ENTITY FRAMEWEORK ŞI MAPAREA BAZELOR DE DATE

MURUG Alexandru

Universitatea Tehnică a Moldovei

Abstract: *această lucrare descrie cadrul entity framework (în continuare ef), ce este o platformă de programare pentru a spori nivelul de abstracție logic (relațional) la nivelul conceptual (entitate) și, astfel reduce semnificativ impedanța pentru serviciile de aplicații și date, cum ar fi raportarea, analiza și replicare datelor. Modelul conceptual de date se face printr-un runtime care implementează un model relațional estins (entity data model alias edm), care cuprinde entități și relații ca și concepte de primă clasă, un limbaj de interogare pentru edm, un motor cuprinzător de cartografiere care interpretează datele din modelul conceptul în cel logic (relațional). Edm are și un set de instrumente bazate pe modele care ajută la crearea entitate-obiect, obiect-xml și transformarea în entitate-xml.*

Cuvinte cheie: *SQL Server, Entity Framework, C#, .Net, object-relational mapping, Entity SQL.*

Introducere

Maparea relațional-obiectuală sau obiectual-relațională sau O/RM este o tehnică de programare care leagă bazele de date de limbajele de programare orientate-obiect, creând o bază virtuală de obiecte. Există pachete comerciale și gratuite disponibile pe piață care efectuează maparea între obiect și relațional, deși unii programatori preferă să-și realizeze și utilizeze propriul cod pentru mapare.

Un instrument tipic ORM generează clase pentru interacțiunea bazei de date pentru aplicația dumneavoastră așa cum se arată în figura 1.

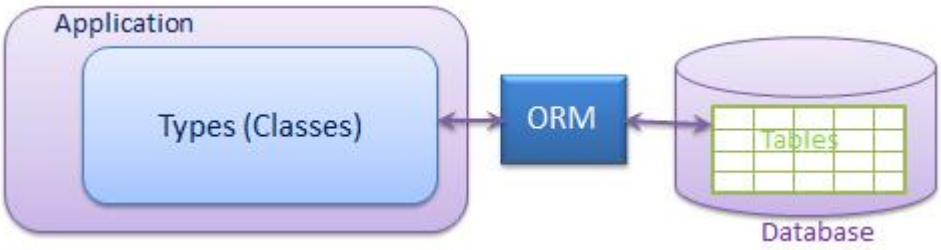

Fig. 1. Generarea claselor din baza de date

Entity Framework este un Framework open source de mapare relațional obiectuală pentru platforma .Net. Per ansamblu este un set de tehnologii în ADO.Net ce susține dezvoltarea de aplicații software orientate pe date. Arhitecții și dezvoltatorii de aplicații orientate pe date s-au străduit să atingă două obiective foarte diferite:

- Trebuie să modeleze entitățile, relațiile și logica de business a problemei pe care o rezolvă;
- Trebuie să lucreze cu motorul de date folosit pentru a stoca și regăsi datele;

Acest Framework le permite să lucreze cu date sub forma unor obiecte și proprietăți specifice domeniului, cum ar fi clientul, sau adresa clientului fără sa fie preocupați de tabelele care stau la baza bazei de date sau coloanele unde vor fi stocate aceste date. Astfel dezvoltatorii pot lucra la un nivel superior de abstractizare a datelor și pot crea și întreține aplicații orientate pe date cu mai puține linii de cod decât aplicațiile tradiționale.

Prima versiune, EFv1 a fost inclusă în .Net Framework 3.5 Service Pack 1 și Visual Studio 2008 Service Pack 1 și lansată în August 2008. Ultima versiune este EF 6.1.3 și este cunoscută formal ca EF7 (1).

ADO.Net Entity Framework folosește o variantă de limbaj structurat de interogare denumit Entity SQL (2), ce a fost creat pentru a scrie interogări și update-uri asupra entităților și relațiilor la nivel conceptual.

1. Arhitectura EF

Următoarea figură prezintă arhitectura generală a cadrului Entity. Mai jos se va descrie componentele arhitecturii în mod individual.

