



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

# PROIECT DE SEMESTRU

la disciplina  
Introducere în Baze de Date

# UNIVERSITATE

**An universitar:** 2019 – 2020

**Proiect de semestru**

**Facultatea:** Automatică și Calculatoare

**Departamentul:** Calculatoare

**Specializarea:** Calculatoare și Tehnologia

Informației

**Coordonator:** Cornea Georgiana Bianca

**Data predării:** 15.01.2021

**Studenti:** Bozdog Raluca – Delia

Lazea Dragoș – Bogdan

**Grupa:** 30223



# **Cuprins**

|  |           |
|--|-----------|
| <b>1. Introducere.....</b>   | <b>3</b>  |
| <b>2. Analiza cerințelor utilizatorilor (specificațiile de proiect).....</b> | <b>3</b>  |
| 2.1. Ipoteze specifice domeniului ales pentru proiect.....                   | 3         |
| 2.2. Organizarea structurată a cerințelor utilizatorului.....                | 5         |
| 2.3. Determinarea și caracterizarea profilurilor de utilizator.....          | 5         |
| <b>3. Modelul de date și descrierea acestuia.....</b>                        | <b>7</b>  |
| 3.1. Entități și atributele acestora – Tabele.....                           | 7         |
| 3.2. Diagrama EER/UML pentru modelul de date complet.....                    | 10        |
| 3.3. Proceduri, triggeri și evenimente.....                                  | 11        |
| 3.3.1. Proceduri.....  | 11        |
| 3.3.2. Triggeri.....   | 19        |
| 3.3.3. Evenimente.....   | 21        |
| 3.4. Normalizarea datelor.....   | 21        |
| 3.5. Cod MySQL.....  | 22        |
| <b>4. Detalii de implementare.....</b>                                       | <b>22</b> |
| <b>5. Concluzii. Limitări și dezvoltări ulterioare.....</b>                  | <b>25</b> |



## **1. Introducere**

Proiectul presupune dezvoltarea unui sistem informatic destinat gestiunii unei platforme de studiu. Scopul aplicației îl reprezintă acela de a facilita accesul la baza de date a unei universități, prin dezvoltarea unei interfețe grafice intuitive. Aplicația poate fi accesată, pe baza unui proces de autentificare, de trei tipuri de utilizatori: studenți, profesori, administratori. Această aplicație vine drept răspuns atât la nevoile studenților și ale profesorilor referitoare la activitățile didactice pe care aceștia le desfășoară, cât a celor din cadrul personalului administrativ al universității, prin oferirea unui număr generos de funcționalități fiecărui tip de utilizator.

Instrumentele software folosite la dezvoltarea aplicației:

- **MySQL Workbench 8.0** – crearea și gestionarea bazei de date, folosind limbajul MySQL, și crearea diagramei UML a bazei de date
- **JDBC - AP** – suport pentru susținerea conexiunii bazei de date cu mediul de programare Java
- **Eclipse IDE for Java Developers – 2020.09** – mediu de programare în limbajul Java, folosit la crearea interfeței grafice
- **StarUML** – realizarea diagramei UML a aplicației

## **2. Analiza cerințelor utilizatorilor (specificatiile de proiect)**

### **2.1. Ipoteze specifice domeniului ales pentru proiect**

Baza de date trebuie să modeleze în mod corect toate entitățile și relațiile dintre acestea astfel încât aceasta să poată susține activitatea unei universități. Crearea și dezvoltarea bazei de date pe

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

care aplicația este fundamentată are drept puncte de plecare următoarele ipoteze:

- Aplicația este destinată spre a fi utilizată de trei tipuri de utilizatori: studenți, profesori și administratori.
- Un utilizator este unic identificat printr-un id, având o serie de alte attribute generale: CNP, nume, prenume, adresă, număr de telefon, e-mail, cont IBAN și numărul de contract.
- Un utilizator de tip student va avea, în particular, attributele specifice referitoare la anul de studiu și numărul de ore pe care trebuie să le susțină.
- Pentru utilizatorul de tip administrator se vor reține suplimentar: cursurile predate, numărul minim și numărul maxim de ore pe care le poate preda și departamentul din care acesta face parte.
- Cursurile sunt predate de mai mulți profesori și au una sau mai multe tipuri de activități (curs, seminar, laborator), o descriere și un număr maxim de studenți participanți.
- Studenții sunt evaluați cu note pentru fiecare tip de activitate și primesc o notă finală, reprezentând media ponderată a tipurilor de activități, împărțirea procentuală pe fiecare tip de activitate în parte fiind stabilită de către profesor.
- Fiecare activitate se desfășoară recursiv între două date, pe o anumită perioadă de timp.
- Există grupuri de studiu pentru diverse materii, destinate aprofundării cunoștințelor din domeniul materiei respective. În cadrul unei activități desfășurate de un grup de studiu poate fi invitat și un profesor pentru a oferi studenților explicațiile necesare.
- Activitățile de grup sunt caracterizate de un număr minim de studenți participanți.



