UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**PROIECT**

**BAZE DE DATE**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU LAURENȚIU

STUDENT:

DINU ANDREEA-VIOLETA

BUCUREȘTI

2022

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**BIBLIOTECĂ**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU LAURENȚIU

STUDENT:

DINU ANDREEA-VIOLETA

BUCUREȘTI

2022

**Cuprins**

1.Prezentarea modelului (din lumea reală) și a regulilor acestuia…………………………….…...5

2.Diagrama entitate-relație………………………………………………………………………...6

2.1. Reprezentarea diagramei entitate-relatie ………………...………………………………..6

2.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților. ……………..….7

2.2.1 CITITOR…………………………………………………………………………...7

2.2.2. CARTE…………………………………………………………………………..…7

2.2.3 AUTOR………………………………………………………………………….....9

2.2.4. EDITURĂ………………………………………………………………………….9

2.2.5. SECȚIUNE………………………………………………………………………...9

2.2.6. SALĂ\_LECTURĂ………………………………………………………………..11

2.2.7. ANGAJAT………………………………………………………………………..12

2.2.8. ROL……………………………………………………………………………….13

3. Diagrama Conceptuală………………………………………………………………………...12

3.1. Reprezentarea diagramei conceptuale..…………………………………………………...12

3.2. Descrierea constrângerilor de integritate…………………………………………………13

3.2.1.Tabelul **CITITOR**……………………………………………………………..…..14

3.2.2.Tabelul **FISA\_IMPRUMUT**……………….……………………………….……..15

3.2.3. Tabelul **CARTE**…..……………..……………..……………..……………..….…15

3.2.4.Tabelul **SCRIS** …..………..……………..……………..……………..……..…….16

3.2.5.Tabelul **AUTOR** …..……..……………..……………..……………..……………16

3.2.6. Tabelul **EDITURA**…..………..……………..……………..……………..………17

3.2.7.Tabelul **SECTIUNE** …..……..……………..……………..………………………17

3.2.8.Tabelul **SALĂ\_LECTURA**…..………..……………..……………..………,……18

3.2.9.Tabelul **ANGAJAT**………………………….……………………………......…...18

3.2.10.Tabelul **ROL**………..…………………….………………………………………19

3.3. Scheme relationale….……………………………………………………………………20

3.3.1. Schemele relaționale atașate diagramei conceptuale…………...………………….20

3.3.2. Descrierea constrângerilor **ON DELETE**…………………………………………21

4.Scriptul SQL…………………………………………………………………………..………22

4.1. Introducere………………………………………………………………………………22

4.2. Crearea tabelelor, inclusiv a constrângerilor………………………………..……………22

4.2.1. Crearea tabelului **CITITOR**….…..……………..……………..……………..…..22

4.2.2. Crearea tabelului **EDITURA**…………………………………..…………………23

4.2.3. Crearea tabelului **SALA\_LECTURA**…………………………………..…………24

4.2.4.Crearea tabelului **SECTIUNE**…………………………………………..…………24

4.2.5. Crearea tabelului **CARTE**.. ……………………………………………………….25

4.2.6. Crearea tabelului **FISA\_IMPRUMUT**.. ………………………………………….26

4.2.7. Crearea tabelului **AUTOR**….……………………………………………………..26

4.2.8. Crearea tabelului **SCRIS**…………………………………..………………………27

4.2.9. Crearea tabelului **ROL**…………………………………………………………….27

4.2.10. Crearea tabelului **ANGAJAT**………………………………………………...……28

4.3. Introducerea datelor în baza de date…………………………………………………..29

4.3.1. Introducerea datelor în tabelul **CITITOR**………..…………..……………………29

4.3.2. Introducerea datelor în tabelul **EDITURA**………………………………………..30

4.3.3. Introducerea datelor în tabelul **SALA\_LECTURA** ………..…………………...…30

4.3.4. Introducerea datelor în tabelul **SECTIUNE**………………………………………31

4.3.5. Introducerea datelor în tabelul **CARTE**……………………………………………31

4.3.6. Introducerea datelor în tabelul **FISA\_IMPRUMUT**……………………………...32

4.3.7. Introducerea datelor în tabelul **AUTOR**…………………………………………...33

4.3.8. Introducerea datelor în tabelul **SCRIS**……………………………………………..34

4.3.9. Introducerea datelor în tabelul **ROL**…………………………………….…………34

4.3.10. Introducerea datelor în tabelul **ANGAJAT**………………………………………34

1.Prezentarea modelului (din lumea reală) și a regulilor acestuia.

O bibliotecă este o colecție de documente ([cărți](https://ro.wikipedia.org/wiki/Carte)) organizată după anumite criterii, o instituție de cultură în care se adună, se organizează și se păstrează aceste colecții pentru a fi folosite de cititori. Bibliotecile, indiferent de felul și mărimea lor, furnizează pe lângă documentele din colecțiile sale utile nevoilor de studiu, relaxare ori înțelegerii dezbaterilor publice, a actualitãții, a problemelor de ordin istoric, filozofic, științific sau din alte domenii și serviciile bibliotecarilor, instruiți în găsirea, selectarea, circulația, organizarea și interpretarea informațiilor.

     În cadrul acestui proiect am implementat un model de baze de date pentru o bibliotecă în care cărțile sunt organizate în secțiuni, în funcție de domeniul de activitate la care fac referire, secțiunile sunt amenajate în săli de lectură de care se ocupă angajațice indeplinesc diferite funcții, iar despre cărțile pe care cititorii le pot imprumută de la bibliotecă se cunosc informații referitoare la autori, titlu, anul aparitiei, categoria din care fac parte, număr de pagini și editura care a contribuit la tipărire.De asemenea, se rețin și detalii referitoare la imprumuturile pe care cititorii le fac, cum ar fi data la care a fost imprumutată o anumită carte și data la care a fost returnată(daca este cazul). Reguli ce stau la baza desfășurării corecte a activității bibliotecii sunt următoarele:

* **CITITOR**

- un cititor poate imprumuta una sau mai multe cărți

- cititorul poate restituii cărțile pe care le-a împrumutat în aceeasi zi(data\_imprumut să coincida cu data\_returnare)

* **CARTE**

-o carte poate fi împrumutată de mai mulți cititori

-cărțile sunt oraganizate pe secțiuni(domenii/categorii)

-o carte poate fi scrisă de mai mulți autori

-o carte este tipărită de o singură editură

-orice carte se găsește în inventarul bibliotecii

-o carte poate există în biblioteca fără o secțiune în care să fie încadrată

* **AUTOR**

-un autor poate scrie mai multe cărți

-un autor poate scrie cărți ce aparțin de secțiuni diferite

* **EDITURĂ**

-o editură poate tipări mai multe cărți

-o editură poate tipări cărți ce fac parte din secțiuni diferite

* **SECȚIUNE**

-o secțiune conține mai multe cărți

-o secțiune este amplasată într-o singură sala de lectură

-o secțiune nu poate există fără o sală de lectură în care să fie amplasată

* **SALĂ DE LECTURĂ**

-o sală de lectură poate conține mai multe secțiuni de cărți sau poate să nu conțină nici o secțiune

-nu există două sau mai multe săli de lectură cu secțiuni comune

-o sală de lectură se află în grijă mai multor angajați

* **ANGAJAT**

-un angajat își îndeplinește rolul într-o singură sala de lectură

-un angajat poate avea un singur rol(funcție).

