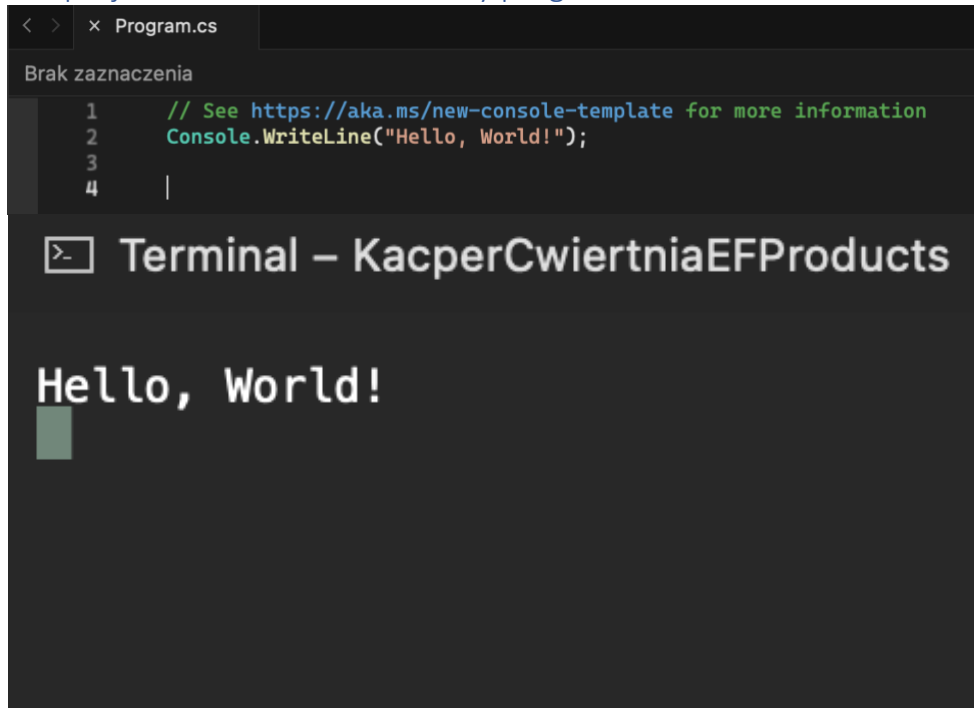


Sprawozdanie EntityFramework

Kacper Ćwiertnia

1.

Stworzyłem projekt i uruchomiłem testowy program.

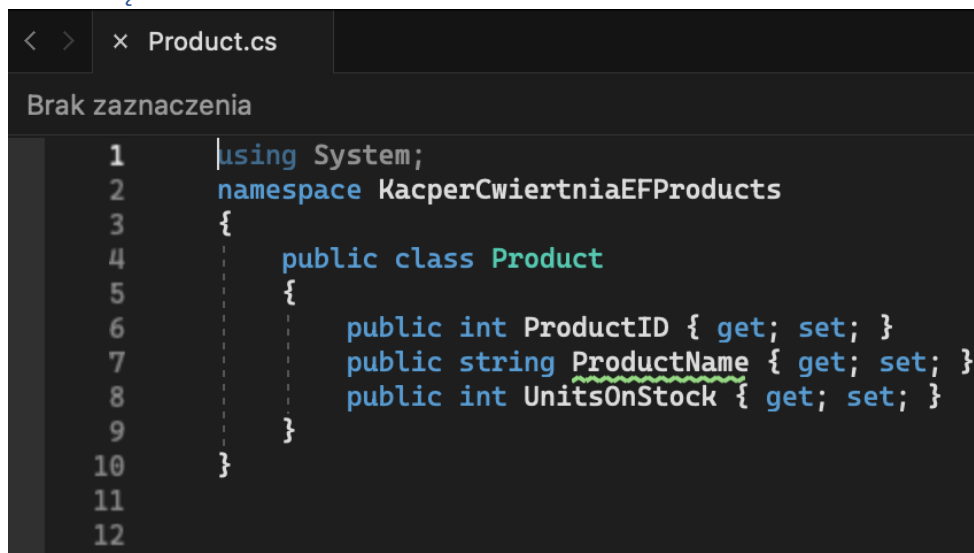


```
< > x Program.cs
Brak zaznaczenia
1 // See https://aka.ms/new-console-template for more information
2 Console.WriteLine("Hello, World!");
3
4 |

Terminal – KacperCwiertniaEFProducts

Hello, World!
|
```

Stworzyłem klasę Product



```
< > x Product.cs
Brak zaznaczenia
1 using System;
2 namespace KacperCwiertniaEFProducts
3 {
4     public class Product
5     {
6         public int ProductID { get; set; }
7         public string ProductName { get; set; }
8         public int UnitsOnStock { get; set; }
9     }
10 }
11
12
```

Stworzyłem klasę ProductContext

```
ProductContext.cs

ProductContext > Brak zaznaczenia

1  using System;
2  namespace KacperCwiertniaEFProducts
3  {
4      public class ProductContext: DbContext
5      {
6      }
7  }
8
9
10
```