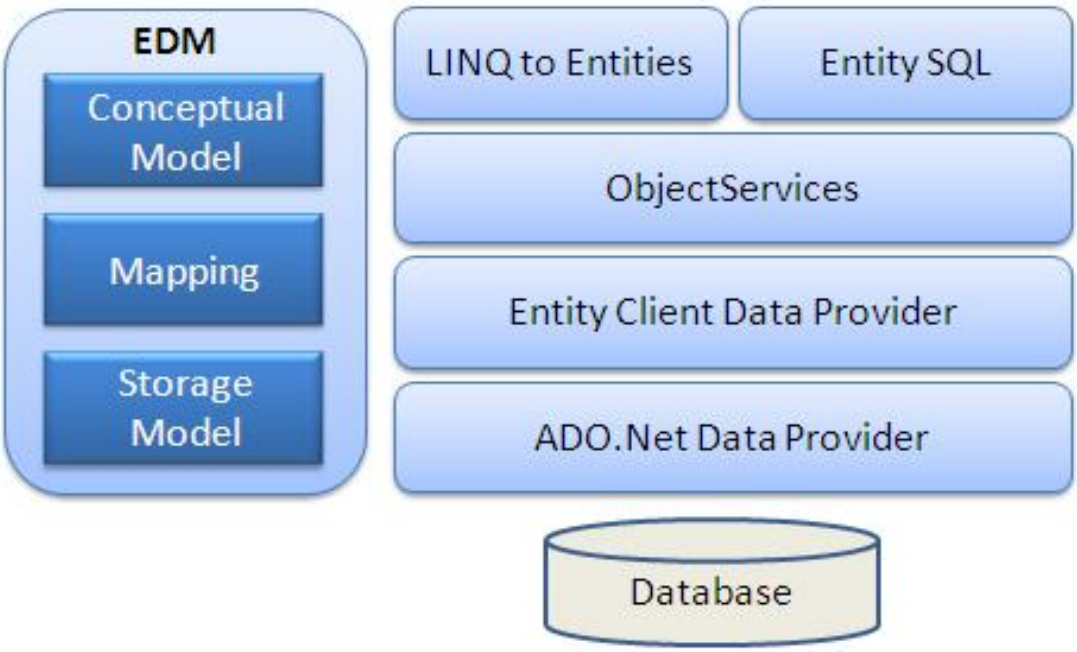

Fig. 2. Arhitectura Entity Framework (3)

- **EDM** (Entitate Date model): EDM este format din trei părți principale - Modelul conceptual, cartografiere și modelul de stocare.
- **Conceptual Model**: Modelul conceptual conține clasele de model și relațiile lor. Acest lucru va fi independent de design-ul tabelului bazei de date.
- **Storage Model**: Modelul de stocare este modelul de proiectare a bazei de date, care include tabele, vizualizări, proceduri stocate, precum și relațiile și cheile.
- **Mapping**: cartografiere constă din informații despre modul în care modelul conceptual este mapat la modelul de stocare.
- **LINQ to Entities**: LINQ entități este un limbaj de interogare utilizat pentru a scrie interogări împotriva modelului de obiect. Returnează entități, care sunt definite în modelul conceptual.
- **Entity SQL**: entitate SQL este un alt limbaj de interogare la fel ca LINQ entități. Cu toate acestea, este un pic mai dificilă decât L2E și dezvoltator va trebui să-l învețe separat.
- **Object Service**: Serviciul de obiect este un punct principal de intrare pentru accesarea datelor din baza de date și returnarea lor. Serviciul de obiect este responsabil pentru materializarea, care este procesul de conversie a datelor returnate de la un furnizor de date client entitate (strat următor) la o structură de obiect entitate.
- **Entity Client Data Provider**: Principala responsabilitate a acestui strat este de a converti L2E sau entitate interogări SQL într-o interogare SQL care este înțeleasă de către baza de date. Comunică cu furnizorul de date ADO.Net, care la rândul său, trimite sau preia datele din baza de date.
- **ADO.Net Data Provider**: Acest strat comunică cu baza de date folosind standardul ADO.Net.Entity Data Model (EDM) specifică modelul conceptual al datelor prin intermediul *Entity-Relationship model*, ce se ocupă în principal de entități și asocieri din care acestea fac parte. În Visual Studio EDM Wizard inițial generează o mapare 1:1 a schemei bazei de date la schema conceptuală. În schema relațională elementele sunt compuse din tabele, în care cheile primare și cheile externe țin tabelele corelate în strânsă legătură. În contrast *Entity Types* definește schema conceptuală a datelor. Tipurile de entități sunt o agregare a unor tipuri de câmpuri multiple, fiecare câmp mapând o coloană din baza de date, câmpuri ce pot conține informații din mai multe tabele fizice. Schema logică și maparea ei pe schema fizică este reprezentată ca un EDM și specificată ca un fișier XML.

2. Trei abordări de dezvoltare ale EF

EF susține trei abordări diferite de dezvoltare, vezi următoarea figură.