* **ROL**

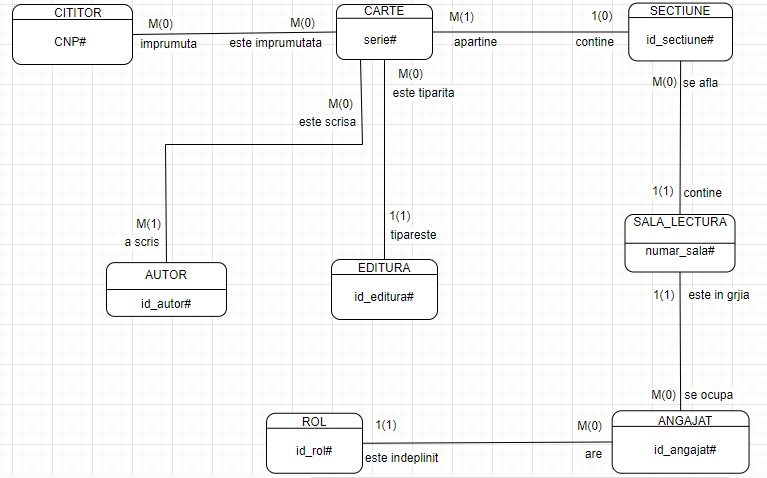
-enititatea rol definește funcțiile pe care angajații le pot avea în cadrul bibliotecii, respectiv în sălile de lectură care le-au fost atribuite

-un rol poate fi asociat mai multor angajați

-un angajat nu poate avea mai multe roluri

2. Diagrama entitate-relație.

2.1. Reprezentarea diagramei entitate-relatie



*Figura 1*-*Diagrama entitate-relatie*

2.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților.

* + 1. **CITITOR**
  + Atribute:

-**CNP**:primary key, cod numeric personal utilizat pentru identificarea cititorului în fișa de lectură, tip de date: șir de caractere;

-**nume**:numele cititorului, tip de date: șir de caractere;

-**prenume**:prenumele cititorului, tip de date: șir de caractere;

-**data\_nastere:**data de nastere a cititorului, tip de date: date;

-**nr\_telefon**:numarul de telefon al cititorului, tip de date: șir de caractere.

* + Chei:

-**CNP**:primary key.

* + Relații și cardinalități:

**CITITOR M(0) - CARTE M(0) relaţie many-to-many, care va conduce la apariţia unui tabel asociativ, numit FISA\_IMPRUMUT.**

-un cititor poate împrumuta una sau mai multe cărți

-o carte poate fi împrumutată de mai mulți cititori

-o carte poate să nu fie împrumutată

-un cititor poate să nu împrumute nici o carte

Cardinalitate maximală: Cate cărți poate împrumuta un cititor?=> multe

De câți citatori poate fi împrumutată o carte?=> mulți

Cardinalitate minimală: Câte cărți trebuie să împrumute un cititor ?=> 0

De câți cititori trebuie să fie împrumutată o carte?=> 0

**2.2.2.**  **CARTE**

* + Atribute:

-**serie**:primary key, pentru identificarea cartilor, tip de date: număr întreg;

-**tiltu**:titlul cartii, tip de date: șir de caractere;

-**id\_sectiune**:foreign key, pentru identificarea sectiunii de care apartine cartea, tip de date: număr întreg;

-**id\_editura**:foreign key, pentru identificarea editurii de care a fost tiparită cartea

-**numar\_pagini**: numarul de pagini pe care o carte il are, tip de date: număr întreg

-**an\_publicare**: anul în care cartea a fost publicata, tip de date: date

* + Chei:

-**serie**:primary key;

-**id\_sectiune**:foreign key;

-**id\_editura**:foreign key;

* + Relații și cardinalități:

**CITITOR M(0) - CARTE M(0): relaţie many-to-many, care va conduce la apariţia unui tabel asociativ, numit FISA\_IMPRUMUT.**

-un cititor poate împrumuta una sau mai multe cărți

-o carte poate fi împrumutată de mai mulți cititori

-o carte poate să nu fie împrumutată

-un cititor poate să nu împrumute nici o carte

**CARTE M(0) – EDITURĂ 1(1)**

-o carte trebuie să fie tipărită de o singură editură

-o editură poate tipări mai multe cărți

-o carte nu poate să existe fără o editură care să o tipărească

-o editură poate să nu tipărească nici o carte

Cardinalitate maximală: Cate cărți poate tipari o editura?=> multe

De câte edituri poate fi tiparita o carte?=>1

Cardinalitate minimală: Cate carti trebuie sa tipareasca o editura ?=> 0

De cate edituri trebuie sa fie tiparita o carte?=> 1

**CARTE M(1) – SECȚIUNE 1(0)**

-o carte poate să se regăsească într-o singură secțiune

-o secțiune trebuie să conțină una sau mai multe cărți

-o secțiune nu poate să existe dacă nu conține cel puțin o carte

-o carte poate să nu aparțină unei secțiuni

Cardinalitate maximală: De câte sectiuni poate apartine o carte?=> 1

Cate carti poate contine o sectiune?=> multe

Cardinalitate minimală: De cate sectiune trebuie sa apartina o carte ?=> 0

Câte carti trebuie sa contina o sectiune?=> 1

**CARTE M(0) – AUTOR M(1) relaţie many-to-many, care va conduce la apariţia unui tabel asociativ, numit SCRIS.**

-o carte poate fi scrisă de mai mulți autori

-un autor poate scrie mai multe cărți

-o carte nu poate sa existe fara un autor

Cardinalitate maximală: De câți autori poate fi scrisă o carte?=> mulți

Câte cărti poate scrie un autor?=> multe

Cardinalitate minimală: De câți autori trebuie sa fie scrisă o carte?=> 1

Câte cărti trebuie să scrie un autor?=> 0

**2.2.3 AUTOR**

* + Atribute:

-**id\_autor**: primary key pentru identificarea autorilor în baza de date, tip de date: număr întreg;

-**nume**: numele autorului, tip de date: șir de caractere;

-**prenume**: prenumele autorului, tip de date: șir de caractere;

-**data\_nastere**: data de naștere a autorului, tip de date: date.

* + Chei:

-**id\_autor**: primary key.

* + Relații și cardinalități:

**CARTE M(0) – AUTOR M(1): relaţie many-to-many, care va conduce la apariţia unui tabel asociativ, numit SCRIS.**

-o carte poate fi scrisă de mai mulți autori

-un autor poate scrie mai multe cărți

**2.2.4. EDITURĂ**

* + Atribute:

-**id\_editura**:primary key pentru identificarea editurii în baza de date, tip de date: număr întreg;

-**nume**: numele editurii, tip de date: sir de caractere;

-**oras**: orasul in care se afla sediul editurii, tip de date: șir de caractere;

-**strada**: strada pe care se afla sediul editurii, tip de date: șir de caractere;

-**nr\_telefon**: numarul de contact al editurii, tip de date: șir de caractere;

-**email**:adresa de email a editurii, tip de date: șir de caractere.

* + Chei:

-**id\_editura**:primary key.

