

Rețea de socializare pentru cititori

Realizat de Buzatu Giulian

Grupa 152

Cuprins

| | |
|---|----|
| Utilitatea modelului în viața reală | 2 |
| Constrângeri impuse asupra modelului..... | 2 |
| Descrierea entităților | 3 |
| Descrierea relațiilor | 4 |
| Descrierea atributelor entităților | 6 |
| Diagrama Entitate-Relație | 12 |
| Diagrama Conceptuală..... | 13 |
| Scheme relaționale | 13 |
| corespunzătoare diagramei conceptuale | 13 |
| Realizarea normalizării până la forma normală 3..... | 14 |
| Forma normală 1..... | 14 |
| Forma normală 2..... | 15 |
| Forma normală 3..... | 16 |
| Crearea de secvențe pentru popularea înregistrărilor în tabele..... | 18 |
| Crearea tabelor și inserarea datelor | 22 |
| Cereri SQL complexe..... | 45 |
| Operații de actualizare și de ștergere | 49 |
| Cereri SQL outer-join, operația division și analiza top-n | 53 |
| Optimizarea unei cereri | 55 |
| Cererea neoptimizată | 56 |
| Cererea optimizată | 57 |
| Realizarea normalizării până la forma normală 5..... | 59 |
| BCNF..... | 59 |
| FN4..... | 59 |
| FN5 | 59 |
| Denormalizarea..... | 59 |

Utilitatea modelului în viața reală

În lumea contemporană, oamenii se conectează unii cu ceilalți, în cea mai mare parte, prin intermediul Internetului. Mulți dintre aceștia caută să întâlnească persoane noi, cu preocupări și gusturi comune cu ale lor. De aceea, o rețea de socializare bazată pe lectură, care implică profiluri de utilizator, posibilitatea de a lăsa câte o recenzie pentru fiecare carte citită, aflarea mai multor informații despre o anumită carte, precum editura, autorul, editorul și traducătorul cărții. Pe lângă recenziile pe care le poate lăsa utilizatorul cărților pe care le-a citit, acesta poate face o postare în cadrul aplicației, în care să își exprime sentimentele despre lectură, să realizeze clasamente cu cărțile sale preferate, etc. Utilizatorii pot crea grupuri cu tematică specifică, în care să socializeze și să își împărtășească părerile despre cărțile pe care le-au citit sau pe care vor să le citească, autorii preferați și alte subiecte referitoare la lectură, dar nu numai. Astfel, această rețea de socializare ajută oamenii cărora le place să citească să se cunoască și să dezvolte prietenii pe care le-ar fi fost greu să le creeze altfel în mediul online.

Constrângeri impuse asupra modelului

- Fiecare carte aparține în una sau mai multe categorii, iar fiecare categorie conține una sau mai multe cărți.
- Fiecare editură a publicat una sau mai multe cărți. Fiecare carte este publicată de una sau mai multe edituri.
- Fiecare traducător a tradus una sau mai multe cărți, iar fiecare carte are cel mult un singur traducător.
- O carte are unul sau mai mulți autori, iar un autor a scris una sau mai multe cărți.
- Un editor a editat una sau mai multe cărți, iar o carte are unul sau mai mulți editori.
- Un autor colaborează cu unul sau mai mulți editori, iar un editor poate lucra cu unul sau mai mulți autori.
- Fiecare recenzie este specifică unei cărți, dar o carte poate avea zero sau mai multe recenzii.
- O carte poate fi citită de zero sau mai mulți utilizatori, iar un utilizator a citit zero sau mai multe cărți.
- Un utilizator poate fi în zero sau mai multe grupuri, iar un grup conține cel puțin un utilizator.
- O postare este caracteristică unui utilizator, iar acesta poate realiza una sau mai multe postări.

- În cadrul rețelei de socializare, există mai multe tipuri de postări, mai precis, cele normale, pe care utilizatorul le poate face pe pagina sa, și recenziile, care sunt specifice și cărților.
- Fiecare postare poate avea zero sau mai multe comentarii, dar fiecare comentariu este particular pentru o postare, cum o recenzie este un fel de postare, același lucru se aplică și pentru ea.

Descrierea entităților

Pentru modelul de date referitor la rețeaua de socializare pentru cititori, structurile CATEGORIE, AUTOR, TRADUCATOR, AUTOR, EDITOR, CARTE, UTILIZATOR, GRUP, POSTARE, COMENTARIU reprezintă entități, iar structura RECENZIE este o subentitate a entității POSTARE.

Toate entitățile care vor fi prezentate sunt independente, cu excepția entităților dependente POSTARE și COMENTARIU și a subentității RECENZIE.

CATEGORIE = colecție de cărți cu aceeași tematică, care ajută la realizarea de recomandări către utilizator, deoarece se pot recomanda cărți care se află în aceleași categorii cu cărțile deja citite de utilizator. Cheia primară a acestei entități este *id_categorie*.

