UTILIZAREA EFICIENTĂ A BD PE PLATFORMA .NET

BÎTCA Ernest

Universitatea Tehnică a Moldovei

***Abstract**: În articol este descrisă utilizarea unui Micro ORM - Dapper. Sunt indicați pașii de instalare și de utilizare. Totodată, sunt indicate cazurile cînd trebuie folosit un Micro ORM și cazurile cînd trebuie utilizat un ORM. Toate acestea au ca scop folosirea eficientă a bazelor de date pe platforma .NET. Este descrisă performanța utilizării și totodată securitatea interogărilor. Utilizarea Dapper este foarte simplă și eficientă, metodele sunt intuitive și scalabile sub diferite tipuri de date. Odată cu folosirea Dapper se reduce volumul codului din program, devenind mai citibil și mai ușor de editat.*

***Cuvinte cheie:** ORM, Micro ORM, Dapper, .NET, SQL.*

1. ORM, Micro ORM

1.1 ORM (Object Relational Mapping) reprezintă o tehnologie de programare care leagă bazele de date cu conceptul orientat pe obiect a limbajelor de programare (Nhibernate, Entity Framework). Prin alte cuvinte, ORM reprezintă proiectarea modelului relațional într-o clasă.

În figura 1 se observă conceptul relațional și cel orientat pe obiecte. Schema *Persons* are o cheie externă *AdressId* ce indică spre tabela *Adresses* (atributul Id).

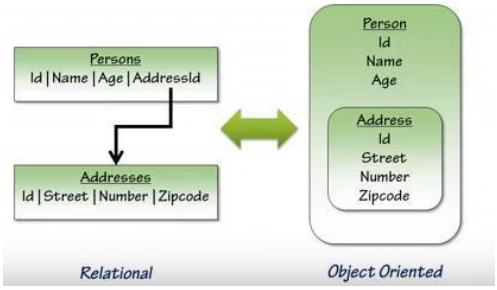

Fig. 1. Conceptul ORM

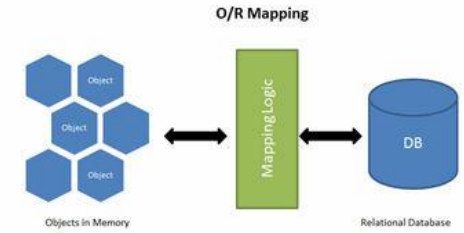

Fig. 2. Principiul de lucru a mapării

1.2 Micro ORM (Micro Object Relational Mapping) – este o tehnologie similară cu ORM, doar că are unele limitări în funcționare cu scopul de a simplifica lucrul, de a o face mai portabilă și mai rapidă (Dapper).

Method	Duration
Hand coded (using a `SqlDataReader`)	47ms
Dapper `ExecuteMapperQuery`	49ms
PetaPoco	52ms
NHibernate SQL	104ms
Entity framework	631ms

Fig. 3. Performanța ORM, Micro ORM

Au fost efectuate teste pentru a determina performanța, în cazul dat s-a testat viteza de mapare pentru o iterare (repetată de 500 de ori) pentru comanda SELECT. Liderul în testul dat s-a dovedit a fi Dapper. Din această cauză, în continuare se va descrie lucrul cu acest instrument pe platforma .NET.

1.3 Neajunsurile Micro ORM sunt:

-Lipsa sistemului încorporat de depozitare a datelor (Cache), care trebuie implementat manual.
-Nu are designer. Majoritatea ORM au un designer care permite crearea modelului, vizualizarea relatiilor.
-Nu sunt migrații. Nu se suportă conceptul Code-First.

2. Instalare Dapper

2.1 Instrumentele necesare sunt:

- Microsoft SQL Server (cu o bază de date deja populată). Microsoft Visual Studio.

Pasul 1.1 - Crearea unui proiect simplu în Visual Studio (Console Application).
Pasul 1.2 – Adăugarea pachetelor.