## **2.2. Organizarea structurată a cerințelor utilizatorului**

Baza de date trebuie să satisfacă următoarele funcționalități:

- Stocarea informațiilor referitoare la toate cele trei tipuri de utilizatori;
- Stocarea informațiilor referitoare la materiile predate în cadrul universității;
- Stocarea detaliilor referitoare la acitivitățile de predare din cadrul fiecărei materii;
- Stocarea informațiilor referitoare la grupurile de studiu și la activitățile desfășurate în cadrul acestora;
- Inserarea de noi informații referitoare la utilizatori, materii și activități, prin folosirea de proceduri și triggere specifice sau prin simple interogări SQL;
- Căutarea, modificarea și ștergerea informațiilor deja existente, utilizând o serie de proceduri stocate sau interogări SQL;

Aplicația Java trebuie să satisfacă următoarele funcționalități:

- Afișarea informațiilor din baza de date;
- Căutarea unor informații în baza de date;
- Prelucrarea bazei de date: modificarea informațiilor reținute în baza de date prin apelul procedurilor stocate în cadrul acesteia;
- Adăugarea de noi înregistrări în baza de date: crearea unui nou cont de utilizator, adăugarea unei materii etc.;
- Ștergerea unor înregistrări: ștergerea informațiilor referitoare la utilizatori, renunțarea participării la un anumit curs;

## **2.3. Determinarea și caracterizarea profilurilor de utilizatori**

Funcționalitățile utilizatorului de tip student:

- Căutare curs;



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

- Înscriere la curs;
- Vizualizare note;
- Vizualizare grupuri și membri;
- Trimitere de mesaje pe grup;
- Vizualizare/Descărcare activități curente/din viitor;

Funcționalitățile utilizatorului de tip profesor:

- Adăugare și programare activități;
- Gestionare ponderi note;
- Notare studenți;
- Vizualizare liste studenți și descărcare cataloage;
- Vizualizare/Descărcare activități curente/din viitor;

Funcționalitățile utilizatorului de tip administrator:

- Adăugare, modificare, ștergere de informații în/din baza de date, referitoare la studenți și profesori;
- Căutare utilizatori după nume;
- Filtrare utilizatori după tip;
- Asignare profesori la cursuri;
- Căutare după numele cursului;

Funcționalitățile utilizatorului de tip super-administrator:

- Funcționalitățile utilizatorilor de tip super-administrator sunt aceleași cu cele ale utilizatorilor de tip administrator, cu deosebirea că aceștia pot opera modificări și ștergeri și asupra informațiilor referitoare la administratori.



### **3. Modelul de date și descrierea acestuia**

#### **3.1. Entități și atributele acestora – Tabele**

**utilizator** – Stochează informațiile referitoare la utilizatorii înregistrați în baza de date, indiferent de tipul acestora; atribute: *id*, *cnp*, *nume*, *prenume*, *adresa*, *nrTelefon*, *email*, *iban*, *nrContract*, *tipUtilizator*, *parola*;

**student** – Stochează informațiile specifice utilizatorului de tip student, fiind legat de tabela utilizator prin cheia străină *idUtilizator*; atribute: *id*, *idUtilizator*, *anStudiu*, *nrOre*;

**profesor** – Stochează informațiile specifice utilizatorului de tip profesor, fiind legat de tabela utilizator prin cheia străină *idUtilizator*; atribute: *id*, *idUtilizator*, *nrMinOre*, *nrMaxOre*, *idDepartament*;

**administrator** – Stochează informațiile specifice utilizatorului de tip administrator, fiind legat de tabela utilizator prin cheia străină *idUtilizator*; atribute: *id*, *idUtilizator*;

