UNIVERSITATEA BUCURESTI

FACULTATEA DE MATEMATICA SI INFORMATICA

DEPARTAMENTUL CALCULATOARE SI TEHNOLOGIA INFORMATIEI

**PROIECT BAZE DE DATE**

**MANAGEMENTUL LIBRARIILOR**

Profesor coordinator: Student:

Vasile Silviu-Laurentiu Stoian Vlad

**Cuprins**

Prezentarea bazei de date…………………pagina 3

Regulile bazei de date……………………….pagina 3

Diagrama Entitate – Relatie………………pagina 4

Diagrama Conceptuala……………………..pagina 10

Scriptul din MySQL…………………………..pagina 17

1. **Prezentarea bazei de date**

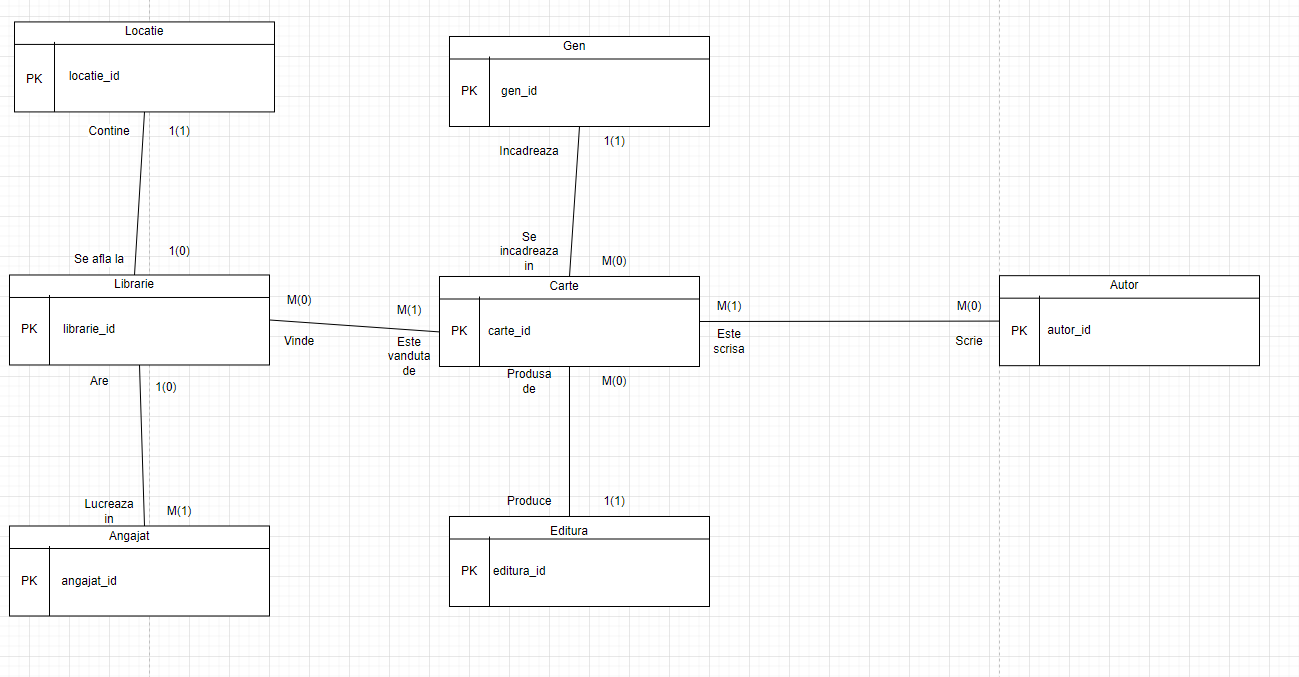
Pentru proiectul de la disciplina Baze de Date am ales tema ,,Managementul librariilor”. Baza de date inregistreaza librarii la nivel national.

Toate librariile au nevoie de o baza de date pusa la punct daca acestea vor sa fie organizate. Astfel, si angajatii si potentialii clineti pot aveea acces la informatii legate de stocurile magazinzelor si alte detalii cu precizie, lucru ce usureaza munca ambelor parti. O baza de date este esentiala pentru majoritatea intreprindrilor moderne.

De aceea am realizat o baza de date care are ca scop gestionarea datelor legate de librariile afiliate, mai exact cartile care sunt pe stoc, autorii cartilor respective, genurile literare, editurile, librariile, locatiile si angajatii.