EDITURA = companie care se ocupă cu publicarea cărților, care reține data înființării editurii și denumirea acesteia. Cheia primară a acestei entități este *id_editura*.

TRADUCĂTOR = persoană care traduce cărțile dintr-o limbă în limba română. Cheia primară a acestei entități este *id_traducator*.

AUTOR = persoană care scrie cărți în mai multe limbi, care lucrează cu mai mulți editori pentru îmbunătățirea lecturilor pe care le-a conceput. Cheia primară a acestei entități este *id_autor*.

EDITOR = persoană care ajută la îmbunătățirea conținutului unei cărți redactate de către autorul, sau autorii, acesteia. Cheia primară a acestei entități este *id_editor*.

CARTE = scriere cu un anumit subiect, tipărită și legată sau broșată în volum, care are unul sau mai mulți autori, publicată de o editură și care se încadrează în una sau mai multe categorii. Cheia primară a acestei entități este *id_carte*.

UTILIZATOR = persoana care își creează un cont, după care se poate adăuga grupurilor care îi atrag interesul, să scrie postări sau să lase comentariile la postările altor utilizatori. Cheia primară a acestei entități este *id_utilizator*.

GRUP = grup alcătuit din una sau mai multe persoane, care are ca scop reuniunea mai multor oameni, într-un context online, pentru a discuta diverse subiecte legate de cărți. Cheia primară a acestei entități este *id_grup*.

POSTARE = modalitate de exprimare a unei păreri sau de publicare a unui text personal în cadrul rețelei de socializare. Deoarece este o entitate dependentă de entitatea UTILIZATOR, cheia primară a acestei entități este compusă din *id_utilizator* și *id_postare*.

RECENZIE = un tip special de postare, care poate fi realizată doar pentru o carte existentă în baza de date. Aceasta este o subentitate a entității POSTARE și este dependentă și de entitatea CARTE, deci cheia primară a acestei entități este compusă din *id_utilizator*, *id_postare* și *id_carte*.

COMENTARIU = modalitate de exprimare a unei păreri în cadrul unei postări realizate de un alt utilizator, sau chiar de sine însuși, pentru a continua subiectul introdus de postare. Această entitate este dependentă de entitatea POSTARE, prin urmare, cheia primară a acestei entități este compusă din *id_utilizator*, *id_postare* și *id_comentariu*.

Descrierea relațiilor

TRADUCATOR_traduce_CARTE = relație dintre entitățile TRADUCATOR și CARTE, reprezentând cărțile pe care un traducător le transpune din limba în care este scrisă cartea într-o nouă limbă. Relația are cardinalitate minimă 0:1 (un traducător traduce cel puțin o carte, dar o carte poate să nu aibă niciun traducător) și cardinalitate maximă 1:m (un traducător poate traduce mai multe cărți, iar o carte are cel mult un traducător).

EDITURA_publica_CARTE = relație dintre entitățile EDITURA și CARTE, reprezentând cărțile pe care o editură le tipărește pentru publicul larg. Relația are cardinalitate minimă 1:1 (o editură publică cel puțin o carte, iar o carte este publicată de cel puțin o editură) și cardinalitate maximă m:n (o editură poate publica mai multe cărți, iar o carte poate apărea la mai multe edituri pe parcursul timpului).

CARTE_apartine_CATEGORIE = relație dintre entitățile CARTE și CATEGORIE, reprezentând categoriile din care face parte o carte. Relația are cardinalitate minimă 0:1 (o categorie poate să nu conțină nicio carte, iar o carte

aparține cel puțin unei categorii) și cardinalitate maximă $m:n$ (o categorie poate conține mai multe cărți, iar o carte poate face parte din mai multe categorii).

CARTE_are_RECENZIE = relație dintre entitatea CARTE și subentitatea RECENZIE, reprezentând recenziile pe care le are o carte. Relația are cardinalitate minimă $1:0$ (o carte poate să nu aibă nicio recenzie, dar o recenzie nu poate exista fără o carte pentru care să fie scrisă) și cardinalitate maximă $1:m$ (o recenzie este scrisă pentru o singură carte, dar o carte poate avea mai multe recenzii).

UTILIZATOR_citeste_CARTE = relație dintre entitățile UTILIZATOR și CARTE, reprezentând cărțile pe care un utilizator le-a citit. Relația are cardinalitate minimă $0:0$ (o carte poate să nu aibă niciun cititor, iar un utilizator poate să nu fi citit nicio carte până acum) și cardinalitate maximă $m:n$ (o carte poate fi citită de mai mulți utilizatori, iar un utilizator poate citi mai multe cărți).

UTILIZATOR_apartine_GRUP = relație dintre entitățile UTILIZATOR și GRUP, reprezentând grupurile în care este înscris un utilizator. Relația are cardinalitate minimă $1:0$ (un utilizator poate să nu aparțină niciunui grup, dar fiecare grup are cel puțin un utilizator) și cardinalitate maximă $m:n$ (un utilizator poate aparține mai multor grupuri, iar un grup poate avea mai mulți utilizatori).

