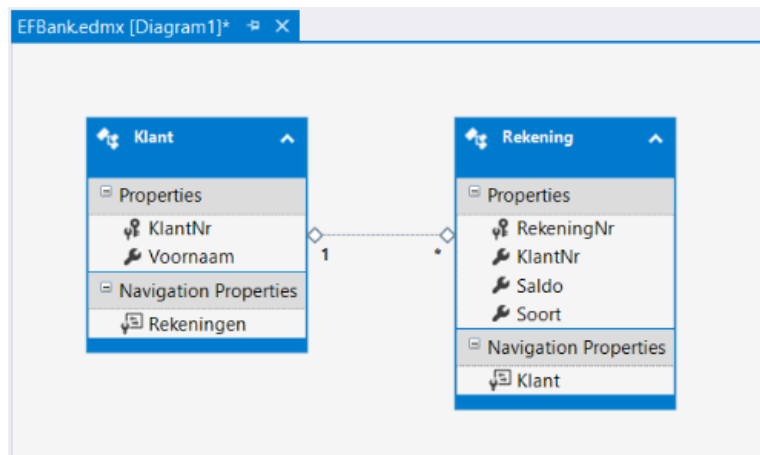


## 19 VOORBEELDOPLOSSINGEN

### 19.1 TAAK 01 : BANK MAKEN

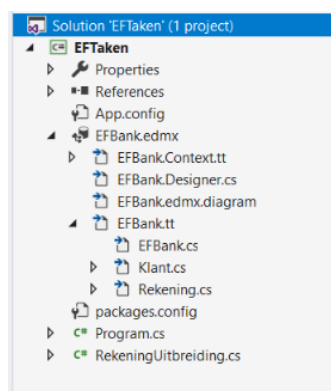


- De class **Klanten** is hernoemd naar **Klant**
- De class **Rekeningen** is hernoemd naar **Rekening**
- De navigation property **Klanten** van de class **Rekening** is hernoemd naar **Klant**.

De method **Storten** van de class **Rekening** in een apart bestand **RekeningUitbreiding.cs**

```

namespace EFTaken
{
    public partial class Rekening
    {
        public void Storten(decimal bedrag)
        {
            this.Saldo += bedrag;
        }
    }
}
  
```



### 19.2 TAAK 02 : KLANTEN EN HUN REKENINGEN

```

.....
using System.Linq;
.....
  
```

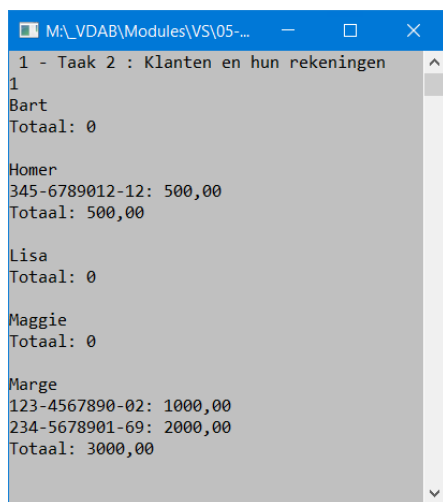
```
using (var entities = new EFBankEntities())
{
    var query = from klant in entities.Klanten.Include("Rekeningen")
                orderby klant.Voornaam
                select klant;

    foreach (var klant in query)
    {
        Console.WriteLine(klant.Voornaam);
        var totaleSaldo = Decimal.Zero;

        foreach (var rekening in klant.Rekeningen)
        {
            Console.WriteLine("{0}: {1}", rekening.RekeningNr, rekening.Saldo);
            totaleSaldo += rekening.Saldo;
        }

        Console.WriteLine("Totaal: {0}", totaleSaldo);
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



### 19.3 TAAK 03 : ZICHTREKENING TOEVOEGEN

