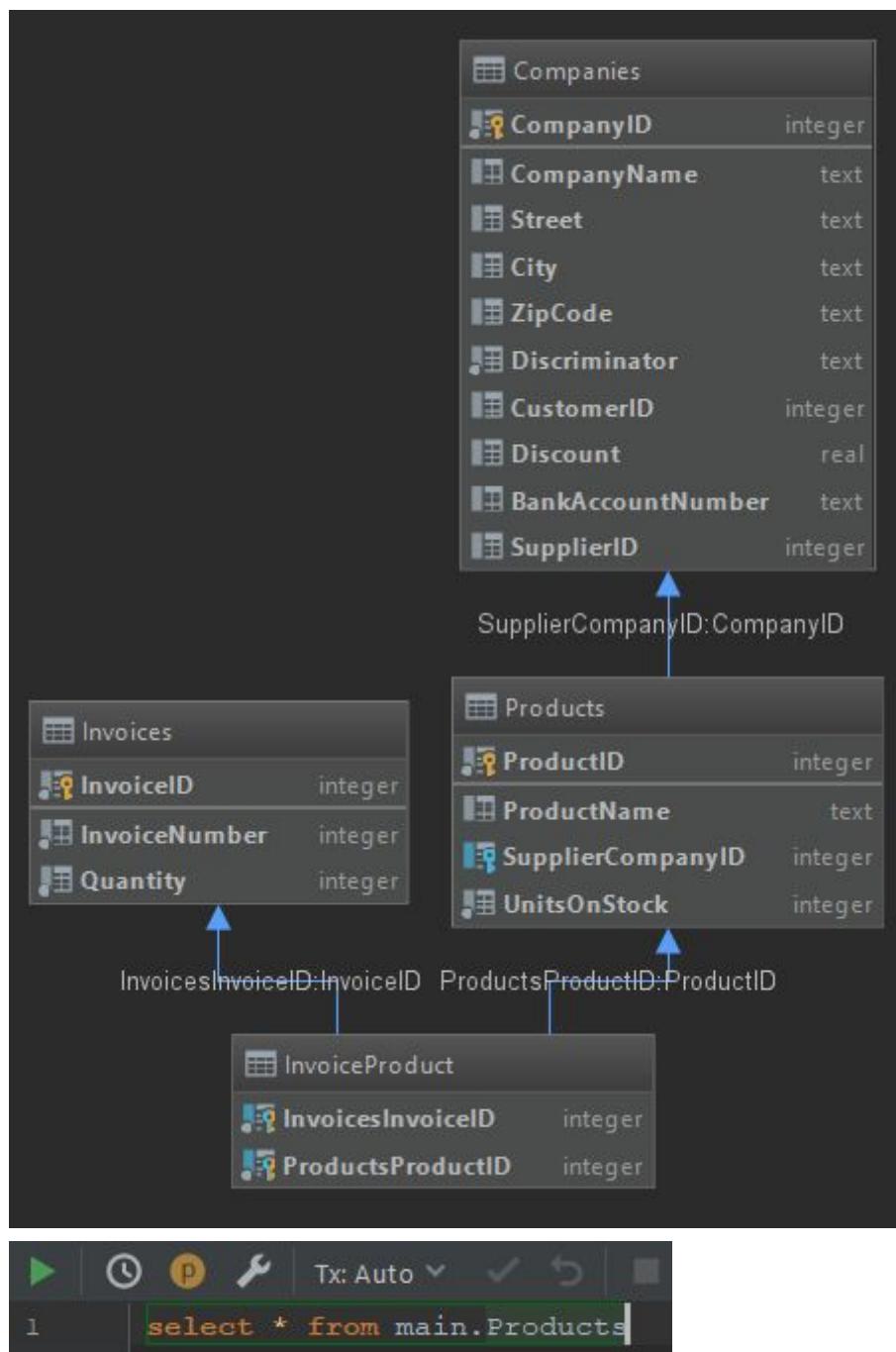
```

Poniżej lista produktów zarejestrowanych w naszej bazie danych
Cegła
Pustak
Deska dębową dł. 2m
Biała emulsja
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
BudB 1231231230555444
Mrówka 3211231230555444
Poniżej lista klientów zarejestrowanych w naszej bazie danych
Januszex 20
DomBud 10

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts\bin\Debug\netcoreapp3.1\BartoszKordek2EFProducts.exe (proces 12100) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

- schemat bazy danych w DataGrip



Tx: Auto ✓ ↻