**superadministrator** – Stochează informațiile specifice utilizatorului de tip super-administrator, fiind legat de tabela utilizator prin cheia străină *idUtilizator*; atribute: *id*, *idUtilizator*;



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

**materie** – Stochează informațiile referitoare la o materie înregistrată în baza de date; attribute: id, *nume*, *descriere*;

**activitatepredare** – Stochează informațiile referitoare la o activitate de predare din cadrul unei materii; attribute: id, *idProf*, *idProfCurs*, *idMaterie*, *tipActivitate*, *pondere*, *dataStart*, *dataFinal*, *recurenta*, *ora*, *nrMaxStudenti*, *nrCurentStudent*;

**inscriereactivitatipredare** – Stochează informațiile referitoare la studenții înscriși activitățile de predare aferente materiilor la care sunt înscriși; attribute: id, *idStudent*, *idActivitate*, *nota*;

**catalog** – Stochează informațiile referitoare la notele finale obținute de studenți la materiile la care aceștia sunt înscriși; attribute: id, *idStudent*, *idMaterie*, *nota*;

**departament** – Stochează informațiile referitoare la departamentele din cadrul facultății; attribute: id, *nume*;

**grupstudiu** – Stochează informațiile referitoare grupurile de studiu existente; attribute: id, *idMaterie*, *nume*;

**listagrupuri** – Stochează informațiile referitoare la grupurile de studiu la care sunt înscriși studenții din baza de date; attribute: id, *idStudent*, *idGrup*;



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

**activitategrup** – Stochează informațiile referitoare la activitățile desfășurate în cadrul grupurilor de studiu; attribute: id, *idGrupStudiu*, *idProfesor*, *dateDesfasurare*, *ora*, *durata*, *tempExpirare*, *nrMinStudenti*, *nrCurentStudenti*, *momentCreare*;

**listaactivitati** – Stochează informațiile referitoare la activitățile de grup disponibile fiecărui student; attribute: id, *idStudent*, *idActivitateGrup*;

**mesaje** – Stochează informațiile referitoare mesajele trimise de studenți în fiecare dintre grupurile de studiu existente; attribute: id, *idUtilizator*, *idGrup*, *mesaj*;

**mesajeroare** – Stochează mesajele de eroare ce pot apărea la utilizarea neconformă a aplicației sau a bazei de date; attribute: id, *mesaj*;

**atribuirecalendargrup** – Realizează corespondența dintre un utilizator și activitățile de grup; attribute: id, *idUtilizator*, *idCalendar*;