```
using (var entities = new EFBankEntities())
{
    var query = from klant in entities.Klanten orderby klant.Voornaam select klant;

    foreach (var klant in query)
    {
        Console.WriteLine("{0}: {1}", klant.KlantNr, klant.Voornaam);
    }
    try
    {
        Console.Write("KlantNr:");

        var klantNr = int.Parse(Console.ReadLine());
        var klant = entities.Klanten.Find(klantNr);

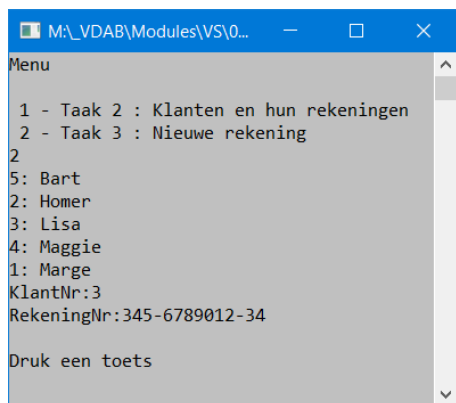
        if (klant == null)
        {
            Console.WriteLine("Klant niet gevonden");
        }
        else
        {
            Console.Write("RekeningNr:");
        }
    }
}
```

```

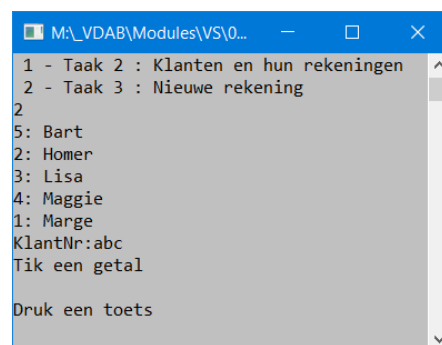
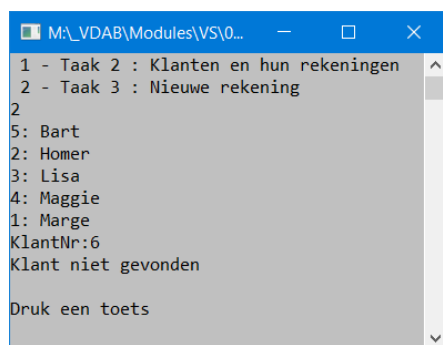
        var rekeningNr = Console.ReadLine();
        var rekening = new Rekening { RekeningNr = rekeningNr, Soort = "Z" };

        klant.Rekeningen.Add(rekening);
        entities.SaveChanges();
    }
}
catch (FormatException)
{
    Console.WriteLine("Tik een getal");
}
}

```



	RekeningNr	KlantNr	Saldo	Soort
	123-4567890-02	1	1000,00	Z
	234-5678901-69	1	2000,00	S
	345-6789012-12	2	500,00	S
▶	345-6789012-34	3	0,00	Z
*	NULL	NULL	NULL	NULL



## 19.4 TAAK 04 : STORTEN

```

using (var entities = new EFBankEntities())
{
    Console.Write("RekeningNr:");
    var rekeningNr = Console.ReadLine();
    var rekening = entities.Rekeningen.Find(rekeningNr);

    if (rekening == null)
    {
        Console.WriteLine("Rekening niet gevonden");
    }
}

```

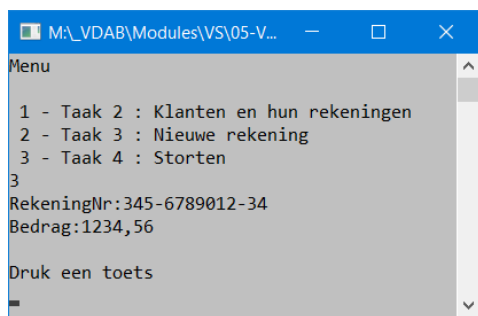
```

else
    try
    {
        Console.Write("Bedrag:");
        var bedrag = decimal.Parse(Console.ReadLine());