* + Relații și cardinalități:

**CARTE M(0) – EDITURĂ 1(1)**

-o carte trebuie să fie tipărită de o singură editură

-o editură poate tipări mai multe cărți

-o carte nu poate să existe fără o editură care să o tipărească

-o editură poate să nu tipărească nici o carte

Cardinalitate maximală: Cate cărți poate tipari o editura?=> multe

De câte edituri poate poate fi tiparita o carte?=>1

Cardinalitate minimală: Cate carti trebuie sa tipareasca o editura ?=> 0

De cate edituri trebuie sa fie tiparita o carte?=> 1

**2.2.5. SECȚIUNE**

* + Atribute:

-**id\_sectiune**: primary key pentru identificarea unei categorii de carti, tip de date: număr întreg;

-**nume\_domeniu**: numele domeniului(sectiunii) de carti, tip de date: sir de caractere;

-**numar\_sala**: foreign key pentru identificare salii de lectură in care se afla sectiunea, tip de date: număr întreg;

-**nr\_raft**: atribut util în identificarea locului în care este amenajată o anumită sectiune în cadrul bibliotecii, tip de date: număr întreg;

-**nr\_rand**: atribut util în identificarea locului în care este amenajată o anumită sectiune în cadrul bibliotecii, tip de date: număr întreg.

* + Chei:

-**id\_sectiune**: primary key;

-**numar\_sala**: foreign key.

* + Relații și cardinalități:

**SECTIUNE 1(0) - CARTE M(1)**

-o sectiune trebuie sa conțină una sau mai multe cărți

-o carte poate să se regasească intr-o singură sectiune

-o sectiune nu poate sa existe daca nu conține cel putin o carte

-o carte poate sa nu aparțină unei secțiuni

Cardinalitate maximală: Cate cărți poate contine o sectiune?=> multe

De câte sectiuni poate apartine o carte?=>1

Cardinalitate minimală: Cate cărți trebuie sa contina o sectiune?=> 1

De câte sectiuni trebuie sa apartina o carte?=>0

**SECTIUNE M(0) – SALA DE LECTURA 1(1)**

-o secțiune trebuie să se află într-o singură sală de lectură

-o sală de lectură poate conține una sau mai multe secțiuni de cărti

-o secțiune nu poate există fără o sala de lectură în care să fie amenajată

-o sală de lectură poate să nu contină nici o secțiune

Cardinalitate maximală: Câte secțiuni poate contine o sala?=> multe

De câte sali poate apartine o secțiune?=>1

Cardinalitate minimală: Câte secțiuni trebuie sa contina o sala?=> 0

De câte sali trebuie sa apartina o secțiune?=>1

**2.2.6. SALĂ\_LECTURĂ**

* + Atribute:

-**nr\_sală**: primary key pentru identificarea unei săli de lectură, tip de date: număr întreg;

-**nr\_locuri**: atribut ce definește locurile disponibile cititorilor pentru lectură/studiu într-o anumită sala, tip de date: număr întreg.

* + Chei:

-**nr\_sala**:primary key.

* + Relații și cardinalități:

**SALĂ DE LECTURĂ 1(1) – SECTIUNE M(0)**

-o sală de lectură poate conține una sau mai multe secțiuni de cărți

-o secțiune trebuie să se află într-o singură sala de lectură

-o sală de lectură poate să nu conțină nici o secțiune

-o secțiune nu poate exista fără o sală de lectură în care să fie amenajată

Cardinalitate maximală: Câte secțiuni poate contine o sala?=> multe

De câte sali poate apartine o secțiune?=>1

Cardinalitate minimală: Câte secțiuni trebuie sa contina o sala?=> 0

De câte sali trebuie sa apartina o secțiune?=>1

**ANGAJAT M(0) - SALA DE LECTURA 1(1)**

-un angajat trebuie să se ocupe de o singură sală  de lectură

-o sală de lectură poate fi în grijă unuia sau mai multor angajați

-un angajat nu se poate ocupă de mai multe săli de lectură

-un angajat nu poate există fără o sală de lectură de care să se ocupe

* + 1. **ANGAJAT**
  + Atribute:

-**id\_angajat**: primary key pentru identificarea unui angajat, tip de date: număr întreg;

-**nume**: numele angajatului, tip de date: sir de caractere;

-**prenume**: prenumele angajatului, tip de date: sir de caractere;

-**data\_nastere**: data de nastere a angajatului, tip de date: date;

-**data\_angajarii**: data angajarii in biblioteca, tip de date: date;

-**telefon**:contact pentru angajat, tip de date: sir de caractere

-**rol\_id**:foreign key pentru identificarea rolului pe care un angajat il indeplineste în sala de lectura la care a fost asociat, tip de date: număr întreg;

-**nr\_sala**:foreign key pentru identificarea salii de care trebuie să se ocupe un angajat, tip de date: număr întreg.

* + Chei:

-**id\_angajat**: primary key;

-**rol\_id**:foreign key;

-**nr\_sala**:foreign key.

* + Relații și cardinalități:

**ANGAJAT M(0) - SALA DE LECTURA 1(1)**

-un angajat trebuie să se ocupe de o singură sala de lectură

-o sala de lectură poate fi în grijă unuia sau mai multor angajați

-un angajat nu se poate ocupă de mai multe săli de lectură

-un angajat nu poate există fără o sala de lectură de care să se ocupe

Cardinalitate maximală: De cate sali de lectura se poate ocupa un angajat?=> 1

Cati angajati se pot ocupa de o sala de lectura?=>multi

Cardinalitate minimală: De cate sali de lectura trebuie sa se ocupe un angajat?=> 1

Cati angajati trebuie sa se ocupe de o sala de lectura?=>0

**ROL 1(1) – ANGAJAT M(0)**

-un rol poate fi îndeplinit de mai mulți angajați

-un angajat trebuie să aibă un singur rol

-un angajat nu poate există fără un rol care să îi fie atribuit

-un rol poate să nu fie îndeplinit de nici un angajat

Cardinalitate maximală: De cati angajati poate fi indeplinit un rol?=>multi

Cate roluri poate indeplini un angajat?=> 1

Cardinalitate minimală: De cati angajati trebuie sa fie indeplinit un rol?=>0

Cate roluri trebuie sa indeplineasca un angajat?=>1

**2.2.8. ROL**

* + Atribute:

-**rol\_id:**primary key pentru identificarea unei functii în cadrul bibliotecii, tip de date: număr întreg;

-**titlu\_rol**:denumirea rolului, tip de date: sir de caractere;

-**max\_sal**:salariul maxim pe care angajatii cu un anumit rol il pot castiga, tip de date: număr întreg;

-**min\_sal**:salariul minim pe care angajatii cu un anumit rol il pot castiga , tip de date: număr întreg.