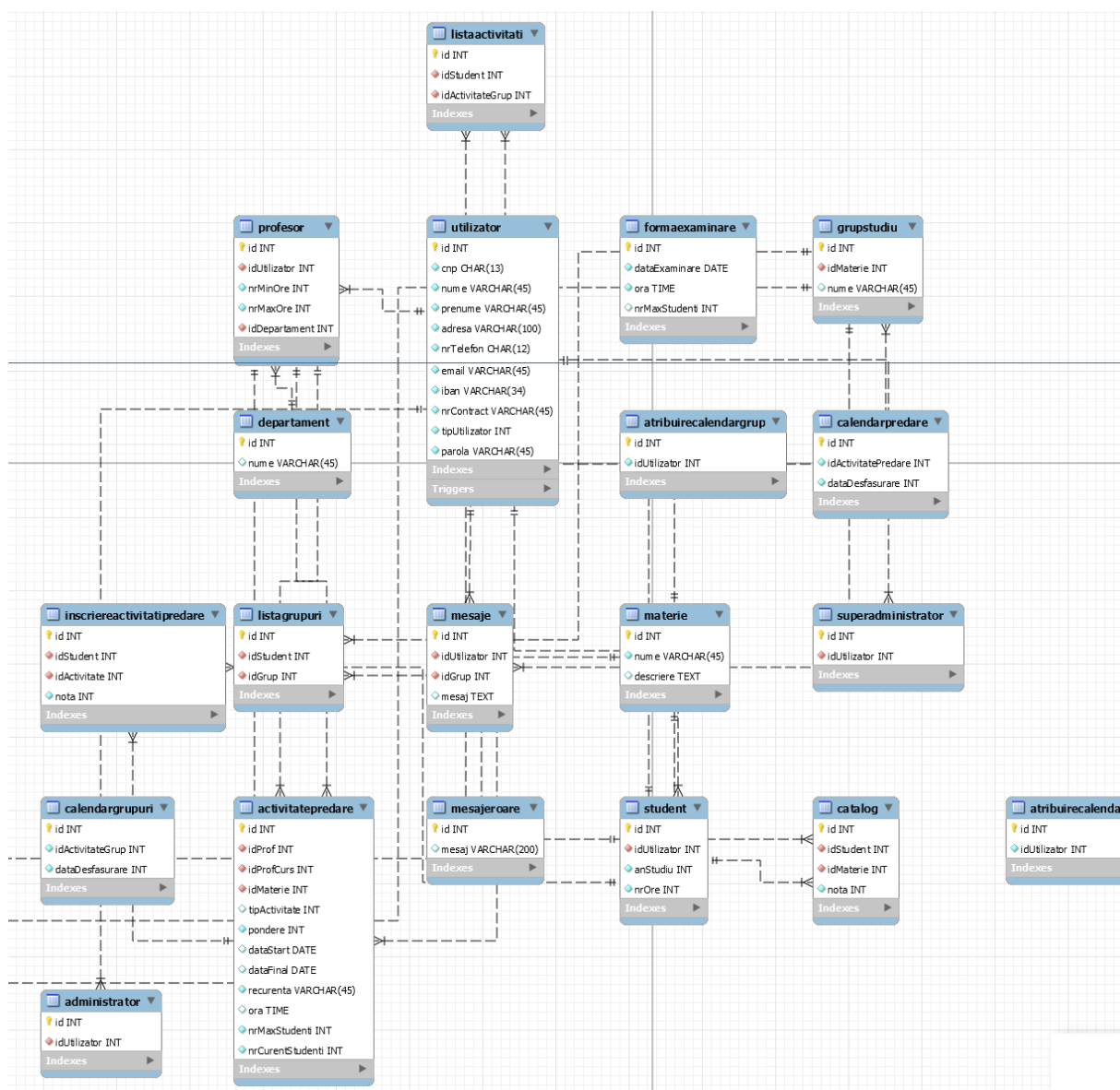
**atribuirecalendarpredare** – Realizează corespondența dintre un utilizator și activitățile de predare la care acesta este înscris; attribute: id, *idUtilizator*, *idCalendar*;



**calendargrupuri** – Reține activitățile de grup planificate; attribute: id, *idActivitateGrup*, *dataDesfasurare*;

**calendarpredare** – Reține activitățile de predare planificate; attribute: id, *idActivitatePredare*, *dataDesfasurare*;

### 3.2. Diagrama EER/UML pentru modelul de date complet





### 3.3. Proceduri și triggere

#### 3.3.1. Proceduri

- `adaugareUser(cnp char(13), nume varchar(45), prenume varchar(45), adresa varchar(100), telefon char(12), email varchar(45), iban varchar(34), contract varchar(45), tip int, out mesajerr varchar(200))`

*-- procedură pentru adăugarea unui nou utilizator în baza de date*

- `autentificare(  
    emailUser varchar(45),  
    parolaUser varchar(45),  
    out numeUser varchar(45),  
    out prenumeUser varchar(45),  
    out cnpUser char(13),  
    out telefonUser char(12),  
    out adrEmailUser varchar(45),  
    out adresaUser varchar(100),  
    out ibanUser varchar(34),  
    out contractUser varchar(45),  
    out tipUser int,  
    out mesajerr varchar(200))`

*-- procedură folosită la autentificarea unui utilizator în aplicație*

- `asignareProfCurs(  
    numeProf varchar(45),  
    prenumeProf varchar(45),  
    contractProf varchar(45),  
    numeCurs varchar(45),  
    tipAct varchar(45),  
    out mesajerr varchar(200)`



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

)

*-- procedură pentru asignarea unui profesor la un curs*

```
▪ creareCont(
    email varchar(45),
    parola varchar(45),
    confirmaParola varchar(45),
    out mesajEroare varchar(200)
)
```

*-- procedură pentru modificarea parolei după crearea contului*

```
▪ incarcareProcentuala(
    idMaterie int,
    idProfCurs int,
    procentCurs int,
    procentLab int,
    procentSeminar int,
    out mesajEroare varchar(200)
)
```