Instaluję pakiet Microsoft.EntityFrameworkCore

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet add package Microsoft.EntityFrameworkCore
Determining projects to restore...
Writing /var/folders/ls/f7qp8vy907g8_dfj80g5d_6w0000gn/T/tmpea90yF.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/6.0.100/certificates/cacert.pem'
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/6.0.100/certificates/cacert.pem'
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore' into project '/Users/kacpercwiertnia/Projects/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts.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 158ms
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/page/0.0.1-alpha/3.1.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/page/0.0.1-alpha/3.1.1 158ms
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/page/3.1.4/6.0.0-rc.1.21452
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/page/3.1.4/6.0.0-rc.1.21452 158ms
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/page/6.0.0-rc.1.21452
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore/page/6.0.0-rc.1.21452 158ms
info : Restoring packages for /Users/kacpercwiertnia/Projects/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts.csproj...
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.entityframeworkcore/index.json
info : OK https://api.nuget.org/v3-flatcontainer/microsoft.entityframeworkcore/index.json 202ms
```

Rozwiązanie

- ▼ KacperCwiertniaEFProducts
 - ▼ KacperCwiertniaEFProducts
 - Usługi połączone
 - ▼ Zależności
 - ▼ Struktury
 - Microsoft.NETCore.App – 7.0.4
 - ▼ NuGet
 - ▼ Microsoft.EntityFrameworkCore – 7.0.4
 - Product.cs
 - ProductContext.cs
 - Program.cs

Przy pomocy menu kontekstowego dodaje pakiet do klasy.

```
< > • ProductContext.cs
Brak zaznaczenia
1      using System;
2      using Microsoft.EntityFrameworkCore;
3
4      namespace KacperCwiertniaEFProducts
5      {
6          public class ProductContext : DbContext
7          {
8              ...
9          }
10     }
11
12     |
```

Instaluję dotnet-ef.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet tool install --global dotnet-ef
Tools directory '/Users/kacpercwiertnia/.dotnet/tools' is not currently on the PATH environment variable.
If you are using zsh, you can add it to your profile by running the following command:

cat << \EOF >> ~/.zprofile
# Add .NET Core SDK tools
export PATH="$PATH:/Users/kacpercwiertnia/.dotnet/tools"
EOF

And run 'zsh -l' to make it available for current session.

You can only add it to the current session by running the following command:

export PATH="$PATH:/Users/kacpercwiertnia/.dotnet/tools"

You can invoke the tool using the following command: dotnet-ef
Tool 'dotnet-ef' (version '7.0.4') was successfully installed.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

Instaluję Microsoft.EntityFrameworkCore.Design

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet add package Microsoft.EntityFrameworkCore.Design
Determining projects to restore...
Writing /var/folders/ls/f7qp8vy907g8_djf80g5d_6w0000gn/T/tmpAqWrc.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.201/certificates/cacert.pem'
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.201/certificates/cacert.pem'
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore.Design' into project '/Users/kacpercwiertnia/Projects/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts.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 509ms
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/page/0.0.1-alpha/3.1.4/6.0.0-rc.1.21452.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/page/0.0.1-alpha/3.1.4/6.0.0-rc.1.21452.1
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/page/3.1.5/6.0.0-rc.1.21452.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/page/3.1.5/6.0.0-rc.1.21452.1
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/page/6.0.0-rc.2.21480.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.design/page/6.0.0-rc.2.21480.1
info : Restoring packages for /Users/kacpercwiertnia/Projects/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts.csproj...
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.entityframeworkcore.design/index.json
```

