



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
CATEDRA CALCULATOARE**

# **PROIECT**

la disciplina  
Introducere in Baze de Date

## **Gestiunea unei biblioteci**

**Bendre Roxana, Moldovan Teodora**

**An academic :2017 – 2018**



## **Cuprins**

### **1.Introducere**

- Tema proiectului
- Specificație proiect

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

- Tabele
- Normalizarea datelor
- Diagrama EER

### **3.Detalii de implementare**

- Interogări
- Obiecte de tip vedere
- Triggere
- Proceduri stocate
- Users

### **4.Interfata Java**

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



## 1.Introducere

### Tema proiectului

Proiectul constă în realizarea unei baze de date cu ajutorul căreia să se gestioneze o bibliotecă, cu elementele necesare: membri, angajați, cărți, săli de studiu și evenimente, comenzi într-o manieră organizată și automatizată. Pentru a facilita accesul la baza de date, aceasta este însoțită de o interfață grafică.

### Specificație proiect

Proiectul constă într-o bază de date implementată în MySQL Workbench și o interfață grafică realizată în Eclipse Java Oxygen care oferă membrilor bibliotecii posibilitatea de a vizualiza cartile, operele, informații despre alți membri precum și să împrumute, să facă rezervări pentru anumite cărți, să returneze cărți și să anuleze rezervări, să vadă informații despre sălile de studiu și locurile disponibile în acestea și să vadă când au loc evenimente în cadrul bibliotecii. Pe de altă parte, angajații bibliotecii pot vizualiza toate aceste informații, dar nu pot împrumuta cărți sau să facă rezervări. În schimb, aceștia pot adăuga membri în sălile de studiu, deoarece fiecare membru este înregistrat atunci când intră într-o sală și pot adăuga opere noi.

Baza de date este creată astfel încât să redea cât mai fidel modul de lucru real al unei biblioteci.

- Fiecare carte din bibliotecă este unic identificată prin ISBN și poate să fie sub formă tipărită sau de manuscris. Pentru cele tipărite este important să se cunoască editura, iar pentru manuscrise anul în care au fost redactate. Despre cărți de cunoscut limba în care este scrisă, numărul total de copii, numărul de copii disponibile, categoria din care face parte, statusul (dacă este disponibilă, rezervată sau împrumutată), locația acesteia în bibliotecă și tariful de penalizare pe o zi, în caz de întârziere la returnare.
- Biblioteca ține o evidență a operelor în baza de date. Acestea nu există fizic neapărat, dar o carte nu poate exista dacă nu există și opera asociată acesteia. Despre fiecare operă se cunosc titlul și autorul.
- Membrii bibliotecii sunt identificați printr-un ID unic. Despre aceștia se cunosc numele, prenumele, adresa, ocupația, data nașterii, sexul și adresa de email. Adresa de email este utilizată pentru ca un membru să poată intra în aplicație. Membrii pot să împrumute

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
CATEDRA CALCULATOARE**

cari, sa le rezerve, sa le restituie si sa anuleze rezervările facute. Se retin de asemenea informatii despre imprumuturile curente, istoricul de imprumuturi si rezervari. Pentru fiecare dintre acestea se cunosc datele importante (inceput, final, data returnării) .

- Biblioteca dispune de sali de studiu in care membrii pot sa vina sa invete. Accesul membrilor in sala de studiu este inregistrat de catre un angajat si se pune in evidenta data accesului in sala. Pentru fiecare sala de studiu este inregistrat numarul de locuri disponibile, numar care variaza in functie de membrii care sunt in sala.
- Fiecare angajat al bibliotecii are un ID unic, nume , prenume si se cunoaste data angajarii acestuia.
- Biblioteca organizeaza evenimente din diverse domenii la care membrii pot participa.

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

### Descrierea bazei de date relaționale

#### Tabele

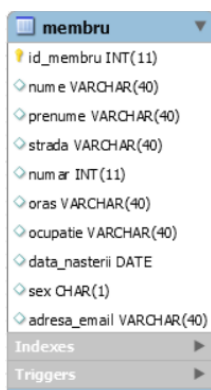
##### *Tabela membru:*

-conține informațiile despre fiecare membru înscris la bibliotecă.

1. Id\_membru – cheie primară
2. Nume – numele membrului
3. Prenume – prenumele membrului
4. Strada – adresa la care locuiește membrul
5. Numar – adresa la care locuiește membrul
6. Oras - orașul în care s-a născut membrul
7. Ocupatie – cu ce se ocupă fiecare membru
8. Data\_nasterii – data în care s-a născut membrul, determină vârsta unui membru
9. Sex – sexul unui membru, feminin sau masculin
10. Adresa\_email – email-ul fiecarui membru

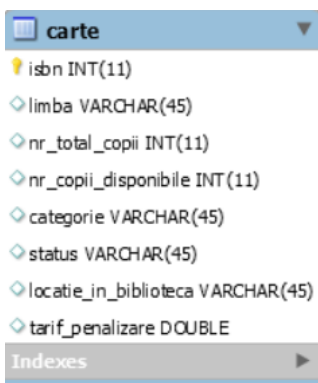
**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**
*Tabela carte:*

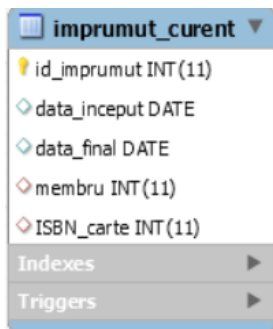
-conține informațiile despre fiecare carte.

1. ISBN – cheie primară, reprezintă codul de identificare al cărților
2. Limba – limba în care este scrisă cartea
3. Nr\_total\_copii - numărul de copii pentru fiecare carte
4. Nr\_copii\_disponibile - numărul de copii disponibile în bibliotecă
5. Categorie – genul de literatură
6. Status - dacă este disponibilă sau nu în bibliotecă
7. Locatie\_in\_biblioteca - în ce zonă se află cartea
8. Tarif\_penalizare – este tariful aplicat în cazul în care nu se returnează cartea la timp

*Tabela imprumut\_curent:*

-conține informațiile necesare pentru împrumutarea curentă a unei cărți.