*-- procedură stabilirea procentului pe care fiecare tip de activitate îl are din nota finală la materia respectivă*

```
▪ calculMedie (idMaterie int, idStudent int)
    -- procedură pentru calculul mediei unui student la o materie
```

```
▪ creareGrup(idMaterie int, numeGrup varchar(45))
    -- procedură pentru crearea unui nou grup de studiu
```

```
▪ inscriereGrup(idStudent int, idGrup int, out mesajEroare
    varchar(200))
    -- procedură pentru înscrierea într-un nou grup de studiu
```



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

- **toateGrupurile()**  
*-- procedură pentru extragerea din baza de date a tuturor grupurilor de studiu*
  
- **adaugareNota(idStudent int, idMaterie int, tipActivitate int, nota float, out mesajEroare varchar(200))**  
*-- procedură pentru adăugarea unei note la o activitate de predare la o materie*
  
- **inscriereCurs (idStudent int, idMaterie int, out mesajEroare varchar(200))**  
*-- procedură pentru înscrierea unui student la un curs*
  
- **renuntareCurs(idStudent int, idMaterie int, mesajEroare varchar(200))**  
*-- procedură pentru a renunța la participarea la un curs*
  
- **modificareDate(**  
     **emailUser varchar(45),**  
     **newNume varchar(45),**  
     **newPrenume varchar(45),**  
     **newCnp char(13),**  
     **newTelefon char(12),**  
     **newEmail varchar(45)**  
     **newAdresa varchar(100),**  
     **newIban char(34),**  
     **newNrContract varchar(45),**  
     **out mesajerr varchar(200)**  
     **)**  
*-- procedură pentru modificarea informațiilor referitoare la utilizatorii înregistrați în baza de date*

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

- `stergereInformatii(  
emailUser varchar(45),  
out mesajerr varchar(200)  
)`  
*-- procedură pentru ștergerea informațiilor despre un utilizator din baza de date*
- `adaugareDetaliiProfesor(  
emailUser varchar(45),  
minOre int,  
maxOre int,  
dep varchar(45),  
out mesajerr varchar(200)  
)`  
*-- procedură pentru adăugarea informațiilor specifice pentru utilizatorul de tip profesor*
- `adaugareDetaliiStudent(  
emailUser varchar(45),  
an int,  
ore int,  
out mesajerr varchar(200)  
)`  
*-- procedură pentru adăugarea informațiilor specifice pentru utilizatorul de tip student*
- `adaugareMaterie(umeMat varchar(45), descr text, out  
mesajerr varchar(200))`  
*-- procedură pentru adăugarea unei noi materii în baza de date*



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

- **trimitereMesaj(**  
     **msj varchar(200),**  
     **idUser int,**  
     **idGr varchar (45)**  
     **)**  
     *-- procedură pentru trimiterea unui mesaj în chat-ul unui grup de studiu*
  
- **descarcareCatalog()**  
     *-- procedură pentru descărcarea catalogului sub forma de fișier*
  
- **vizualizareNoteStudent(idStudent int, out mesajEroare int)**  
     *-- procedură pentru vizualizarea notelor de către un student*
  
- **vizualizareNoteProfesor(idProf int, out mesajEroare varchar(200))**  
     *-- procedură pentru vizualizarea notelor de către un profesor*
  
- **cautareUtilizator(ume varchar(45), prenume varchar(45), tipUtilizator varchar(45))**  
     *-- procedură pentru adăugarea informațiilor specifice pentru utilizatorul de tip profesor*
  
- **cautareCurs(ume varchar(45), out mesajEroare varchar(200))**  
     *-- procedură pentru căutarea unui curs*
  
- **afisareMembri(idGrup int)**  
     *-- procedură pentru extragerea membrilor dintr-un grup de studiu*
  
- **afisareMesaje(idGrup int)**



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

*-- procedură pentru extragerea mesajelor trimise pe chat-ul unui grup de studiu*

- **inserareMesajNou(idUtilizator int, idGrup int, mesaj text)**

*-- procedură pentru adăugarea informațiilor specifice pentru utilizatorul de tip profesor*

- **afisareStudenti(idActivitate int)**

*-- procedură pentru extragerea din baza de date a studenților care sunt înscriși la o activitate de predare*