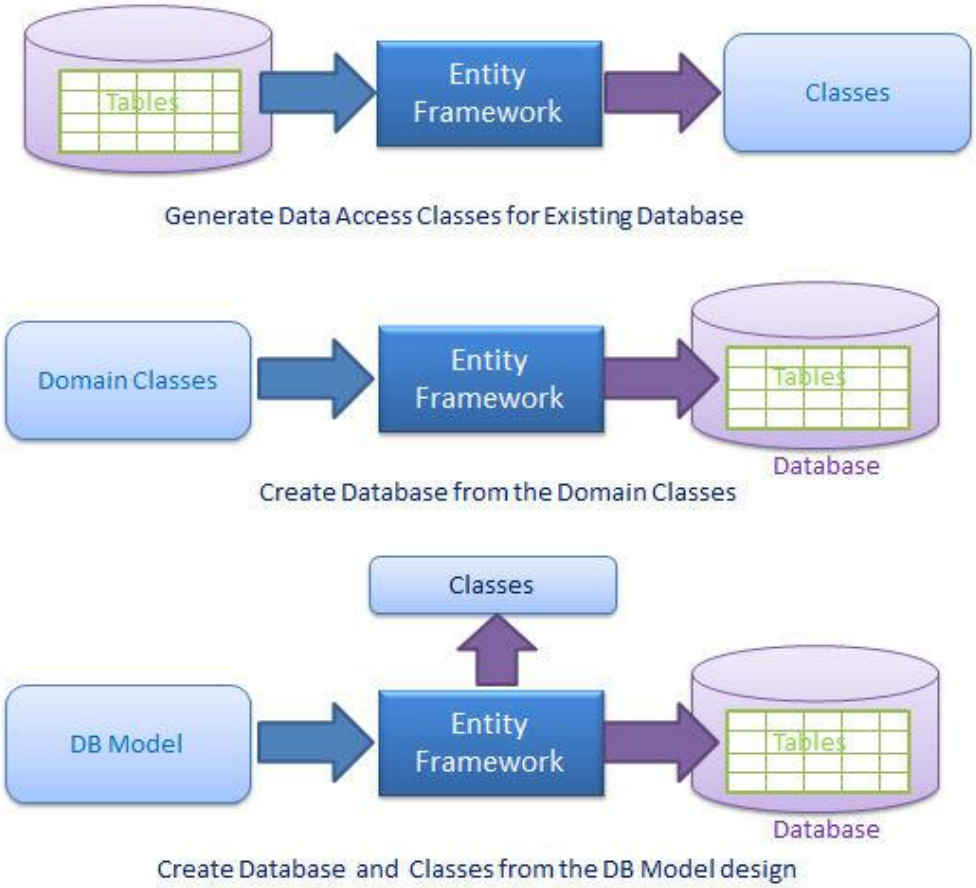

Fig. 3. Ilustrarea modelelor de utilizare a Framework-ului

2.1 Code First

În abordarea Code First, evită complet lucrul cu designerul vizual (EDMX). Aici mai întâi se scriu Clasele POCO, iar apoi se creează baza de date din clasele create.

Dezvoltatorii care urmează principiile calei Domain-Driven Design (DDD), preferă să înceapă prin scrierea codului pentru clasele de domeniu și apoi generarea bazei de date necesare pentru a persista datele.

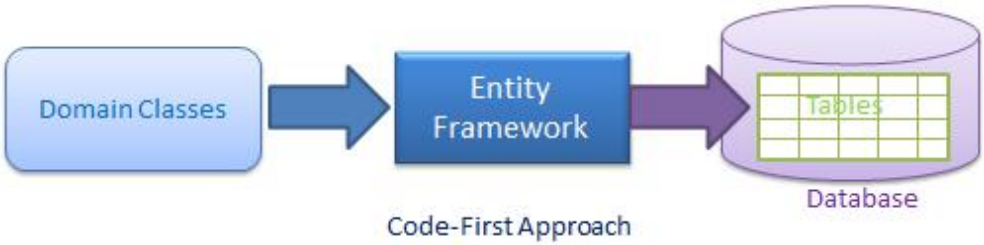

Fig. 4. Diagrama pentru abordarea Code-First

2.2 Model First

În abordarea Model First se creează în primul rând elementele, relațiile și ierarhiile de moștenire direct pe suprafața de proiectare a EDMX-ului și apoi se generează baza de date de la model.

Așa că, în abordarea Model First, se adaugă noul model ADO.NET Entity Data Model și se selectează Empty EF Designer model în Entity Data Model Wizard.

Este posibil de a crea o entitate, asociere și moștenirea pe un designer gol făcând clic dreapta pe suprafața de designer -> Add New -> Entity (vezi figura 10).

Se poate adăuga, de asemenea, proprietăți în entitatea generată făcând clic dreapta pe entitate -> Add New -> Scalar property.