1. **Regulile bazei de date**
2. Toate copiile unei anumite carti impart acelasi id.
3. Mai multi autori pot face o colaborare pentru o anumita carte.
4. O carte poate sa aiba un autor necunoscut.
5. Pentru ca o persoana sa fie considerata un autor de carti, aceasta trebuie sa fi scris minimum o carte.
6. Un gen nu trebuie sa incadreze neaparat vreo carte din baza de date; de asemenea un gen poate incadra mai multe carti.
7. O carte este incadrata intr-un singur gen.
8. Pot exista edituri care nu au produs nicio carte; de asemenea o editura poate produce mai multe carti.
9. O carte este produse de o singura editura.
10. O carte poate sa nu fie vanduta in vreo librarie; de asemenea o carte poate sa fie vanduta in mai multe librarii.
11. O librarie poate vinde una sau mai multe carti.
12. O librarie se alfa la o singura locatie.
13. O locatie contine maximum o librarie (deoarece exista locatii de interes pentru viitoare librarii in baza de date).
14. O librarie are unul sau mai multi angajati.
15. Exista doua tipuri de angajati: cei care lucreaza intr-o singura librarie si cei care lucreaza in afara librariilor (ex. livratori)
16. In tabelul vinde\_p, daca data\_disponibilitate este NULL atunci cartea din aceasta intrare nu mai este pe stoc; asta inseamna ca aceasta carte a fost vanduta in trecut in libraria din aceasta intrare dar acum nu mai este vanduta
17. **Diagrama Entitate-Relatie**

Reprezentarea diagramei:



**Descrierea entitatilor, atributelor si a cheilor:**

1. Tabelul Autor:

Tabelul Autor stocheaza informatii legate de autorii care se afla in baza de date:

* PK autor\_id – identificatorul unic al autorului
* nume\_familie – numele de familie al autorului
* prenume – prenumele autorului (daca are mai multe se trece doar unul)
* data\_nastere – data nasterii autorului

1. Tabelul Carte:

Tabelul Carte stocheaza informatii legate de cartile care se afla in baza de date:

* PK carte\_id – identificatorul unic al cartii
* FK gen\_id – id-ul genului ce incadreaza cartea
* FK editura\_id – id-ul editurii care produce cartea
* titlu – titlul cartii
* numar\_pagini – numarul de pagini al cartii
* tip\_coperta – tipul de coperta al cartii (paperback sau hardcover)
* limba\_text – limba in care este scrisa cartea
* an\_publicare – anul de publicare al cartii

1. Tabelul Gen:

Tabelul Gen stocheaza informatii legate de genurile literare care se afla in baza de date:

* PK gen\_id – identificatorul unic la genului
* nume\_gen – numele genului literar

1. Tabelul Editura:

Tabelul Editura stocheaza informatii legate de editurile care se afla in baza de date:

* PK editura\_id – identificatorul unic al editurii
* nume\_editura – numele editurii
* tara\_origine – tara in care s-a fondat editura
* nr\_telefon – un numar de telefon prin care poti contacta editura
* link\_site – link-ul prin care poti accesa site-ul editurii

1. Tabelul Librarie:

Tabelul Librarie stocheaza informatii legate de librariile care se afla in baza de date:

* PK librarie\_id – identificatorul unic al librariei
* FK locatie\_id ­– id-ul locatiei la care se afla librarie
* nume – numele librariei
* nr\_telefon – numarul de telefon prin care poti contacta librarie
* email – email-ul prin care poti contacta libraria

1. Tabelul Locatie

Tabelul Locatie stocheaza informatii legate de locatiile librariilor actuale dar si locatiile de interes pentru librariile din viitor:

* PK locatie\_id – identificatorul unic al locatiei
* oras – orasul in care se afla locatia
* judel – judetul in care se afla locatia
* nume\_strada – numele strazii in care se afla locatia
* nr\_strada – numarul strazii in care se afla locatia

1. Tabelul Angajat

Tabelul Angajat stocheaza informatii legate de angajatii care lucreaza in cadrul librariilor, dar si informatii legate de angajatii care lucreaza in afara acestora:

* PK angajat\_id – identificatorul unic al angajatului
* FK librarie\_id – id-ul librariei in care lucreaza angajatul (NULL daca acesta nu lucreaza in cadrul vreunei librarii)
* nume – numele angajatului
* prenume – prenumele angajatului
* data\_nasterii – data de nastere a angajatului
* nr\_telefon – numarul de telefon al angajatului
* email – email-ul angajatului
* salariu – salariul angajatului

**Descrierea relatiilor si a cardinalitatilor:**

* CARTE – AUTOR

Autorii scriu cartile.

Cartile sunt scrise de catre autori.

