

Database

Obiective

După completarea acestui laborator veți dobândi următoarele cunoștințe:

- Veți putea să vă conectați o baza de date folosind ADO.NET și Entity Framework.
- Veți putea să scrieți query-uri LINQ complexe ce presupun lucrul cu mai multe tabele

Deși lucrul cu baze de date nu este în programa de examen și a fost atins numai tangential prin folosirea LINQ, bazele de date sunt o componentă importantă a aplicațiilor din ziua de azi și dorim să vă însușiți câteva cunoștințe legate de acestea.

Condiții prealabile

- Trebuie să aveți instalat Microsoft SQL Server și Microsoft SQL Management Studio 2012 Express.

Scenariul de laborator

Veți implementa un sistem centralizat de gestiune centralizat al cărților din biblioteci și împrumuturilor acestora de către utilizatori. Acesta menține înregistrări cu toate bibliotecile abonate și toate cartile din acestea. Sistemul mai ține o evidență a tuturor utilizatorilor abonați cu toate împrumuturile acestora precum și numărul de telefon și adresa de email ale acestora, pentru cazul în care nu returnează cărțile la timp. O carte împrumutată trebuie returnată în termen de 14 zile altfel se aplică penalități după următoarea regulă:

- Dacă nu se depășește o săptămână: 1 RON/zi
- Dacă nu se depășesc 30 de zile: 5 RON/săptămână
- Altfel: 50 RON/lună

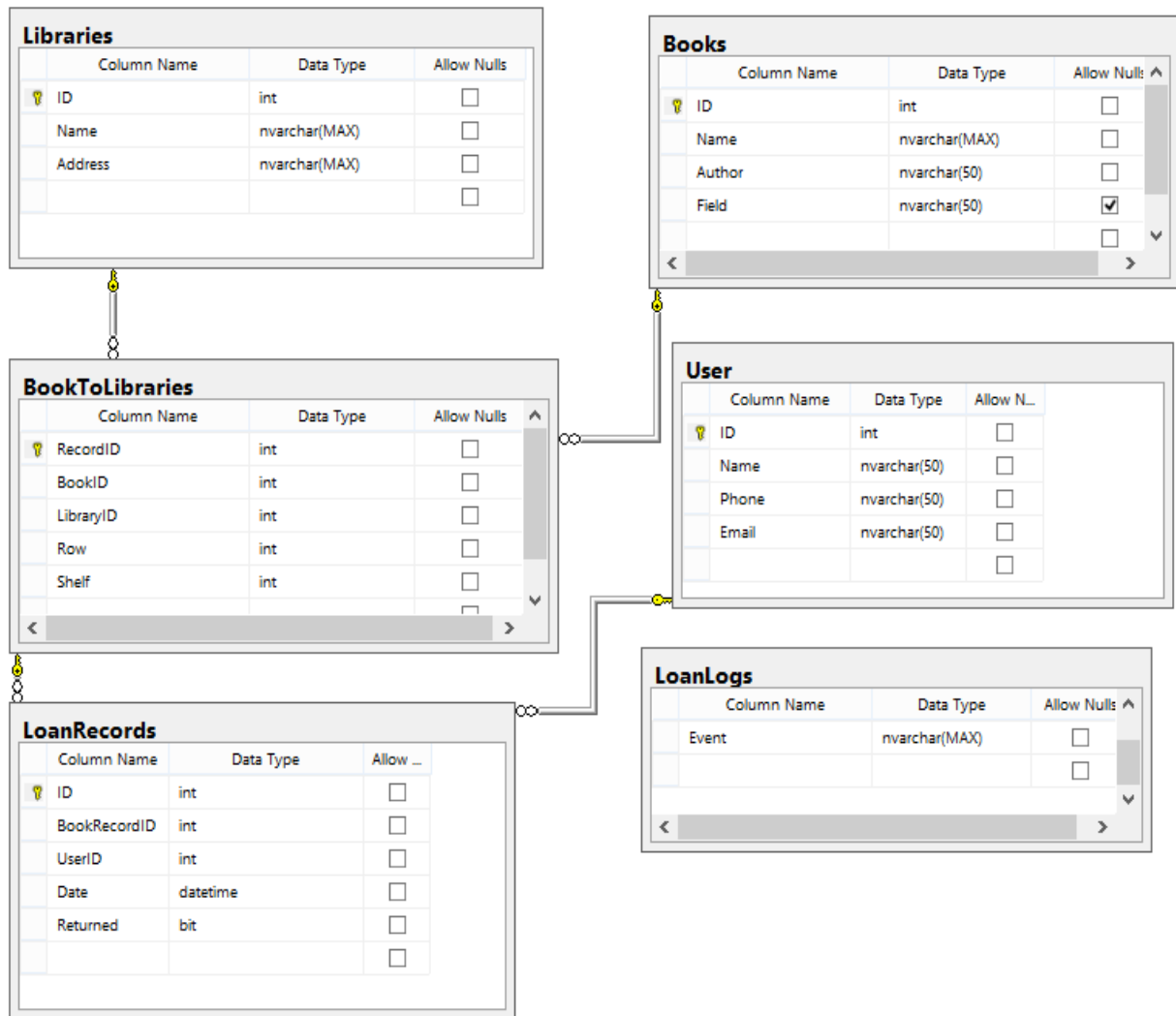
Utilizatori primesc la abonare un card cu numele și ID-ul acestora, credențiale folosite pentru autentificarea în sistem. Un utilizator poate căuta o carte după nume, autor sau domeniu în orice bibliotecă abonată la sistem și o poate împrumuta în cazul în care acesta este disponibilă. De asemenea utilizatorul poate vedea toate împrumuturile pe care le-a făcut de-a lungul timpului începând cu cea mai recentă, precum și dacă a returnat cartea. În plus, acesta poate vedea toate împrumuturile pe care le-a făcut a căror dată scadentă pentru returnare a trecut și numărul de zile de la această dată.