* + Chei:

-**rol\_id:**primary key;

* + Relații și cardinalități:

**ROL 1(1) – ANGAJAT M(0)**

-un rol poate fi îndeplinit de mai mulți angajați

-un angajat trebuie să aibă un singur rol

-un angajat nu poate există fără un rol care să îi fie atribuit

-un rol poate să nu fie îndeplinit de nici un angajat

Cardinalitate maximală: De cati angajati poate fi indeplinit un rol?=>multi

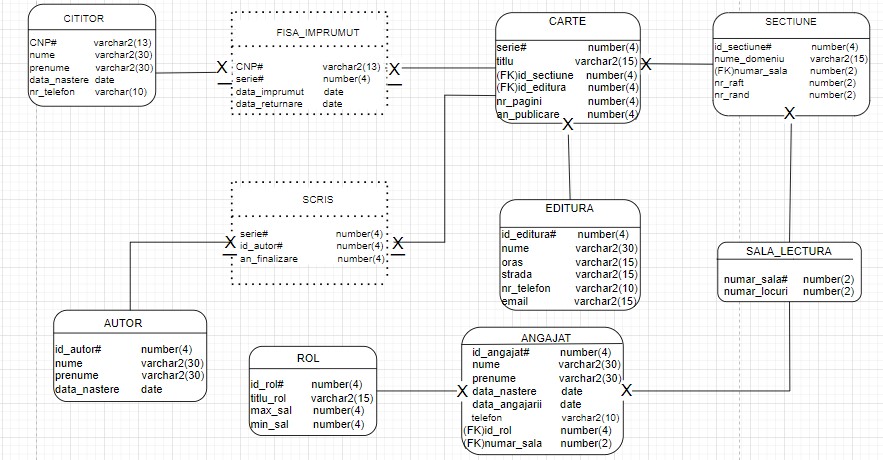
Cate roluri poate indeplini un angajat?=> 1

Cardinalitate minimală: De cati angajati trebuie sa fie indeplinit un rol?=>0

Cate roluri trebuie sa indeplineasca un angajat?=>1

3. Diagrama Conceptuală

3.1. Reprezentarea diagramei conceptuale



*Figura 2- Diagrama Conceptuala*

3.2. Descrierea constrângerilor de integritate.

3.2.1. Tabelul **CITITOR**

• Constrângeri de tip **PRIMARY KEY**

* CITITOR\_CNP\_PK pentru atributul CNP

• Constrângeri de tip **NOT NULL**

* CITITOR\_nume\_NN pentru atributul nume

>o persoană nu poate exista fără nume

* CITITOR\_prenume\_NN pentru atributul prenume

>o persoană nu poate exista fără prenume

* CITITOR\_datanastere\_NN pentru atributul data\_nastere

> data nasterii este obligatorie

* CITITOR\_nrtelefon\_NN pentru nr\_telefon

> datele de contact sunt obligatorii

• Constrângeri de tip **UNIQUE**

* CITITOR\_nrtelefon\_U, pentru atributul nr\_telefon

>nu pot exista mai multi cititori cu același număr de telefon.

• Constrângeri de tip **CHECK**

* CITITOR\_nrtelefon\_C, pentru atributul telefon

> prin funcția LENGTH() verific dacă numărul de telefon are 10 cifre.

* CITITOR\_CNP\_C, pentru atributul CNP

> prin funcția LENGTH() verific dacă CNP-ul are 13 cifre.

3.2.2. Tabelul **FISA\_IMPRUMUT–**tabel asociativ intre tabelul **CITITOR** si tabelul **CARTE**

• Constrângeri de tip **PRIMARY KEY**

* FISA\_IMPR\_PK pentru atributele CNP, serie

• Constrângeri de tip **FOREIGN KEY**

* FISA\_IMPR\_CNP\_FK, pentru atributul CNP

**>** face legătura cu tabelul CITITOR (CNP);

* FISA\_IMPR\_serie\_FK, pentru atributul serie

**>** face legătura cu tabelul CARTE (serie);

• Constrângeri de tip **NOT NULL**

* FISA\_dataimprumut\_NN pentru atributul data\_imprumut

>determina data la care o anumita carte a fost imprumutata

3.2.3. Tabelul **CARTE**

• Constrângeri de tip **PRIMARY KEY**

* CARTE\_serie\_PK pentru atributul serie

• Constrângeri de tip **FOREIGN KEY**

* CARTE\_idsectiune\_FK, pentru atributul id\_sectiune

**>** face legătura cu tabelul SECTIUNE (id\_sectiune)

**>**determina sectiunea de care apartine cartea

* CARTE\_ideditura\_FK, pentru atributul id\_editura

**>** face legătura cu tabelul EDITURA (id\_editura)

**>**determina editura care a tiparit cartea

• Constrângeri de tip **NOT NULL**

* CARTE\_titlu\_NN pentru atributul titlu

>o carte nu poate exista fara un titlu

* CARTE\_ideditura\_NN pentru atributul id\_editura

>determina editura care a tiparit cartea

* CARTE\_nrpagini\_NN pentru atributul nr\_pagini

>o carte trebuie sa aiba un anumit numar de pagini

* CARTE\_anpublicare\_NN pentru atributul id\_sectiune

>determina anul in care cartea a fost publicata

3.2.4.Tabelul **SCRIS –** tabel asociativ intre tabelul **CARTE** si tabelul **AUTOR**

• Constrângeri de tip **PRIMARY KEY**

* SCRIS\_PK pentru atributele serie,id\_autor

• Constrângeri de tip **FOREIGN KEY**

* SCRIS\_idautor\_FK, pentru atributul id\_autor

**>** face legătura cu tabelul AUTOR(id\_autor);

* SCRIS\_serie\_FK, pentru atributul serie

**>** face legătura cu tabelul CARTE (serie);

3.2.5.Tabelul **AUTOR**

• Constrângeri de tip **PRIMARY KEY**

* AUTOR\_idautor\_PK pentru atributele id\_autor

• Constrângeri de tip **NOT NULL**

* AUTOR\_nume\_NN pentru atributul nume

> o persoană nu poate exista fără nume

* AUTOR\_prenume\_NN pentru atributul prenume

> o persoană nu poate exista fără prenume

* AUTOR\_datanastere\_NN pentru atributul data\_nastere

> o persoană nu poate exista fără o data de nastere

3.2.6. Tabelul **EDITURA**

• Constrângeri de tip **PRIMARY KEY**

* EDITURA\_ideditura\_PK pentru atributele id\_editura

• Constrângeri de tip **NOT NULL**

* EDITURA\_ nume\_NN pentru atributul nume

> o editura trebuie sa aiba un nume

* EDITURA\_ oras\_NN pentru atributul oras

> sediul unei edituri se afla intr-un oras

* EDITURA\_ strada\_NN pentru atributul strada

> sediul unei edituri se afla pe o strada

* EDITURA\_nrtelefon\_NN pentru nr\_telefon

> datele de contact sunt obligatorii

* EDITURA\_email\_NN pentru email

> datele de contact sunt obligatorii

• Constrângeri de tip **UNIQUE**

* EDITURA\_nrtelefon\_U, pentru atributul nr\_telefon

>nu pot exista mai multe edituri cu același număr de telefon.

* EDITURA\_email\_U, pentru atributul email;

> nu pot exista mai multe edituri cu aceeași adresă de email.

• Constrângeri de tip **CHECK**

* EDITURA\_nrtelefon\_C, pentru atributul telefon

> prin funcția LENGTH() verific dacă numărul de telefon are 10 cifre.