Cardinalitate maximala:

Cate carti poate scrie un autor? – multe

Cati autori poate avea o carte? – mai multi

Cardinalitate minimala:

Cate carti trebuie sa scrie un autor? – 1

De cati autori trebuie sa fie scrisa o carte? – 0

* CARTE – GEN

Cartile se incadreaza in genuri literare.

Genurile incadreaza carti.

Cardinalitate maximala:

Cate carti se pot incadra intr-un gen literar? – multe

De cate genuri poate o carte sa fie incadrata? – 1

Cardinalitate minimala:

Cate carti trebuie sa fie incadrate intr-un gen literar? – 0

De cate genuri trebuie o carte sa fie incadrata? – 1

* CARTE – EDITURA

Cartile sunt produse de edituri.

Editurile produc carti.

Cardinalitate maximala:

Cate carti pot fi produse de o editura? – multe

De cate edituri poate o carte sa fie produse? – 1

Cardinalitate minimala:

Cate carti trebuie sa fie produse de o editura? – 0

De cate edituri trebuie o carte sa fie produsa? – 1

* CARTE – LIBRARIE

Cartile sunt vandute de librarii.

Librariile vand carti.

Cardinalitate maximala:

Cate carti pot fi vandute de o librarie? – multe

De cate librarii poate o carte sa fie vanduta? – multe

Cardinalitate minimala:

Cate carti trebuie sa fie vandute de o librarie? – 1

De cate librarii trebuie o carte sa fie vanduta? – 0

* LIBRARIE – LOCATIE

Librariile se afla la anumite locatii.

Locatiile contin librarii.

Cardinalitate maximala:

Cate librarii pot fi continute de o locatie? – 1

La cate locatii se poate afla o librarie? – 1

Cardinalitate minimala:

Cate librarii trebuie sa fie continute de o locatie? – 0

La cate locatii trebuie sa se afle o librarie? – 1

* LIBRARIE – ANGAJAT

Libraria are angajati.

Angajatii lucreaza in librarii.

Cardinalitate maximala:

In cate librarii poate un angajat sa lucreze? – 1

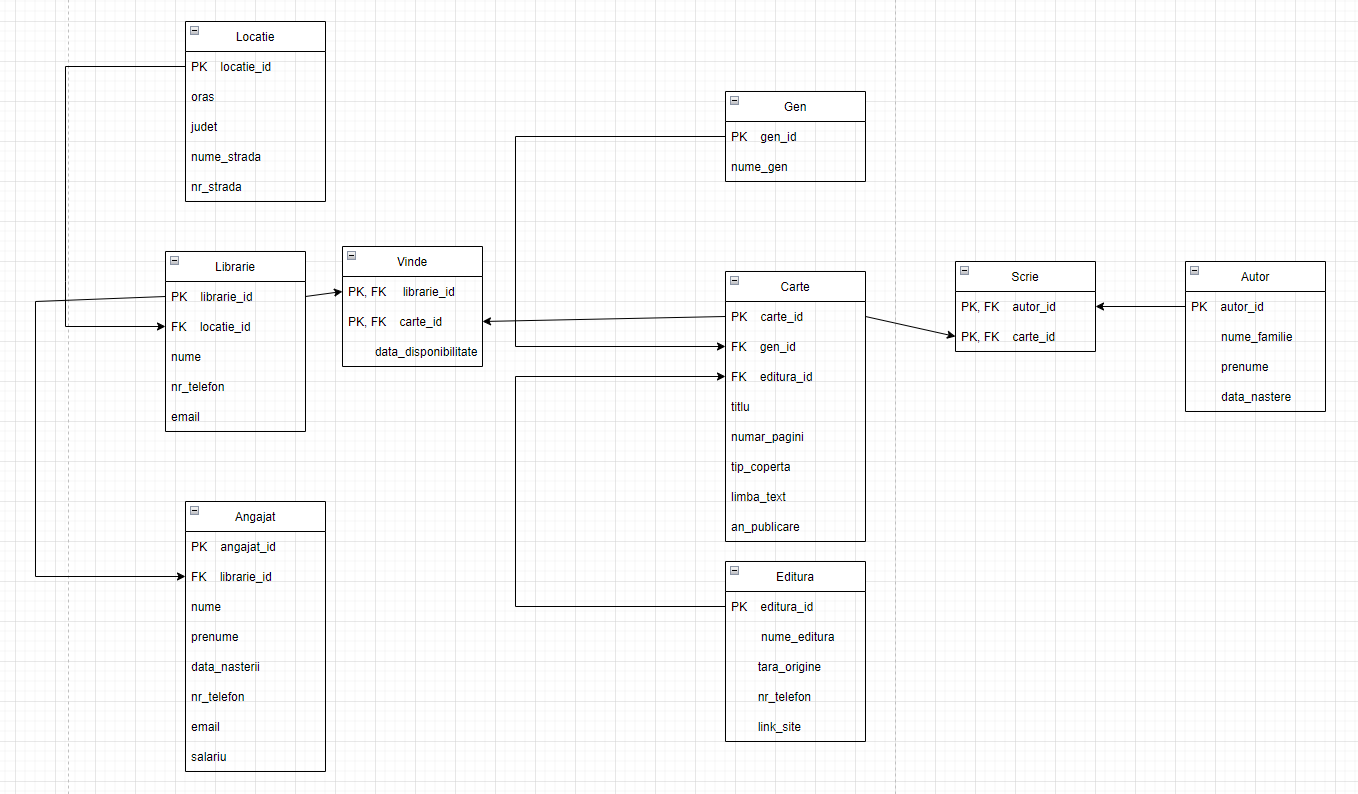
Cati angajati pot lucra intr-o librarie – multi