Instaluję Microsoft.EntityFrameworkCore.Sqlite

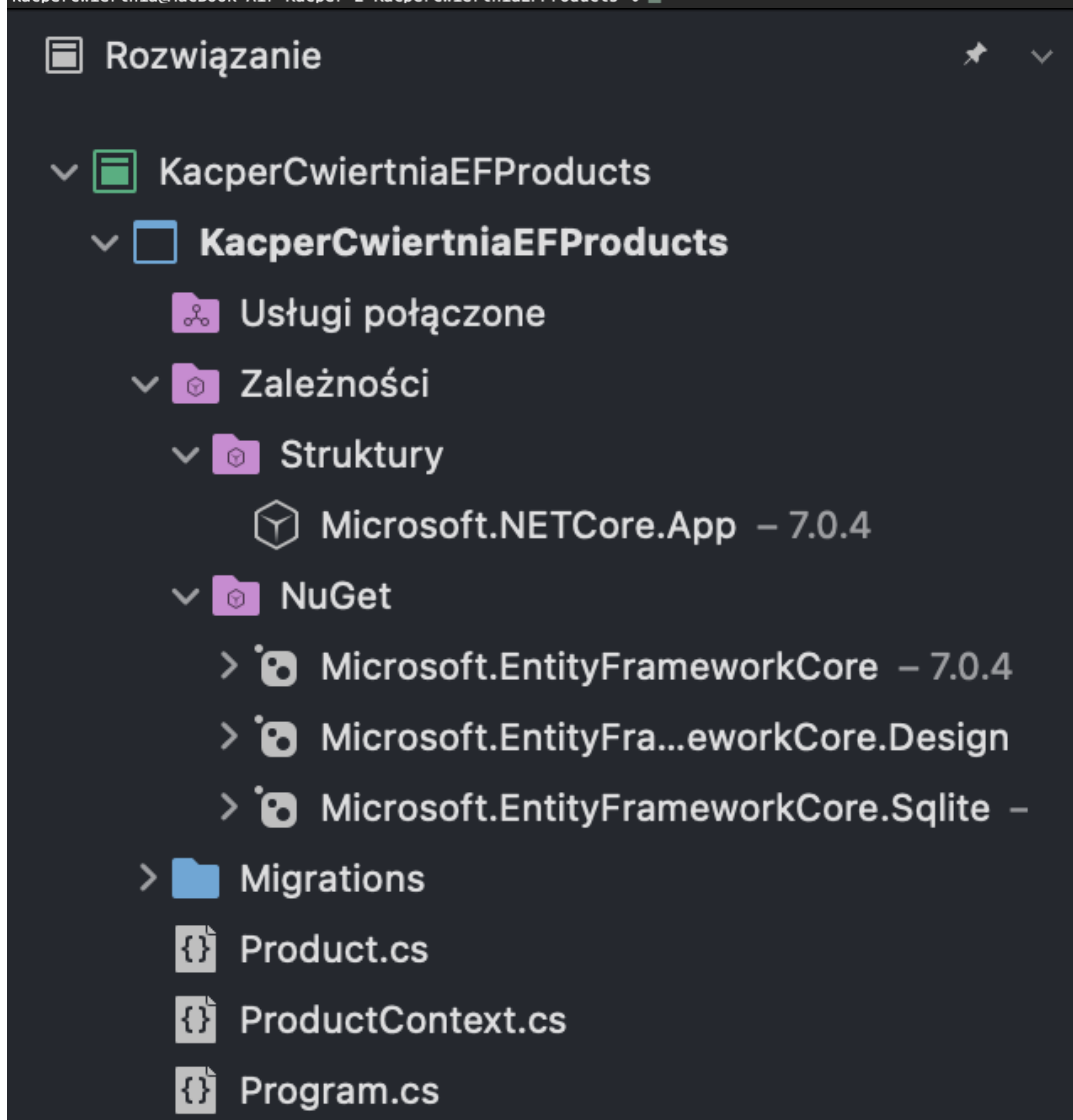
```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet add package Microsoft.EntityFrameworkCore.Sqlite
Determining projects to restore...
Writing /var/folders/ls/f7qp8vy907g8_djf80g5d_6w0000gn/T/tmpfIEsXw.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.201/certificates/cacert.pem'
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.201/certificates/cacert.pem'
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore.Sqlite' into project '/Users/kacpercwiertnia/Projects/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts.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 324ms
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/page/0.0.1-alpha/3.1.4/6.0.0-rc.1.21452.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/page/0.0.1-alpha/3.1.4/6.0.0-rc.1.21452.1
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/page/3.1.4/6.0.0-rc.1.21452.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/page/3.1.4/6.0.0-rc.1.21452.1
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/page/6.0.0-rc.1.21452.1
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityframeworkcore.sqlite/page/6.0.0-rc.1.21452.1
info : Restoring packages for /Users/kacpercwiertnia/Projects/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts/KacperCwiertniaEFProducts.csproj...
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.entityframeworkcore.sqlite/index.json
```

Edytuję klasę ProductContext.

```
< > • ProductContext.cs
Brak zaznaczenia
1  using System;
2  using Microsoft.EntityFrameworkCore;
3
4  namespace KacperCwiertniaEFProducts
5  {
6      public class ProductContext : DbContext
7      {
8          public DbSet<Product> Products { get; set; }
9
10         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
11         {
12             base.OnConfiguring(optionsBuilder);
13             optionsBuilder.UseSqlite("DataSource=ProductsDatabase");
14         }
15     }
16 }
```

