Proiect Baze de Date

Aplicație cu baze de date relaționale având interfață vizuală

Tudor-Florian Vasiloiu

331 AC



# Aplicația aleasă: Evidența angajaților într-o universitate.

1. Definirea cerințelor

În cadrul unei universități, există mai multe facultăți. Este necesar un tabel pentru a ține evidența acestora. De asemenea, în cadrul unei facultăți, sunt înregistrați studenți ce participă la cursurile oferite de facultatea respectivă.

Am luat exemplul faculății noastre, ce împarte studenții pe mai multe specializări și am considerat că o specializare poate fi urmată în cadrul mai multor facultăți.

Cursurile la care studenții participă sunt predate de mai mulți profesori. Deci între Materie ( sau Subject, așa cum am trecut în tabel ) și profesori există o relație many to many, întrucât un profesor poate preda mai multe materii.

Încă o relație many to many este și între studenți și cursuri deoarece mai mulți studenți participă la mai multe cursuri.

# Proiectarea bazei de date

Ținând seama de ce am prezentat mai sus, am proiectat baza de date în acest fel: tabelele necesare sunt Faculties, Specializations, Students, Student\_Subject(tabel de legătură), Subjects, Subject\_Teacher(tabel de legătură) și Teachers.

Între Faculties și Specializations există o relație many to one, fiindcă o specializare poate exista în cadrul mai multor facultăți. Totodată, mai multe materii pot fi predate la o specializare, deci între Specializations și Subjects este o relatie one to many.

În cadrul unei faculăți, sunt mai mulți studenți, așadar între Faculties și Students există o relație one to many iar mai mulți studenți sunt repartizați la o anumită specializare, astfel că între Students și Specializations este o relație many to one.

Am creat tabele de legătură Student\_Subject și Subject\_Teacher fiindcă sunt necesare pentru realizarea relației many to many între tabelele Students și Subjects, respectiv Subjects și Teacher.

Diagram

Description automatically generated

1. Aplicația cu interfață cu utilizatorul

Aplicația pe care am realizat-o este scrisă folosind limbajul Java, în mediul de dezvoltare Eclipse IDE.

Interogările regăsite în cadrul aplicației sunt următoarele:

* Afișează toate facultatile
  + select \* from Faculties
* Afișează facultatea cu ID-ul specificat
  + select \* from Faculties where Faculty\_ID=" + *faculty\_ID*
* Șterge facultatea cu ID-ul specificat
  + delete from Faculties where Faculty\_ID=?
* Inserează o facultate specificând toate datele
  + insert into Faculties (Faculty\_ID, Name, Establishment\_year) values(?,?,?)
* Afișează numărul de facultăți
  + select count(Faculty\_ID) as number from Faculties
* Afișează Facultatile la care media notelor studentilor este mai mare decat media universitatii
  + select F.Name as Nume from Faculties F join Students S on F.Faculty\_ID = S.Faculty\_ID group by F.Name having avg(S.Grade) > (select avg(Grade) from Students)
* Afișează toți studenții
  + select \* from Students
* Afișează studentul cu ID-ul specificat
  + select \* from Students where Student\_ID=" + *student\_ID*
* Șterge studentul cu ID-ul specificat
  + delete from Students where Student\_ID=?
* Inserează un student specificând toate datele
  + insert into Students (Student\_ID, Faculty\_ID, Specialization\_ID, First\_Name, Last\_Name, Year, Series, Batch, Registration\_Number, Contact, Grade) values(?,?,?,?,?,?,?,?,?,?,?)
* Afișează toți profesorii
  + select \* from Teachers
* Afișează profesorii care predau la facultatea specificată
  + select \* from Teachers where Faculty\_ID=(select Faculty\_ID from Faculties where Name=? )
* Inserează un profesor cu datele următoare
  + insert into Teachers (Teacher\_ID , Faculty\_ID, First\_Name, Last\_Name, Salary, Contact) values(?,?,?,?,?,?)
* Șterge un profesor cu ID-ul specificat
  + delete from Teachers where Teacher\_ID=?
* Afișează numărul de profesori
  + select count(Teacher\_ID) as number from Teachers
* Afișează numărul de profesori care predau o materie
  + select count(Teacher\_ID) as number from Subject\_Teacher where Subject\_ID=(select top 1 Subject\_ID from Subjects where Name = ?)
* Afișeaza profesorii care predau o materie
  + select First\_Name as prenume, Last\_Name as nume from Teachers T join Subject\_Teacher ST on ST.Subject\_ID=(select top 1 Subject\_ID from Subjects where Name = ?)
* Afișează toate materiile
  + select distinct Name from Subjects
* Afișează toate materiile de la o anumită specializare
  + select distinct Su.Name from Subjects Su join Specializations Sp on Su.Specialization\_ID=(select distinct Sp.Specialization\_ID from Specializations where Sp.Name=?)
* Afișează profesorii care predau o materie (specificată)
  + select First\_Name as prenume, Last\_Name as nume from Teachers T join Subject\_Teacher ST on ST.Subject\_ID=(select Top 1 Subject\_ID from Subjects where Name=?)
* Afișează studenții care fac o materie
  + select \* from Students St join Student\_Subject SS on SS.Subject\_ID=(select Subject\_ID from Subjects where Name=?)
* Afișează toate specializările
  + select \* from Specializations
* Afișează toate specializările din cadrul unei facultăți
  + select \* from Specializations where Faculty\_ID=(select Faculty\_ID from Faculties where Name=?)
* Afișează numărul de specializări din cadrul unei facultăți
  + select count(Specialization\_ID) as number from Specializations where Faculty\_ID=(select Faculty\_ID from Faculties where Name=?)
* Afișează numărul de specializări din cadrul universității
  + select count(Specialization\_ID) as number from Specializations
* Modifică numele facultății cu ID-ul (specificat)
  + update Faculties set Name = ? where Faculty\_ID = ?
* Modifică grupa și seria studentului cu numele specificat
  + update Students set Series = ?, Batch = ? where Last\_Name = ?
* Afișează studenții care participă la cursuri predate de cel puțin 2 profesori
  + select S.First\_Name as prenume, S.Last\_Name as nume, S.Batch as grupa, S.Series as serie from Students S join Student\_Subject SS on S.Student\_ID = SS.Student\_ID where SS.Subject\_ID in (select A.Subject\_ID from Subjects A join Subject\_Teacher St on St.Subject\_ID = A.Subject\_ID group by A.Subject\_ID having count(St.Teacher\_ID) > 2)
* Grupa cu cei mai mulți studenți cu media peste 9
  + select S1.Batch as Grupa from Students S1 inner join Students S2 on S1.Batch = S2.Batch group by S1.Batch having avg(S1.Grade) > 9 and count(S1.Student\_ID) >= (select count(S1.Student\_ID) from Students S1 inner join Students S2 on S1.Batch = S2.Batch group by S1.Batch having avg(S1.Grade) > 9)
* Afișati facultățile care au cel puțin 15 studenți la o specializare cu cel puțin 2 materii
  + select F.Name from Faculties F join Students S on F.Faculty\_ID = S.Faculty\_ID group by F.Name having count(S.Student\_ID) >= 15 and count(S.Specialization\_ID) >= 2
* Specializările care au cel puțin două materii predate de același profesor
  + select S.Name from Specializations S join Faculties F on S.Faculty\_ID = F.Faculty\_ID where (select count(T.Teacher\_ID)from Teachers T join Subject\_Teacher ST on T.Teacher\_ID = ST.Teacher\_ID) > 2

Acestea sunt interogările din cadrul aplicației.

1. Interfața cu utilizatorul

Aplicația se deschide rulând codul Java numit Login.java, iar interfața care apare pe ecran este aceasta:

Graphical user interface

Description automatically generated

Se introduc datele pentru a avea acces la interfața prin care se interacționează cu baza de date.

După introducerea datelor, acestea sunt verificate iar în cazul în care au fost introduse date corecte, va fi afișat pe ecran un mesaj

Graphical user interface, application

Description automatically generated

Interfața aceasta se închide în mod automat și se deschide interfața prin care pot aduce modificări bazei de date.

Graphical user interface, application

Description automatically generated

Aceasta este pagina ce apare prima dată și se referă la tabela Faculties. Utilizatorul poate naviga între paginile ce se referă la alte tabele, apâsând pe numele tabelei din partea de sus a ecranului.

Numele butoanelor sunt sugestive și descriu succint funcționalitatea lor.

În continuare, am sa afișez fiecare pagină în parte, fără să descriu funționalitatea fiecărui buton.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Probabil ați observat că atât butoanele din josul paginii cât și ecranul din partea dreaptă sunt comune tuturor tabelelor.

Graphical user interface, text, application

Description automatically generated

Shape, square

Description automatically generated

Acest lucru se datorează faptului că pe ecranul alb sunt afișate rezultatele interogărilor asupra bazei de date iar butoanele nu depind de un tabel anume, deoarece acestea execută interogări complexe asupra bazei de date Universitate. Tocmai de aceea nu le-am asociat unei tabele anume, ci sunt independente de pagina pe care utilizatorul se află.