Cardinalitate minimala:

In cate librarii trebuie un angajat sa lucreze? – 0

Cati angajati trebuie sa lucreze intr-o librarie? – 1

1. **Diagrama Conceptuala**



**Descrierea constrangerilor de integritate**

Tabelul locatie:

* Constrangeri de tip Primary Key:

locatie\_id\_pk\_p pentru atributul locatie\_id

* Constrangeri de tip Not Null:

oras\_not\_null\_p pentru atributul oras (locatia trebuie obligatoriu sa aiba un oras)

judet\_not\_null pentru atributul judet (locatia trebuie obligatoriu sa aiba un judet)

nume\_strada\_not\_null\_p pentru atributul nume\_strada (locatia trebuie obligatoriu sa aiba un nume de strada)

nr\_strada\_not\_null\_p pentru atributul nr\_strada (locatia trebuie obligatoriu sa aiba un numar de strada)

Tabelul librarie:

* Constrangeri de tip Primary Key:

librarie\_id\_pk\_p pentru atributul librarie\_id

* Constrangeri de tip Not Null:

nume\_librarie\_not\_null\_p pentru atributul nume (libraria trebuie obligatoriu sa aiba un nume)

locatie\_id\_not\_null\_p pentru atributul locatie\_id nume (libraria trebuie obligatoriu sa aiba o locatie)

nr\_tel\_lib\_not\_null\_p pentru atributul nr\_telefon nume (libraria trebuie obligatoriu sa aiba un numar de telefon)

email\_lib\_not\_null\_p pentru atributul email nume (libraria trebuie obligatoriu sa aiba un email)

* Constrangeri de tip Unique:

locatie\_id\_unica\_p pentru atributul locatie\_id (fiecare librarie trebuie sa aiba o locatie unica)

nr\_telefon\_librarie\_p pentru atributul nr\_telefon (fiecare librarie trebuie sa aiba un numar de telefon unic)

email\_librarie\_p pentru atributul email (fiecare librarie trebuie sa aiba un email unic)

* Constrangeri de tip Foreign Key:

locatie\_id\_fk\_in\_librarie\_p pentru atributul locatie\_id (aceast atribut provine din tabelul locatie, asa ca face legatura intre tabelul librarie si tabelul locatie, si reprezinta locatia la care se afla libraria)

Tabelul angajat:

* Constrangeri de tip Primary Key:

angajat\_id\_pk\_p pentru atributul angajat\_id

* Constrangeri de tip Not Null:

nume\_ang\_not\_null\_p pentru atributul nume (fiecare angajat trebuie sa aiba un nume)

prenume\_ang\_not\_null\_p pentru atributul prenume (fiecare angajat trebuie sa aiba un prenume)

data\_nasterii\_ang\_not\_null\_p pentru atributul data\_nasterii (fiecare angajat trebuie sa aiba o data a nasterii)

nr\_tel\_ang\_not\_null\_p pentru atributul nr\_telefon (fiecare angajat trebuie sa aiba un numar de telefon)

email\_ang\_not\_null\_p pentru atributul email (fiecare angajat trebuie sa aiba un email)

salariu\_ang\_not\_null\_p pentru atributul salariu (fiecare angajat trebuie sa aiba un salariu)

* Constrangeri de tip Unique:

nr\_telefon\_ang\_unique\_p pentru atributul nr\_telefon (fiecare angajat trebuie sa aiba un numar de telefon unic)

email\_ang\_unique\_p pentru atributul email (fiecare angajat trebuie sa aiba un email unic)

* Constrangeri de tip Foreign Key:

librarie\_id\_angajat\_fk\_p pentru atributul librarie\_id (aceast atribut provine din tabelul librarie, astfel face legatura dintre tabelul angajat si tabelul librarie, si reprezinta libraria in care lucreaza angajatul)