Sistemul mai oferă o funcționalitate administratorului. Acesta poate vizualiza toate împrumuturile din toate bibliotecile a căror dată scadentă a trecut, împreună cu numărul de zile de la data scadentă, penalizarile și datele de contact ale utilizatorului.

➤ Exercițiul 1- Crearea bazei de date

Deschideți Microsoft SQL Management Studio și conectați-vă la serverul local de baze de date. Deschideți o nouă fereastră de query-uri. **File -> New -> Query with Current Connection**

Copiați în aceasta fereastră scriptul SQL din fișierul Database.sql și rulați scriptul. Acesta vă va crea baza de date suport pentru acest laborator. Mai jos, găsiți schema acestora.



○ Structura tabelor

Așa cum este reprezentată și în diagramă, baza de date conține 6 tabele, după cum urmează:

- **Books :**
Tabel ce menține date despre toate cărțile din sistem. Se rețin ID-ul, numele, autorul și domeniul cărții.
- **Libraries :**

Tabel ce conține toate bibliotecile abonate la sistem. Pentru fiecare se reține ID-ul, numele și adresa completă.

- **User :**

Tabel ce conține toți utilizatorii abonați la sistem. Pentru fiecare se reține un ID unic și numele, credențiale trecute și pe cardul primit la abonare și folosite pentru autentificarea în sistem, precum și numărul de telefon și adresa de email.

- **BookToLibraries :**

Tabel de legătură între tabela Libraries și Books. Practic, aici se mențin pentru fiecare bibliotecă (prin LibraryID), cărțile asociate (prin BookID). Tot aici se află și informații despre amplarea cărții dată de rândul (coloana Row) și raftul (coloana Shelf) pe care aceasta se află.

- **LoanRecords**

Tabel în care se mențin toate împrumuturile utilizatorilor. Aici se reține ID-ul împrumutului, cartea și biblioteca din care a fost împrumutată prin coloana BookRecordID ce trimite la tabela BookToLibraries, data la care a fost făcut împrumutul și dacă a fost returnată sau nu.

- **LoanLogs**

Tabela de logging care reține în format plain text evenimentele de împrumut și returnare a cărților de utilizator. Înregistrările sunt introduse de către triggerele de pe tabela LoanRecords.

- **Descrierea constrângerilor de integritate**

Următoarele constrângeri de integritate au fost definite pe coloanele din baza de date:

Primary Key: Fiecare tabelă conține o coloană ID, identificator unic în tabela respectivă. În afară de tabela User, în toate celelalte coloana ID are și constrângerea **Identity**, ceea ce înseamnă că valorile se vor autoincrementa.

Not Null: În afară de coloana Field din tabelul Books, toate coloanele din tabele au această constrângere, nu se permite inserarea valorii NULL.

Default: Dacă nu este specificată o valoare la inserarea unui nou rând în tabelul LoanRecords pentru coloana Returned aceasta va fi setată implicit la False în loc de NULL.

Foreign Key: Aceste constrângeri au fost definite pentru stabilirea relațiilor între tabele după cum urmează:

1. În tabelul BookToLibraries:

- a. BookID este o cheie străină către coloana ID din tabelul Books
- b. LibraryID este o cheie străină către coloana ID din tabelul Libraries

Cu ajutorul acestora se reține pentru fiecare bibliotecă ce cărți deține.

2. În tabelul LoanRecords

- a. BookRecordID este o cheie străină către coloana RecordID din tabelul BookToLibraries
- b. UserID este o cheie străină către coloana ID din tabelul Users

Cu ajutorul acestora se reține pentru fiecare împrumut de către ce utilizator a fost făcut și identifică cartea și libraria din care a fost împrumutată.

➤ Exercițiul 2 – Conectarea la baza de date

Deschideți proiectul Visual Studio din arhiva de pornire.

Pentru a ne conecta la baza de date vom folosi ADO.NET, împreună cu Entity Framework.

Click dreapta pe proiect (**LibrariesDB**) -> **Add** -> **New Item** -> **ADO.NET Entity Data Model**. Numiți-l LibraryDB și dați click pe Add.

În dialogul **Choose Model Contents** selectați **EF Designer from database**. Apoi click pe Next.

În dialogul **Choose Your Data Connection** dați click pe **New Connection...**. Introduceți numele serverului dumneavoastră local în campul **Server name**. Este același server la care v-ați conectat în pasul anterior. În câmpul **Connect to a database**, lăsați opțiunea implicită **Select or enter a database name** și alegeți din lista asociată baza de date **LibraryDB**, cea creată la pasul anterior. Click pe OK, click pe Next.

În dialogul **Choose Your Database Objects and Settings** bifați **Tables** și **Stored Procedures and Functions**. Acest lucru va face ca Entity Framework să vă creeze clase și metode pe care va mapa tabelele din baza de date și, respectiv, procedurile stocate și funcțiile.

Click pe Finish.

În acest moment ar trebui să aveți în un nou Item în partea dreapta numit LibraryDB.edmx. Dacă mergeți pe acesta, ar trebui să vedeți o diagramă cu tabele asemănătoare cu cea de mai sus. Diferența este că acelea sunt tabele în baza de date, iar acestea sunt clase în programul vostru pe care se vor mapa datele din tabelele asociate.

➤ Exercițiul 3 – Implementare

Urmăriți TODO-urile din cod pentru a completa implementarea aplicației.