- **sugestiiParticipanti(idGrup int)**

*-- procedură pentru afișarea unor sugestii de participanți la grupurile de studiu*

- **getNumGrup(idGrup int, out numeGrup varchar(45))**

*-- procedură pentru determinarea grupului de studiu care are id-ul idGrup*

- **getNumStudent(idStudent int, out numeStudent varchar(45), out prenumeStudent varchar(45))**

*-- procedură pentru determinarea numelui studentului cu id-ul idStudent*

- **getNumMaterie(idMaterie int, out numeMaterie varchar(45))**

*-- procedură pentru determinarea numelui materiei cu id-ul idMaterie*

- **toateCursurileStudentului(idStudent int)**

*-- procedură pentru determinarea tuturor cursurilor la care este înscris studentul cu id-ul idStudent*





**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

- **creareActivitate(**  
     **idGrup int,**  
     **dataDesfasurare date,**  
     **ora time,**  
     **durata float,**  
      **timpExpirare float,**  
     **minStudenti int,**  
     **out mesajEroare varchar(200),**  
     **out idActivitate int)**  
     *-- procedură pentru programarea unei noi activități de grup*
  
- **totiProfesorii()**  
     *-- procedură pentru selectarea tuturor profesorilor*
  
- **toateActivitatileCuProfesor()**  
     *-- procedură pentru selectarea tuturor activităților la care este asignat un profesor oarecare*
  
- **toateActivitatileFaraProfesor()**  
     *-- procedură pentru determinarea tuturor activităților de grup la care nu este invitat niciun profesor*
  
- **inscriereActivitateGrup(idStudent int, idActivitate int,**  
     **out mesajEroare varchar(200))**  
     *-- procedură pentru înscrierea unui student la o activitate dintr-un grup de studiu*
  
- **toateGrupurileStudentului(idStudent int)**  
     *-- procedură pentru selectarea tuturor grupurilor în care este înscris studentul cu id-ul idStudent*



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

- **toateMateriileProfesorului(idProf int)**  
*-- procedură pentru determinarea numelui studentului cu id-ul idStudent*
  
- **vizualizareNoteMaterie(idProfCurs int, out mesajEroare varchar(200))**  
*-- procedură pentru vizualizarea notelor tuturor studenților de la materia predată de profesor cu id-ul idProfCurs*
  
- **vizualizareNoteLaborator(idProf int, idMaterie int, out mesajEroare varchar(200))**  
*-- procedură pentru vizualizarea tuturor notelor de la un laborator de la materia cu id-ul idMaterie, predat de profesorul cu id-ul idProf*
  
- **vizualizareNoteSeminar(idProf int, idMaterie int, out mesajEroare varchar(200))**  
*-- procedură pentru vizualizarea tuturor notelor de la un seminar de la materia cu id-ul idMaterie, predat de profesorul cu id-ul idProf*
  
- **vizualizareNoteCurs(idProfCurs int, idMaterie int, out mesajEroare varchar(200))**  
*-- procedură pentru vizualizarea tuturor notelor de la un curs de la materia cu id-ul idMaterie, predat de profesorul cu id-ul idProfCurs*
  
- **getDate(**  
    **emailUser varchar(45),**  
        **out numeUser varchar(45),**  
        **out prenumeUser varchar(45),**  
        **out adrEmailUser varchar(45),**  
        **out cnpUser char(13),**  
        **out telefonUser char(12),**  
        **out adresaUser varchar(100),**


**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

```

    out ibanUser char(34),
    out nrContractUser varchar(45)
)
-- procedură pentru selectarea informațiilor stocate despre utilizatorul
cu email-ul emailUser

▪   getTipUser(emailUser varchar(45), out tipUser int)
    -- procedură pentru tipului utilizatorului cu email-ul emailUser

▪   getInfoStudent(emailUser varchar(45), out an int, out ore
    int)
    -- procedură pentru selectarea informațiilor specifice stocate despre
    studentul cu email-ul emailUser

▪   getInfoProfesor(emailUser varchar(45), out nrMin int, out
    nrMax int, out idDep varchar(45))
    -- procedură pentru selectarea specifice stocate despre profesorul cu
    email-ul emailUser

▪   getNumeDep(idDep int, out numeDep varchar(45))
    -- procedură pentru selectarea numelui departamentului cu id-ul idDep

▪   getIdDep(numDep varchar(45), out idDep int)
    -- procedură pentru id-ului departamentului cu numele numeDep