```
1 | select * from main.Products
```

Output main.Products

	ProductID	ProductName	SupplierCompanyID	UnitsOnStock
1	1	Cegła	1	2
2	2	Pustak	1	3
3	3	Deska dębową dł. 2m	2	5
4	4	Biała emulsja	2	1

Tx: Auto ✓ ↻

```
1 | select * from main.Companies
```

Output main.Companies

	CompanyID	CompanyName	Street	City	ZipCode	Discriminator	CustomerID	Discount	BankAccountNumber	SupplierID
1	1	BudB	Wodna	Warszawa	00-223	Supplier	<null>	<null>	1231231230555444	0
2	2	Mrówka	Cienista	Rzeszów	36-778	Supplier	<null>	<null>	3211231230555444	0
3	3	Januszex	Rajska	Kraków	30-201	Customer	0	20	<null>	<null>
4	4	DomBud	Kręta	Warszawa	00-051	Customer	0	10	<null>	<null>

Tx: Auto

```
1 | select * from main.Invoices
```

⌚ 2 rows Tx: Auto

	InvoicelD	InvoiceNumber	Quantity
1	1	56664	1
2	2	57885	2

Tx: Auto

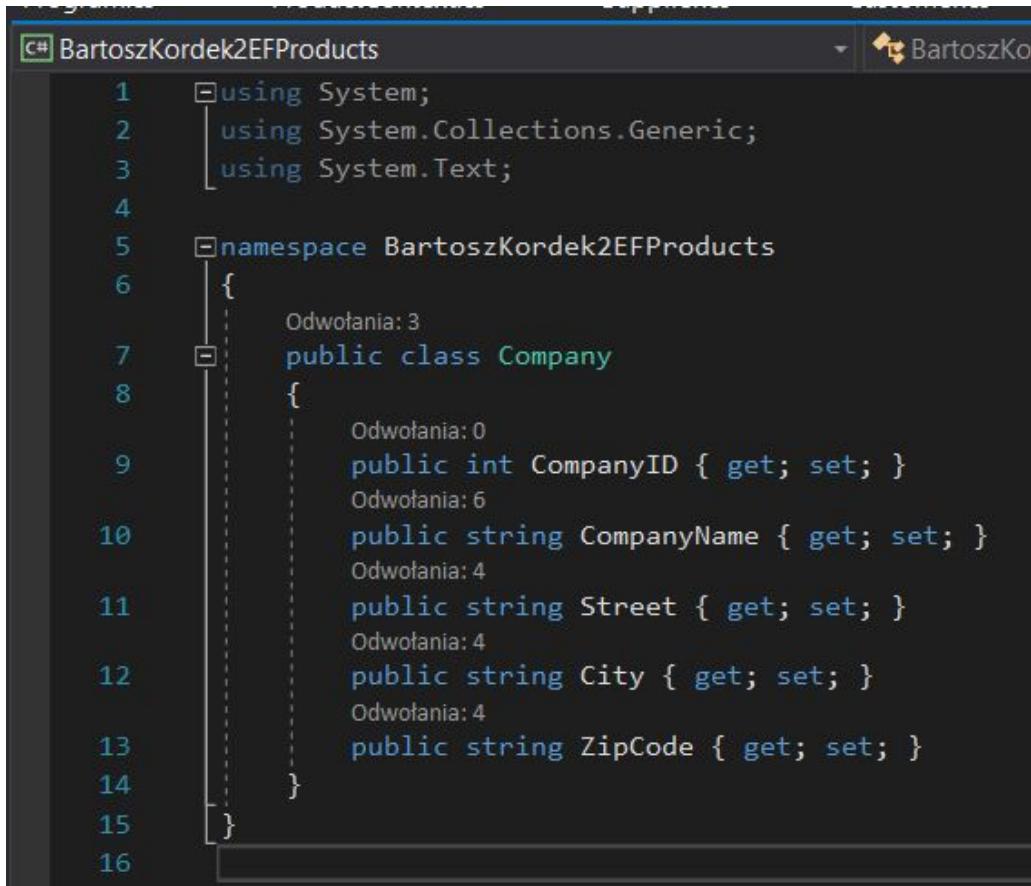
```
1 | select * from main.InvoiceProduct
```

⌚ 6 rows Tx: Auto

	InvoicesInvoicelD	ProductsProductID
1	1	1
2	1	2
3	2	2
4	1	3
5	2	3
6	2	4

VII. Dziedziczenie - Table per table

- klasa *Company* wygląda tak samo jak w przypadku dziedziczenia Table per hierarchy



The screenshot shows a code editor in Visual Studio with the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordek2EFProducts
6  {
7      public class Company
8      {
9          public int CompanyID { get; set; }
10         public string CompanyName { get; set; }
11         public string Street { get; set; }
12         public string City { get; set; }
13         public string ZipCode { get; set; }
14     }
15 }
16
```