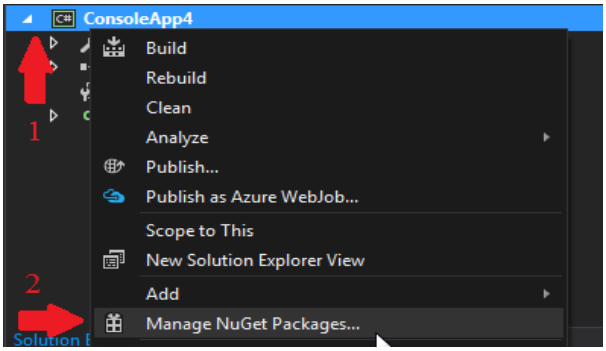

Fig. 4. Instalare

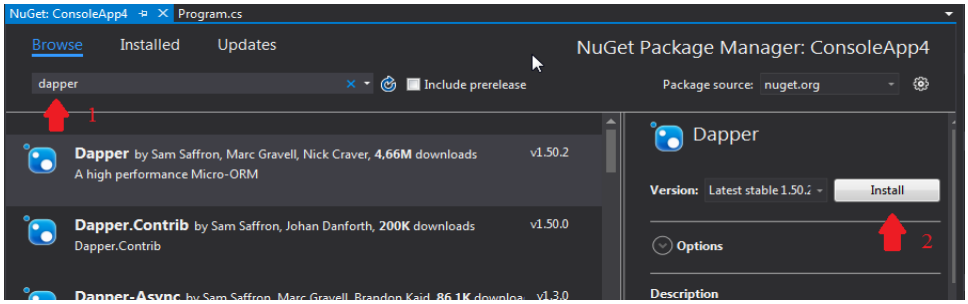

Fig. 5. Instalare

2.2 Conectarea cu BD

La acest capitol, Dapper este deja instalat, însă nu este conexiunea cu baza de date. Se consideră o bază de date care corespunde urmatorului model(Figura 6).

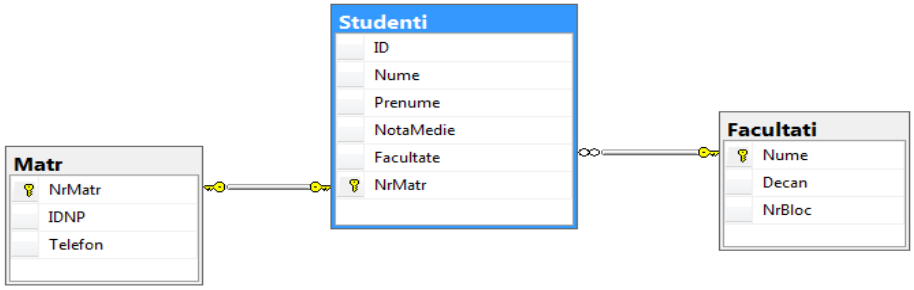

Fig. 6. Diagrama bazei de date

Pasul 1 – Adăugarea conexiunii cu baza de date(Figura 7)

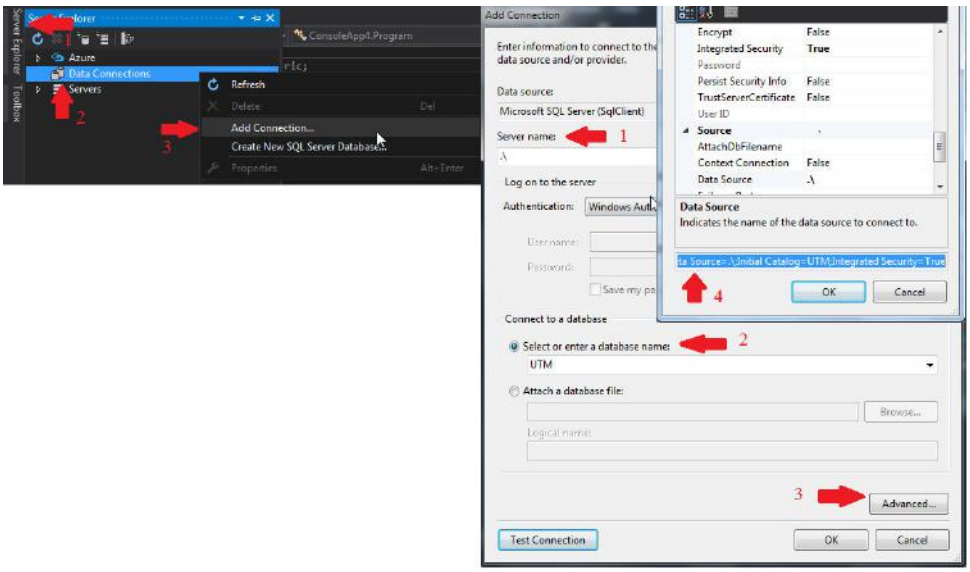

Fig. 7. Conectare pasul

În pasul următor trebuie să se indice anume cu care BD se face conexiunea, dar totodată se copie și șirul de conexiune care se va folosi în program pentru a permite programului să comunice cu BD.

Pasul 2 – deschiderea conexiunii în program (Figura 8)

```
namespace ConsoleApp4
{
    class Program
    {
        static void Main(string[] args)
        {
            using (SqlConnection conn =
                new SqlConnection(@"Data Source =.\; Initial Catalog = UTM; Integrated Security = True"))
            {
                // aici lucram cu datele din BD
            }
        }
    }
}
```

Fig. 8. Conectare

3. Interogări simple

După ce a fost instalat Dapper Micro ORM și după ce a fost creată o conecțiune cu baza de date, nu rămîne altceva decât utilizarea. Se presupune că trebuie să se extragă și să se afișeze informația studentului, NrMatr al căruia este introdus de la tastatură. Înainte de a executa această interogare trebuie să se reproducă echivalentul modelului relațional în conceptul orientat pe obiect (Figura 9).

```
class Student
{
    public string Nume { get; set; }
    public string Prenume { get; set; }
    public decimal NotaMedie { get; set; }
    public string Facultate { get; set; }
    public string Grupa { get; set; }
    public int NrMatr { get; set; }
}
```