```

### 3.3.2.Triggere

- **inserare\_student**  
 -- trigger care la inserarea în tabela de utilizatori a unui utilizator de tip student, inserează o nouă tupă în tabela student pentru a putea reține informațiile specifice acestui tip de utilizator



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

▪ **inserare\_profesor**

*-- trigger care la inserarea în tabela de utilizatori a unui utilizator de tip profesor, inserează o nouă tupă în tabela profesor pentru a putea reține informațiile specifice acestui tip de utilizator*

▪ **inserare\_administrator**

*-- trigger care la inserarea în tabela de utilizatori a unui utilizator de tip administrator, inserează o nouă tupă în tabela administrator pentru a putea reține informațiile specifice acestui tip de utilizator*

▪ **inserare\_superadministrator**

*-- trigger care la inserarea în tabela de utilizatori a unui utilizator de tip super-administrator, inserează o nouă tupă în tabela super-administrator pentru a putea reține informațiile specifice acestui tip de utilizator*

▪ **generare\_calendar\_grup after**

*-- trigger care la inserarea unei noi activități din cadrul unui grup în tabela activitategrup de studiu, plasează activitatea în calendar la data corespunzătoare*

▪ **generare\_calendar\_predare**

*-- trigger care la inserarea în tabela de activitatepredare, plasează activitatea inserată în calendar, la toate datele curpinse între data de început a activității și data de final a acesteia, ținând cont de recurența acesteia*



### 3.3.3. Evenimente

- **verificareActivitateGrup**

-- eveniment care șterge o activitate din cadrul unui grup atunci când timpul de desfășurare al acesteia a trecut sau când, cu un număr de ore prestabilit înainte de desfășurarea activității, nu s-a atins numărul minim de participanți

- **verificareListaActivitati**

-- eveniment care șterge activitățile de grup și din lista de activități, în aceleași condiții ca verificareActivitateGrup

### 3.4. Normalizarea datelor

Baza de date se află în FN1 deoarece toate atributele din toate relațiile prin care aceasta este definită conțin doar valori atomice, adică valorile atributelor sunt folosite întotdeauna ca un întreg și nu se vor utiliza niciodată doar porțiuni din acestea.

Fie tabela *utilizator(id, cnp, nume, prenume, adresa, nrTelefon, email, iban, nrContract, tipUtilizator, parola)*, având drept mulțime de atribute mulțimea  $U = \{id, cnp, nume, prenume, adresa, nrTelefon, email, iban, nrContract, tipUtilizator, parola\}$ . Mulțimea dependențelor funcționale este  $F = \{id \rightarrow \text{toate celelalte atribute}, cnp \rightarrow \text{toate celelalte atribute}, email \rightarrow \text{toate celelalte atribute}, nrContract \rightarrow \text{toate celelalte atribute}\}$ .

Așadar, cheile candidat sunt: id, cnp, email, nrContract, care sunt, prin urmare, atribute prime, mulțimea atributelor neprime fiind

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

*{nume, prenume, adresa, nrTelefon, iban, tipUtilizator, parola}.*

Deoarece mulțimea cheilor candidat nu conține chei compuse, relația *utilizator(U)* se află în FN2. Întrucât toate atributele neprime (care nu sunt chei candidat) sunt total dependente de una dintre cheile candidat, putem afirma că relația *utilizator(id, cnp, nume, prenume, adresa, nrTelefon, email, iban, nrContract, tipUtilizator, parola)* se află în FN3.

Demonstrația decurge la fel pentru fiecare dintre tabelele bazei de date, așadar baza de date se află în FN3.

### **3.5. Cod MySQL**

Codul MySQL pentru crearea și popularea bazei de date, dar și pentru crearea procedurilor, trigger-elor și event-urilor va fi atașat în **Anexa SQL** inclusă în arhiva proiectului.

## **4. Detalii de implementare**

Codul în limbajul de programare Java pentru crearea interfeței grafice va fi anexat în arhiva proiectului.