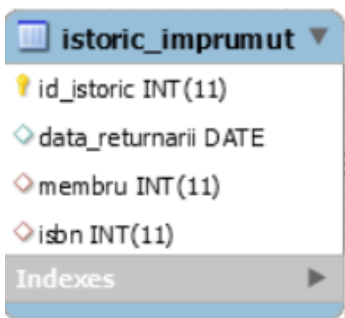
1. Id\_imprumut – cheie primară
2. Data\_inceput – data la care se împrumută cartea
3. Data\_final – data până la care este împrumutată cartea
4. Membru – cheie străină din tabela Membru
5. Isbn\_carte – cheie străină din tabela Carte



*Tabela istoric\_imprumut:*

-conține datele cărților împrumutate.

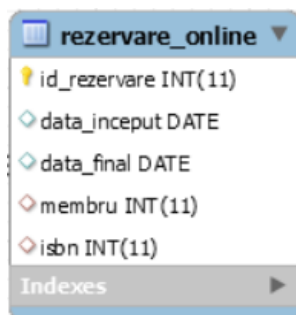
1. Id\_istoric – cheie primară
2. Data\_returnarii – data la care s-a returnat cartea
3. Membru – cheie străină din tabela Membru
4. Isbn – cheie străină din tabela Carte



*Tabela rezervare\_online:*

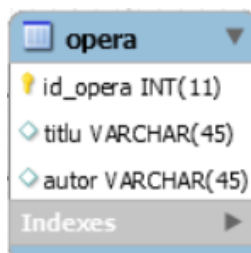
-conține informațiile cu privire la rezervarea online a unei cărți de către un membru al bibliotecii.

1. Id\_rezervare – cheie primară
2. Data\_inceput – data din care se rezervă cartea
3. Data\_final – data până la care trebuie returnată
4. Membru – cheie străină din tabela Membru
5. Isbn – cheie străină din tabela Carte

*Tabela opera:*

-conține informațiile unei opere.

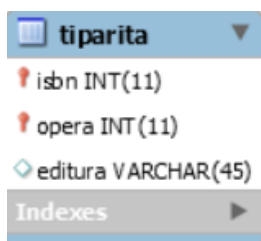
1. Id\_opera – cheie primară
2. Titlu – titlul operei
3. Autor – autorul operei

*Tabela tiparita:*

-este de fapt rezultatul unei relații M:N între tabelele "carte" și "opera".

-conține informațiile despre cărțile tipărite.

1. Isbn – cheie străină din tabela Carte
2. Opera – cheie străină din tabela Opera
3. Editura – numele editurii cărții

*Tabela manuscris:*

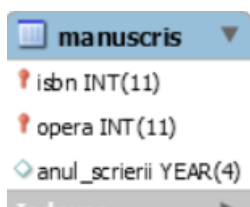
-este de fapt rezultatul unei relații M:N între tabelele "carte" și "opera".

-conține informațiile despre manuscrise.

1. Isbn – cheie străină din tabela Carte
2. Opera – cheie străină din tabela Opera



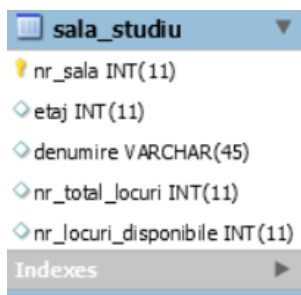
3. Anul\_scrierii – anul în care este scris manuscrisul



*Tabela sala\_studiu:*

-conține informațiile necesare despre fiecare sală de studiu din cadrul bibliotecii.

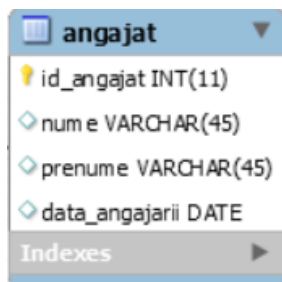
1. Nr\_sala – cheie primară
2. Etaj – etajul unde se află sala
3. Denumire – denumirea sălii
4. Nr\_total\_locuri - numărul de locuri din sală
5. Nr\_locuri\_disponibile - numărul de locuri disponibile din sală



*Tabela angajat:*

-conține informațiile despre angajații bibliotecii.

1. Id\_angajat – cheie primară
2. Nume – numele angajatului
3. Prenume – prenumele angajatului
4. Data\_angajarii – data la care s-a angajat



*Tabela acces\_sala:*

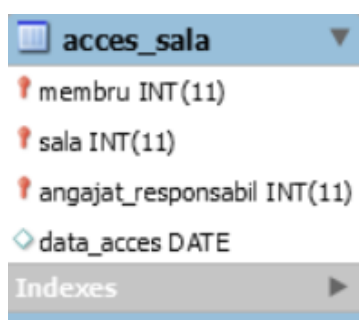




-este de fapt rezultatul unei relații M:N între tabelele "membru" și "sala\_studiu" și rezultatul unei relații M:N între tabelele "membru" și "angajat".

-conține informațiile despre accesul în sală.

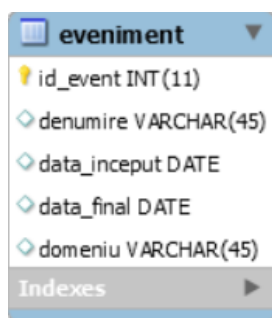
1. Membru – cheie străină din tabela Membru
2. Sala – cheie străină din tabela Sala\_studiu
3. Angajat\_responsabil – cheie străină din tabela Angajat
4. Data\_acces – data în care ai acces la sală



*Tabela eveniment:*

-conține informațiile cu privire la evenimentele din cadrul bibliotecii.