Tabelul gen:

* Constrangeri de tip Primary Key:

gen\_id\_gen\_pk\_p pentru atributul gen\_id

* Constrangeri de tip Not Null:

nume\_gen\_not\_null\_p pentru atributul nume\_gen (fiecare gen are nevoie de un nume)

* Constrangeri de tip Unique:

nume\_gen\_unique\_p pentru atributul nume\_gen (fiecare gen are nevoie de un nume unic)

Tabelul editura:

* Constrangeri de tip Primary Key:

editura\_id\_editura\_pk\_p pentru atributul editura\_id

* Constrangeri de tip Not Null:

nume\_editura\_not\_null\_p pentru atributul nume\_editura (fiecare editura are nevoie de un nume)

nr\_telefon\_editura\_not\_null\_p pentru atributul nr\_telefon (fiecare editura are nevoie de un numar de telefon)

link\_site\_editura\_not\_null\_p pentru atributul link\_site (fiecare editura are nevoie de de un link pentru site-ul lor)

* Constrangeri de tip Unique:

nr\_telefon\_editura\_unique\_p pentru atributul nr\_telefon (fiecare editura are nevoie de un numar de telefon unic)

link\_site\_editura\_unique\_p pentru atributul link\_site site (fiecare editura are nevoie de de un link pentru site-ul lor unic)

Tabelul autor:

* Constrangeri de tip Primary Key:

autor\_id\_autor\_pk\_p pentru atributul autor\_id

Tabelul carte:

* Constrangeri de tip Primary Key:

carte\_id\_carte\_pk\_p pentru atributul carte\_id

* Constrangeri de tip Not Null:

gen\_id\_carte\_not\_null\_p pentru atributul gen\_id (fiecare carte are nevoie de un gen literar)

editura\_id\_carte\_not\_null\_p pentru atributul editura\_id (fiecare carte are nevoie de o editura)

titlu\_carte\_not\_null\_p pentru atributul titlu (fiecare carte are nevoie de un titlu)

numar\_pagini\_carte\_not\_null\_p pentru atributul numar\_pagini (fiecare carte are nevoie de un numar de pagini)

tip\_coperta\_carte\_not\_null\_p pentru atributul tip\_coperta (fiecare carte are nevoie de un tip de coperta)

limba\_text\_carte\_not\_null\_p pentru atributul limba\_text (fiecare carte are nevoie de o limba in care este scris continutul cartii)

* Constrangeri de tip Foreign Key:

gen\_id\_carte\_fk\_p pentru atributul gen\_id (aceast atribut provine din tabelul gen, astfel face legatura dintre tabelul carte si tabelul gen, si reprezinta genul care incadreaza cartea)

editura\_id\_carte\_fk\_p pentru atributul editura\_id (aceast atribut provine din tabelul editura, astfel face legatura dintre tabelul carte si tabelul editura, si reprezinta editura care produce cartea)

Tabelul vinde:

* Constrangeri de tip Primary Key:

pk\_vinde\_p pentru atributele librarie\_id si carte\_id (in acest tabel avem o cheie primara compusa)

* Constrangeri de tip Foreign Key:

librarie\_id\_vinde\_fk\_p pentru atributul librarie\_id (aceast atribut provine din tabelul librarie si reprezinta libraria care vinde cartea)

carte\_id\_vinde\_fk\_p pentru atributul carte\_id (aceast atribut provine din tabelul carte si reprezinta cartea care este vanduta de o librarie)

Tabelul scrie:

* Constrangeri de tip Primary Key:

pk\_scrie\_p pentru atributele autor\_id si carte\_id (in acest tabel avem o cheie primara compusa)

* Constrangeri de tip Foreign Key:

autor\_id\_scrie\_fk\_p pentru atributul autor\_id (aceast atribut provine din tabelul autor si reprezinta autorul care a scris cartea)

carte\_id\_scrie\_fk\_p (aceast atribut provine din tabelul carte si reprezinta cartea care a fost scrisa de catre un autor)

**Schemele relationale sunt:**

* locatie: (PK) locatie\_id, oras, judet, nume\_strada, nr\_strada
* librarie: (PK) librarie\_id, (FK) locatie\_id, nume, nr\_telefon, email
* angajat: (PK) angajat\_id, (FK) librarie\_id, nume, prenume, data\_nasterii, nr\_telefon, email, salariu
* vinde: (PK) (FK) librarie\_id, (PK) (FK) carte\_id, data\_disponibilitate
* gen: (PK) gen\_id, nume\_gen
* carte: (PK) carte\_id, (FK) gen\_id, (FK) editura\_id, titlu, numar\_pagini, tip\_coperta, limba\_text, an\_publicare
* editura: (PK) editura\_id, nume\_editura, tara\_origine, nr\_telefon, link\_site
* scrie: (PK) (FK) autor\_id, (PK) (FK) carte\_id
* autor: (PK) autor\_id, nume\_familie, prenume, data\_nastere