Câteva dintre elementele interfeței grafice, realizate în Eclipse:



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

**LogIn** – panou ce permite logarea în aplicație

**MeniuProfesor** – panou ce ilustrează posibilele funcționalități ale utilizatorului de tip profesor (în mod similar sunt construite meniurile pentru celelalte tipuri de utilizatori)

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

**AsignareCursuri** – panou ce permite unui administrator asigneze un profesor la o activitate de predare (curs, seminar, laborator) la o materie specificată

**Asignare**

Nume profesor:

Prenume profesor:

Materie:

Activitate:

curs  
curs  
laborator  
seminar

**Asignare**

**CautareUtilizator** – panou ce permite administratorului logat în aplicație să caute utilizatorii după nume și să îi filtreze după tip

**Căutare utilizator**

nume:

prenume:

tip utilizator:

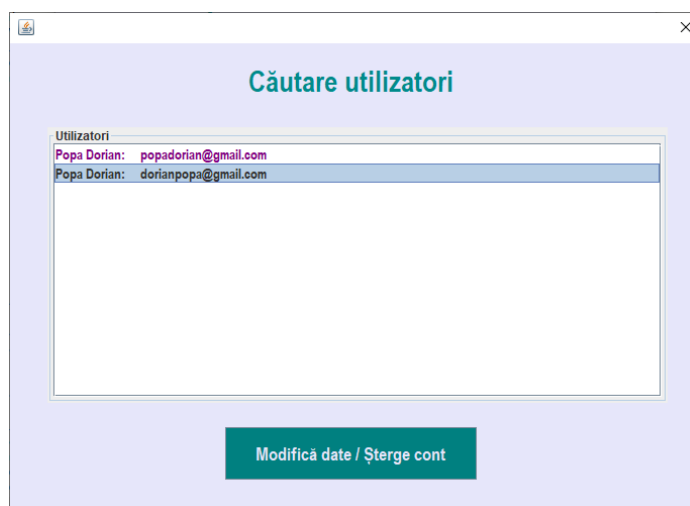
\*toate  
student  
profesor  
\*toate

**Cautare**





**UtilizatoriGasiti** – panou ce afișează utilizatorii căutați după nume și filtrați după tip și permite utilizatorului să îi șteargă din baza de date sau să modifice informațiile referitoare la aceștia



## **5. Concluzii. Limitări și dezvoltări ulterioare**

Întrucât tehnologia câștigă tot mai mult teren odată cu digitalizarea majorității domeniilor profesionale, o variantă digitalizată a mediului academic permite gestionarea flexibilă a activităților didactice, a interacțiunii dintre cadrele didactice și studenți, dar și menținerea consistenței informațiilor, prin actualizarea permanentă a datelor stocate. Astfel, se oferă acces imediat la informațiile necesare desfășurării activităților academice, precum vizualizarea catalogului în format electronic sau planificarea activităților în cadrul unui calendar.

De asemenea, aceasta susține comunicarea între utilizatori, furnizând posibilitatea transmiterii de mesaje și creării de grupuri de studiu. Fiind construită pe o bază de date, cantitatea de informație pe care aceasta o suportă este considerabilă, fapt ce rezultă în posibilitatea utilizării acesteia

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

și în contextul unei universități cu un număr foarte mare de studenți, constituindu-se totodată drept un mijloc de susținere a activităților didactice în contextul realității actuale, în care activitățile didactice se desfășoară parțial sau total în mediul online.

O posibilă direcție de dezvoltare ulterioară a aplicației poate fi reprezentată de implementarea unui sistem de transimțiere a notificărilor către utilizatori după desfășurarea unor evenimente care prezintă interes pentru aceștia: spre exemplu, studentul să primească o notificare în momentul în care i s-a adăugat o notă în catalog.

O altă limitare a aplicației este bazată pe faptul că, la un volum foarte mare de informații stocate în baza de date, viteza de efectuare a operațiilor poate fi destul de scăzută, limitare ce ar putea fi redusă prin dezvoltarea de indecși care să faciliteze viteza de acces la informațiile din baza de date.

De asemenea, aplicația ar mai putea fi dezvoltată în sensul introducerii, pe lângă ferestrele deja existente ce conțin mesaje de eroare, a unor ferestre de dialog care să marcheze realizarea cu succes a unor operații în cadrul aplicației: spre exemplu, la logare ar putea apărea o fereastră cu mesajul „Autentificare cu succes!”, care să dispară după un anumit interval de timp.