

Crearea unui model pentru o bază de date existentă în Entity Framework Core

EF Core nu suportă design vizual pentru DB model. Pentru a obține clasele pentru entități și pentru context, trebuie să folosim inginerie inversă (reverse engineering) folosind comanda **Scaffold-DbContext**. Această comandă va crea entitățile și clasele de context pe baza schemei bazei de date actuale.

Pentru acest exemplu vom folosi baza de date Store, creată într-o sesiune anterioară.

1. Se creează o aplicație .NET Core Console.

2. Instalare EF Core

În aplicația creată la pasul anterior, vom instala EF Core folosind Package Manager Console. Pentru aceasta vom selecta Tools -> NuGet Package Manager -> Package Manager Console și vom executa comanda

```
PM> Install-Package Microsoft.EntityFrameworkCore.SqlServer
```

3. Crearea claselor pentru entități și context

Folosind **NuGet Package Manager Console**, vom crea clasele pentru entități.

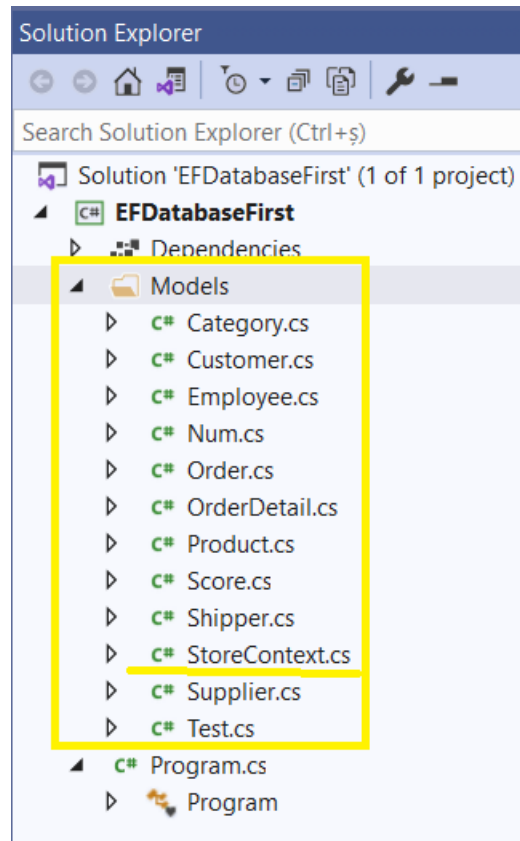
```
PM> Scaffold-DbContext "Server=.;Database=Store;Trusted_Connection=True;"  
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

Comanda conține parametri pentru specificarea serverului de baze de date (Server=.), numele bazei de date (Database=Store), tipul de autentificare (Trusted_Connection = true specifică autentificare Windows), numele providerului (Microsoft.EntityFrameworkCore.SqlServer) și calea unde vor fi salvate clasele realizate (OutputDir Models).

Nota. Dacă comanda nu este recunoscută, atunci introduceți următoarea comandă înaintea:

```
PM> Install-Package Microsoft.EntityFrameworkCore.Tools
```

După rularea comenzii, în soluție se creează clase pentru fiecare entitate în parte.



De exemplu, pentru tabela **Customer** este generate următoare clasă de model:

```
public partial class Customer
{
    public Customer()
    {
        Orders = new HashSet<Order>();
    }

    public int Custid { get; set; }
    public string Companyname { get; set; }
    public string Contactname { get; set; }
    public string Contacttitle { get; set; }
    public string Address { get; set; }
    public string City { get; set; }
    public string Region { get; set; }
    public string Postalcode { get; set; }
    public string Country { get; set; }
    public string Phone { get; set; }
    public string Fax { get; set; }

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

Clasa **StoreContext** va fi folosită pentru a salva sau a obține date din DB.

4. Afișarea datelor

```
static void Main(string[] args)
{
    var context = new StoreContext();
    foreach (var customer in context.Customers)
    {
        Console.WriteLine("{0} - {1} - {2} - {3}",
            customer.Companyname, customer.Contactname,
            customer.Address, customer.City);
    }

    Console.ReadLine();
}
```