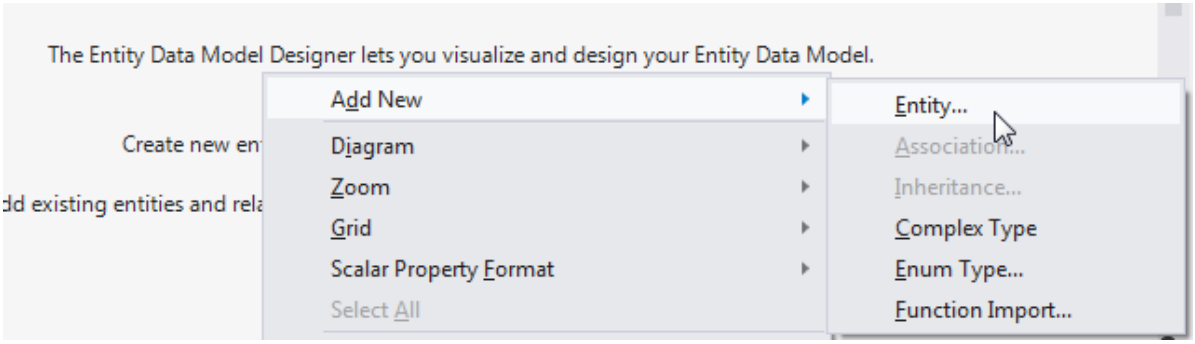

Fig. 10. Adăugarea unei noi entități

După crearea entităților necesare, asocierilor şi moştenirea pe suprafața de proiectare a modelului gol (Emty Model/Mode First), se poate utiliza designerului pentru generarea bazei de date din mode selectând din meniu contextual „Generate database from model" pentru a genera script-ul DDL.

2.3 DataBase First. Crearea Entity Data Model (EDM)

Se va crea un EDM pentru baza de date SchoolDB şi se va explica blocurile de bază. Pentru aceasta trebuie de adăugat EDM-ul, făcând click dreapta pe proiect în Solution Explorer -> Add -> click pe New Items şi se selectează ADO.NET Entity Data Model (4) şi inserează numele modelului şi se face clic pe butonul Add.

Se va alege EF Designer from database şi se face clic pe butonul Next. În următoare fereastră se alege baza de date care se va folosi din conexiunile existente, dacă nu există nici o conexiune creați una nou apăsând pe butonul New Connection. Următorul pas va afişa toate tabele, vizualizări şi procedurile memorate (SP), în baza de date. Selectați tabele, vizualizări şi SP dorite, păstrați căsuțele implicite selectate şi faceți clic pe Finish. Aveți posibilitatea să modificați modelul namespace dacă doriți.După apăsarea butonului Finish în proiect se va adăuga fişierul School.edmx. Apăsând pe el se va deschide diagrama schemei bazei de date.

Concluzii

Entity Framework oferă trei abordări diferite pentru a face față cu modelul dezvoltatorului şi fiecare are propriile sale argumente pro şi contra. EF are încă cota sa de probleme şi nu este larg acceptat încă. Se poate face mai bun prin contribuția la dezvoltarea versiunii următoare. EF poate fi potrivit pentru proiecte noi, atunci când dezvoltatorii ştiu şi iau în considerare limitările ORM şi de a dezvolta proiectele lor.

În cazul în care aveți nevoie de un designer vizual pentru modelele Entity Framework (Database-First şi Model-First) şi " Update Model From Database" se poate fi rezolvată printr-un Entity Framework mode Designer terț, de exemplu, Devart Entity Developer.

EF uşurează munca programatorului considerabil oferindu-i posibilitate să lucreze mai mult asupra altor activități având mapare bazelor de date gata, nepierzând timpul pentru a o crea cu propriile forțe.

Cu atât mai mult EF oferă posibilitatea de a crea baza de date din proiectul deja scris, ceea ce este o modalitate foarte convenabilă dezvoltatorilor care vor să treacă stocare datelor din aplicațiile existente în baze de date. Cu atât mai mult că EF oferă posibilitate de a lucra nu doar cu SQL Sever dar şi cu celelalte baze de date precum Oracle sau Firebird. (5).

Bibliografie

1. Microsoft. Entity Framework Version History. *Data Developer Center.* [Interactiv] [Citat: 16 05 2016.] https://msdn.microsoft.com/en-us/data/jj574253.aspx.

2. Microsoft. Entity SQL Overview. *MSDN.* [Interactiv] [Citat: 13 05 2016.] https://msdn.microsoft.com/en-us/library/bb387145(v=vs.110).aspx.

3. EntityFrameworkTutorial. Entity Framework Architecture. *EntityFrameworkTutorial.* [Interactiv] [Citat: 17 05 2016.] http://www.entityframeworktutorial.net/EntityFramework-Architecture.aspx.

4. Microsoft. ADO.NET Entity Data Model Designer. *MSDN.* [Interactiv] [Citat: 17 05 2016.] https://msdn.microsoft.com/en-us/library/cc716685(v=vs.100).aspx.

5. .NET Foundation. NuGet Gallery | Firebird Entity Framework Provider 5.0.5. *NuGet.* [Interactiv] [Citat: 20 05 2016.] https://www.nuget.org/packages/EntityFramework.Firebird/.