        if (bedrag <= Decimal.Zero)
        {
            Console.WriteLine("Tik een positief bedrag");
        }
        else
        {
            rekening.Storten(bedrag);
            entities.SaveChanges();
        }
    }
    catch (FormatException)
    {
        Console.WriteLine("Tik een bedrag");
    }
}

```

	RekeningNr	KlantNr	Saldo	Soort
	123-4567890-02	1	1000,00	Z
	234-5678901-69	1	2000,00	S
	345-6789012-12	2	500,00	S
▶	345-6789012-34	3	0,00	Z
*	NULL	NULL	NULL	NULL



	RekeningNr	KlantNr	Saldo	Soort
	123-4567890-02	1	1000,00	Z
	234-5678901-69	1	2000,00	S
	345-6789012-12	2	500,00	S
▶	345-6789012-34	3	1234,56	Z
*	NULL	NULL	NULL	NULL

## 19.5 TAAK 05 : KLANT VERWIJDEREN

```

try
{
    Console.Write("KlantNr:");
    var klantNr = int.Parse(Console.ReadLine());

    using (var entities = new EFBankEntities())
    {
        var klant = entities.Klanten.Find(klantNr);

        if (klant == null)
        {
            Console.WriteLine("Klant niet gevonden");
        }
        else
        {

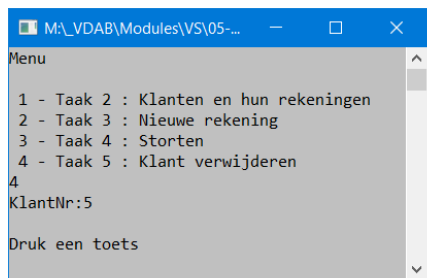
```

```

        if (klant.Rekeningen.Count != 0)
        {
            Console.WriteLine("Klant heeft nog rekeningen");
        }
        else
        {
            entities.Klanten.Remove(klant);
            entities.SaveChanges();
        }
    }
}
}
catch (FormatException)
{
    Console.WriteLine("Tik een getal");
}
}

```

	KlantNr	Voornaam
	1	Marge
	2	Homer
	3	Lisa
	4	Maggie
▶	5	Bart
*	NULL	NULL



	KlantNr	Voornaam
	1	Marge
	2	Homer
	3	Lisa
	4	Maggie
▶*	NULL	NULL

## 19.6 TAAK 06 : OVSCHRIJVEN

Class SaldoOntoereikendException.cs

```

using System;

namespace EFTaken
{
    class SaldoOntoereikendException: Exception
    {
    }
}

```

Een extra method in de partial class Rekening (RekeningUitbreiding.cs)

```

namespace EFTaken
{
    public partial class Rekening

```

```
{
    public void Storten(decimal bedrag)
    {
        this.Saldo += bedrag;
    }

    public void Overschrijven(Rekening naarRekening, decimal bedrag)
    {
        if (this.Saldo < bedrag)
        {
            throw new SaldoOntoereikendException();
        }
        else
        {
            this.Saldo -= bedrag;
            naarRekening.Saldo += bedrag;
        }
    }
}
```

### Program.cs

```
.....
using System.Transactions;
.....

Console.WriteLine("RekeningNr. van rekening:");
var vanRekeningNr = Console.ReadLine();

Console.WriteLine("RekeningNr. naar rekening:");
var naarRekeningNr = Console.ReadLine();

try
{
    Console.WriteLine("Bedrag:");
    var bedrag = decimal.Parse(Console.ReadLine());

    if (bedrag <= decimal.Zero)
        Console.WriteLine("Tik een positief bedrag");
    else
    {
        var transactionOptions = new TransactionOptions
        {
            IsolationLevel = IsolationLevel.RepeatableRead
        };

        using (var transactionScope = new TransactionScope(
            TransactionScopeOption.Required, transactionOptions))
        {
            using (var entities = new EFBankEntities())
            {
                var vanRekening = entities.Rekeningen.Find(vanRekeningNr);