1. Id\_even – cheie primară
2. Denumire – denumirea evenimentului
3. Data\_inceput – data la care începe un eveniment
4. Data\_final – data la care se sfârșește un eveniment
5. Domeniu – domeniul evenimentului

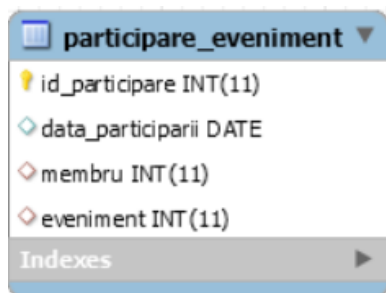


*Tabela participare\_eveniment:*

-este de fapt rezultatul unei relații M:N între tabelele "membru" și "eveniment".

-conține informațiile despre participarea la evenimente.

1. Id\_participare – cheie primară
2. Data\_participarii – data la care se participă la eveniment
3. Membru – cheie străină din tabela Membru
4. Eveniment – cheie străină din tabela Eveniment



## Normalizarea datelor

Baza de date se afla in nivelul de normalizare Boyce-Codd. Conform cercetatorului Chris Date, acest nivel de normalizare poate fi rezumat astfel: "(1)Fiecare atribut trebuie (2)sa reprezinte o informatie (3)despre cheie, (4)intreaga cheie, (5)si nimic in afara cheii". In cazul bazei noastre de date:

1. Fiecare atribut este atomic si nu depinde de alte attribute
2. Fiecare atribut reprezinta o singura informatie
3. Exista attribute chei, reprezentate de id-ul unic al tabelor
4. Cheia primara este suficienta ca sa determine celelalte attribute ale tabelor
5. Cheia primara este singurul atribut care determina celelalte attribute, nu exista un alt atribut care poate determina in mod unic o tupla.

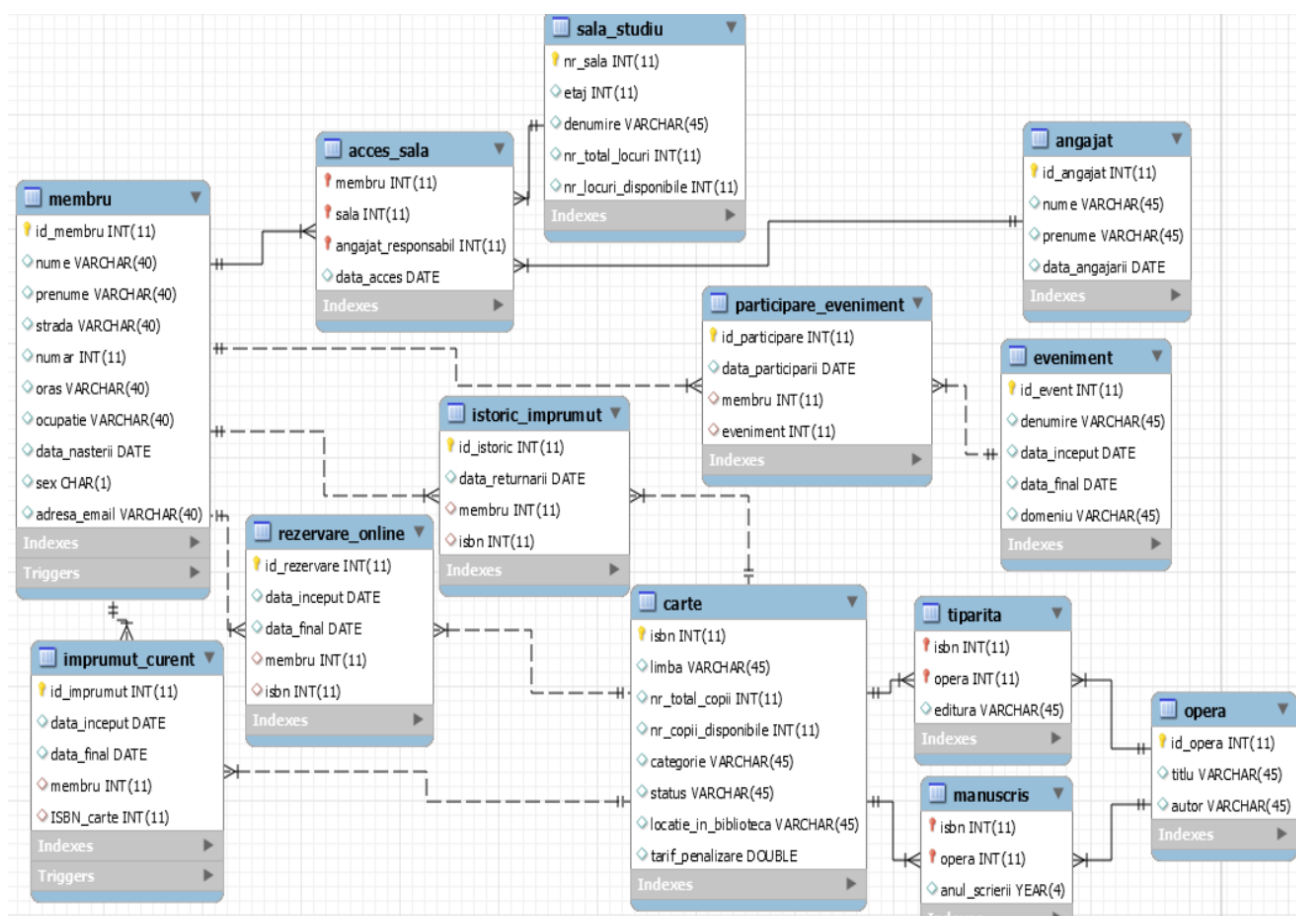


**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

## Diagrama EER





### 3. Detalii de implementare

#### MySQL

##### Interogări

În continuare se prezintă o serie de interogări ce demonstrează cum se folosește bază de date:

