

Laborator 5

1. Demo Entity Framework

Varianta in care se pleaca de la o BD pre-existenta. Creati o BD goala Northwind in serverul de baze de date. Rulati apoi in ea scriptul SQL de la <https://github.com/microsoft/sql-server-samples/blob/master/samples/databases/northwind-pubs/instnwnd.sql>

1. Se creeaza proiect (de tip consola), se descarca prin NuGet pachetul Entity Framework. In serverul SQL se creeaza baza de date Northwind, folosind scriptul SQL de la linkul specificat mai sus.
2. Se genereaza automat clasele plecand de la tabele: se creeaza un director Domain Model, se adauga un item de tip ADO.NET Entity Data Model prin varianta Code First from database, se precizeaza detaliile pentru stringul de conexiune. Specificati numele stringului de conexiune diferit de numele modelului, e.g. NorthwindModelConStr (Figura 2). Din tabelele incarcate excludeti tabela diagrams, daca exista (Figura 3).

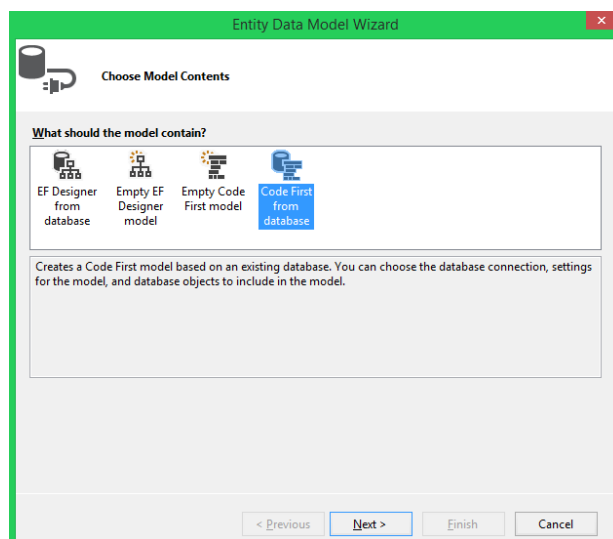


Figura 1. Alegerea strategiei "database first"

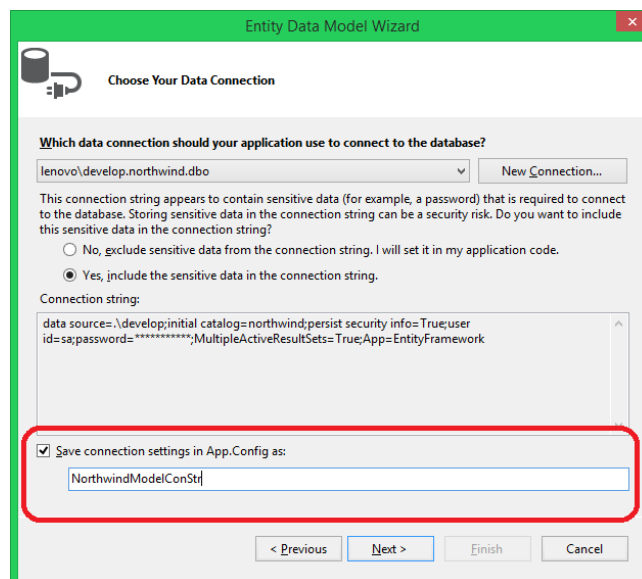


Figura 2. Specificare stringului de conexiune

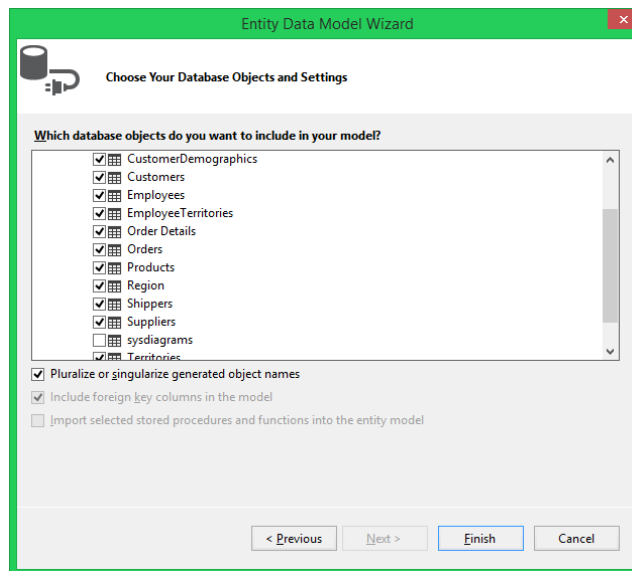


Figura 3. Alegerea tabelelor pentru care se vor genera clase

3. Clasele rezultate raman in folderul DomainModel; clasa NorthwindModel se muta in folderul DataMapper:

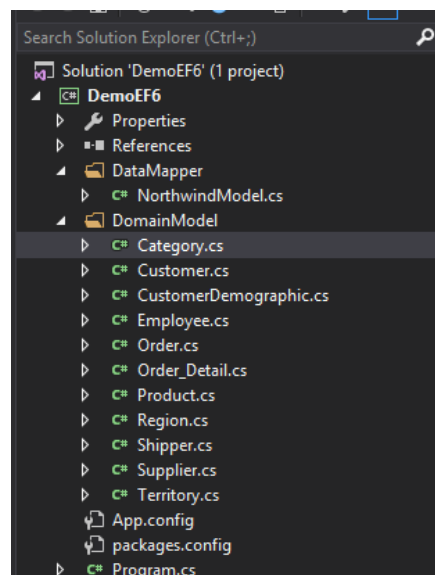


Figura 4. Structurarea codului in cadrul proiectului

4. Cod CRUD – demo la ora.

Exemplu: Solutia din directorul DemoEF_FromDatabase.

