Exercise 6 - Lathund & Dokumentation

Skapad med förlaga från {Youtube video} *Getting Started with Entity Framework Core | Entity Framework Core 101*
I denna övning skapar vi en Customer-databas:

Installera plug-in

Högerklicka i projektet i Solution Explorer och välj Manage NuGet package
Installera:

- microsoft.EntityFrameworkCore
- microsoft.EntityFrameworkCore.Sql
- microsoft.EntityFrameworkCore.Tools
- möjligen också microsoft.EntityFrameworkCore.Design

Models

Skapa mappen ‘Entities’ (Filmen säger models, men Hans indikerar att det är bäst att hålla isär) - eller varför inte ‘EntityModels’

Skapa klassen **Customer** i ovanstående mapp.

Customer:

```
public class Customer
{
	public int Id { get; set; }   //Id blir automagiskt PK

	[Required]
	public string Name { get; set; }
}
```

option: Markera att **Id** är en PK genom [KEY] (men det behövs inte för just nyckelordet **Id**)
Notera att **Namn** är nödvändigt med [Required]

```
#nullable enable/disable - Ett C# direktiv som kan användas för att
ge motsvarande SQL-egenskap (Not null) . Kan på enskild property
upphävas med hjälp av ‘?’ efter typen (ex: string?)
```

Notera att här används #-direktiv för att slå på och av c# nullable. OM jag fattar rätt, kan man göra undantag genom att sätt frågetecken efter typen, som vid **Address** och **Phone** nedan:

Sista genererade koden rör klassen order, som enklast skapas med intellisense …
Denna skapar ett “one 2 many” förhållande (Men jag fattar inte hur!)

Kod till klassen Customer

```
public class Customer
{
	#nullable enable

	public int Id { get; set; }
	public string FirstName { get; set; }
	public string LastName { get; set; }
	public string? Address { get; set; }
	public string? Phone { get; set; }

	#nullable disable

	public ICollection<Order> Orders { get; set; }
}
```

Data context

DB-filen

Skapa en folder ‘Data’ i projektet
Skapa en databasfil

DBContext

Skapa en ny klass i Data-katalogen: t.ex. DBContext.cs

Kod till DBContext-filen

```
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;
using CodeFirstDB.EntityModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeFirstDB.Data
{
    public class CodeFirstContext : DbContext
    {
        public DbSet<Address> Adresses { get; set; }
        public DbSet<Customer> Customers { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\C-Sharp-fundamentals\cs\Exercise6\CodeFir
stDB\Data\CodeFirstDBFile.mdf;Integrated Security=True;Connect Timeout=30");
        }
    }
}
```

Migration

Packet Manager Console

Skapa databasen

Starta Packet Manager Console via menyn Tools - NuGet Packet Manager - Packet Manager Console

Kommando för att generera script: Add-Migration [migration name] t.ex. Initial-Create

Kontrollera scriptet - Redan här kan man se om något håller på att bli fel.

Kommando för att köra scriptet: Update-Database