1. /\* numărul fetelor înscrise la bibliotecă \*/

```
select count(*) from membru where sex='F';
```

2. /\* anul celui mai vechi manuscris \*/

```
select min(manuscris.anul_scrierii) as 'Anul manuscris vechi'
from manuscris, carte, opera
where opera.id_opera=manuscris.opera and carte.isbn=manuscris.isbn;
```

3. /\* titlul cărților din Zona de literatură \*/

```
select distinct opera.titlu
from opera, tiparita, manuscris, carte
where (opera.id_opera=tiparita.opera and carte.isbn=tiparita.isbn and
locatie_in_biblioteca='Zona
literatura') or (carte.isbn=manuscris.isbn and opera.id_opera=manuscris.opera and
locatie_in_biblioteca='Zona literatura');
```

4. /\* numărul de cărți tipărite care sunt disponibile \*/

```
select count(carte.isbn)
from carte, tiparita
where carte.isbn=tiparita.isbn and carte.status='Disponibil';
```

5. /\* numărul membrilor care au rezervare de la data de 09.12.2017 până la data 13.12.2017 \*/

```
select count(membru.ume)
from membru, rezervare_online
where membru.id_membru=rezervare_online.membru and data_inceput>='2017-12-09' and
data_final<='2017-12-13';
```

6. /\* lista membrilor care participă la evenimentul ShoeBox în data de 04.12.2017 \*/

```
select nume, prenume
from membru, eveniment, participare_eveniment
where membru.id_membru=participare_eveniment.membru and
participare_eveniment.data_participarii='2017-12-04' and eveniment.denumire='ShoeBox';
```

7. /\* lista id-urilor membrilor care au făcut rezervări online \*/

```
select membru.id_membru
from membru, rezervare_online
where membru.id_membru=rezervare_online.membru;
```



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
CATEDRA CALCULATOARE**

8. /\* ISBN-ul cărților restante până la data curentă, numărul de zile restante și tariful de penalizare \*/

```
select carte.isbn,(curdate()-imprumut_curent.data_final) as 'Zile restante',((curdate()
-imprumut_curent.data_final)*carte.tarif_penalizare) as 'Penalizare'
from carte,imprumut_curent,istoric_imprumut
where carte.isbn=imprumut_curent.isbn_carte and curdate()>=imprumut_curent.data_final and
carte.isbn not in (select istoric_imprumut.isbn from istoric_imprumut);
```

9. /\* numele si prenumele membrilor cărora li s-a permis accesul la sală de către angajatul Ifrim Raluca \*/

```
select membru.numa,membru.prenume
from membru,acces_sala,angajat
where membru.id_membru=acces_sala.membru and
acces_sala.angajat_responsabil=angajat.id_angajat and angajat.numa='Ifrim' and
angajat.prenume='Raluca';
```

10. /\* lista denumirilor sălilor de studiu ordonate alfabetic \*/

```
select sala_studiu.denumire
from sala_studiu
order by sala_studiu.denumire asc;
```

11. /\* numele angajaților și numele membrilor care au avut acces la sala Victor Babes \*/

```
select angajat.numa as 'Nume angajat',angajat.prenume as 'Prenume angajat',membru.numa as '
Nume membru',membru.prenume as 'Prenume membru'
from angajat,acces_sala,membru,sala_studiu
where angajat.id_angajat=acces_sala.angajat_responsabil and
membru.id_membru=acces_sala.membru and sala_studiu.nr_sala=acces_sala.sala
and sala_studiu.denumire='Victor Babes';
```

12. /\* membri ordonați după vârstă \*/

```
select numa,prenume,(year(curdate())-year(membru.data_nasterii)) as 'Varsta'
from membru
order by (year(curdate())-year(membru.data_nasterii)) asc;
```

13. /\* numele și numărul de cărți din fiecare zonă \*/

```
select carte.locatie_in_biblioteca,count(isbn) as 'Numar carti'
from carte
group by locatie_in_biblioteca;
```

14. /\* titlul cărților care se găsesc în cel puțin două exemplare \*/

```
select distinct opera.titlu
from opera,tiparita,manuscris,carte
where (opera.id_opera=tiparita.opera and carte.isbn=tiparita.isbn and nr_copii>=2)
or (carte.isbn=manuscris.isbn and opera.id_opera=manuscris.opera and nr_copii>=2);
```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
CATEDRA CALCULATOARE**

15. /\* numele membrilor care au participat la mai mult de un eveniment \*/

```
select membru.nume,membru.prenume
from membru,participare_eveniment
where membru.id_membru=participare_eveniment.membru group by
participare_eveniment.membru having count(participare_eveniment.membru)>1;
```

16. /\* numele membrilor care au împrumutat aceeași carte \*/

```
select membru.nume
from membru
where membru.id_membru in (select istoric_imprumut.membru from istoric_imprumut where
istoric_imprumut.isbn in(select carte.isbn from carte));
```

*Obiecte de tip vedere*

1. /\* Aflați în ce sală de studiu sunt disponibile peste 20 de locuri \*/

```
drop view if exists locuri_in_sala;
create view locuri_in_sala as
select denumire as 'Denumire sala', nr_locuri_disponibile as 'Locuri disponibile'
from sala_studiu
where nr_locuri_disponibile>20;
```

2. /\* Numărul de evenimente care încep în luna decembrie \*/

```
drop view if exists evenimente_decembrie;
create view evenimente_decembrie as
select distinct count(id_event) as 'Numar evenimente'
from eveniment
where month(data_inceput)=12;
```

3. /\* Cărțile tipărite în limba română \*/

```
drop view if exists tiparite_rom;
create view tiparite_rom as
select titlu as 'Carti tiparite in limba romana'
from carte,tiparita,opera
where carte.isbn=tiparita.isbn and opera.id_opera=tiparita.opera and carte.limba='Romana';
```

