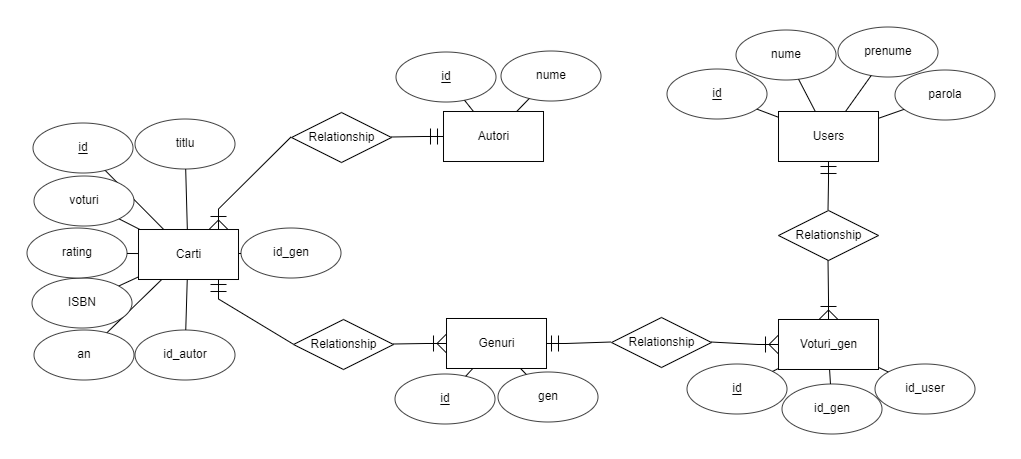
**BookMate**

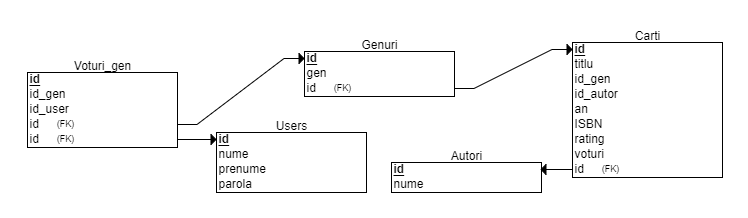
**Despre aplicație**

O platformă unde utilizatorii pot căuta, descoperi și descărca cărți. Fiecare carte va avea un autor și un gen principal în care se încadrează, iar fiecare gen va avea un anumit număr de voturi primite de la diverși utilizatori.

Utilizatorii pot oferi un rating cărților și pot primi recomandări pe baza unui algoritm care analizează preferințele și istoricul utilizatorului, cât și utilizatori cu gusturi similare.

**EntityRelationship Schema**

****

****

Diagramă realizată utilizând <https://erdplus.com/>.

**Forma normală a bazei de date**

Aceasta este cel puțin BCNF deoarece pentru fiecare dependență funcțională X -> Y, X este super-cheie.

Mai mult de atât, este 4NF deoarece nu are dependențe multivaluate care ar duce la repetarea datelor. Toate datele sunt modularizate și împărțite în mai multe tabele pentru a evita redundanța.

**Script creare și populare**

DROP TABLE carti CASCADE CONSTRAINTS

/

DROP TABLE genuri CASCADE CONSTRAINTS

/

DROP TABLE voturi\_gen CASCADE CONSTRAINTS

/

DROP TABLE autori CASCADE CONSTRAINTS

/

DROP TABLE users CASCADE CONSTRAINTS

/

CREATE TABLE carti (

id INT NOT NULL PRIMARY KEY,

titlu VARCHAR2(126) NOT NULL,

id\_gen INT NOT NULL,

id\_autor INT NOT NULL,

an INT NOT NULL,

ISBN INT NOT NULL,

rating FLOAT NOT NULL,

voturi INT NOT NULL

)

/

CREATE TABLE genuri (

id INT NOT NULL PRIMARY KEY,

gen VARCHAR2(126) NOT NULL

)

/

CREATE TABLE voturi\_gen (

id INT NOT NULL PRIMARY KEY,

id\_gen INT NOT NULL,

id\_user INT NOT NULL

)