3.2.7.Tabelul **SECTIUNE**

• Constrângeri de tip **PRIMARY KEY**

* SECTIUNE\_idsectiune\_PK pentru atributele id\_sectiune

• Constrângeri de tip **FOREIGN KEY**

* SECTIUNE\_numarsala\_FK, pentru atributul numar\_sala

**>** face legătura cu tabelul SALA\_LECTURA(numar\_sala);

**>**determina in ce sala de lectura este amenajata sectiunea

• Constrângeri de tip **NOT NULL**

* SECTIUNE\_ numedomeniu\_NN pentru atributul nume\_domeniu

> o sectiune trebuie sa aiba un nume

* SECTIUNE\_ numarsala\_NN pentru atributul numar\_sala

> o sectiune trebuie sa apartina unei sali de lectura

* SECTIUNE\_ nrraft\_NN pentru atributul nr\_raft

> determina raftul unde se regaseste sectiunea in sala de lectura

* SECTIUNE\_ nrrand\_NN pentru atributul nr\_rand

> determina randul de unde incepe sectiunea intr-un anumit raft

3.2.8.Tabelul **SALĂ\_LECTURĂ**

• Constrângeri de tip **PRIMARY KEY**

* SALA\_LECTURA\_PK, pentru atributul numar\_sala

• Constrângeri de tip **NOT NULL**

* SALA\_numarlocuri\_NN pentru atributul numar\_locuri

> determina numarul de locuri disponibile pentru studiu/lectura dintr-o sala

3.2.9.Tabelul **ANGAJAT**

• Constrângeri de tip **PRIMARY KEY**

* ANGAJAT\_idangajat\_PK, pentru atributul id\_angajat

• Constrângeri de tip **FOREIGN KEY**

* ANGAJAT\_idrol\_FK, pentru atributul id\_rol

**>** face legătura cu tabelul ROL(id\_rol)

**>** determina functia pe care angajatul o are

* ANGAJAT\_numarsala\_FK, pentru atributul numar\_sala

**>** face legătura cu tabelul SALA\_LECTURA (numar\_sala)

**>**determina sala de care se ocupa angajatul

• Constrângeri de tip **NOT NULL**

* ANGAJAT\_idrol\_NN, pentru atributul id\_rol

> determina rolul pe care angajatul in indeplineste in sala de lectura atribuita

* ANGAJAT\_numarsala\_NN, pentru atributul numar\_sala

**>** determina sala de care se ocupa angajatul

* ANGAJAT\_nume\_NN pentru atributul nume

>o persoană nu poate exista fără nume

* ANGAJAT\_prenume\_NN pentru atributul prenume

>o persoană nu poate exista fără prenume

* ANGAJAT\_datanastere\_NN pentru atributul data\_nastere

> data nasterii este obligatorie

* ANGAJAT\_dataangajarii\_NN pentru atributul data\_angajarii

> data angajarii este obligatorie

* ANGAJAT\_telefon\_NN pentru atributul telefon

> datele de contact sunt obligatorii

* Constrângeri de tip **UNIQUE**
* ANGAJAT\_telefon\_U, pentru atributul nr\_telefon

>nu pot exista mai multi angajati cu același număr de telefon.

• Constrângeri de tip **CHECK**

* ANGAJAT\_telefon\_C, pentru atributul telefon

> prin funcția LENGTH() verific dacă numărul de telefon are 10 cifre.

3.2.10.Tabelul **ROL**

• Constrângeri de tip **PRIMARY KEY**

* ROL\_idrol\_PK, pentru atributul id\_rol

• Constrângeri de tip **NOT NULL**

* ROL\_titlurol\_NN, pentru atributul titlu\_rol

> un rol nu poate exista fara un titlu

* ROL\_maxsal\_NN, pentru atributul max\_sal

**>** determina salariul maxim ce poate fi obtinut pe un anumit rol

* ROL\_minsal\_NN, pentru atributul min\_sal

**>** determina salariul minim ce poate fi obtinut pe un anumit rol

* 1. Scheme relationale.

3.3.1. Schemele relaționale atașate diagramei conceptuale sunt:

* CITITOR(CNP#, nume, prenume, data\_nastere, nr\_telefon)
* FISA\_IMPRUMUT(CNP#(FK), serie#(FK), data\_imprumut, data\_returnare)
* CARTE(serie#, titlu, id\_sectiune(FK), id\_editura(FK), nr\_pagini, an\_publicare)
* SCRIS(serie#(FK), id\_autor#(FK), an\_finalizare)
* AUTOR(id\_autor#, nume, prenume, data\_nastere)
* EDITURA(id\_editura#, nume, oras, strada, nr\_telefon, email)
* SECTIUNE(id\_sectiune#, nume\_domeniu, numar\_sala(FK), nr\_raft, nr\_rand)
* SALA\_LECTURA(numar\_sala#, numar\_locuri)
* ANGAJAT(id\_angajat#, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol(FK), numar\_sala(FK))
* ROL(id\_rol#, titlu\_rol, max\_sal, min\_sal)

3.3.2. Descrierea constrângerilor **ON DELETE**.

Există 9 constrângeri ON DELETE, câte una pentru fiecare constrângere de tip FOREIGN KEY. Acestea sunt:

* FISA\_IMPR\_CNP\_FK, **ON DELETE CASCADE**

> dacă un cititor este șters, atunci toate cartile asociate acelui citator vor fi sterse din fisa de imprumut

* FISA\_IMPR\_serie\_FK, **ON DELETE CASCADE**

>daca o carte este stearsa, atunci, in fisa de imprumut cititorul asociat cartii respective ramane cu restul cartilor pe care le-a imprumutat, daca este cazul(vor fi sterse intrarile ce prezentau si aceasta carte)

* CARTE\_idsectiune\_FK, **ON DELETE SET NULL**

>cand o sectiune este stearsa din baza de data, cartile care erau in sectiunea respectivă vor ramane in baza de date, dar fără sa fie incadrate intr-o sectiune

>o carte poate sa nu aparțină unei secțiuni

* CARTE\_ideditura\_FK, **ON DELETE CASCADE**

>cand o editura este stearsa din baza de date, toate cartile care aveau id-ul editurii respective vor fi sterse

>o carte nu poate sa existe fara o editura care sa o tiparească, astfel ca ON DELETE SET NULL nu era potrivit

* SCRIS\_serie\_FK, **ON DELETE CASCADE**

> daca o carte este stearsă, atunci, autorul asociat cartii respective ramane cu restul cărtilor pe care le-a scris, dacă este cazul(vor fi sterse intrarile ce prezentau si aceasta carte)

* SCRIS\_idautor\_FK, **ON DELETE CASCADE**

> dacă un autor este șters, atunci toate cartile asociate acelui autor vor fi sterse din tabel

* SECTIUNE\_numarsala\_FK, **ON DELETE CASCADE**

>daca o sală de lectură este stearsă din baza de date, toate sectiunile care apartineau acelei săli vor fi sterse

>o sectiune nu poate sa existe fără o sală de lectură in care să fie amenajată

* ANGAJAT\_idrol\_FK, **ON DELETE CASCADE**

>daca un rol este sters, toti angajatii care aveau acel rol vor fi stersi

>un angajat nu poate există fără un rol care să îi fie atribuit

* ANGAJAT\_numarsala\_FK, **ON DELETE CASCADE**

>daca o sala de lectura este stearsa din baza de date, toti angajatii care se ocupau de sala respectiva vor fi stersi

> un angajat nu poate exista fără o sala de lectură de care să se ocupe

1. Scriptul SQL

4.1. Introducere

Scriptul SQL, salvat cu numele scriptsql\_DINU\_ANDREEA\_VIOLETA\_GR261.sql, a fost scris folosind Oracle SQL Developer.

4.2. Crearea tabelelor, inclusiv a constrângerilor

Tabelele au fost create cu ajutorul instrucțiunii **CREATE TABLE nume\_tabel( );** care face parte din setul de instrucțiuni (Data Definition Language) ale SQL.