                if (vanRekening == null)
                    Console.WriteLine("Van rekening niet gevonden");
                else
                {
                    var naarRekening = entities.Rekeningen.Find(naarRekeningNr);

                    if (naarRekening == null)
                        Console.WriteLine("Naar rekening niet gevonden");
                    else
                    {
                        try
                        {
                            vanRekening.Overschrijven(naarRekening,
                                                            bedrag);
                            entities.SaveChanges();
                            transactionScope.Complete();
                        }
                    }
                }
            }
        }
    }
}
```

```

        catch (SaldoOntoereikendException)
        {
            Console.WriteLine("Saldo ontoereikend");
        }
    }
}

catch (FormatException)
{
    Console.WriteLine("Tik een bedrag");
}
}

```

(Vergeet niet: Add Reference..., Assemblies, Framework, **System.Transactions** aanvinken)

	RekeningNr	KlantNr	Saldo	Soort
	123-4567890-02	1	1000,00	Z
	234-5678901-69	1	2000,00	S
►	345-6789012-12	2	500,00	S
	345-6789012-34	3	1234,56	Z
*	NULL	NULL	NULL	NULL



	RekeningNr	KlantNr	Saldo	Soort
	123-4567890-02	1	1000,00	Z
	234-5678901-69	1	1900,00	S
►	345-6789012-12	2	600,00	S
	345-6789012-34	3	1234,56	Z
*	NULL	NULL	NULL	NULL

## 19.7 TAAK 07 : KLANT WIJZIGEN

## Aanpassingen aan het edm

Je wijzigt de property **Concurrency Mode** van de property **Voornaam** van de class **Klant** naar **Fixed**.

Opmerking:

Je hoeft dit niet te doen voor de property **KlantNr.**

De bijbehorende kolom is een **autonumber**-kolom en kan dus niet gewijzigd worden bij het wijzigen van een record.

```
using System.Data.Entity.Infrastructure;
```

```

. . . . .
Console.Write("KlantNr:");

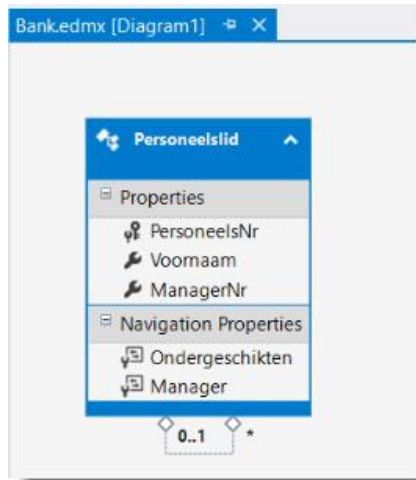
try
{
    var klantNr = int.Parse(Console.ReadLine());

    using (var entities = new EFBankEntities())
    {
        var klant = entities.Klanten.Find(klantNr);

        if (klant == null)
        {
            Console.WriteLine("Klant niet gevonden");
        }
        else
        {
            Console.Write("Voornaam:");
            klant.Voornaam = Console.ReadLine();
            entities.SaveChanges();
        }
    }
}
catch (DbUpdateConcurrencyException)
{
    Console.WriteLine("Een andere gebruiker wijzigde deze klant");
}
catch (FormatException)
{
    Console.WriteLine("Tik een getal");
}

```

## 19.8 TAAK 08 : PERSONEEL



```

using (var entities = new EFBankEntities())
{
    var hoogstenInHierarchie = (from personeelslid in entities.Personeel
                                where personeelslid.Manager == null
                                select personeelslid).ToList();