/

CREATE TABLE autori (

id INT NOT NULL PRIMARY KEY,

nume VARCHAR2(126) NOT NULL

)

/

CREATE TABLE users (

id INT NOT NULL PRIMARY KEY,

nume VARCHAR2(126) NOT NULL,

prenume VARCHAR2(126) NOT NULL,

parola VARCHAR2(126) NOT NULL

)

/

--cod inspirat din scriptul initial de creare a tabelelor pentru studenti

SET SERVEROUTPUT ON;

DECLARE

TYPE varr IS VARRAY(1000) OF varchar2(255);

lista\_nume varr := varr('Ababei','Acasandrei','Adascalitei','Afanasie','Agafitei','Agape','Aioanei','Alexandrescu','Alexandru','Alexe','Alexii','Amarghioalei','Ambroci','Andonesei','Andrei','Andrian','Andrici','Andronic','Andros','Anghelina','Anita','Antochi','Antonie','Apetrei','Apostol','Arhip','Arhire','Arteni','Arvinte','Asaftei','Asofiei','Aungurenci','Avadanei','Avram','Babei','Baciu','Baetu','Balan','Balica','Banu','Barbieru','Barzu','Bazgan','Bejan','Bejenaru','Belcescu','Belciuganu','Benchea','Bilan','Birsanu','Bivol','Bizu','Boca','Bodnar','Boistean','Borcan','Bordeianu','Botezatu','Bradea','Braescu','Budaca','Bulai','Bulbuc-aioanei','Burlacu','Burloiu','Bursuc','Butacu','Bute','Buza','Calancea','Calinescu','Capusneanu','Caraiman','Carbune','Carp','Catana','Catiru','Catonoiu','Cazacu','Cazamir','Cebere','Cehan','Cernescu','Chelaru','Chelmu','Chelmus','Chibici','Chicos','Chilaboc','Chile','Chiriac','Chirila','Chistol','Chitic','Chmilevski','Cimpoesu','Ciobanu','Ciobotaru','Ciocoiu','Ciofu','Ciornei','Citea','Ciucanu','Clatinici','Clim','Cobuz','Coca','Cojocariu','Cojocaru','Condurache','Corciu','Corduneanu','Corfu','Corneanu','Corodescu','Coseru','Cosnita','Costan','Covatariu','Cozma','Cozmiuc','Craciunas','Crainiceanu','Creanga','Cretu','Cristea','Crucerescu','Cumpata','Curca','Cusmuliuc','Damian','Damoc','Daneliuc','Daniel','Danila','Darie','Dascalescu','Dascalu','Diaconu','Dima','Dimache','Dinu','Dobos','Dochitei','Dochitoiu','Dodan','Dogaru','Domnaru','Dorneanu','Dragan','Dragoman','Dragomir','Dragomirescu','Duceac','Dudau','Durnea','Edu','Eduard','Eusebiu','Fedeles','Ferestraoaru','Filibiu','Filimon','Filip','Florescu','Folvaiter','Frumosu','Frunza','Galatanu','Gavrilita','Gavriliuc','Gavrilovici','Gherase','Gherca','Ghergu','Gherman','Ghibirdic','Giosanu','Gitlan','Giurgila','Glodeanu','Goldan','Gorgan','Grama','Grigore','Grigoriu','Grosu','Grozavu','Gurau','Haba','Harabula','Hardon','Harpa','Herdes','Herscovici','Hociung','Hodoreanu','Hostiuc','Huma','Hutanu','Huzum','Iacob','Iacobuta','Iancu','Ichim','Iftimesei','Ilie','Insuratelu','Ionesei','Ionesi','Ionita','Iordache','Iordache-tiroiu','Iordan','Iosub','Iovu','Irimia','Ivascu','Jecu','Jitariuc','Jitca','Joldescu','Juravle','Larion','Lates','Latu','Lazar','Leleu','Leon','Leonte','Leuciuc','Leustean','Luca','Lucaci','Lucasi','Luncasu','Lungeanu','Lungu','Lupascu','Lupu','Macariu','Macoveschi','Maftei','Maganu','Mangalagiu','Manolache','Manole','Marcu','Marinov','Martinas','Marton','Mataca','Matcovici','Matei','Maties','Matrana','Maxim','Mazareanu','Mazilu','Mazur','Melniciuc-puica','Micu','Mihaela','Mihai','Mihaila','Mihailescu','Mihalachi','Mihalcea','Mihociu','Milut','Minea','Minghel','Minuti','Miron','Mitan','Moisa','Moniry-abyaneh','Morarescu','Morosanu','Moscu','Motrescu','Motroi','Munteanu','Murarasu','Musca','Mutescu','Nastaca','Nechita','Neghina','Negrus','Negruser','Negrutu','Nemtoc','Netedu','Nica','Nicu','Oana','Olanuta','Olarasu','Olariu','Olaru','Onu','Opariuc','Oprea','Ostafe','Otrocol','Palihovici','Pantiru','Pantiruc','Paparuz','Pascaru','Patachi','Patras','Patriche','Perciun','Perju','Petcu','Pila','Pintilie','Piriu','Platon','Plugariu','Podaru','Poenariu','Pojar','Popa','Popescu','Popovici','Poputoaia','Postolache','Predoaia','Prisecaru','Procop','Prodan','Puiu','Purice','Rachieru','Razvan','Reut','Riscanu','Riza','Robu','Roman','Romanescu','Romaniuc','Rosca','Rusu','Samson','Sandu','Sandulache','Sava','Savescu','Schifirnet','Scortanu','Scurtu','Sfarghiu','Silitra','Simiganoschi','Simion','Simionescu','Simionesei','Simon','Sitaru','Sleghel','Sofian','Soficu','Sparhat','Spiridon','Stan','Stavarache','Stefan','Stefanita','Stingaciu','Stiufliuc','Stoian','Stoica','Stoleru','Stolniceanu','Stolnicu','Strainu','Strimtu','Suhani','Tabusca','Talif','Tanasa','Teclici','Teodorescu','Tesu','Tifrea','Timofte','Tincu','Tirpescu','Toader','Tofan','Toma','Toncu','Trifan','Tudosa','Tudose','Tuduri','Tuiu','Turcu','Ulinici','Unghianu','Ungureanu','Ursache','Ursachi','Urse','Ursu','Varlan','Varteniuc','Varvaroi','Vasilache','Vasiliu','Ventaniuc','Vicol','Vidru','Vinatoru','Vlad','Voaides','Vrabie','Vulpescu','Zamosteanu','Zazuleac');