4.2.1. Crearea tabelului **CITITOR**

CREATE TABLE CITITOR(

CNP VARCHAR(13)

CONSTRAINT CITITOR\_CNP\_PK PRIMARY KEY

CONSTRAINT CITITOR\_CNP\_C CHECK (LENGTH(CNP)=13),

nume VARCHAR(30)

CONSTRAINT CITITOR\_nume\_NN NOT NULL,

prenume VARCHAR(30)

CONSTRAINT CITITOR\_prenume\_NN NOT NULL,

data\_nastere DATE

CONSTRAINT CITITOR\_datanastere\_NN NOT NULL,

nr\_telefon VARCHAR2(10)

CONSTRAINT CITITOR\_nrtelefon\_NN NOT NULL

CONSTRAINT CITITOR\_nrtelefon\_C CHECK (LENGTH(nr\_telefon)=10)

CONSTRAINT CITITOR\_nrtelefon\_U UNIQUE

);

4.2.2. Crearea tabelului **EDITURA**

Am creat secvența EDITURA\_ideditura\_SEQ pentru a asigna valori cheii primare id\_editura.

CREATE TABLE EDITURA(

id\_editura NUMBER(4)

CONSTRAINT EDITURA\_ideditura\_PK PRIMARY KEY,

nume VARCHAR2(30)

CONSTRAINT EDITURA\_nume\_NN NOT NULL,

oras VARCHAR(15)

CONSTRAINT EDITURA\_oras\_NN NOT NULL,

strada VARCHAR(15)

CONSTRAINT EDITURA\_strada\_NN NOT NULL,

nr\_telefon VARCHAR(10)

CONSTRAINT EDITURA\_nrtelefon\_NN NOT NULL

CONSTRAINT EDITURA\_nrtelefon\_C CHECK (LENGTH(nr\_telefon)=10)

CONSTRAINT EDITURA\_nrtelefon\_U UNIQUE,

email VARCHAR(15)

CONSTRAINT EDITURA\_email\_NN NOT NULL

CONSTRAINT EDITURA\_email\_U UNIQUE

);

CREATE SEQUENCE EDITURA\_ideditura\_SEQ

START WITH 1000

INCREMENT BY 1;

4.2.3. Crearea tabelului **SALA\_LECTURA**

CREATE TABLE SALA\_LECTURA(

numar\_sala NUMBER(2)

CONSTRAINT SALA\_LECTURA\_PK PRIMARY KEY,

numar\_locuri NUMBER(2)

CONSTRAINT SALA\_numarlocuri\_NN NOT NULL

);

4.2.4. Crearea tabelului **SECTIUNE**

Am creat secvența SECTIUNE\_idsectiune\_SEQ pentru a asigna valori cheii primare id\_sectiune.

CREATE TABLE SECTIUNE(

id\_sectiune NUMBER(4)

CONSTRAINT SECTIUNE\_idsectiune\_PK PRIMARY KEY,

nume\_domeniu VARCHAR2(15)

CONSTRAINT SECTIUNE\_numedomeniu\_NN NOT NULL,

numar\_sala NUMBER(2)

CONSTRAINT SECTIUNE\_numarsala\_NN NOT NULL

CONSTRAINT SECTIUNE\_numarsala\_FK REFERENCES SALA\_LECTURA(numar\_sala) ON DELETE CASCADE,

nr\_raft NUMBER(2)

CONSTRAINT SECTIUNE\_nrraft\_NN NOT NULL,

nr\_rand NUMBER(2)

CONSTRAINT SECTIUNE\_nrrand\_NN NOT NULL

);

CREATE SEQUENCE SECTIUNE\_idsectiune\_SEQ

START WITH 2000

INCREMENT BY 1;

4.2.5. Crearea tabelului **CARTE**

Am creat secvența CARTE\_serie\_SEQ pentru a asigna valori cheii primare id\_carte.

CREATE TABLE CARTE(

serie NUMBER(4)

CONSTRAINT CARTE\_serie\_PK PRIMARY KEY,

titlu VARCHAR2(15)

CONSTRAINT CARTE\_titlu\_NN NOT NULL,

id\_sectiune NUMBER(4)

CONSTRAINT CARTE\_idsectiune\_FK REFERENCES SECTIUNE(id\_sectiune) ON DELETE SET NULL,

id\_editura NUMBER(4)

CONSTRAINT CARTE\_ideditura\_FK REFERENCES EDITURA(id\_editura) ON DELETE CASCADE

CONSTRAINT CARTE\_ideditura\_NN NOT NULL,

nr\_pagini NUMBER(4)

CONSTRAINT CARTE\_nrpagini\_NN NOT NULL,

an\_publicare NUMBER(4)

CONSTRAINT CARTE\_anpublicare\_NN NOT NULL

);

CREATE SEQUENCE CARTE\_serie\_SEQ

START WITH 1000

INCREMENT BY 1;

4.2.6. Crearea tabelului **FISA\_IMPRUMUT**

CREATE TABLE FISA\_IMPRUMUT(

CNP VARCHAR(13)

CONSTRAINT FISA\_IMPR\_CNP\_FK REFERENCES CITITOR(CNP) ON DELETE CASCADE,

serie NUMBER(4)

CONSTRAINT FISA\_IMPR\_serie\_FK REFERENCES CARTE(serie) ON DELETE CASCADE,

data\_imprumut DATE

CONSTRAINT FISA\_dataimprumut\_NN NOT NULL,

data\_returnare DATE,

CONSTRAINT FISA\_IMPR\_PK PRIMARY KEY (CNP,serie)

);

4.2.7. Crearea tabelului **AUTOR**

Am creat secvența AUTOR\_serie\_SEQ pentru a asigna valori cheii primare id\_autor.

CREATE TABLE AUTOR (

id\_autor NUMBER(4)

CONSTRAINT AUTOR\_idautor\_PK PRIMARY KEY,

nume VARCHAR(30)

CONSTRAINT AUTOR\_nume\_NN NOT NULL,

prenume VARCHAR(30)

CONSTRAINT AUTOR\_prenume\_NN NOT NULL,

data\_nastere DATE

CONSTRAINT AUTOR\_datanastere\_NN NOT NULL

);

CREATE SEQUENCE AUTOR\_serie\_SEQ

START WITH 1

INCREMENT BY 1;

4.2.8. Crearea tabelului **SCRIS**

CREATE TABLE SCRIS (

serie NUMBER(4)

CONSTRAINT SCRIS\_serie\_FK REFERENCES CARTE(serie) ON DELETE CASCADE,

id\_autor NUMBER(4)

CONSTRAINT SCRIS\_idautor\_FK REFERENCES AUTOR(id\_autor) ON DELETE CASCADE,

an\_finalizare NUMBER(4),

CONSTRAINT SCRIS\_PK PRIMARY KEY (serie,id\_autor)

);

4.2.9. Crearea tabelului **ROL**

Am creat secvența ROL\_idrol\_SEQ pentru a asigna valori cheii primare id\_rol.

CREATE TABLE ROL(

id\_rol NUMBER(4)

CONSTRAINT ROL\_idrol\_PK PRIMARY KEY,

titlu\_rol VARCHAR2(15)

CONSTRAINT ROL\_titlurol\_NN NOT NULL,

max\_sal NUMBER(4)

CONSTRAINT ROL\_maxsal\_NN NOT NULL,

min\_sal NUMBER(4)

CONSTRAINT ROL\_minsal\_NN NOT NULL

);

CREATE SEQUENCE ROL\_idrol\_SEQ

START WITH 1

INCREMENT BY 1;

4.2.10. Crearea tabelului **ANGAJAT**

Am creat secvența ANGAJAT\_idangajat\_SEQ pentru a asigna valori cheii primare id\_angajat.