    Afbelden(hoogstenInHierarchie, 0);
}

```

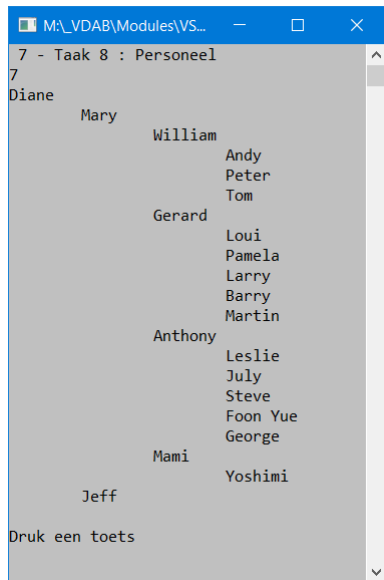
```

static void Afbelden(List<Personeelslid> personeel, int insprong)
{
    foreach (var personeelslid in personeel)

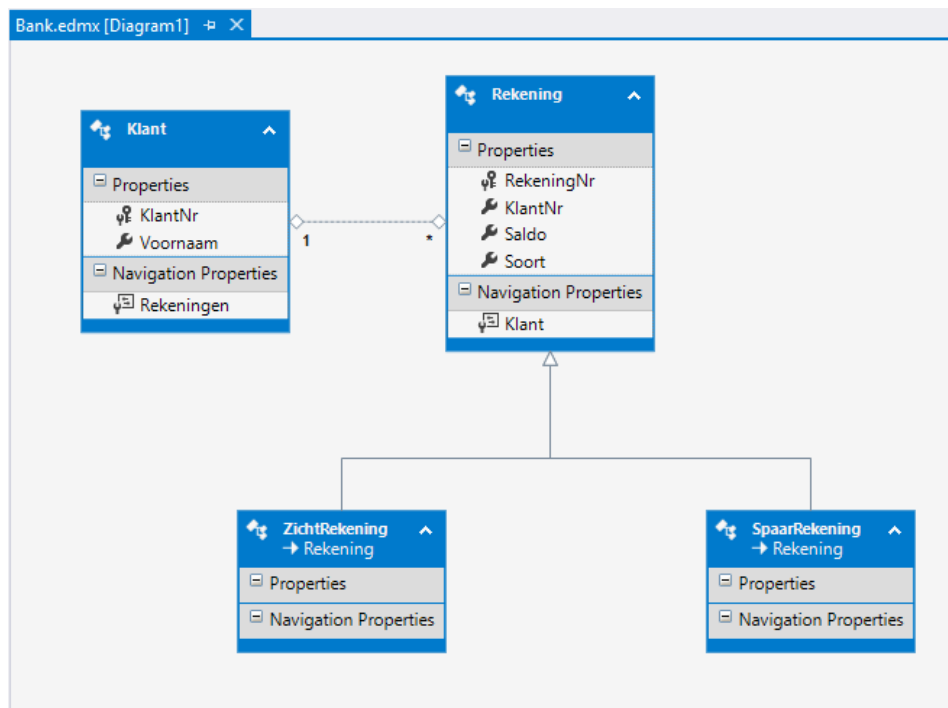
```



```
{  
    Console.WriteLine(new String('\t', insprong));  
    Console.WriteLine(personneelslid.Voornaam);  
  