lista\_prenume\_fete varr := varr('Adina','Alexandra','Alina','Ana','Anca','Anda','Andra','Andreea','Andreia','Antonia','Bianca','Camelia','Claudia','Codrina','Cristina','Daniela','Daria','Delia','Denisa','Diana','Ecaterina','Elena','Eleonora','Elisa','Ema','Emanuela','Emma','Gabriela','Georgiana','Ileana','Ilona','Ioana','Iolanda','Irina','Iulia','Iuliana','Larisa','Laura','Loredana','Madalina','Malina','Manuela','Maria','Mihaela','Mirela','Monica','Oana','Paula','Petruta','Raluca','Sabina','Sanziana','Simina','Simona','Stefana','Stefania','Tamara','Teodora','Theodora','Vasilica','Xena');

lista\_prenume\_baieti varr := varr('Adrian','Alex','Alexandru','Alin','Andreas','Andrei','Aurelian','Beniamin','Bogdan','Camil','Catalin','Cezar','Ciprian','Claudiu','Codrin','Constantin','Corneliu','Cosmin','Costel','Cristian','Damian','Dan','Daniel','Danut','Darius','Denise','Dimitrie','Dorian','Dorin','Dragos','Dumitru','Eduard','Elvis','Emil','Ervin','Eugen','Eusebiu','Fabian','Filip','Florian','Florin','Gabriel','George','Gheorghe','Giani','Giulio','Iaroslav','Ilie','Ioan','Ion','Ionel','Ionut','Iosif','Irinel','Iulian','Iustin','Laurentiu','Liviu','Lucian','Marian','Marius','Matei','Mihai','Mihail','Nicolae','Nicu','Nicusor','Octavian','Ovidiu','Paul','Petru','Petrut','Radu','Rares','Razvan','Richard','Robert','Roland','Rolland','Romanescu','Sabin','Samuel','Sebastian','Sergiu','Silviu','Stefan','Teodor','Teofil','Theodor','Tudor','Vadim','Valentin','Valeriu','Vasile','Victor','Vlad','Vladimir','Vladut');