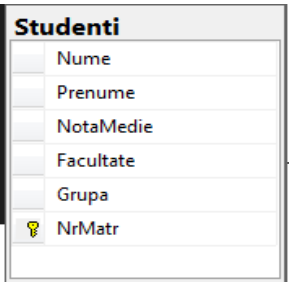

Fig. 9. Modelul de clasă pentru relația Studenți

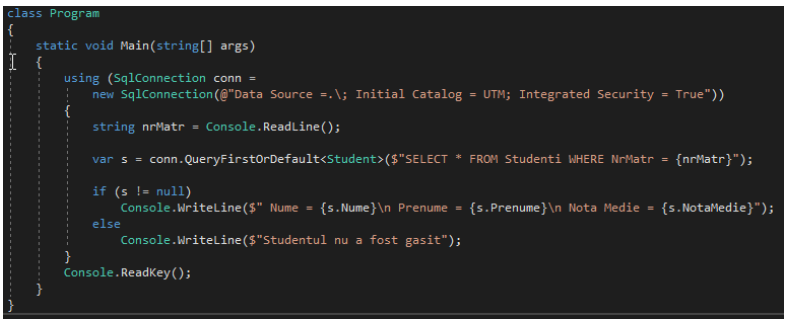

Fig. 10. Exemplu de program

4. Prevenirea SQL Injection

Prin SQL injection nu se înțelege altceva decât introducerea unor comenzi dăunătoare în interogările programului dat, prin mediul exterior. În cazul dat, dacă utilizatorul va introduce următoarele date:

4421322; DELETE FROM Studenti; Întreaga interogare va fi de forma:

SELECT * FROM Studenti WHERE NrMatr = 4421322; DELETE FROM Studenti;

Ca rezultat se pierd datele din BD. Cu scopul de a preveni astfel de injectări se folosesc interogări parametrizate. Prin urmare interogările sunt înlocuite astfel:

conn.QueryFirstOrDefault<Student>($"SELECT * FROM Studenti WHERE NrMatr = {nrMatr}");

conn.QueryFirstOrDefault<Student>($"SELECT * FROM Studenti WHERE NrMatr = @NrMatr",new { NrMatr = nrMatr});

5. Tipuri de interogări Dapper

Execute – execută o comandă şi returnează numărul de rînduri afectate. De obicei, este folosită pentru execuția procedurilor, comenzilor INSERT, UPDATE, DELETE.

Query – execută o interogare şi mapează rezultatul.

QueryFirst – excută o interogare şi mapează primul rezultat.

QueryFirstOrDefault – excută o interogare şi mapeaza primul rezultat, în caz că nu sunt elemente, returnează valoarea default.

QuerySingle – execută o interogare şi mapează primul rezultat, dacă sunt mai mult elemente decît unul, aruncă excepție.

QuerySingleOrDefault – execută o interogare şi mapeaza primul rezultat, dacă sunt mai mult elemente decît unul, afişează excepție, dacă secvența este pustie, returnează valoarea default.

QueryMultiple – execută mai multe interogări din aceiaşi comandă şi mapează rezultatul.

Concluzii

În lucrarea dată s-a trasat o paralelă dintre ORM şi Micro ORM, s-au arătat avantajele şi dezavantajele. Totuşi, dacă se dorește o comunicare cu BD mai rapidă, trebuie să se sacrifice cu unele lucruri. Dapper este foarte uşor de folosit şi este foarte rapid, permite şi maparea relațiilor cu chei externe. În lucrare au fost reflectate cazurile de bază a folosirii interogărilor. Totodată, a fost reprezentată şi instalarea pe pași a bibliotecii date. În concluzie se poate spune că folosirea eficientă a bazelor de date constă şi în modul de comunicare cu baza de date. Viteza nu trebuie limitată la acest capitol. Astfel, se poate garanta utilizatorului o experiență plăcută şi satisfăcătoare.

Bibliografie

1. V.Cotelea. M.Cotelea, Baze de date. Chişinău, 2016.
2. *Dapper documentation,* [Resursă electronică].-Regim de acces: http://www.dapper-tutorial.net 05.11.2017
3. *About Dapper,* [Resursă electronică].-Regim de acces: http://www.github.com/StackExchange/Dapper 05.11.2017
4. *Dapper Wiki,* [Resursă electronică].-Regim de acces: https://en.wikipedia.org/wiki/Dapper_ORM 04.11.2017
5. *Dapper HabraHabr,* [Resursă electronică].-Regim de acces: - https://habrahabr.ru/post/116862/ 04.11.2017