CREATE TABLE ANGAJAT (

id\_angajat NUMBER(4)

CONSTRAINT ANGAJAT\_idangajat\_PK PRIMARY KEY,

nume VARCHAR2(30)

CONSTRAINT ANGAJAT\_nume\_NN NOT NULL,

prenume VARCHAR2(30)

CONSTRAINT ANGAJAT\_prenume\_NN NOT NULL,

data\_nastere DATE

CONSTRAINT ANGAJAT\_datanastere\_NN NOT NULL,

data\_angajarii DATE

CONSTRAINT ANGAJAT\_dataangajarii\_NN NOT NULL,

telefon VARCHAR2(10)

CONSTRAINT ANGAJAT\_telefon\_NN NOT NULL

CONSTRAINT ANGAJAT\_telefon\_U UNIQUE

CONSTRAINT ANGAJAT\_telefon\_C CHECK (LENGTH(telefon)=10),

id\_rol NUMBER(4)

CONSTRAINT ANGAJAT\_idrol\_FK REFERENCES ROL(id\_rol) ON DELETE CASCADE

CONSTRAINT ANGAJAT\_idrol\_NN NOT NULL,

numar\_sala NUMBER(4)

CONSTRAINT ANGAJAT\_numarsala\_FK REFERENCES SALA\_LECTURA(numar\_sala) ON DELETE CASCADE

CONSTRAINT ANGAJAT\_numarsala\_NN NOT NULL

);

CREATE SEQUENCE ANGAJAT\_idangajat\_SEQ

START WITH 3000

INCREMENT BY 1;

4.3. Introducerea datelor în baza de date

4.3.1. Introducerea datelor în tabelul **CITITOR**

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('2950603291234', 'Micu', 'Adriana', TO\_DATE('03-06-1995', 'DD-MM-YYYY'), '0725463995');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('1991203291335', 'Dima', 'Marian', TO\_DATE('03-12-1999', 'DD-MM-YYYY'), '0723473975');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('1001011291437', 'Craciun', 'Adelin', TO\_DATE('11-10-2000', 'DD-MM-YYYY'), '0721174995');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('6021120291739', 'Iacob', 'Crina', TO\_DATE('20-11-2002', 'DD-MM-YYYY'), '0721173335');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('1871222291741', 'Dumitrescu', 'Ion', TO\_DATE('22-12-1978', 'DD-MM-YYYY'), '0721444435');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('1881214291741', 'Ion', 'Mihai', TO\_DATE('14-12-1988', 'DD-MM-YYYY'), '0712345435');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('2960315292756', 'Pavel', 'Maria', TO\_DATE('15-03-1996', 'DD-MM-YYYY'), '0712345123');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('1851201291743', 'Crin', 'Andrei', TO\_DATE('01-12-1985', 'DD-MM-YYYY'), '0778945435');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('2930904291461', 'Iris', 'Mihaela', TO\_DATE('04-09-1993', 'DD-MM-YYYY'), '0712349631');

INSERT INTO CITITOR (CNP, nume, prenume, data\_nastere, nr\_telefon) VALUES

('5040904291321', 'Florea', 'Evelin', TO\_DATE('04-09-2004', 'DD-MM-YYYY'), '0712354631');

4.3.2. Introducerea datelor în tabelul **EDITURA**

INSERT INTO EDITURA(id\_editura, nume, oras, strada, nr\_telefon, email) VALUES

(EDITURA\_ideditura\_SEQ.NEXTVAL,'Corint','Bucuresti','Mihai Eminescu','0737555999','cedit@gmail.com');

INSERT INTO EDITURA(id\_editura, nume, oras, strada, nr\_telefon, email) VALUES

(EDITURA\_ideditura\_SEQ.NEXTVAL,'Hyperion','Craiova','Florilor','0737123456','hyper@gmail.com');

INSERT INTO EDITURA(id\_editura, nume, oras, strada, nr\_telefon, email) VALUES

(EDITURA\_ideditura\_SEQ.NEXTVAL,'Bookzone','Bucuresti','Sos.Berceni','0744423456','bzone@gmail.com');

INSERT INTO EDITURA(id\_editura, nume, oras, strada, nr\_telefon, email) VALUES

(EDITURA\_ideditura\_SEQ.NEXTVAL,'Polirom','Iasi','Bd.Copou','0744423555','polir@gmail.com');

INSERT INTO EDITURA(id\_editura, nume, oras, strada, nr\_telefon, email) VALUES

(EDITURA\_ideditura\_SEQ.NEXTVAL,'Litera','Bucuresti','Moeciu','0744141792','ledit@gmail.com');

INSERT INTO EDITURA(id\_editura, nume, oras, strada, nr\_telefon, email) VALUES

(EDITURA\_ideditura\_SEQ.NEXTVAL,'Carminis','Pitesti','Exercitiu','0736641122','carmi@gmail.com');

4.3.3. Introducerea datelor în tabelul **SALA\_LECTURA**

INSERT INTO SALA\_LECTURA(numar\_sala,numar\_locuri) VALUES (1,12);

INSERT INTO SALA\_LECTURA(numar\_sala,numar\_locuri) VALUES (2,20);

INSERT INTO SALA\_LECTURA(numar\_sala,numar\_locuri) VALUES (3,15);

INSERT INTO SALA\_LECTURA(numar\_sala,numar\_locuri) VALUES (4,25);

INSERT INTO SALA\_LECTURA(numar\_sala,numar\_locuri) VALUES (5,10);

4.3.4. Introducerea datelor în tabelul **SECTIUNE**

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)VALUES (SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Istorie',1,2,1);

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)VALUES (SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Geografie',2,3,2);

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)VALUES (SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Fictiune',3,2,1);

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)VALUES (SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Non-fictiune',4,1,1);

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)VALUES (SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Bibliografii',1,2,3);

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)

VALUES (SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Romance',3,3,1);

INSERT INTO SECTIUNE(id\_sectiune, nume\_domeniu, numar\_sala, nr\_raft, nr\_rand)VALUES(SECTIUNE\_idsectiune\_SEQ.NEXTVAL,'Matematica',5,1,1);

4.3.5. Introducerea datelor în tabelul **CARTE**

INSERT INTO CARTE(serie, titlu, id\_sectiune, id\_editura, nr\_pagini, an\_publicare)

VALUES (CARTE\_serie\_SEQ.NEXTVAL, 'Atlas Europa',2001,1000,80,2013);

INSERT INTO CARTE(serie, titlu, id\_sectiune, id\_editura, nr\_pagini, an\_publicare)

VALUES(CARTE\_serie\_SEQ.NEXTVAL, 'Cartea Noptii',2002,1002,464,2022);

INSERT INTO CARTE(serie, titlu, id\_sectiune, id\_editura, nr\_pagini, an\_publicare)

VALUES (CARTE\_serie\_SEQ.NEXTVAL, 'Sapiens',2000,1004,384,2017);

INSERT INTO CARTE(serie, titlu, id\_sectiune, id\_editura, nr\_pagini, an\_publicare)

VALUES (CARTE\_serie\_SEQ.NEXTVAL, 'Atomic Habits',2003,1004,271,2019);

INSERT INTO CARTE(serie, titlu, id\_sectiune, id\_editura, nr\_pagini, an\_publicare)

VALUES (CARTE\_serie\_SEQ.NEXTVAL, 'Me before you',2005,1004,416,2016);

INSERT INTO CARTE(serie, titlu, id\_sectiune, id\_editura, nr\_pagini, an\_publicare)

