INTERACȚIUNEA DINTRE ENTITY FRAMEWORK ȘI BAZE DE DATE

Victor COVRIG

Departamentul Ingineria Software și Automatică, grupa TI-192, Facultatea Calculatoare, Informatică și Microelectronică, Universitatea Tehnică a Moldovei, Chișinău, Republica Moldova

Autorul corespondent: Victor COVRIG, e-mail: covrig.victor@isa.utm.md

Conducător științific: Dorian SARANCIUC, DISA, FCIM, UTM

Rezumat: *În aceasta lucrare se definește noțiunea de ORM și Entity Framework și se enumeră modalitățile de interacțiune dintre bazele de date si Entity Framework, cu descrierea amanunțită a fiecăruia din ele, dar și se aduce exemple de utilizare în dependență de necesitățile dezvoltatorului. Aceste metode de interacțiune sunt: Code First, Database First și Model First.*

Cuvinte-cheie: *ORM, Database First, Model First, Code First, Entity Data Model*

Introducere

Object Relational Mapper-ul reprezintă un instrument ce realizeaza interactiunea cu baza de date prin intermediul paradigmei obiect-orientată. Conceptul de ORM este implementat utilizand librării (framework-uri) în limbaje ce suportă paradigma obiect-orientată. Exemple de librării ce implementează ORM-ul sunt: Entity Framework (limbajul C#) Sequelize (limbajul JavaScript), Laravel Eloquent (limbajul PHP). Unul din cele mai cunoscute și des utilizate librării ORM ce utilizează tehnologia .Net este Entity Framework [1].

Entiy Framework permite de a abstractiza baza de date și de a lucra cu datele independent de tipul de păstrare lor. În cadrul lucrului direct cu o bază de date se operează cu tabele, indecși, chei primare și secundare. La nivelul de concept ce propune Entity Framework, lucrul cu bazele de date se efectuează prin intermediul obiectelor. Concepția centrală a Entity Framework este entitatea. Entitatea reprezintă un set de date, ce sunt asociate unui anumit obiect [2].

Un alt concept de bază este Entity Data Model (EDM). Acest model suprapune clasele entităților cu tabelele reale. Exista 3 abordări ale EDM: Database First, Model First și Code First.

1. Code First

Code First este modalitatea în care EDM generează baza de date utilizând codul claselor .Net și utilizează Entity Framework pentru a "migra" baza de date atunci când atributele sunt modificate [3]. Această modalitate este potrivită atunci când nu există încă o bază de date a aplicației. Printre beneficiile abordării Code First se numără:

- Codul corespunde întodeauna cu schema tabelelor, astfel necorespondența în tipurile atributelor, definiția tabelelor sunt minime;
- Oricine poate genera tabele dacă cunoaște C#, chiar dacă nu are prea multe cunoștințe in C# (pe de altă parte, este și un dezavantaj, deoarece pot fi stabilite incorect contrângerile de cheie, de tuplu etc.);
- Este o modalitate universală, întrucât lucrează cu diferite sisteme de gestiune a bazelor de date, precum PostgresSQL, MySQL, SQL Server.

Urmatoarea diagramă (Fig. 1) afișează modul în care funcționeaza Code First.