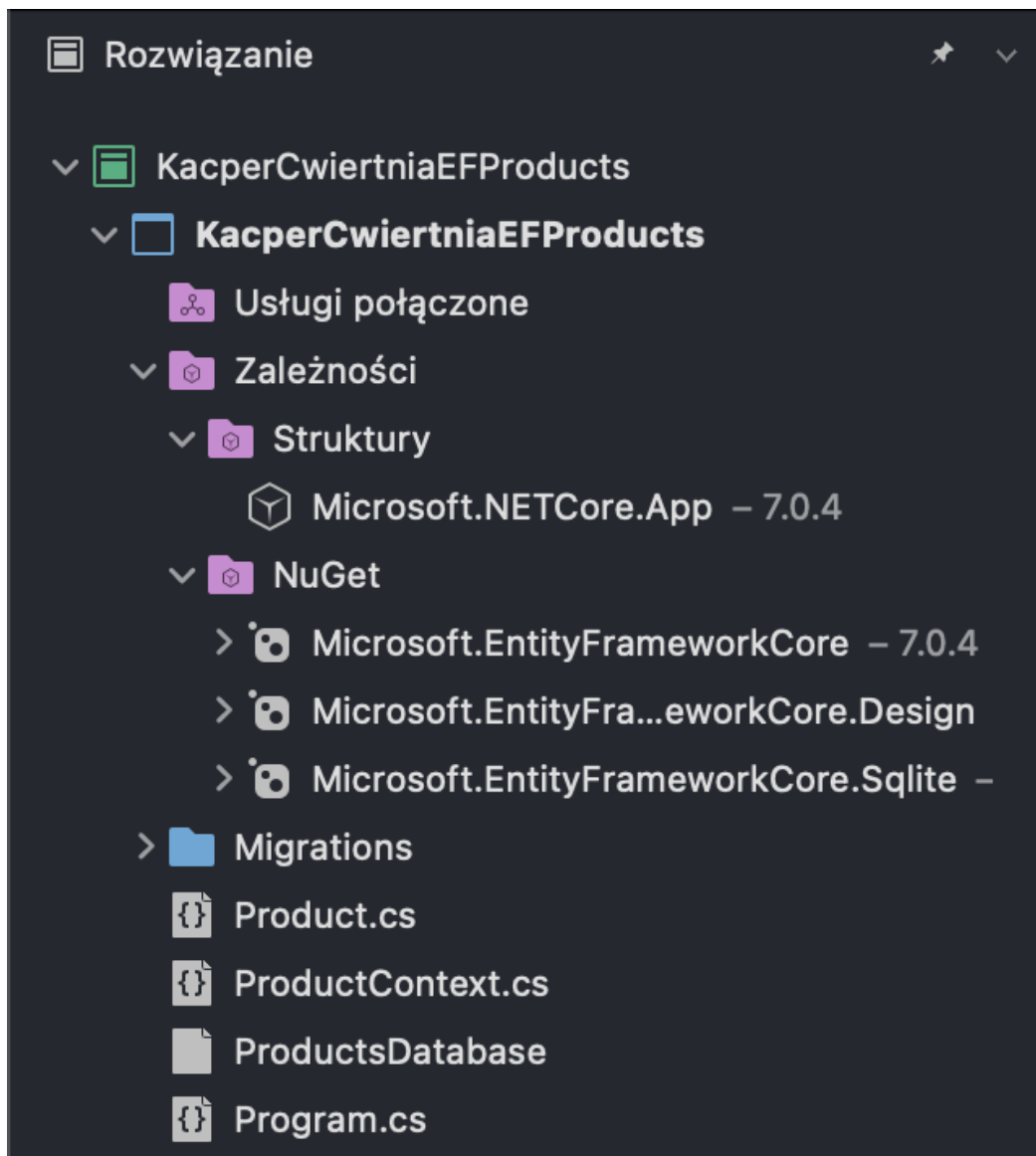
Migruję model.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add InitProductDatabase
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

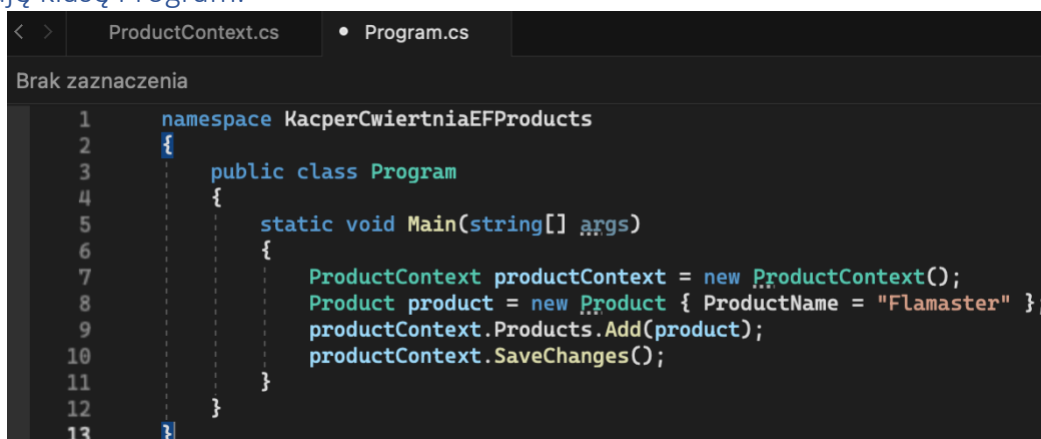


Dokonuję update bazy danych.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230405170804_InitProductDatabase'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```



Edytuję klasę Program.



Po kompilacji programu flamaster jest widoczny w tabeli.

```
kacpercwiertnia@MacBook-Air-Kacper-2 net7.0 % sqlite3 ProductsDatabase
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
sqlite> .tables
Products          __EFMigrationsHistory
sqlite> select * from Products;
1|Flamaster|0
sqlite>
```

Edytuję klasę Program.