**Descrierea constrangerilor on delete:**

* locatie\_id\_fk\_in\_librarie\_p on delete cascade

cand o locatie este stearsa din baza de date, toate librariile care aveau codul locatiei respective vor fi sterse deoarece nu exista librarii fara locatii

* librarie\_id\_angajat\_fk\_p on delete set null

cand o librarie este stearsa din baza de date, atributul librarie\_id din tabelul angajati isi va schimba valoarea in NULL, deoarece atunci cand o librarie este scoasa din baza de date angajatii care lucrau in aceasta vor ocupa job-uri in afara libariilor (ex. livrator), astfel vor ramane in baza de date

* gen\_id\_carte\_fk\_p on delete cascade

cand un gen literar este sters din baza de date, toate cartile care aveau codul genului literar respectiv vor fi sterse deoarece nu exista carti fara gen literar

* editura\_id\_carte\_fk\_p on delete cascade

cand o editura este stearsa din baza de date, toate cartile care aveau codul editurii respective vor fi sterse deoarece nu exista carti fara editura

* librarie\_id\_vinde\_fk\_p on delete cascade

cand o librarie este stearsa din baza de date, toate intrarile care contineau id-ul librariei respective vor fi sterse din tabelul vinde, deoarece libraria a fost stearsa si nu mai poate vinde vreo carte

* carte\_id\_vinde\_fk\_p on delete cascade

cand o carte este stearsa din baza de date, toate intrarile care contineau id-ul cartii respective vor fi sterse din tabelul vinde, deoarece cartea a fost stearsa si nu mai poate fi vanduta in vreo librarie

* autor\_id\_scrie\_fk\_p on delete cascade

cand un autor este stears din baza de date, toate intrarile care contineau id-ul autorului respectiv vor fi sterse din tabelul scrie, deoarece autorul a fost sters si nu mai poate avea carti scrise de el

* carte\_id\_scrie\_fk\_p on delete cascade

cand o carte este stearsa din baza de date, toate intrarile care contineau id-ul cartii respective vor fi sterse din tabelul scrie, deoarece cartea a fost stearsa si nu mai poate fi scrisa de vreun autor

1. **Scriptul din MySQL**

Scriptul este salvat cu numele scriptsql\_Stoian\_Vlad\_261.

drop table scrie\_p;

drop table vinde\_p;

drop table carte\_p;

drop table gen\_p;

drop table editura\_p;

drop table autor\_p;

drop table angajat\_p;

drop table librarie\_p;

drop table locatie\_p;

create table locatie\_p (

locatie\_id int,

oras varchar(20) constraint oras\_not\_null\_p check(oras is not null),

judet varchar(20) constraint judet\_not\_null\_p check(judet is not null),

nume\_strada varchar(20) constraint nume\_strada\_not\_null\_p check(nume\_strada is not null),

nr\_strada int constraint nr\_strada\_not\_null\_p check(nr\_strada is not null),

constraint locatie\_id\_pk\_p primary key(locatie\_id)

);

insert into locatie\_p

values(1001, 'Constanta', 'Constanta', 'Strada Eliberarii', 12);

insert into locatie\_p

values(1002, 'Sector 3 ', 'Bucuresti', 'Nicolae Grigorescu', 21);

insert into locatie\_p

values(1003, 'Sector 4 ', 'Bucuresti', 'Panselelor', 23);

insert into locatie\_p

values(1004, 'Constanta', 'Constanta', 'Caragiale', 12);

insert into locatie\_p

values(1005, 'Timisora', 'Timis', '1 Decembrie 1918', 11);

create table librarie\_p (

librarie\_id int,

locatie\_id int,

nume varchar(20) constraint nume\_librarie\_not\_null\_p check(nume is not null),

nr\_telefon int,

email varchar(20),

constraint locatie\_id\_fk\_in\_librarie\_p foreign key(locatie\_id)

references locatie\_p(locatie\_id) on delete cascade,

constraint locatie\_id\_unica\_p unique(locatie\_id),

constraint locatie\_id\_not\_null\_p check(locatie\_id is not null),

constraint nr\_tel\_lib\_not\_null\_p check(nr\_telefon is not null),

constraint email\_lib\_not\_null\_p check(email is not null),

constraint librarie\_id\_pk\_p primary key(librarie\_id),

constraint nr\_telefon\_librarie\_p unique(nr\_telefon),

constraint email\_librarie\_p unique(email)