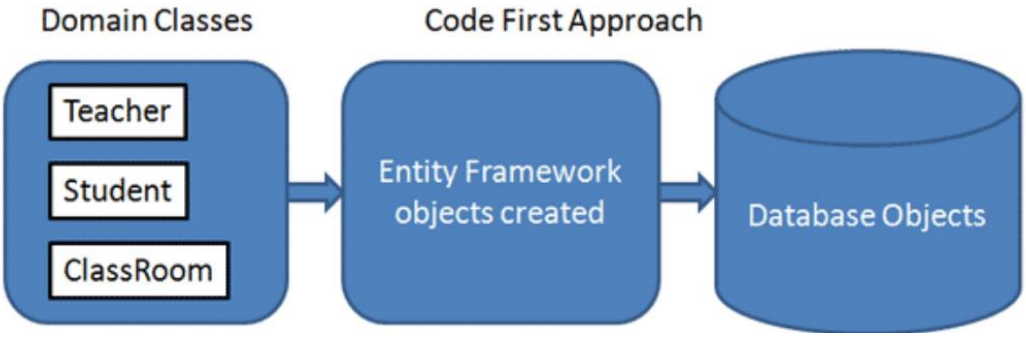

Figura 1. Modul de funcționare Code First

Pe partea de aplicație sunt prezente clasele de domeniu – clasele ce definesc atributele și contrângerile de atribute și de domeniu al fiecărui tabel al bazei de date. Aceste clase interacționează cu clasa de context (DbContext) ce reprezintă punctul principal de interacțiune al bazei de date cu clasele de domeniu. Prin intermediul mecanismului Code First, inclus in cadrul Entity Framework este definit codul de adaugare a tuplurilor noi în tabel și salvarea schimbărilor. După executarea acestui cod este creată baza de date împreună cu toate tabelele definite în cod.

2. Database First

Database First este prima abordare a EDM ce a aparut în Entity Framework. Această abordare este potrivită pentru cazul în care baza de date este deja creată. Acesta abordare prezintă avantaje, comparativ cu Code First în cazul în care avem deja un număr mare de tabele (peste 50):

- Evitarea introducerii manuale a fiecarei definiții a tabelelor, ca în cazul Code First;
- Nu putem utiliza migrațiile deoarece baza de date deja există, iar schimbarile în atribute trebuie gestionate în cod SQL.

Nu este recomandat de a schimba clasele pe care le generează Entity Framework, deoarece acestea vor fi regenerate dupa ce este efectuată urmatoarea rulare a comenzii. Dacă pentru crearea claselor este utilizat IDE-ul Visual Studio, clasele sunt generate pe baza unui fișier edmx, care este afisat cu ajutorul unui „designer”, care este creat odata cu accesarea obțiunii: „Generate from database”.

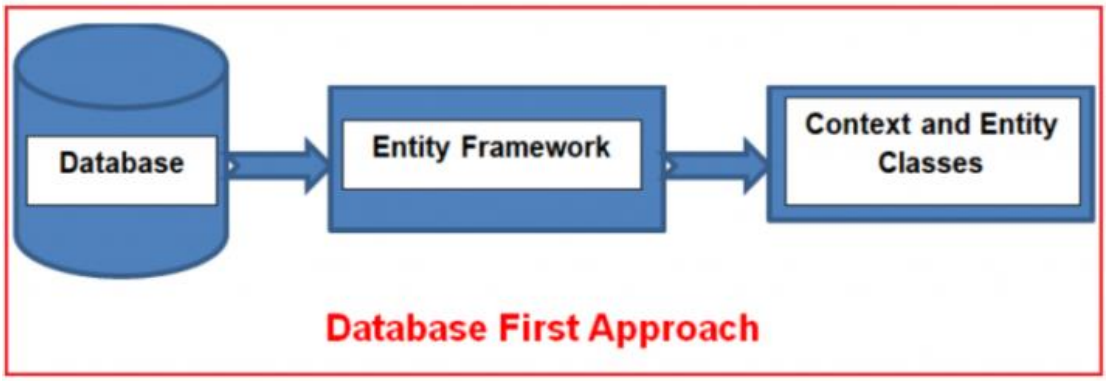

Figura 2. Schema de funcționare Database First

3. Model First

Model First poate fi comparat cu Code First, deoarece ambele abordări contribuie la crearea bazei de date utilizând modele. Însa, Model First nu doar genereaza baza de date, dar și codul claselor de domeniu și contextul bazei de date. Code First presupune crearea manuală a claselor, însă Model First genereaza în mod automat toate datele necsare, chiar și adaugă suportul Entity Framework în aplicație, fără a utiliza configurarea manuală cu ajutorul pachetelor NuGet, așa cum este necesar la Code First [5].

Model First utilizează designer-ul EF, ce prezintă modele din fișiere edmx. Acest deginer este prezent in Visual Studio pentru a genera baza de date si clasele. In figura ce urmeaza este prezentat un exemplu de modele contruite in interfața grafică (Fig. 3). Trebuie menționat că constrângerile de domeniu și de atribut, dar și relațiile între tabele sunt setate tot în interfața grafică.