lista\_titluri varr := varr ('Maitreyi','O mie noua sute optzeci si patru','Batranul si marea','Portretul lui Dorian Gray','Don Quijote de la Mancha','Ferma animalelor','Lolita','Micul print','Puterea prezentului','Inteligenta emotionala','Pe culmile disperarii','Tropicul capricornului','Strainul','Procesul','Alchimistul','Amintiri din pribegie','Harry Potter');

lista\_genuri varr := varr ('autobiografie','romanta','nuvela','fictiune','satira','dezvoltare','filozofie','fantezie');

lista\_autori varr := varr ('Mircea Eliade','George Orwell','Ernest Hemingway','Miguel de Cervantes','Vladimir Nabokov','Antoine de Saint-Exupery','Eckhart Tolle','Daniel Goleman','Emil Cioran','Henry Miller','Albert Camus','Paulo Coelho','Neagu Djuvara','J. K. Rowling');

v\_nume VARCHAR2(255);

v\_prenume VARCHAR2(255);

v\_prenume1 VARCHAR2(255);

v\_prenume2 VARCHAR2(255);

v\_titlu VARCHAR2(255);

v\_id\_gen INT;

v\_id\_autor INT;

v\_an NUMBER(4, 0);

v\_isbn NUMBER(13, 0);

v\_rating FLOAT;

v\_voturi INT;

v\_id\_user INT;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Inserarea a 100.000 useri.');

FOR v\_i IN 1..100000 LOOP

v\_nume := lista\_nume(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_nume.count))+1);

IF (DBMS\_RANDOM.VALUE(0,100)<50) THEN

v\_prenume1 := lista\_prenume\_fete(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_prenume\_fete.count))+1);

LOOP

v\_prenume2 := lista\_prenume\_fete(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_prenume\_fete.count))+1);

exit when v\_prenume1<>v\_prenume2;

END LOOP;

ELSE

v\_prenume1 := lista\_prenume\_baieti(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_prenume\_baieti.count))+1);

LOOP

v\_prenume2 := lista\_prenume\_baieti(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_prenume\_baieti.count))+1);

exit when v\_prenume1<>v\_prenume2;

END LOOP;

END IF;

IF (DBMS\_RANDOM.VALUE(0,100)<60) THEN

IF LENGTH(v\_prenume1 || ' ' || v\_prenume2) <= 20 THEN

v\_prenume := v\_prenume1 || ' ' || v\_prenume2;

END IF;

else

v\_prenume:=v\_prenume1;

END IF;

INSERT INTO users VALUES (v\_i, v\_nume, v\_prenume, 'hash');

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Inserarea a 100.000 carti.');

FOR v\_i IN 1..100000 LOOP

v\_titlu := lista\_titluri(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_titluri.count))+1);

v\_id\_gen := TRUNC(DBMS\_RANDOM.VALUE(0,lista\_genuri.count))+1;

v\_id\_autor := TRUNC(DBMS\_RANDOM.VALUE(0,lista\_autori.count))+1;

v\_an := TRUNC(DBMS\_RANDOM.VALUE(1200, 2019));

v\_isbn := TRUNC(DBMS\_RANDOM.VALUE(1000000000000, 9999999999999));

v\_rating := DBMS\_RANDOM.VALUE(1, 5);

v\_voturi := TRUNC(DBMS\_RANDOM.VALUE(500, 10000));

INSERT INTO carti VALUES (v\_i, v\_titlu, v\_id\_gen, v\_id\_autor, v\_an, v\_isbn, v\_rating, v\_voturi);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Simularea a 1.000.000 voturi de callback.');