    if (personneelslid.Ondergeschikten.Count != 0)  
    {  
        Afbeelden(personneelslid.Ondergeschikten.ToList(), insprong + 1);  
    }  
}
```

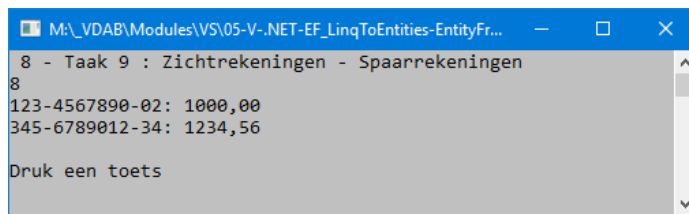


## 19.9 TAAK 09 : ZICHTREKENINGEN – SPAARREKENINGEN



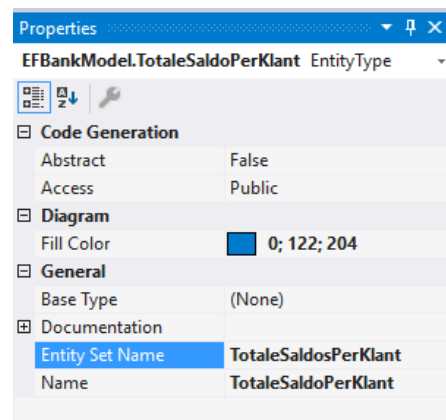
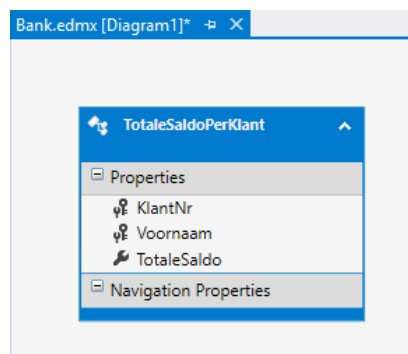
```
using (var entities = new EFBankEntities())
{
    var query =    from rekening in entities.Rekeningen
                   where rekening is ZichtRekening
                   select rekening;

    foreach (var zichtrekening in query)
    {
        Console.WriteLine("{0}: {1}", zichtrekening.RekeningNr, zichtrekening.Saldo);
    }
}
```



RekeningNr	KlantNr	Saldo	Soort
123-4567890-02	1	1000,00	Z
234-5678901-69	1	1900,00	S
345-6789012-12	2	600,00	S
345-6789012-34	3	1234,56	Z

## 19.10 TAAK 10 : TOTALE SALDO PER KLANT



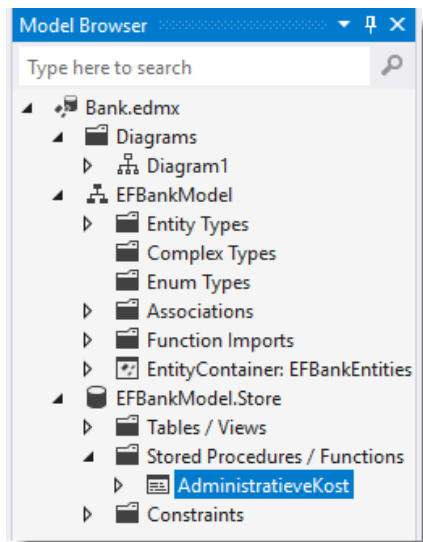
De property **Entity Set Name** is hernoemd naar **TotaleSaldosPerKlant**.

```
using (var entities = new EFBankEntities())
{
    var query =    from totaleSaldoPerKlant
                   in entities.TotaleSaldosPerKlant
                   orderby totaleSaldoPerKlant.Voornaam
                   select totaleSaldoPerKlant;

    foreach (var totaleSaldoPerKlant in query)
    {
        Console.WriteLine("{0}: {1}", totaleSaldoPerKlant.Voornaam,
            totaleSaldoPerKlant.TotaleSaldo);
    }
}
```

## 19.11 TAAK 11: ADMINISTRATIEVE KOST

### Bank.edmx



### Bank.Context.cs

```
//-----  
// <auto-generated>  
//   This code was generated from a template.  
//  
//   Manual changes to this file may cause unexpected behavior in your application.  
//   Manual changes to this file will be overwritten if the code is regenerated.  
// </auto-generated>  
//-----  
  
namespace EFTaken  
{  
    using System;  
    using System.Data.Entity;  
    using System.Data.Entity.Infrastructure;  
    using System.Data.Entity.Core.Objects;  
    using System.Linq;  
  
    public partial class EFBankEntities : DbContext  
    {  
        . . . . .  
  