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

Po kompilacji programu w konsoli widzimy Flamaster.

```
>- Terminal – KacperCwiertniaEFProducts

Flamaster
```

Edytuję klasę Program.

```
ProductContext.cs x Program.cs
Program > Main(string[] args)
1 namespace KacperCwiertniaEFProducts
2 {
3     public class Program
4     {
5         static void Main(string[] args)
6         {
7             Console.WriteLine("Podaj nazwę produktu");
8             string prodName = Console.ReadLine();
9
10            Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
11            ProductContext productContext = new ProductContext();
12            Product product = new Product { ProductName = prodName };
13            productContext.Products.Add(product);
14            productContext.SaveChanges();
15
16            var query = from prod in productContext.Products
17                        select prod.ProductName;
18
19            foreach(var pName in query)
20            {
21                Console.WriteLine(pName);
22            }
23        }
24    }
25 }
```


Po kompilacji programu w konsoli widzimy nowo dodane produkty.

```
> Terminal – KacperCwiertniaEFProducts

Podaj nazwę produktu
Flamaster
Poniżej lista porduktów zarejestrowanych w naszej bazie danych
Resorak
Zeszyt
Flamaster
█
```

Tworzę klasę Supplier.

```
1 using System;
2 namespace KacperCwiertniaEFProducts
3 {
4     public class Supplier
5     {
6         public int SupplierID { get; set; }
7         public string CompanyName { get; set; }
8         public string Street { get; set; }
9         public string City { get; set; }
10    }
11 }
```

Edytuję klasę Product.

```
1 using System;
2 namespace KacperCwiertniaEFProducts
3 {
4     public class Product
5     {
6         public int ProductID { get; set; }
7         public string ProductName { get; set; }
8         public int UnitsOnStock { get; set; }
9         public Supplier Supplier { get; set; }
10    }
11 }
```

Edytuję klasę ProductContext.

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

Dokonuję migracji zmian i updateuję bazę.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add ProductDatabase2
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230420143254_ProductDatabase2'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

Edytuję klasę Program.

```
1 namespace KacperCwiertniaEFProducts
2 {
3     public class Program
4     {
5         static void Main(string[] args)
6         {
7             Console.WriteLine("Podaj nazwę produktu");
8             string prodName = Console.ReadLine();
9
10            Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych");
11            ProductContext productContext = new ProductContext();
12            Product product = new Product { ProductName = prodName };
13            Supplier supplier = new Supplier { CompanyName = "GigaZabawki", Street = "Buryka", City = "Kraków" };
14            productContext.Suppliers.Add(supplier);
15            product.Supplier = supplier;
16            productContext.Products.Add(product);
17            productContext.SaveChanges();
18
19            var query = from prod in productContext.Products
20                        select prod;
21
22            foreach(var pName in query)
23            {
24                Console.WriteLine(pName.ProductName+" "+pName.Supplier.CompanyName);
25            }
26        }
27    }
28 }
```

Supplier dodaje się do produktu.

```
Terminal – KacperCwiertniaEFProducts

Podaj nazwę produktu
Autk
Poniżej lista produktów zarejestrowanych w naszej bazie danych
Autk GigaZabawki
```

2.

Edytuję klasę Product.

```
using System;
namespace KacperCwiertniaEFProducts
{
    public class Product
    {
        public int ProductID { get; set; }
        public string ProductName { get; set; }
        public int UnitsOnStock { get; set; }
    }
}
```


Edytuję klasę Supplier.

```
using System;
namespace KacperCwiertniaEFProducts
{
    public class Supplier
    {
        public int SupplierID { get; set; }
        public string CompanyName { get; set; }
        public string Street { get; set; }
        public string City { get; set; }

        public ICollection<Product> Products { get; set; }
    }
}
```

Dokonuję migracji zmian i updateuję bazę.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add ProductDatabase3
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230420154542_ProductDatabase3'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

Edytuję klasę Program.

```
1 namespace KacperCwiertniaEFProducts
2 {
3     public class Program
4     {
5         static void Main(string[] args)
6         {
7             Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z produktami");
8             ProductContext productContext = new ProductContext();
9             Product product = new Product { ProductName = "Kredki" };
10            Product product2 = new Product { ProductName = "Pisaki" };
11            Product product3 = new Product { ProductName = "Pluszka" };
12            ICollection<Product> products = new List<Product>();
13            productContext.Products.Add(product);
14            productContext.Products.Add(product2);
15            productContext.Products.Add(product3);
16            products.Add(product);
17            products.Add(product2);
18            products.Add(product3);
19            Supplier supplier = new Supplier { CompanyName = "ZabawkiDoSzkoly", Street = "Poselska", City = "Kraków",
20            Products = products };
21            productContext.Suppliers.Add(supplier);
22            productContext.SaveChanges();
23
24            var query = from supp in productContext.Suppliers
25                        select supp;
26
27            foreach(var supp in query)
28            {
29                foreach(var prod in supp.Products){
30                    Console.WriteLine(supp.CompanyName + " " + prod.ProductName);
31                }
32            }
33        }
34    }
35 }
```

Supplier ma produkty.

Terminal – KacperCwiertniaEFProducts

```
Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z produktami
ZabawkiDoSzkoly Kredki
ZabawkiDoSzkoly Pisaki
ZabawkiDoSzkoly Pluszka
```

3.