4. /\* Toate informațiile despre cărți\*/

```
drop view if exists Carti;
create view Carti as
select
carte.isbn,opera.titlu,opera.autor,carte.categorie,carte.status,carte.locatie_in_biblioteca,carte.tarif_pen
alizare
from carte,opera,tiparita
```



```

where carte.isbn=tiparita.isbn and tiparita.opera=opera.id_opera
group by opera.titlu
union
select
carte.isbn,opera.titlu,opera.autor,carte.categorie,carte.status,carte.locatie_in_biblioteca,carte.tarif_pen
alizare
from carte,opera,manuscris
where carte.isbn=manuscris.isbn and manuscris.opera=opera.id_opera
group by opera.titlu asc;

```

#### 5. /\* Informații împrumuturi \*/

```

drop view if exists Imprumuturi;
create view Imprumuturi as
select
imprumut_curent.id_imprumut,imprumut_curent.data_inceput,imprumut_curent.data_final,membru.n
ume,membru.prenume,opera.titlu,opera.autor
from imprumut_curent,membru,carte,opera,tiparita
where imprumut_curent.membru=membru.id_membru and imprumut_curent.ISBN_carte=carte.isbn
and carte.isbn=tiparita.isbn and tiparita.opera=opera.id_opera
group by membru.ume asc
union
select
imprumut_curent.id_imprumut,imprumut_curent.data_inceput,imprumut_curent.data_final,membru.n
ume,membru.prenume,opera.titlu,opera.autor
from imprumut_curent,membru,carte,opera,manuscris
where imprumut_curent.membru=membru.id_membru and imprumut_curent.ISBN_carte=carte.isbn
and carte.isbn=manuscris.isbn and manuscris.opera=opera.id_opera
group by membru.ume asc;

```

#### 6. /\* Informații rezervări \*/

```

drop view if exists Rezervari;
create view Rezervari as
select
rezervare_online.id_rezervare,rezervare_online.data_inceput,rezervare_online.data_final,membru.nu
me,membru.prenume,opera.titlu,opera.autor
from rezervare_online,membru,carte,opera,tiparita
where rezervare_online.membru=membru.id_membru and rezervare_online.ISBN=carte.isbn and
carte.isbn=tiparita.isbn and tiparita.opera=opera.id_opera
group by membru.ume asc
union
select
rezervare_online.id_rezervare,rezervare_online.data_inceput,rezervare_online.data_final,membru.nu
me,membru.prenume,opera.titlu,opera.autor
from rezervare_online,membru,carte,opera,manuscris

```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

where rezervare\_online.membru=membru.id\_membru and rezervare\_online.ISBN=carte.isbn and  
 carte.isbn=manuscris.isbn and manuscris.opera=opera.id\_opera  
 group by membru.numa asc;

7. /\* Informații istoric împrumuturi \*/

```
drop view if exists Istorice;
create view Istorice as
select
istoric_imprumut.id_istoric,istoric_imprumut.data_returnarii,membru.numa,membru.prenume,opera.ti
tlu,opera.autor
from istoric_imprumut,membru,carte,opera,tiparita
where istoric_imprumut.membru=membru.id_membru and istoric_imprumut.ISBN=carte.isbn and
carte.isbn=tiparita.isbn and tiparita.opera=opera.id_opera
group by istoric_imprumut.id_istoric asc
union
select
istoric_imprumut.id_istoric,istoric_imprumut.data_returnarii,membru.numa,membru.prenume,opera.ti
tlu,opera.autor
from istoric_imprumut,membru,carte,opera,manuscris
where istoric_imprumut.membru=membru.id_membru and istoric_imprumut.ISBN=carte.isbn and
carte.isbn=manuscris.isbn and manuscris.opera=opera.id_opera
group by istoric_imprumut.id_istoric asc;
```

8. /\* Informații tiparite \*/

```
drop view if exists Informații_tiparite;
create view Informații_tiparite as
select
opera.titlu,opera.autor,carte.limba,carte.nr_total_copii,carte.nr_copii_disponibile,carte.categor
ie,carte.locatie_in_biblioteca,carte.tarif_penalizare,tiparita.editura
from carte,opera,tiparita
where carte.isbn=tiparita.isbn and tiparita.opera=opera.id_opera
group by opera.titlu;
```

9. /\* Informații manuscrise \*/

```
drop view if exists Informații_manuscrise;
create view Informații_manuscrise as
select
opera.titlu,opera.autor,carte.limba,carte.nr_total_copii,carte.nr_copii_disponibile,carte.categor
ie,carte.locatie_in_biblioteca,carte.tarif_penalizare,manuscris.anul_scrierii
from carte,opera,manuscris
where carte.isbn=manuscris.isbn and manuscris.opera=opera.id_opera
group by opera.titlu;
```





10. /\* Informatii membrii \*/  
 drop view if exists Informatii\_membrii;  
 create view Informatii\_membrii as  
 select id\_membru,nume,prenume,ocupatie,adresa\_email  
 from membru  
 group by id\_membru;
  
11. /\* Informatii acces \*/  
 drop view if exists Informatii\_acces;  
 create view Informatii\_acces as  
 select membru.nume as Nume\_Membru,membru.prenume as  
 Prenume\_Membru,acces\_sala.sala,angajat.nume as Nume\_angajat,angajat.prenume as  
 Prenume\_angajat,acces\_sala.data\_acces  
 from membru,acces\_sala,angajat  
 where membru.id\_membru=acces\_sala.membru and  
 angajat.id\_angajat=acces\_sala.angajat\_responsabil  
 order by acces\_sala.data\_acces asc;