FOR v\_i IN 1..1000000 LOOP

v\_titlu := lista\_titluri(TRUNC(DBMS\_RANDOM.VALUE(0,lista\_titluri.count))+1);

v\_id\_gen := TRUNC(DBMS\_RANDOM.VALUE(0,lista\_genuri.count))+1;

v\_id\_user := TRUNC(DBMS\_RANDOM.VALUE(1,100000))+1;

INSERT INTO voturi\_gen VALUES (v\_i, v\_id\_gen, v\_id\_user);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Inserarea genurilor.');

FOR v\_i IN 1..8 LOOP

INSERT INTO genuri VALUES (v\_i, lista\_genuri(v\_i));

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Inserarea autorilor.');

FOR v\_i IN 1..14 LOOP

INSERT INTO autori VALUES (v\_i, lista\_autori(v\_i));

END LOOP;

END;

**Descrierea structurilor de date**

* tabela ‘Cărți’ - conține informațiile asociate tuturor cărților
  + id - cheie de indexare unică în tabel
  + titlu - titlul cărții sub formă de string - varchar2
  + id\_autor - cheie străină către tabela ‘Autori’
  + id\_gen - cheie străină către tabela ‘Genuri’
  + an - anul apariției - int
  + ISBN - șir unic de referință - poate fi folosit ca și hash pentru numele fișierului - va identifica în mod unic pdf-ul cărții în sistemul de fișiere - int
  + rating - media voturilor acordate cărții respective - float
  + voturi - numărul de voturi acordate, pentru a putea actualiza corect matematic rating-ul - int
* tabela ‘Autori’
  + id - cheie de indexare unică în tabel (referențiată în tabela ‘Cărți’)
  + nume - numele autorului - varchar2
* tabela ‘Genuri’
  + id - cheie de indexare unică în tabel (referențiată în tabela ‘Cărți’ și în tabela ‘Voturi\_gen’
  + gen - genul abordat în conținutul cărții - varchar2
* tabela ‘Users’
  + id - cheie de indexare unică în tabel (referențiată în tabela ‘Voturi\_gen’)
  + nume - numele utilizatorului - varchar2
  + prenume - prenumele utilizatorului - varchar2
  + parola - în acest câmp va fi stocat doar un hash al parolei, care va împiedica pe oricine (din sistem sau din afara lui) să poată decripta parola utilizatorului - verificarea făcându-se doar comparând output-ul funcției de hash cu intrarea din această tabelă, atunci când utilizatorul se autentifică.
* tabela ‘Voturi\_gen’
  + id - cheie de indexare unică în tabel
  + id\_gen - cheie străină către tabela ‘Genuri’
  + id\_user - cheie străină către tabela ‘Users’

**Descrierea funcțiilor si procedurilor**

* GetBrothers (user\_id u) - va interoga și corela datele din tabele pentru a afla care utilizatori au preferințele cele mai asemănătoare cu userul dat
* GetTopBooks (id\_gen g) - va întoarce lista cu cele mai bune cărți în ordinea rating-ului pentru un anumit gen
* VisitBook (id\_carte c, user\_id u) - funcție de callback care se va apela mereu când un utilizator interacționează cu o anumită carte, pentru a contoriza interesul său față de atributele ei
* GetBookLink (id\_carte c) - va calcula un hash pe ISBN-ul cărții pentru a întoarce calea de pe disk către pdf-ul asociat
* CheckPassword (user\_id u, varchar2 password) - va întoarce true sau false dacă parola primită ca parametru (în plaintext) corespunde hash-ului parolei utilizatorului dat ca parametru
* GetRecommendations (user\_id u) - va întoarce o listă cu noi cărți (pe care utilizatorul nu le-a mai întâlnit până acum), folosind și prima funcție din acest paragraf - GetBrothers. Pentru fiecare carte se poate calcula un coeficient de interes - bazat pe numărul de voturi, valoarea rating-ului, valoarea rating-ului raportat la numărul de voturi, popularitatea genului ei printre cei mai activi utilizatori, numărul total de vizualizări raportat la anul apariției, experiența autorului ei, rating-ul altor cărți scrise de același autor, numărul de accesări într-un interval setat de timp, numărul de apariții în istoricul ultimelor 100 cărți vizitate global.