The code defines a class named *Company* with four properties: *CompanyID*, *CompanyName*, *Street*, and *ZipCode*. The *CompanyName* and *Street* properties have annotations indicating they are referenced 3 and 6 times respectively. The *City* and *ZipCode* properties also have annotations indicating they are referenced 4 times each.

- klasa *Customer* bez zmian

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordek2EFProducts
6  {
7      class Customer:Company
8      {
9          public int CustomerID { get; set; }
10         public double Discount { get; set; }
11     }
12 }
13
```

- klasa *Supplier* również bez zmian

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace BartoszKordek2EFProducts
6  {
7      public class Supplier:Company
8      {
9          public int SupplierID { get; set; }
10         public String BankAccountNumber { get; set; }
11         public string CompanyName { get; set; }
12         public string Street { get; set; }
13         public string City { get; set; }
14         public ICollection<Product> Products { get; set; }
15     }
16 }
17
```

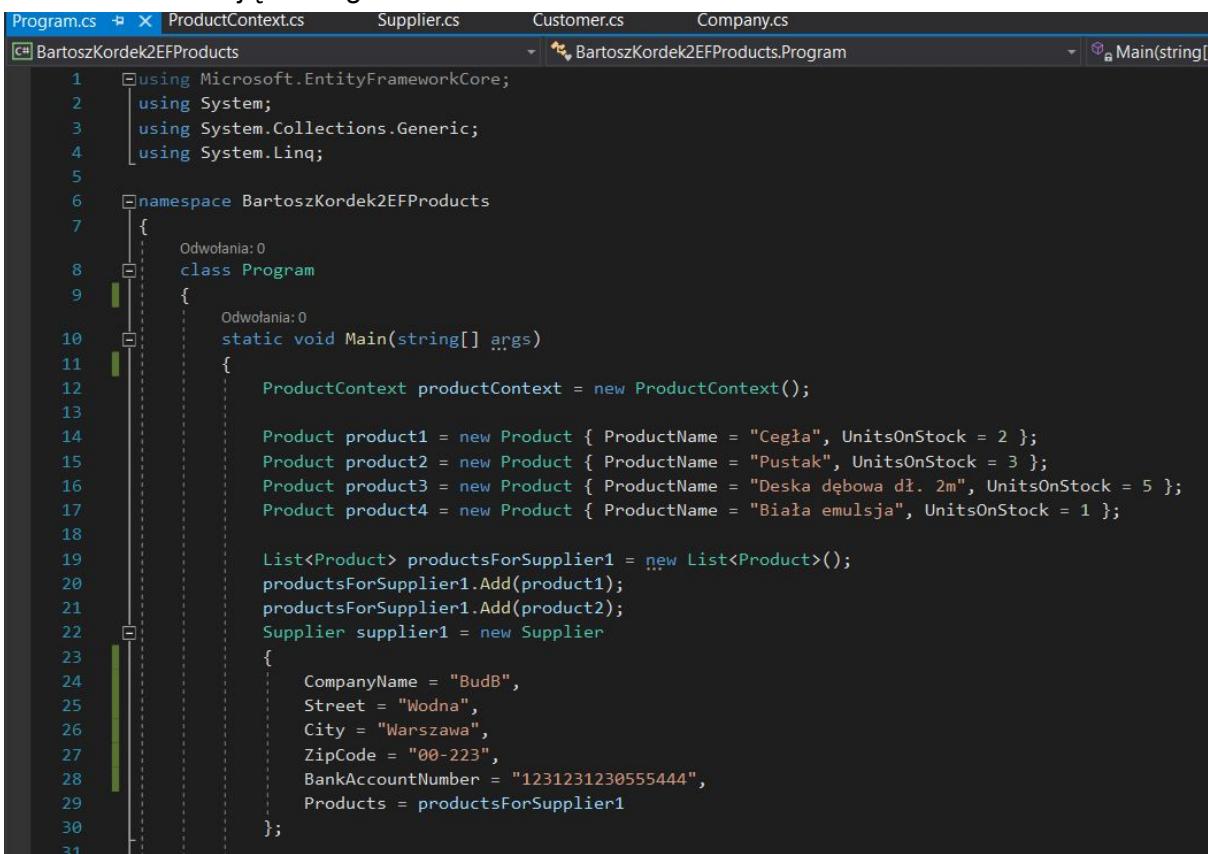
- utworzenie migracji i aktualizacja bazy danych

```
C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts>dotnet ef migrations add InheritanceTablePerTable
Build started...
Build succeeded.
An operation was scaffolded that may result in the loss of data. Please review the migration for accuracy.
Done. To undo this action, use 'ef migrations remove'

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts>dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201201202042_InheritanceTablePerTable'.
Done.

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts>
```