*Triggere*

1. /\* Trigger inserare în istoric după restituire carte \*/

```
drop trigger if exists inserare_istoric;
delimiter //
create trigger inserare_istoric after delete on imprumut_curent
for each row begin
insert into istoric_imprumut(data_returnarii,membru,isbn)
values (curdate(),old.membru,old.isbn_carte);
end //
delimiter ;
```

2. /\* Trigger pentru anunț penalizare la returnare \*/

```
drop trigger if exists penalizare;
delimiter //
create trigger penalizare after delete on imprumut_curent
for each row begin
set @tarif=null;
select carte.tarif_penalizare into @tarif from carte where carte.isbn=old.isbn_carte;
if ((curdate()-old.data_final)>0) then
set @message_text=concat('Aveti de platit penalizare: ',(curdate()-old.data_final)*@tarif,' lei');
signal sqlstate value '45000'
set message_text=@message_text;
end if;
end //
delimiter ;
```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

### 3. /\* Trigger verificare constrângeri vârstă membrii noi\*/

```
drop trigger if exists verificare_membru;
delimiter //
create trigger verificare_membru before insert on membru
for each row begin

    if(new.data_nasterii<'1900-01-01'or new.data_nasterii>'2013-01-01') then
    signal sqlstate '45000'
        set message_text='Varsta nu este corespunzatoare pt a deveni membru al bibliotecii';
    end if;
end //
delimiter ;
```

### *Proceduri stocate*

#### 1. /\* Adăugare membru nou \*/

```
drop procedure if exists adaugare_membru_nou;
delimiter //
create procedure adaugare_membru_nou(ume varchar(40),prenume varchar(40),strada
varchar(40),numar int,oras varchar(40),ocupatie varchar(40),data_nasterii date,sex
char(1),adresa_email varchar(40))
begin
    start transaction;
    set @email=null;
    select @email:=membru.adresa_email from membru where
membru.adresa_email=adresa_email;
    if((@email is null) and (ume is not NULL) and (prenume is not NULL) and (strada is not
NULL) and (numar is not NULL) and (oras is not NULL) and (ocupatie is not NULL) and
(data_nasterii is not NULL) and (sex is not NULL) and (adresa_email is not NULL)) then
        begin
            insert into membru(ume,prenume,strada,numar,oras,ocupatie,data_nasterii,sex,adresa_email)
values
                (ume,prenume,strada,numar,oras,ocupatie,data_nasterii,sex,adresa_email);
            commit;
        end;
    else rollback;
    end if;
end //

delimiter ;
```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

## 2. /\* Adăugare împrumut nou\*/

```

drop procedure if exists adaugare_imprumut_nou;
delimiter //
create procedure adaugare_imprumut_nou(membru int,isbn int)
begin
start transaction;
set @id_membru=null;
    select @id_membru:=membru.id_membru from membru where membru.id_membru=membru;
    if(@id_membru is not null) then
begin
set @isbn_c=null;
    select @isbn_c:=carte.isbn from carte where carte.isbn=isbn;
    if(@isbn_c is not null) then
begin
set @deja_imprumutat=null;
    select @deja_imprumutat:=carte.status from carte where carte.isbn=isbn;
    if(@deja_imprumutat='Disponibil') then
begin
insert into imprumut_curent(data_inceput,data_final,membru,isbn_carte) values
    (curdate(),curdate()+14,membru,isbn);
##scadem numarul de copii disponibile
    update carte set nr_copii_disponibile=nr_copii_disponibile-1 where carte.isbn=isbn;
    set @nr_c=null;
    select @nr_c:=carte.nr_copii_disponibile from carte where carte.isbn=isbn;
    if(@nr_c=0) then
begin

update carte set status='Imprumutat' where carte.isbn=isbn;
    commit;
        end;
    end if;
    commit;
    end;
else rollback;
    end if;
    commit;
    end;
else rollback;
    end if;
    commit;
    end;
else rollback;
    end if;

```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

```
end //
delimiter ;
```

### 3. /\* Restituire împrumut \*/

```
drop procedure if exists restituire_imprumut;
delimiter //
create procedure restituire_imprumut(membru int,id_imprumut int)
begin
start transaction;
set @id_membru=null;
select @id_membru:=membru.id_membru from membru where membru.id_membru=membru;
if(@id_membru is not null) then
begin
set @imprumut=null;
select @imprumut:=imprumut_curent.id_imprumut from imprumut_curent where
imprumut_curent.id_imprumut=id_imprumut;
if(@imprumut is not null) then
begin
set @isbn=null;
select @isbn:=isbn_carte from imprumut_curent where
imprumut_curent.id_imprumut=id_imprumut;
delete from imprumut_curent where imprumut_curent.id_imprumut=id_imprumut;
##crestem numarul de copii disponibile
update carte set nr_copii_disponibile=nr_copii_disponibile+1 where carte.isbn=@isbn;
#actualizam disponibilitatea cartii
set @nr_c=null;
select @nr_c:=carte.nr_copii_disponibile from carte where carte.isbn=@isbn;
if(@nr_c>0) then
begin
update carte set status='Disponibil' where carte.isbn=@isbn;

commit;
end;
end if;
commit;
end;
else rollback;
end if;
commit;
end;
else rollback;
end if;

end //
```



delimiter ;

#### 4. /\* Adăugare rezervare nouă \*/

```
drop procedure if exists adaugare_rezervare_noua;
delimiter //
create procedure adaugare_rezervare_noua(membru int,isbn int)
begin
start transaction;
set @id_membru=null;
select @id_membru:=membru.id_membru from membru where membru.id_membru=membru;
if(@id_membru is not null) then
begin
set @isbn_c=null;
select @isbn_c:=carte.isbn from carte where carte.isbn=isbn;
if(@isbn_c is not null) then
begin
set @deja_imprumutat=null;
select @deja_imprumutat:=carte.status from carte where carte.isbn=isbn;
if(@deja_imprumutat='Disponibil') then
begin
insert into rezervare_online(data_inceput,data_final,membru,isbn) values
(curdate(),curdate()+1,membru,isbn);
##scadem numarul de copii disponibile
update carte set nr_copii_disponibile=nr_copii_disponibile-1 where carte.isbn=isbn;
set @nr_c=null;
select @nr_c:=carte.nr_copii_disponibile from carte where carte.isbn=isbn;
if(@nr_c=0) then

begin
update carte set status='Rezervat' where carte.isbn=isbn;
commit;

end;
end if;
commit;
end;
else rollback;
end if;
commit;
end;
else rollback;
end if;
commit;
end;
```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

```

else rollback;
    end if;
end //
delimiter ;