);

insert into librarie\_p

values(1001, 1001, 'Carturesti', 0700000001, 'librarie1@gmail.com');

insert into librarie\_p

values(1002, 1002, 'Alexandria', 0700000002, 'librarie2@gmail.com');

insert into librarie\_p

values(1003, 1003, 'Libraria Nationala', 0700000003, 'librarie3@gmail.com');

insert into librarie\_p

values(1004, 1004, 'Libraria Constantei', 0700000004, 'librarie4@gmail.com');

create table angajat\_p (

angajat\_id int,

librarie\_id int,

nume varchar(20) constraint nume\_ang\_not\_null\_p check(nume is not

null),

prenume varchar(20) constraint prenume\_ang\_not\_null\_p check(prenume is not null),

data\_nasterii date constraint data\_nasterii\_ang\_not\_null\_p check(data\_nasterii is not null),

nr\_telefon int constraint nr\_tel\_ang\_not\_null\_p check(nr\_telefon is not null),

email varchar(20) constraint email\_ang\_not\_null\_p check(email is not null),

salariu int constraint salariu\_ang\_not\_null\_p check(salariu is not null),

constraint nr\_telefon\_ang\_unique\_p unique(nr\_telefon),

constraint email\_ang\_unique\_p unique(email),

constraint librarie\_id\_angajat\_fk\_p foreign key(librarie\_id)

references librarie\_p(librarie\_id) on delete set null,

constraint angajat\_id\_pk\_p primary key(angajat\_id)

);

insert into angajat\_p

values(1001, NULL, 'Popescu', 'Alex', str\_to\_date('01-02-1997', '%d-%m-%Y'), 0700000005, 'angajat5@gmail.com', 3000);

insert into angajat\_p

values(1002, 1001, 'Popescu', 'Andrei', str\_to\_date('03-04-1998', '%d-%m-%Y'), 0700000006, 'angajat6@gmail.com', 3500);

insert into angajat\_p

values(1003, 1001, 'Stoian', 'Mihail', str\_to\_date('06-01-1995', '%d-%m-%Y'), 0700000007, 'angajat7@gmail.com', 5000);

insert into angajat\_p

values(1004, 1002, 'Ion', 'Alexandra', str\_to\_date('01-01-1992', '%d-%m-%Y'), 0700000008, 'angajat8@gmail.com', 4500);

insert into angajat\_p

values(1005, 1003, 'Iancu', 'Gabriel', str\_to\_date('06-09-1991', '%d-%m-%Y'), 0700000009, 'angajat9@gmail.com', 5000);

insert into angajat\_p

values(1006, 1004, 'Geogrica', 'Maria', str\_to\_date('10-12-1989', '%d-%m-%Y'), 0700000010, 'angajat10@gmail.com', 3500);

create table gen\_p (

gen\_id int,

nume\_gen varchar(20) constraint nume\_gen\_not\_null\_p check(nume\_gen is not null),

constraint nume\_gen\_unique\_p unique(nume\_gen),

constraint gen\_id\_gen\_pk\_p primary key(gen\_id)

);

insert into gen\_p

values(1001, 'Horror');

insert into gen\_p

values(1002, 'Sci-Fi');

insert into gen\_p

values(1003, 'Non-fiction');

insert into gen\_p

values(1004, 'Fantasy');

insert into gen\_p

values(1005, 'Fiction');

create table editura\_p (

editura\_id int,

nume\_editura varchar(20) constraint nume\_editura\_not\_null\_p check(nume\_editura is not null),

tara\_origine varchar(20),

nr\_telefon int constraint nr\_telefon\_editura\_not\_null\_p check(nr\_telefon is not null),

link\_site varchar(20) constraint link\_site\_editura\_not\_null\_p check(link\_site is not null),

constraint nr\_telefon\_editura\_unique\_p unique(nr\_telefon),

constraint link\_site\_editura\_unique\_p unique(link\_site),

constraint editura\_id\_editura\_pk\_p primary key(editura\_id)

);

insert into editura\_p

values(1001, 'Litera', 'Romania', 0700000011, 'litera.com');

insert into editura\_p

values(1002, 'Orizont', NULL, 0700000012, 'orizont.com');

insert into editura\_p

values(1003, 'Fun Studios', 'America', 0700000013, 'funcstudios.com');

insert into editura\_p

values(1004, 'Smart Edit', 'Germania', 0700000014, 'smartedit.com');

insert into editura\_p

values(1005, 'Good Reads', NULL, 0700000015, 'goodreads.com');

create table autor\_p (

autor\_id int,

nume\_familie varchar(20),

prenume varchar(20),

data\_nastere date,

constraint autor\_id\_autor\_pk\_p primary key(autor\_id)