- klasa sterująca Program - bez zmian



The screenshot shows a code editor with the following code in the `Program.cs` file:

```
1  using Microsoft.EntityFrameworkCore;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5
6  namespace BartoszKordek2EFProducts
7  {
8      Odwołania:0
9      class Program
10     {
11         Odwołania:0
12         static void Main(string[] args)
13         {
14             Product product1 = new Product { ProductName = "Cegła", UnitsOnStock = 2 };
15             Product product2 = new Product { ProductName = "Pustak", UnitsOnStock = 3 };
16             Product product3 = new Product { ProductName = "Deska dębową dł. 2m", UnitsOnStock = 5 };
17             Product product4 = new Product { ProductName = "Biała emulsja", UnitsOnStock = 1 };
18
19             List<Product> productsForSupplier1 = new List<Product>();
20             productsForSupplier1.Add(product1);
21             productsForSupplier1.Add(product2);
22             Supplier supplier1 = new Supplier
23             {
24                 CompanyName = "BudB",
25                 Street = "Wodna",
26                 City = "Warszawa",
27                 ZipCode = "00-223",
28                 BankAccountNumber = "1231231230555444",
29                 Products = productsForSupplier1
30             };
31 }
```

```
31
32
33     List<Product> productsForSupplier2 = new List<Product>();
34     productsForSupplier2.Add(product3);
35     productsForSupplier2.Add(product4);
36     Supplier supplier2 = new Supplier
37     {
38         CompanyName = "Mrówka",
39         Street = "Cienista",
40         City = "Rzeszów",
41         ZipCode = "36-778",
42         BankAccountNumber = "3211231230555444",