```

### 5. /\* Anulare rezervare online \*/

```

drop procedure if exists anulare_rezervare;
delimiter //
create procedure anulare_rezervare(membru int,id_rezervare int)
begin
start transaction;
set @id_membru=null;
    select @id_membru:=membru.id_membru from membru where membru.id_membru=membru;
    if(@id_membru is not null) then
begin
set @rezervare=null;
select @rezervare:=rezervare_online.id_rezervare from rezervare_online where
rezervare_online.id_rezervare=id_rezervare;
if(@rezervare is not null) then
begin
set @isbn=null;
select @isbn:=isbn from rezervare_online where rezervare_online.id_rezervare=id_rezervare;
delete from rezervare_online where rezervare_online.id_rezervare=id_rezervare;
##crestem numarul de copii disponibile
update carte set nr_copii_disponibile=nr_copii_disponibile+1 where carte.isbn=@isbn;
#actualizam disponibilitatea cartii
set @nr_c=null;
select @nr_c:=carte.nr_copii_disponibile from carte where carte.isbn=@isbn;
if(@nr_c>0) then
begin

update carte set status='Disponibil' where carte.isbn=@isbn;
commit;
        end;
    end if;
commit;
        end;
else rollback;
        end if;
        commit;
        end;
else rollback;
        end if;

```

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

```
end //
delimiter ;
```

### 6. /\* Adăugare operă nouă \*/

```
drop procedure if exists adaugare_opera;
delimiter //
create procedure adaugare_opera(titlu varchar(45),autor varchar(45))
begin
start transaction;
set @title=null;
select @title:=opera.titlu from opera where opera.autor=autor;
if(@title<>titlu || @title is null and titlu is not null and autor is not null) then
begin
insert into opera(titlu,autor) values
(titlu,autor);
commit;
end;
else rollback;
end if;
end //
delimiter ;
```

### 7. /\* Adăugare tipărită nouă \*/

```
drop procedure if exists adaugare_tiparita;
delimiter //
create procedure adaugare_tiparita(isbn int ,limba varchar(40),nr_total_copii int ,nr_copii_disponibile
int ,categorie varchar(40),status varchar(40),locatie_in_biblioteca varchar(40),tarif_penalizare
double,opera int,editura varchar(45))
begin
start transaction;
set @carte=null;
select @carte:=isbn from carte where carte.isbn=isbn;
if(@carte is null ) then
begin
set @opera=null;
select @opera:=id_opera from opera where opera.id_opera=opera;
if(@opera is not null) then
begin
insert into
carte(isbn,limba,nr_total_copii,nr_copii_disponibile,categorie,status,locatie_in_biblioteca,tarif_penali
zare) values
(isbn,limba,nr_total_copii,nr_copii_disponibile,categorie,status,locatie_in_biblioteca,tarif_penalizare)
;
```



```

        insert into tiparita(isbn,opera,editura) values
        (isbn,opera,editura);
    commit;
    end;
else rollback;
end if;
commit;
end;
else rollback;
end if;
end //
delimiter ;

```

#### 8. /\* Adăugare manuscris nou \*/

```

drop procedure if exists adaugare_manuscris;
delimiter //
create procedure adaugare_manuscris(isbn int ,limba varchar(40),nr_total_copii
int ,nr_copii_disponibile int ,categorie varchar(40),status varchar(40),locatie_in_biblioteca
varchar(40),tarif_penalizare double,opera int,anul year)
begin
start transaction;
set @carte=null;
select @carte:=isbn from carte where carte.isbn=isbn;
if(@carte is null ) then
begin
set @opera=null;
select @opera:=id_opera from opera where opera.id_opera=opera;
if(@opera is not null) then
begin
insert into
carte(isbn,limba,nr_total_copii,nr_copii_disponibile,categorie,status,locatie_in_biblioteca,tarif_penali
zare) values
(isbn,limba,nr_total_copii,nr_copii_disponibile,categorie,status,locatie_in_biblioteca,tarif_penalizare)
;

```

```

        insert into manuscris(isbn,opera,anul_scrierii) values
        (isbn,opera,anul);
    commit;
    end;
else rollback;
end if;
commit;

```



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

```

end;
else rollback;
end if;
end //
delimiter ;

```

### 9. /\* Adăugare membru în sala de studiu \*/

```

drop procedure if exists adaugare_acces;
delimiter //
create procedure adaugare_acces(membru int,sala int ,angajat int)
begin
start transaction;
set @id_membru=null;
select @id_membru:=membru.id_membru from membru where membru.id_membru=membru;
set @id_sala=null;
select @id_sala:=sala_studiu.nr_sala from sala_studiu where sala_studiu.nr_sala=sala;
set @id_angajat=null;
select @id_angajat:=angajat.id_angajat from angajat where angajat.id_angajat=angajat;

if(@id_membru is not null and @id_sala is not null and @id_angajat is not null) then
begin
insert into acces_sala(membru,sala,angajat_responsabil,data_acces) values
(membru,sala,angajat,curdate());
commit;
end;
else rollback;
end if;
end //
delimiter ;