);

insert into autor\_p

values(1001, 'Stoian', 'Andrei', str\_to\_date('14-11-2001', '%d-%m-%Y'));

insert into autor\_p

values(1002, 'Alexander', NULL, NULL);

insert into autor\_p

values(1003, 'Popa', 'Eduard', str\_to\_date('11-08-1987', '%d-%m-%Y'));

insert into autor\_p

values(1004, 'Vasile', 'Anca', NULL);

insert into autor\_p

values(1005, 'Stone', 'Emma', str\_to\_date('09-11-1998', '%d-%m-%Y'));

create table carte\_p (

carte\_id int,

gen\_id int constraint gen\_id\_carte\_not\_null\_p check(gen\_id is not null),

editura\_id int constraint editura\_id\_carte\_not\_null\_p check(editura\_id is not null),

titlu varchar(50) constraint titlu\_carte\_not\_null\_p check(titlu is not null),

numar\_pagini int constraint numar\_pagini\_carte\_not\_null\_p check(numar\_pagini is not null),

tip\_coperta varchar(20) constraint tip\_coperta\_carte\_not\_null\_p check(tip\_coperta is not null),

limba\_text varchar(20) constraint limbe\_text\_carte\_not\_null\_p check(limba\_text is not null),

an\_publicare int,

constraint gen\_id\_carte\_fk\_p foreign key(gen\_id)

references gen\_p(gen\_id) on delete cascade,

constraint editura\_id\_carte\_fk\_p foreign key(editura\_id)

references editura\_p(editura\_id) on delete cascade,

constraint carte\_id\_carte\_pk\_p primary key(carte\_id)

);

insert into carte\_p

values(1001, 1001, 1001, 'It', 1200, 'Hard cover', 'Engleza', 2019);

insert into carte\_P

values(1002, 1001, 1001, 'Pet Sematary', 800, 'Paperback', 'Romana', 2020);

insert into carte\_p

values(1003, 1002, 1002, 'Hyperion', 600, 'Paperback', 'Engleza', 2019);

insert into carte\_p

values(1004, 1004, 1004, 'Circe', 400, 'Hard cover', 'Germana', 2017);

insert into carte\_p

values(1005, 1005, 1005, 'A Little Life', 800, 'Paper back', 'Engleza', 2017);

insert into carte\_p

values(1006, 1001, 1001, 'Misery', 700, 'Paper back', 'Romana', 2018);

create table vinde\_p (

librarie\_id int,

carte\_id int,

data\_disponibilitate date,

constraint pk\_vinde\_p primary key(librarie\_id, carte\_id),

constraint librarie\_id\_vinde\_fk\_p foreign key(librarie\_id)

references librarie\_p(librarie\_id) on delete cascade,

constraint carte\_id\_vinde\_fk\_p foreign key(carte\_id)

references carte\_p(carte\_id) on delete cascade

);

insert into vinde\_p

values(1001, 1001, str\_to\_date('14.01.2022', '%d.%m.%Y'));

insert into vinde\_p

values(1001, 1002, str\_to\_date('13.01.2022', '%d.%m.%Y'));

insert into vinde\_p

values(1002, 1003, null);

insert into vinde\_p

values(1002, 1001, null);

insert into vinde\_p

values(1002, 1002, str\_to\_date('12.01.2021', '%d.%m.%Y'));

insert into vinde\_p

values(1003, 1003, str\_to\_date('01.09.2021', '%d.%m.%Y'));

insert into vinde\_p

values(1004, 1004, str\_to\_date('01.06.2020', '%d.%m.%Y'));

create table scrie\_p (

autor\_id int,

carte\_id int,

constraint pk\_scrie\_p primary key(autor\_id, carte\_id),

constraint autor\_id\_scrie\_fk\_p foreign key(autor\_id)

references autor\_p(autor\_id) on delete cascade,

constraint carte\_id\_scrie\_fk\_p foreign key(carte\_id)

references carte\_p(carte\_id) on delete cascade

);

insert into scrie\_p

values(1001, 1001);

insert into scrie\_p

values(1001, 1002);

insert into scrie\_p

values(1002, 1001);

insert into scrie\_p

values(1002, 1002);

insert into scrie\_p

values(1003, 1003);

insert into scrie\_p

values(1003, 1001);

insert into scrie\_p

values(1004, 1004);

insert into scrie\_p

values(1005, 1005);