43         Products = productsForSupplier2
44     };
45
46     Customer customer1 = new Customer
47     {
48         CompanyName = "Januszex",
49         Street = "Rajska",
50         City = "Kraków",
51         ZipCode = "30-201",
52         Discount = 20
53     };
54
55     Customer customer2 = new Customer
56     {
57         CompanyName = "DomBud",
58         Street = "Kreta",
59         City = "Warszawa",
60         ZipCode = "00-051",
61         Discount = 10
62     };

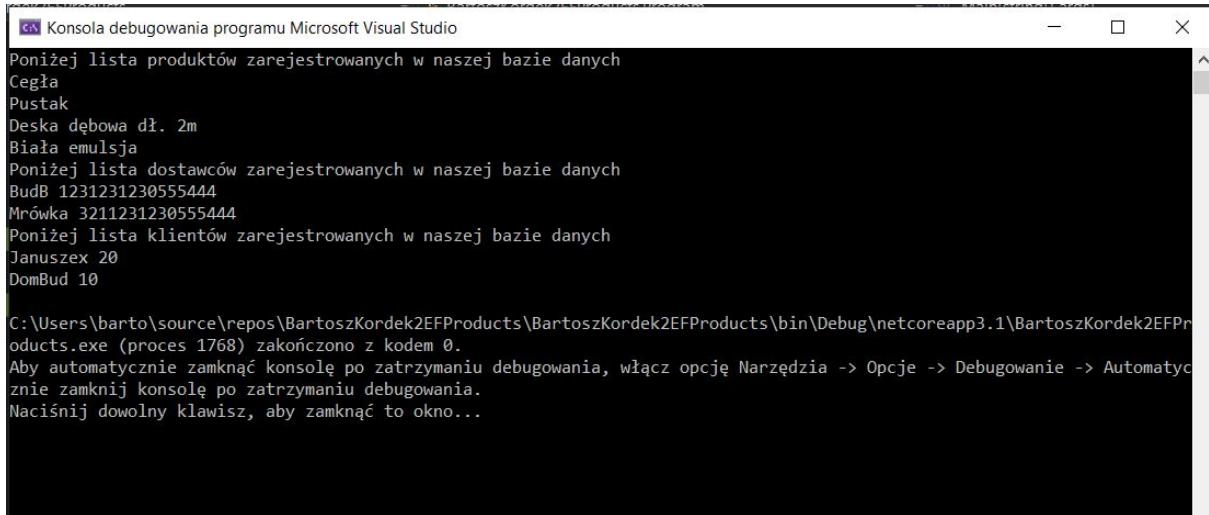
```

```
63
64
65     List<Product> productsForInvoice1 = new List<Product>();
66     productsForInvoice1.Add(product1);
67     productsForInvoice1.Add(product2);
68     productsForInvoice1.Add(product3);
69     Invoice invoice1 = new Invoice { InvoiceNumber = 56664, Quantity = 1, Products = productsForInvoice1 };
70
71     List<Product> productsForInvoice2 = new List<Product>();
72     productsForInvoice2.Add(product2);
73     productsForInvoice2.Add(product3);
74     productsForInvoice2.Add(product4);
75     Invoice invoice2 = new Invoice { InvoiceNumber = 57885, Quantity = 2, Products = productsForInvoice2 };
76
77     /*
78     productContext.Suppliers.Add(supplier1);
79     productContext.Suppliers.Add(supplier2);
80     */
81     productContext.Companies.Add(supplier1);
82     productContext.Companies.Add(supplier2);
83     productContext.Companies.Add(customer1);
84     productContext.Companies.Add(customer2);
85     productContext.Products.Add(product1);
86     productContext.Products.Add(product2);
87     productContext.Invoices.Add(invoice1);
88     productContext.Invoices.Add(invoice2);
89     productContext.SaveChanges();
90