        public virtual int AdministratieveKost(Nullable<decimal> kost)  
        {  
            var kostParameter = kost.HasValue ?  
                new ObjectParameter("Kost", kost) :  
                new ObjectParameter("Kost", typeof(decimal));  
  
            return ((IObjectContextAdapter)this).ObjectContext.ExecuteFunction("AdministratieveKost",  
kostParameter);  
        }  
    }  
}
```

### Program.cs

```
try  
{  
    Console.Write("Kost:");
```

```

        var kost = decimal.Parse(Console.ReadLine());

        using (var entities = new EFBankEntities())
        {
            Console.WriteLine("{0} rekeningen aangepast",
                entities.AdministratieveKost(kost));
        }
    }
    catch (FormatException)
    {
        Console.WriteLine("Tik een getal");
    }
}

```

	RekeningNr	KlantNr	Saldo	Soort
1	123-4567890-02	1	1000.00	Z
2	234-5678901-69	1	1900.00	S
3	345-6789012-12	2	600.00	S
4	345-6789012-34	3	1234.56	Z

```

10 - Taak 11 : Administratieve kost
10
Kost:55
4 rekeningen aangepast
Druk een toets

```

	RekeningNr	KlantNr	Saldo	Soort
1	123-4567890-02	1	945.00	Z
2	234-5678901-69	1	1845.00	S
3	345-6789012-12	2	545.00	S
4	345-6789012-34	3	1179.56	Z

## 19.12 TAAK 12 : CODE FIRST

### 19.12.1 INSTALLEREN EF

### 19.12.2 ARTIKELGROEP

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;

namespace CodeFirstTaak
{
    [Table("Artikelgroepen")]
    public class Artikelgroep
    {
        public int Id { get; set; }
        public string Naam { get; set; }
        public ICollection<Artikel> Artikels { get; set; }
    }
}

```

### 19.12.3 ARTIKEL

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;

namespace CodeFirstTaak
{
    [Table("Artikels")]
    public abstract class Artikel

```

```
{
    public int Id { get; set; }
    public string Naam { get; set; }
    public virtual Artikelgroep Artikelgroep { get; set; }
    public int ArtikelgroepId { get; set; }
    public ICollection<Leverancier> Leveranciers { get; set; }
}
```

#### 19.12.4 LEVERANCIER

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;

namespace CodeFirstTaak
{
    [Table("Leveranciers")]
    public class Leverancier
    {
        public int Id { get; set; }
        public string Naam { get; set; }
        public ICollection<Artikel> Artikels { get; set; }
    }
}
```

#### 19.12.5 FOODARTIKEL

```
using System.ComponentModel.DataAnnotations.Schema;

namespace CodeFirstTaak
{
    [Table("FoodArtikels")]
    public class FoodArtikel : Artikel
    {
        public int Houdbaarheid { get; set; }
    }
}
```

#### 19.12.6 NONFOODARTIKEL

```
using System.ComponentModel.DataAnnotations.Schema;

namespace CodeFirstTaak
{
    [Table("NonFoodArtikels")]
    public class NonFoodArtikel : Artikel
    {
        public int Garantie { get; set; }
    }
}
```

#### 19.12.7 CONTEXT CLASS

```
using System.Data.Entity;

namespace CodeFirstTaak
{
    class EFTaakContext : DbContext
    {
        public DbSet<Artikelgroep> Artikelgroepen { get; set; }
        public DbSet<Artikel> Artikels { get; set; }
        public DbSet<Leverancier> Leveranciers { get; set; }
    }
}
```

#### 19.12.8 APP.CONFIG

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
    http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework"
    type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>

  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7" />
  </startup>

  <connectionStrings>
    <add name="EFTaakContext"
        providerName="System.Data.SqlClient"
        connectionString="Server=localhost;Database=EFTaak;Trusted_Connection=true;"/>
    <!-- connectionString="Server=.\\SQLEXPRESS;Database=EFTaak;Trusted_Connection=true;"/> -->
  </connectionStrings>