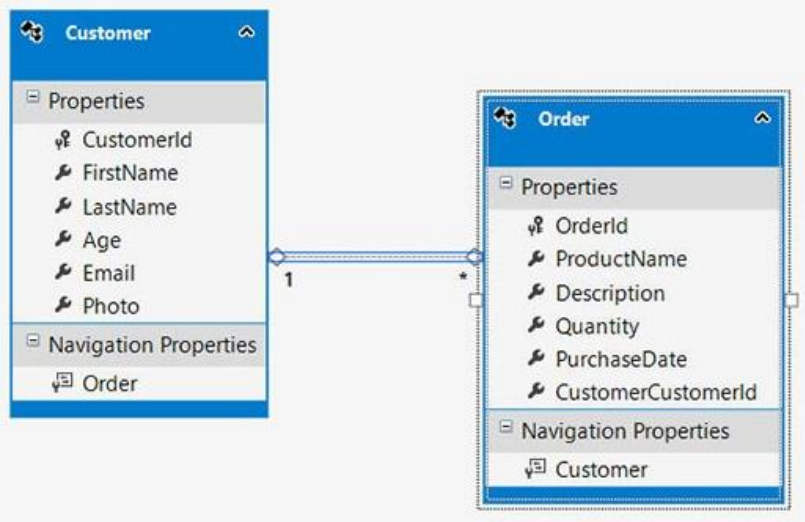

Figura 3. Modele create in EF designer

Următoarea diagramă prezintă schema de funcționare a Model First în generarea claselor de domeniu si bazei de date.

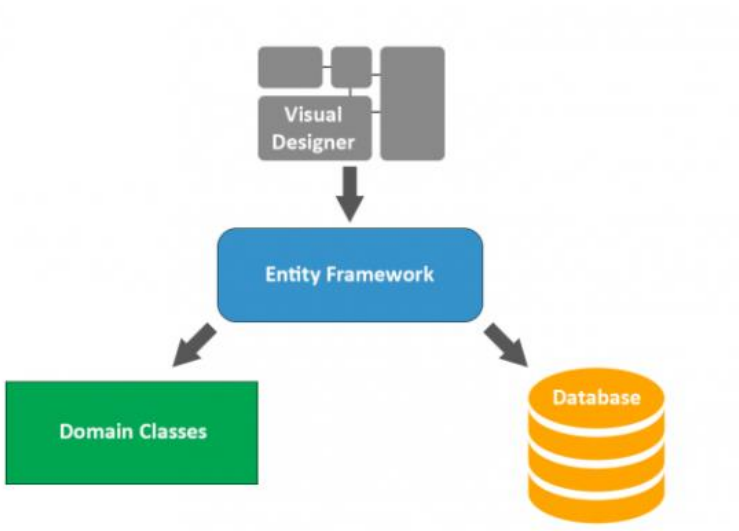

Figura 4. Schema de funcționare Model First

Concluzii

În lucrare au fost expuse toate cele 3 abordări ale Entity Data Model: Code First, Database First și Model First. Toate cele 3 abordări au avantajele și dezavantajele proprii și sunt utilizate în pentru situații diferite. Astfel, Code First este o soluție atunci când nu exista o bază de date, dar există dezvoltatori ce pot crea codul necesar generării bazei de date. Database Model este utilizat când avem deja baza de date creată și este necesar de a genera clasele de domeniu și clasele de context. Model First este o soluție mai universala ce se utilizează pentru a genera în mod automat baza de date și clasele de domeniu și context având la dispoziție doar modelul conceptual al tabelelor bazei de date (atributele tabelelor, contrângeri, relații între tabele).

Bibliografie

1. What is an ORM and Why You Should Use it. [online]. [accesat18.12.2021]. Disponibil: https://blog.bitsrc.io/what-is-an-orm-and-why-you-should-use-it-b2b6f75f5e2a
2. Введение в Entity Framework. [online]. [accesat18.12.2021]. Disponibil: https://metanit.com/sharp/entityframework/1.1.php
3. Code First vs Database First vs Model First – EntityFramework Approaches Explained. [online]. [accesat18.12.2021]. Disponibil: https://dotnetcoretutorials.com/2021/06/26/code-first-vs-database-first-vs-model-first-entityframework-approaches-explained/
4. Reverse Engineering [online]. [accesat 18.12.2021]. Disponibil: https://docs.microsoft.com/en-us/ef/core/managing-schemas/scaffolding?tabs=dotnet-core-cli
5. Использование Model-First. [online]. [accesat 18.12.2021]. Disponibil: https://professorweb.ru/my/entity-framework/6/level1/1_5.php