```

```
91  
92  
93     Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");  
94     var query = from prod in productContext.Products select prod.ProductName;  
95     foreach (var pName in query)  
96     {  
97         Console.WriteLine(pName);  
98     }  
99  
100    Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych");  
101    foreach (Supplier c in productContext.Companies.OfType<Supplier>().Select(s => s))  
102    {  
103        Console.WriteLine(c.CompanyName + " " + c.BankAccountNumber);  
104    }  
105  
106    Console.WriteLine("Poniżej lista klientów zarejestrowanych w naszej bazie danych");  
107    foreach (Customer c in productContext.Companies.OfType<Customer>().Select(s => s))  
108    {  
109        Console.WriteLine(c.CompanyName + " " + c.Discount);  
110    }
```

- wynik

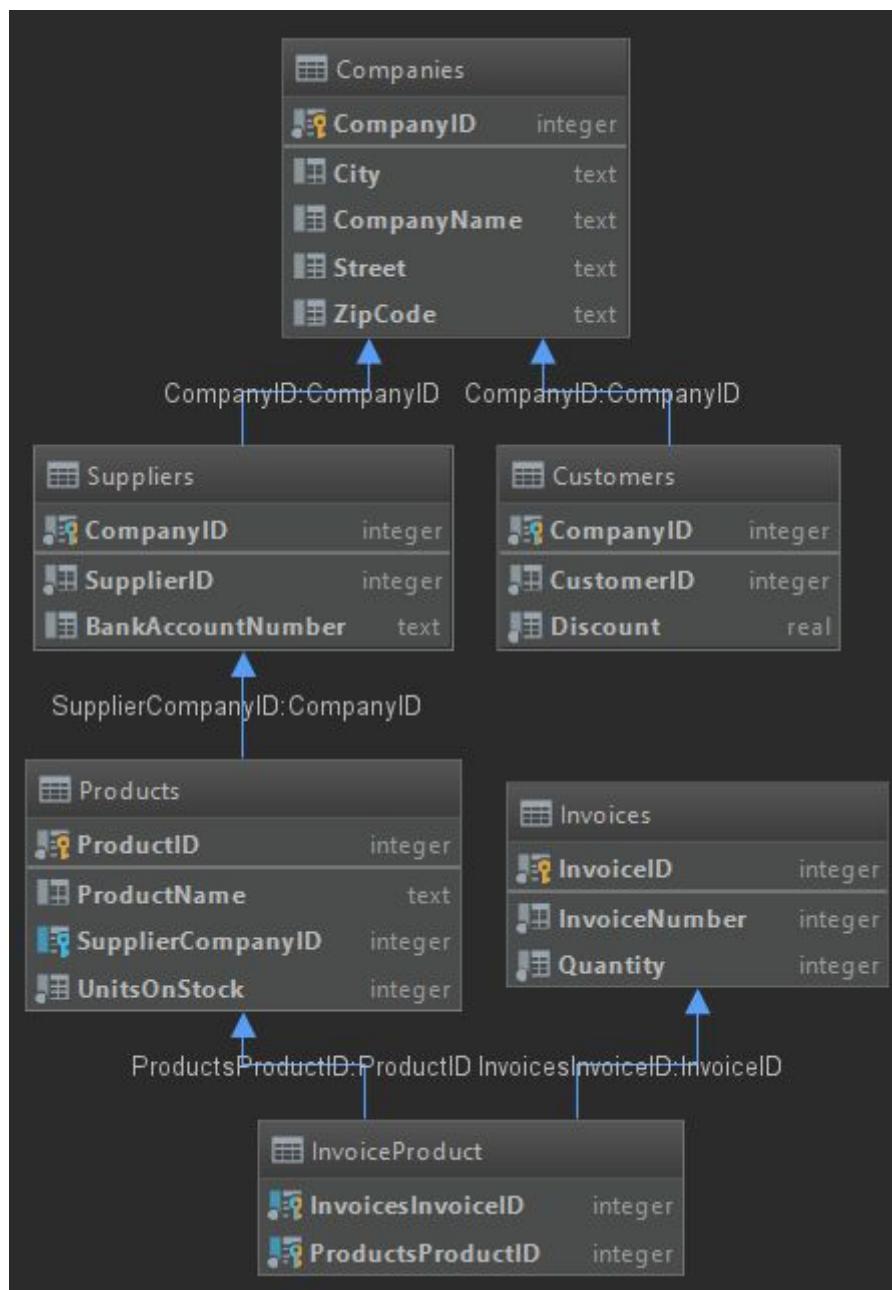


Konsola debugowania programu Microsoft Visual Studio

```
Poniżej lista produktów zarejestrowanych w naszej bazie danych
Cegła
Pustak
Deska dębową dł. 2m
Biała emulsja
Poniżej lista dostawców zarejestrowanych w naszej bazie danych
BudB 1231231230555444
Mrówka 3211231230555444
Poniżej lista klientów zarejestrowanych w naszej bazie danych
Januszex 20
DomBud 10

C:\Users\barto\source\repos\BartoszKordek2EFProducts\BartoszKordek2EFProducts\bin\Debug\netcoreapp3.1\BartoszKordek2EFProducts.exe (proces 1768) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

- schemat bazy danych w DataGrip



Tx: Auto

 1. `select * from main.Companies`

Output main.Companies

	CompanyID	City	CompanyName	Street	ZipCode
1	1	Warszawa	BudB	Wodna	00-223
2	2	Rzeszów	Mrówka	Cienista	36-778
3	3	Kraków	Januszex	Rajska	30-201
4	4	Warszawa	DomBud	Kręta	00-051

```
|▶|⌚|⌚|Tx: Auto|✓|↻|  
1 | select * from main.Customers|
```