  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
    EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient"
      type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
</configuration>
```

## 19.12.9 PROGRAM

```
using System;
using System.Data.Entity;

namespace CodeFirstTaak
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Data.Entity.Database.SetInitializer(
                new DropCreateDatabaseIfModelChanges<EFTaakContext>());

            using (var context = new EFTaakContext())
            {
                var leverancier = new Leverancier
                {
                    Id = 1,
                    Naam = "Marcel Kiekeboe"
                };

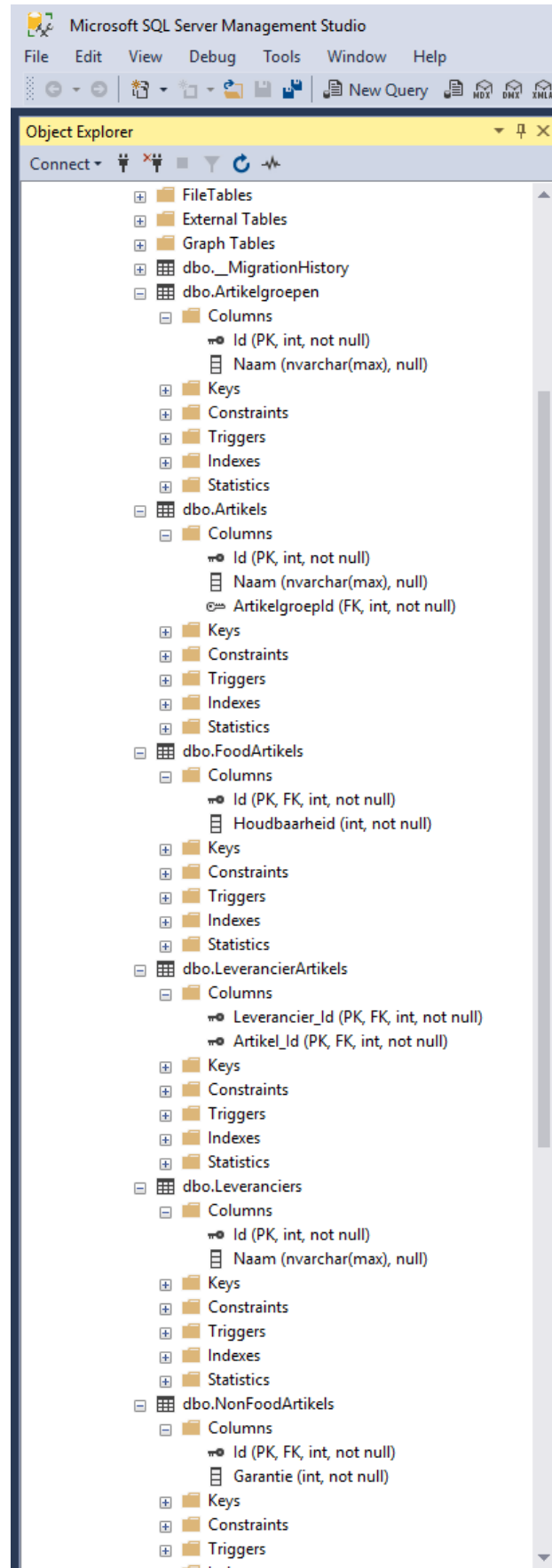
                context.Leveranciers.Add(leverancier);
                context.SaveChanges();

                Console.WriteLine("EFTaak Done");
                Console.ReadKey();
            }
        }
    }
}
```

### 19.12.10 RESULTAAT IN SQL SERVER MANAGEMENT STUDIO

Table **Leveranciers**

	Id	Naam
1	1	Marcel Kiekeboe



## 20 COLOFON

---

Domeinexpertisemanager:	Jean Smits
Moduleverantwoordelijke:	Hans Desmet
Medewerkers:	Hans Desmet Johan Vandaele
Versie:	01-mei-2018
Nummer dotatielijst:	