Edytuję klasę Product.

```
1      using System;
2      namespace KacperCwiertniaEFProducts
3      {
4          public class Product
5          {
6              public int ProductID { get; set; }
7              public string ProductName { get; set; }
8              public int UnitsOnStock { get; set; }
9              public Supplier Supplier { get; set; }
10         }
11     }
```

Dokonuję migracji zmian i updateuję bazę.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add ProductDatabase4
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'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230420163252_ProductDatabase4'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

Edytuję klasę Program.

```
1      namespace KacperCwiertniaEFProducts
2      {
3          public class Program
4          {
5              static void Main(string[] args)
6              {
7                  ProductContext productContext = new ProductContext();
8                  Supplier supplier = new Supplier { CompanyName = "ZabawkiDoSzkoly", Street = "Poselska", City = "Kraków" };
9                  Product product = new Product { ProductName = "Kredki", Supplier = supplier };
10                 Product product2 = new Product { ProductName = "Pisaki", Supplier = supplier };
11                 Product product3 = new Product { ProductName = "Pluszka", Supplier = supplier };
12                 ICollection<Product> products = new List<Product>();
13                 products.Add(product);
14                 products.Add(product2);
15                 products.Add(product3);
16                 supplier.Products = products;
17                 productContext.Products.Add(product);
18                 productContext.Products.Add(product2);
19                 productContext.Products.Add(product3);
20                 productContext.Suppliers.Add(supplier);
21                 productContext.SaveChanges();
22
23                 var query = from supp in productContext.Suppliers
24                             select supp;
25
26                 Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z produktami");
27                 foreach (var supp in query)
28                 {
29                     foreach(var prod in supp.Products){
30                         Console.WriteLine(supp.CompanyName + " " + prod.ProductName);
31                     }
32                 }
33
34                 var query2 = from prod in productContext.Products
35                             select prod;
36
37                 Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych wraz ze sprzedawcą");
38                 foreach (var prod in query2)
39                 {
40                     Console.WriteLine(prod.ProductName + " " + prod.Supplier.CompanyName);
41                 }
42             }
43         }
44     }
```

Supplier ma produkty, a produkty mają suppliera.

```
Terminal – KacperCwiertniaEFProducts

Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z produktami
ZabawkiDoSzkoly Kredki
ZabawkiDoSzkoly Pisaki
ZabawkiDoSzkoly Pluszka
Poniżej lista produktów zarejestrowanych w naszej bazie danych wraz ze sprzedawcą
Kredki ZabawkiDoSzkoly
Pisaki ZabawkiDoSzkoly
Pluszka ZabawkiDoSzkoly
```

4.

Edytuję klasę Product.

```
1  using System;
2  namespace KacperCwiertniaEFProducts
3  {
4      public class Product
5      {
6          public Product(){
7              this.Invoices = new HashSet<Invoice>();
8          }
9
10         public int ProductID { get; set; }
11         public string ProductName { get; set; }
12         public int UnitsOnStock { get; set; }
13         public virtual ICollection<Invoice> Invoices { get; set; }
14     }
15 }
```

Tworzę klasę Invoice.

```
1  using System;
2  namespace KacperCwiertniaEFProducts
3  {
4      public class Invoice
5      {
6          public int InvoideID { get; set; }
7          public int InvoiceNumber { get; set; }
8          public int Quantity { get; set; }
9          public virtual ICollection<Product> Products { get; set; }
10
11         public Invoice(){
12             this.Products = new HashSet<Product>();
13         }
14     }
15 }
```

Edytuję klasę ProductContext.

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

Dokonuję migracji zmian i updateuję bazę.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add ProductDatabase5
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230420175314_ProductDatabase5'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

Edytuję klasę Program.

```
1 namespace KacperCwiertniaEFProducts
2 {
3     public class Program
4     {
5         static void Main(string[] args)
6         {
7             ProductContext productContext = new ProductContext();
8             Invoice invoice = new Invoice { InvoiceNumber = 1 };
9             Invoice invoice2 = new Invoice { InvoiceNumber = 2 };
10            Invoice invoice3 = new Invoice { InvoiceNumber = 3 };
11            Product product = new Product { ProductName = "Kredki" };
12            Product product2 = new Product { ProductName = "Pisaki" };
13            Product product3 = new Product { ProductName = "Pluszka" };
14            invoice.Products.Add(product);
15            invoice.Products.Add(product2);
16            invoice2.Products.Add(product2);
17            invoice2.Products.Add(product3);
18            invoice3.Products.Add(product);
19            invoice3.Products.Add(product3);
20            product.Invoices.Add(invoice);
21            product.Invoices.Add(invoice2);
22            product2.Invoices.Add(invoice2);
23            product2.Invoices.Add(invoice3);
24            product3.Invoices.Add(invoice);
25            product3.Invoices.Add(invoice3);
26            productContext.Products.Add(product);
27            productContext.Products.Add(product2);
28            productContext.Products.Add(product3);
29            productContext.Invoices.Add(invoice);
30            productContext.Invoices.Add(invoice2);
31            productContext.Invoices.Add(invoice3);
32            productContext.SaveChanges();
33
34            var query = from prod in productContext.Products
35                       select prod;
36
37            Console.WriteLine("Poniżej lista produktów zarejestrowanych w naszej bazie danych wraz z fakturami");
38            foreach (var prod in query)
39            {
40                foreach(var invo in prod.Invoices){
41                    Console.WriteLine(prod.ProductName + " " + invo.InvoiceNumber);
42                }
43            }
44
45            var query2 = from invo in productContext.Invoices
46                       select invo;
47
48            Console.WriteLine("Poniżej lista faktur zarejestrowanych w naszej bazie danych wraz z produktami");
49            foreach (var invo in query2)
50            {
51                foreach (var prod in invo.Products)
52                {
53                    Console.WriteLine(invo.InvoiceNumber + " " + prod.ProductName);
54                }
55            }
56        }
57    }
58 }
```