	CompanyID	CustomerID	Discount
1	3	0	20
2	4	0	10

```
|▶|⌚|⌚|Tx: Auto|✓|↻|  
1 | select * from main.Suppliers|
```

	CompanyID	SupplierID	BankAccountNumber
1	1	0	1231231230555444
2	2	0	3211231230555444

```
|▶|⌚|⌚|Tx: Auto|✓|↻|  
1 | select * from main.Products|
```

	ProductID	ProductName	SupplierCompanyID	UnitsOnStock
1	1	Cegła	1	2
2	2	Pustak	1	3
3	3	Deska dębową dł. 2m	2	5
4	4	Biała emulsja	2	1

```
|▶|⌚|⌚|Tx: Auto|✓|↻|  
1 | select * from main.Invoices|
```

	InvoiceID	InvoiceNumber	Quantity
1	1	56664	1
2	2	57885	2

```
|▶|⌚|⌚|Tx: Auto|✓|↻|  
1 | select * from main.InvoiceProduct|
```

	Invoices	Products
	InvoiceID	ProductID
1		1
2		1
3		2
4		1
5		2
6		2

Porównanie strategii dziedziczenia

W metodzie **Table-Per-Hierarchy (TPH)** otrzymujemy tylko jedną tabelę, w tym przypadku *Company*, w której zawarte są rekordy dla klas dziedziczącej (*Customer*, *Supplier*) po klasie głównej. Można zauważyć, że w przypadku nr konta wartość pola dla rekordu klienta jest null, podobnie jak w przypadku zniżki dla dostawcy. W oczy rzuca się jeszcze pole *Discriminator*, które określa jakiej klasy jest dany wiersz tabeli. Problemem tego typu rozwiązania jest niewykorzystanie pól, przez co wiele z nich ma wartość null, co prowadzić może do problemów związanych z integralnością danych. Zaletą z pewnością jest prostota rozwiązania. Nie potrzebujemy wiele tabel, wystarczy jedna.

W metodzie **Table-Per-Table (TPT)** otrzymujemy wiele tabel, które odzwierciedlają zaimplementowane klasy (zbiory danych). Pola klasy bazowej znajdują się w tabelach klas dziedziczących. Nie występują tutaj wartości null (chyba że zostaną zaimplementowane specjalnie dla danej tabeli). Plusem tej metody jest spójność danych. Do minusów można zaliczyć większą złożoność bazy danych.