```

### *Utilizatori*

```

use biblioteca;
DROP USER 'member'@localhost;
CREATE USER 'member'@'localhost';
GRANT SELECT ON biblioteca.* TO 'member'@'localhost';
GRANT INSERT ON biblioteca.* TO 'member'@'localhost';
GRANT UPDATE ON biblioteca.* TO 'member'@'localhost';
GRANT DELETE ON biblioteca.* TO 'member'@'localhost';
GRANT SHOW VIEW ON biblioteca.* TO 'member'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_imprumut_nou TO
'member'@'localhost';

```



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
CATEDRA CALCULATOARE**

```
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_membru_nou TO 'member'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_rezervare_noua TO
'member'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.restituire_imprumut TO 'member'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.anulare_rezervare TO 'member'@'localhost';
```

```
DROP USER 'employee'@localhost;
CREATE USER 'employee'@'localhost';
GRANT SELECT ON biblioteca.* TO 'employee'@'localhost';
GRANT INSERT ON biblioteca.* TO 'employee'@'localhost';
GRANT UPDATE ON biblioteca.* TO 'employee'@'localhost';
GRANT DELETE ON biblioteca.* TO 'employee'@'localhost';
GRANT SHOW VIEW ON biblioteca.* TO 'employee'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_membru_nou TO
'employee'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_tiparita TO 'employee'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_manuscris TO 'employee'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_opera TO 'employee'@'localhost';
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare_acces TO 'employee'@'localhost'
```

#### 4.1 Structura claselor in Java.Diagrama UML





Clasele din Java sunt:

- AccessInStudyRoomInformationView
- AddMemberINStudyRoomView
- ApplicationController
- BooksInformationView
- BorrowBooksView
- BorrowedBooksInformationView
- BorrowHistoryInformationView
- CancelReservationView
- DatabaseConnectionController
- DatabaseConnectionView
- DatabaseLogicController
- EmployeeInformationView
- EventInformationView
- GraphicController
- LogInView
- Main
- MainFrame
- ManuscriptInformationView
- MembersInformationView
- NewLiteraryWorkView
- NewManuscriptView
- NewMembrview
- NewPrintedView
- PrintedInformationView
- ReservationInformationView
- Reservationview
- ReturnBorrowedBooksView
- StartApplicationView
- StudyRoomView



## 4.2 Manual de utilizare/instalare si elemente de securitate

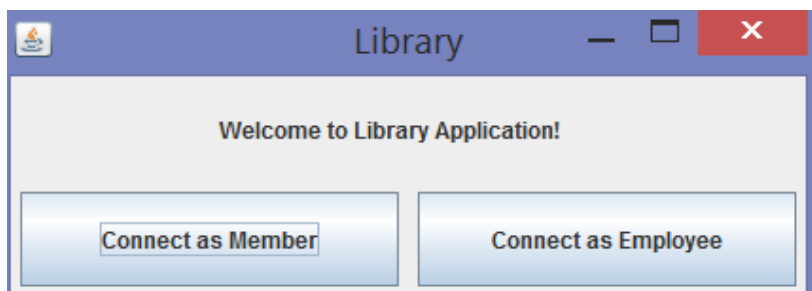
Pentru realizarea unei logari de tip test:

- Membru: UserName: orice adresa de email din baza de date (de ex. [nas.razvan@gmail.com](mailto:nas.razvan@gmail.com)), fara parola
- Angajat: Username:employee, fara parola

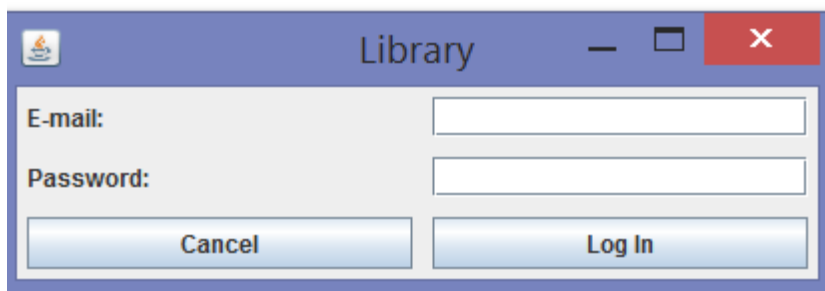
In functie de tipul de logare, accesul la baza de date este diferit: angajatii au voie sa introduca opera, iar apoi carti prin intermediul introducerii unei carti tiparite sau a unui manuscris, sa inregistreze membrii la accesul in sala, in timp ce membrii au voie sa imprumute carti, sa le restituie, sa le rezerve si sa anuleze rezervari. Informatiile din baza de date sunt disponibile prin intermediul interfetei ambelor tipuri de utilizatori.

Acesta logare diferentiata pe tipuri de utilizatori reprezinta elementele de securitate ale aplicatiei, accesul la baza de date fiind diferit, protejand modificarea tabelor de catre utilizatori neautorizati.

Fereastra initiala:



Fereastra de logare:



## DIN CLUJ-NAPOCA



## 5. Concluzii si dezvoltari ulterioare

Aplicatia dezvoltata ofera membrilor inscrisi la biblioteca o solutie rapida si completa de a imprumuta carti atat in timp real, cat si in mediul online, de a avea acces la o sala de studiu, de a participa la evenimente. Baza de date a fost conceputa sa ocupe un spatiu de memorie minim si a fost dotata cu 16 interogari, 3 trigger-e, 11 vederi, 9 proceduri pentru a face interactiunea cu datele cat mai simpla si rapida. Procedurile au fost create cu scopul de a nu fi nevoiti administratorii bazei de date sa-si creeze propriile subprograme ci doar sa apeleze cele deja existente, baza de date fiind usor de modificat.

Dezvoltarea aplicatiilor folosind Java si MySQL este posibila, dar este o varianta de implementare a aplicatiilor intre multe variante. Legat de aplicatia de fata, dupa cum am mai mentionat este demonstrativa, dar metodele folosite aici se pot folosi usor pentru a implementa o aplicatie reala.