Produkty mają wiele faktur, a faktury mają wiele produktów.

```
Poniżej lista produktów zarejestrowanych w naszej bazie danych wraz z fakturami
Kredki 1
Kredki 2
Kredki 3
Pisaki 2
Pisaki 3
Pisaki 1
Pluszka 1
Pluszka 3
Pluszka 2
Poniżej lista faktur zarejestrowanych w naszej bazie danych wraz z produktami
1 Kredki
1 Pisaki
1 Pluszka
2 Pisaki
2 Pluszka
2 Kredki
3 Kredki
3 Pluszka
3 Pisaki
```

5.

Tworzę klasę Company.

```
1 using System;
2 namespace KacperCwiertniaEFProducts
3 {
4     public class Company
5     {
6         public int CompanyID { get; set; }
7         public string CompanyName { get; set; }
8         public string Street { get; set; }
9         public string City { get; set; }
10        public string ZipCode { get; set; }
11    }
12 }
```

Tworzę klasę Customer.

```
1 using System;
2 namespace KacperCwiertniaEFProducts
3 {
4     public class Customer : Company
5     {
6         public double discount { get; set; }
7     }
8 }
```

Edytuję klasę Supplier.

```
1 using System;
2 namespace KacperCwiertniaEFProducts
3 {
4     public class Supplier : Company
5     {
6         public string bankAccountNumber { get; set; }
7     }
8 }
```

Edytuję klasę Program.

```

1 namespace MiereniaEPPProducts
2 {
3     public class Program
4     {
5         static void Main(string[] args)
6         {
7             ProductContext productContext = new ProductContext();
8             Company company1 = new Customer { CompanyName = "firma1", Street = "Stonieczna", City = "Kraków", ZipCode = "32-012", discount = 0.2 };
9             Company company2 = new Customer { CompanyName = "firma2", Street = "Budryka", City = "Kraków", ZipCode = "35-122", discount = 0.1 };
10            Company company3 = new Customer { CompanyName = "firma3", Street = "Lea", City = "Kraków", ZipCode = "35-222", discount = 0.5 };
11            Company company4 = new Supplier { CompanyName = "firma4", Street = "Karmelińska", City = "Kraków", ZipCode = "35-282", bankAccountNumber = "0000000000";
12            Company company5 = new Supplier { CompanyName = "firma5", Street = "Batorego", City = "Kraków", ZipCode = "35-132", bankAccountNumber = "000012310001";
13            Company company6 = new Supplier { CompanyName = "firma6", Street = "Kremerowska", City = "Kraków", ZipCode = "35-332", bankAccountNumber = "123100000";
14            productContext.Companies.Add(company1);
15            productContext.Companies.Add(company2);
16            productContext.Companies.Add(company3);
17            productContext.Customers.Add((Customer)company1);
18            productContext.Customers.Add((Customer)company2);
19            productContext.Customers.Add((Customer)company3);
20            productContext.Companies.Add(company4);
21            productContext.Companies.Add(company5);
22            productContext.Companies.Add(company6);
23            productContext.Suppliers.Add((Supplier)company4);
24            productContext.Suppliers.Add((Supplier)company5);
25            productContext.Suppliers.Add((Supplier)company6);
26            productContext.SaveChanges();
27
28            var query = from comp in productContext.Companies
29                       select comp;
30
31            Console.WriteLine("Poniżej lista firm zarejestrowanych w naszej bazie danych");
32            foreach (var comp in query)
33            {
34                Console.WriteLine(comp.CompanyName);
35            }
36
37            var query2 = from comp in productContext.Companies.OfType<Supplier>()
38                        select comp;
39
40            Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z numerami kont bankowych");
41            foreach (var comp in query2)
42            {
43                Console.WriteLine(comp.CompanyName + " " + ((Supplier)comp).bankAccountNumber);
44            }
45
46            var query3 = from comp in productContext.Customers
47                        select comp;
48
49            Console.WriteLine("Poniżej lista klientów zarejestrowanych w naszej bazie danych wraz z wartością zniżki");
50            foreach (var comp in query3)
51            {
52                Console.WriteLine(comp.CompanyName + " " + comp.discount);
53            }
54        }
55    }
56 }

```

Dokonuję migracji zmian i updateuje bazę.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add ProductDatabase7
^[[OABuild 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'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230423121053_ProductDatabase7'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```


Wpierw wypisuję wszystkie firmy, potem wszystkich supplierow, a na końcu wszystkich customerow.

```
Poniżej lista firm zarejestrowanych w naszej bazie danych
firma1
firma2
firma3
firma4
firma5
firma6
Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z numerami kont bankowych
firma4 000000000001
firma5 000012310001
firma6 123100000001
Poniżej lista klientów zarejestrowanych w naszej bazie danych wraz z wartością zniżki
firma1 0.2
firma2 0.1
firma3 0.5
```

6.