2. Modul de lucru “code first”, migrare

In cazul in care nu se furnizeaza o baza de date, se pot crea clase de tip POCO pentru modelarea logicii de domeniu si se genereaza apoi baza de date. Acest mod de lucru permite chiar si modificarea claselor (sau adaugarea de alte clase) si actualizarea automata a bazei de date – migratii - fara pierderi (majore) de date.

Pasi (exemplificare in DemoEF6CodeFirst)

1. Se creeaza proiect nou, se adauga pachet Entity Framework prin NuGet.

2. Se creeaza intr-un folder "Domain model" al proiectului toate clasele de modelare (Product, Category, Customer, Order, OrderLine)
3. Se adauga in app.config un string de conexiune cu referinta la o BD **care poate sa nu existe!** In cadrul stringului de conexiune se specifica si atributul providerName cu valoarea System.Data.SqlClient, pentru ca EF sa stie cu ce tip de server de BD se va lucra.
4. Se creeaza un folder DataMapper in care se scrie o clasa derivata din DbContext; atentie la forma constructorului, cu referire la numele precizat in stringul de conexiune + proprietati *virtuale* de tip DbSet<Entitate>.
5. Demo: listAllProducts() -> daca baza de date nu exista, atunci ea se creeaza automat cu tabele (momentan destul de naive)
6. Adnotarea claselor POCO pentru a impune constrangeri asupra campurilor (lungime, non-nulabilitate, chei straine nenule etc.)
 - a. Se folosesc adnotari din spatiul de nume System.ComponentModel.DataAnnotations, de exemplu:

```
[Required(ErrorMessage = "The name cannot be null")]
[StringLength(100, MinimumLength = 2, ErrorMessage = "The length must be between 2 and 100")]
public String Name
{
    get;
    set;
}
```

- b. Se declara intentia de migrare, pentru a permite reflectarea modificarilor din clase in baza de date, astfel:

Tools -> NuGet package manager -> package manager console in care se da comanda:

Enable-Migrations

Se remarca crearea in cadrul proiectului VS a folderului Migrations cu fisierele timestamp_InitialCreate.cs si Configuration.cs

Odata rulata comanda, nu mai e nevoie sa se mai execute la modificari ulterioare.

- c. Se cere efectiv propagarea modificarilor prin rularea in package manager console a comenzilor:

Add-Migration (si se va cere specificarea unui nume formal pentru migrare; e util a se specifica un nume care sa rezume modificarile facute)

Update-Database

Se constata modificarea corespunzatoare a coloanelor din baza de date (ne-nule, lungime maxima) si aparitia tabeli __MigrationHistory cu detalii despre toate migrarile efectuate pe baza de date.

Perechea de comenzi se va specifica ori de cate ori apar modificari in cod ce trebuie propagate catre baza de date. Incercati sa rulati mai multe comenzi Add-Migration si doar in final Update-Database pentru a vedea ca baza de date poate fi actualizata de la o versiune mult mai veche.

- d. Detalii: <https://msdn.microsoft.com/en-us/data/jj591621.aspx>; de vazut partea de particularizare a migrarilor si utilitatea metodelor Up() si Down() generate in decursul unei migrari.
- e. Documentati-va despre continutul coloanei Model din tabela dbo. __MigrationHistory.

3. Exercitii

1. Entity Framework: parcurgeti si aplicati pasii din capitolele 2 si 3 din [Programming Entity Framework].
2. Probleme pentru BD northwind:
 - a. Pentru fiecare angajat din tabela Employees, care este numarul de subordonati pe care ii are?
 - b. Pentru fiecare regiune in parte, care sunt teritoriile pe care acestea le contine?
 - c. Pentru fiecare angajat, sa se afiseze lista teritoriilor de care raspunde.
 - d. Cum se poate sterge eficient o inregistrare din tabela Employees, dupa valoarea cheii primare, fara a aduce din baza de date inregistrarea vizata?

Indicatie: urmariti solutiile de la <http://stackoverflow.com/questions/8391520/entity-framework-how-do-i-delete-a-record-by-its-primary-key>

 - e. Care angajati au servit un client oarecare? Tratati situatia in care pentru o comanda nu se stie in BD ce angajat a servit-o.
 - f. Care este valoarea unei comenzi (order) specificate?
 - g. Care este valoarea totala a comenzilor din tabela Orders?
 - h. Pentru fiecare categorie de produse, sa se scrie care este lista de furnizori care o pot aduce.
 - i. Care sunt cele mai scumpe 3 produse (pret unitar) care nu sunt marcate ca "discontinued"?
 - j. Care este furnizorul care da cele mai multe produse? Cum rezolvati situatia de balotaj?
 - k. Care sunt cei trei furnizori care dau cele mai multe categorii de produse?
3. Ce se intampla daca in varianta code first se sterge baza de date de pe server, chiar dupa ce s-au facut cateva migrari?
4. Ce se intampla daca in varianta code first se adauga o proprietate read-only in cadrul unei clase – cum se reflecta ea in structura bazei de date in urma migrarii?
5. Cum se pot folosi adnotarile pentru clasele din domain model in vederea validarii? Investigati clasa System.ComponentModel.DataAnnotations.Validator, <https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.validator?view=net-6.0> si <https://www.infoworld.com/article/3543302/how-to-use-data-annotations-in-csharp.html> .