VALUES(CARTE\_serie\_SEQ.NEXTVAL, 'Matematica M1',2006,1005,336,2006);

4.3.6. Introducerea datelor în tabelul **FISA\_IMPRUMUT**

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('2950603291234',1002,TO\_DATE('10-12-2022', 'DD-MM-YYYY'),TO\_DATE('06-01-2023', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('2950603291234',1003,TO\_DATE('10-12-2022', 'DD-MM-YYYY'),TO\_DATE('06-01-2023', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('1991203291335',1000,TO\_DATE('12-12-2022', 'DD-MM-YYYY'),TO\_DATE('23-12-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('5001011291437',1002,TO\_DATE('11-11-2022', 'DD-MM-YYYY'),TO\_DATE('25-11-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('1881214291741',1002,TO\_DATE('01-12-2022', 'DD-MM-YYYY'),TO\_DATE('08-12-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('5040904291321',1000,TO\_DATE('07-01-2023', 'DD-MM-YYYY'),TO\_DATE('07-01-2023', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('6021120291739',1000,TO\_DATE('12-01-2023', 'DD-MM-YYYY'),NULL);

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('1871222291741',1005,TO\_DATE('03-10-2022', 'DD-MM-YYYY'),TO\_DATE('20-11-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('2960315292756',1004,TO\_DATE('12-02-2022', 'DD-MM-YYYY'),TO\_DATE('20-02-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('1851201291743',1004,TO\_DATE('01-02-2022', 'DD-MM-YYYY'),TO\_DATE('5-02-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('2930904291461',1003,TO\_DATE('17-04-2022', 'DD-MM-YYYY'),TO\_DATE('25-04-2022', 'DD-MM-YYYY'));

INSERT INTO FISA\_IMPRUMUT(CNP, serie, data\_imprumut, data\_returnare) VALUES

('2930904291461',1005,TO\_DATE('26-05-2022', 'DD-MM-YYYY'),TO\_DATE('29-05-2022', 'DD-MM-YYYY'));

4.3.7. Introducerea datelor în tabelul **AUTOR**

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Mandrut', 'Octavian', TO\_DATE('15-01-1960', 'DD-MM-YYYY'));

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Black', 'Holly', TO\_DATE('10-11-1971', 'DD-MM-YYYY'));

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Harari', 'Yuval Noah', TO\_DATE('04-02-1976', 'DD-MM-YYYY'));

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Clear', 'James', TO\_DATE('22-01-1986', 'DD-MM-YYYY'));

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Jojo', 'Moyes', TO\_DATE('04-09-1969', 'DD-MM-YYYY'));

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Burtea', 'Marius', TO\_DATE('04-09-1965', 'DD-MM-YYYY'));

INSERT INTO AUTOR(id\_autor, nume, prenume, data\_nastere) VALUES

(AUTOR\_serie\_SEQ.NEXTVAL,'Burtea', 'Georgeta', TO\_DATE('14-10-1966', 'DD-MM-YYYY'));

4.3.8. Introducerea datelor în tabelul **SCRIS**

INSERT INTO SCRIS(serie, id\_autor, an\_finalizare) VALUES (1000,1,NULL);

INSERT INTO SCRIS(serie, id\_autor, an\_finalizare) VALUES (1001,2,NULL);

INSERT INTO SCRIS(serie, id\_autor, an\_finalizare) VALUES (1002,3,NULL);

INSERT INTO SCRIS(serie, id\_autor, an\_finalizare) VALUES (1004,5,NULL);

INSERT INTO SCRIS(serie, id\_autor, an\_finalizare) VALUES (1005,6,2006);

INSERT INTO SCRIS(serie, id\_autor, an\_finalizare) VALUES (1005,7,2006);

4.3.9. Introducerea datelor în tabelul **ROL**

INSERT INTO ROL(id\_rol, titlu\_rol, max\_sal, min\_sal) VALUES (ROL\_idrol\_SEQ.NEXTVAL, 'Bibliotecar',3500,1525);

INSERT INTO ROL(id\_rol, titlu\_rol, max\_sal, min\_sal) VALUES (ROL\_idrol\_SEQ.NEXTVAL, 'Supraveghetor',2500,1525);

INSERT INTO ROL(id\_rol, titlu\_rol, max\_sal, min\_sal) VALUES (ROL\_idrol\_SEQ.NEXTVAL, 'Organizator',3000,1525);

INSERT INTO ROL(id\_rol, titlu\_rol, max\_sal, min\_sal) VALUES (ROL\_idrol\_SEQ.NEXTVAL, 'Pers Curatenie',2000,1525);

4.3.10. Introducerea datelor în tabelul **ANGAJAT**

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES (ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Mitu','Alina',TO\_DATE('10-11-1971', 'DD-MM-YYYY'),TO\_DATE('10-10-2010', 'DD-MM-YYYY'),'0734003145',1,1);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Nita','Mariana',TO\_DATE('12-02-1970', 'DD-MM-YYYY'),TO\_DATE('05-07-2020', 'DD-MM-YYYY'),'0723400321',2,1);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Popa','Catalin',TO\_DATE('17-06-1990', 'DD-MM-YYYY'),TO\_DATE('12-03-2021', 'DD-MM-YYYY'),'0713400925',3,1);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Calin','Mihaela',TO\_DATE('16-12-1985', 'DD-MM-YYYY'),TO\_DATE('14-08-2022', 'DD-MM-YYYY'),'0723410321',4,1);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Mihai','Andreea',TO\_DATE('10-01-1981', 'DD-MM-YYYY'),TO\_DATE('10-10-2016', 'DD-MM-YYYY'),'0734003149',1,2);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Gica','Paul',TO\_DATE('23-10-1979', 'DD-MM-YYYY'),TO\_DATE('05-07-2021', 'DD-MM-YYYY'),'0723444321',2,2);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Sovarel','Steliana',TO\_DATE('19-05-1982', 'DD-MM-YYYY'),TO\_DATE('12-03-2019', 'DD-MM-YYYY'),'0765400925',3,2);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Dudu','Alexandra',TO\_DATE('12-10-1990', 'DD-MM-YYYY'),TO\_DATE('14-08-2022', 'DD-MM-YYYY'),'0723478921',4,2);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Dumitru','Madalina',TO\_DATE('10-08-1988', 'DD-MM-YYYY'),TO\_DATE('10-10-2017', 'DD-MM-YYYY'),'0734034945',1,3);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Alexandru','Erwin',TO\_DATE('12-02-1994', 'DD-MM-YYYY'),TO\_DATE('29-05-2021', 'DD-MM-YYYY'),'0723433321',2,3);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Ilie','Flori',TO\_DATE('17-11-1990', 'DD-MM-YYYY'),TO\_DATE('12-03-2021', 'DD-MM-YYYY'),'0713147825',3,3);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Grigore','Mihnea',TO\_DATE('16-12-1977', 'DD-MM-YYYY'),TO\_DATE('14-08-2022', 'DD-MM-YYYY'),'0732110321',4,3);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES

(ANGAJAT\_idangajat\_SEQ.NEXTVAL,'Mihai','Daniela',TO\_DATE('06-08-1991', 'DD-MM-YYYY'),TO\_DATE('10-10-2018', 'DD-MM-YYYY'),'0734064745',1,4);

INSERT INTO ANGAJAT(id\_angajat, nume, prenume, data\_nastere, data\_angajarii, telefon, id\_rol, numar\_sala) VALUES