Edytuję klasę Customer.

```
1  using System;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace KacperCwiertniaEFProducts
5  {
6
7      [Table("Company")]
8      public class Company
9      {
10         public int CompanyID { get; set; }
11         public string CompanyName { get; set; }
12         public string Street { get; set; }
13         public string City { get; set; }
14         public string ZipCode { get; set; }
15     }
16 }
```

Edytuję klasę Company.

```
1  using System;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace KacperCwiertniaEFProducts
5  {
6      [Table("Customer")]
7      public class Customer : Company
8      {
9         public double discount { get; set; }
10     }
11 }
```

Edytuję klasę Supplier.

```
1 using System;
2 using System.ComponentModel.DataAnnotations.Schema;
3
4 namespace KacperCwiertniaEFProducts
5 {
6     [Table("Supplier")]
7     public class Supplier : Company
8     {
9         public string bankAccountNumber { get; set; }
10    }
11 }
```

Edytuję klasę Program.

```
1 namespace KacperCwiertniaEFProducts
2 {
3     public class Program
4     {
5         static void Main(string[] args)
6         {
7             ProductContext productContext = new ProductContext();
8             Company company1 = new Customer { CompanyName = "firma1", Street = "Stonieczna", City="Kraków", ZipCode="32-012", discount = 0.2 };
9             Company company2 = new Customer { CompanyName = "firma2", Street = "Budryka", City = "Kraków", ZipCode = "35-122", discount = 0.1 };
10            Company company3 = new Customer { CompanyName = "firma3", Street = "Lea", City = "Kraków", ZipCode = "35-222", discount = 0.5 };
11            Company company4 = new Supplier { CompanyName = "firma4", Street = "Karmelicka", City = "Kraków", ZipCode = "35-282", bankAccountNumber = "0000000000" };
12            Company company5 = new Supplier { CompanyName = "firma5", Street = "Batorego", City = "Kraków", ZipCode = "35-132", bankAccountNumber = "000012310001" };
13            Company company6 = new Supplier { CompanyName = "firma6", Street = "Kremerowska", City = "Kraków", ZipCode = "35-332", bankAccountNumber = "123100000" };
14            productContext.Companies.Add(company1);
15            productContext.Companies.Add(company2);
16            productContext.Companies.Add(company3);
17            productContext.Customers.Add((Customer)company1);
18            productContext.Customers.Add((Customer)company2);
19            productContext.Customers.Add((Customer)company3);
20            productContext.Companies.Add(company4);
21            productContext.Companies.Add(company5);
22            productContext.Companies.Add(company6);
23            productContext.Suppliers.Add((Supplier)company4);
24            productContext.Suppliers.Add((Supplier)company5);
25            productContext.Suppliers.Add((Supplier)company6);
26            productContext.SaveChanges();
27
28            var query = from comp in productContext.Companies
29                        select comp;
30
31            Console.WriteLine("Poniżej lista firm zarejestrowanych w naszej bazie danych");
32            foreach (var comp in query)
33            {
34                Console.WriteLine(comp.CompanyName);
35            }
36
37            var query2 = from comp in productContext.Suppliers
38                        select comp;
39
40            Console.WriteLine("Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z numerami kont bankowych");
41            foreach (var comp in query2)
42            {
43                Console.WriteLine(comp.CompanyName + " " + comp.bankAccountNumber);
44            }
45
46            var query3 = from comp in productContext.Customers
47                        select comp;
48
49            Console.WriteLine("Poniżej lista klientów zarejestrowanych w naszej bazie danych wraz z wartością zniżki");
50            foreach (var comp in query3)
51            {
52                Console.WriteLine(comp.CompanyName + " " + comp.discount);
53            }
54        }
55    }
56 }
```

Dokonyuję migracji zmian i updateuje bazę.

```
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef migrations add ProductDatabase8
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'
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts % dotnet ef database update
Build started...
Build succeeded.
Applying migration '20230423124427_ProductDatabase8'.
Done.
kacpercwiertnia@MacBook-Air-Kacper-2 KacperCwiertniaEFProducts %
```

Wpierw wypisuję wszystkie firmy, potem wszystkich supplierow, a na końcu wszystkich customerow.

```
Poniżej lista firm zarejestrowanych w naszej bazie danych
firma1
firma2
firma3
firma4
firma5
firma6
Poniżej lista dostawców zarejestrowanych w naszej bazie danych wraz z numerami kont bankowych
firma4 0000000000001
firma5 000012310001
firma6 123100000001
Poniżej lista klientów zarejestrowanych w naszej bazie danych wraz z wartością zniżki
firma1 0.2
firma2 0.1
firma3 0.5
```

TPH tworzy pojedynczą tabelę, a TPT tworzy tabelę, dla każdego typu.

TPH.

```
Enter ".help" for usage hints.
sqlite> .table
Companies          Invoices          __EFMigrationsHistory
InvoiceProduct     Products
```

TPT.

```
sqlite> .table
Company            Invoices          __EFMigrationsHistory
Customer           Products
InvoiceProduct     Supplier
```