UTILIZATOR_posteaza_POSTARE = relație dintre entitățile UTILIZATOR și POSTARE, reprezentând postările pe care un utilizator le face pe rețeaua de socializare. Relația are cardinalitate minimă $1:0$ (o postare este realizată neapărat de un utilizator, dar un utilizator poate să nu realizeze nicio postare) și cardinalitate maximă $1:m$ (o postare este realizată de exact un utilizator, dar un utilizator poate să facă mai multe postări).

POSTARE_are_COMENTARIU = relație dintre entitățile POSTARE și COMENTARIU, reprezentând comentariile pe care o postare le are. Relația are cardinalitate minimă $1:0$ (un comentariu este dependent de o postare, deci nu poate exista fără aceasta, dar o postare poate să nu aibă niciun comentariu) și cardinalitatea maximă $1:m$ (un comentariu necesită o postare ca să existe, iar o postare poate să aibă mai multe comentarii).

POSTARE_ISA_RECENZIE = relație specială între supraentitatea POSTARE și subentitatea RECENZIE, reprezentând dacă o postare este sau nu o recenzie. Relația are cardinalitate minimă $1:0$ (o postare poate să nu fie o recenzie, dar o recenzie este neapărat o postare) și cardinalitate maximă $1:1$ (o postare este o recenzie, iar o recenzie trebuie să fie o postare).

AUTOR_lucreaza_cu_EDITOR_la_mai_multe_CARTE = relație de tip 3, între entitățile AUTOR, EDITOR și CARTE, reprezentând cu ce editor a lucrat autorul

în cadrul unei cărți scrise de el. Cardinalitatea minimă este 0:0:0 și cardinalitatea maximă este m:n:p.

Descrierea atributelor entităților

Entitatea independentă CATEGORIE are ca attribute:

id_categorie = variabilă de tip întreg, de lungime maximă 3, NUMBER(3), care reprezintă id-ul unei categorii. Cheia primară a entității.

nume = variabila de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele categoriei. Nu poate fi null.

Entitatea independentă EDITURA are ca attribute:

id_editura = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unei edituri. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele editurii. Nu poate fi null.

data_infiintare = variabilă de tip dată calendaristică, DATE, care reprezintă data înființării editurii respective. Nu poate fi null.

Entitatea independentă TRADUCATOR are ca attribute:

id_traducator = variabilă de tip întreg, de lungime maximă 9, NUMBER(3), care reprezintă id-ul unui traducător. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele traducătorului. Nu poate fi null.

prenume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă prenumele traducătorului. Nu poate fi null.

Entitatea independentă AUTOR are ca attribute:

id_autor = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui autor. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele autorului. Nu poate fi null.

prenume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă prenumele autorului. Nu poate fi null.

data_debut = variabilă de tip dată calendaristică, DATE, care reprezintă data debutului autorului. Nu poate fi null.

Entitatea independentă EDITOR are ca attribute:

id_editor = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui editor. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele editorului. Nu poate fi null.

prenume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă prenumele editorului. Nu poate fi null.

data_debut = variabilă de tip dată calendaristică, DATE, care reprezintă data la care ediorul și-a început cariera. Nu poate fi null.

Entitatea independentă CARTE are ca attribute:

id_carte = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unei cărți. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 50, VARCHAR2(50), care reprezintă numele cărții. Nu poate fi null.

nr_capitole = variabilă de tip întreg, de lungime maximă 3, NUMBER(3), care reprezintă numărul de capitole al cărții. Nu poate fi null.

nr_pagini = variabilă de tip întreg, de lungime maximă 4, NUMBER(4), care reprezintă numărul de pagini al cărții. Nu poate fi null.

isbn = variabilă de tip caracter, de lungime maximă 13, VARCHAR2(13), număr de identificare internațional atribuit fiecărei lucrări publicate. Coloana poate avea fie valori de lungime 10, fie valori de lungime 13. Nu poate fi null.

cost_productie = variabilă de tip întreg, de lungime maximă 2, NUMBER(2), care reprezintă costul, în euro, necesar pentru producerea unui exemplar al cărții. Poate fi null.

pret = variabilă de tip întreg, de lungime maximă 3, NUMBER(3), care reprezintă prețul, în euro, al unui exemplar al cărții. Nu poate fi null.

limba = variabilă de tip caracter, de lungime maximă 15, VARCHAR2(15), care reprezintă limba în care este scrisă sau tradusă cartea. Nu poate fi null.

id_traducator = variabilă de tip întreg, de lungime maximă 3, NUMBER(3), care reprezintă id-ul traducătorului cărții. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TRADUCATOR. Poate fi null.

Entitatea independentă UTILIZATOR are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui utilizator. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele utilizatorului. Nu poate fi null.

prenume = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă prenumele utilizatorului. Nu poate fi null.

nume_cont = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele contului utilizatorului. Nu poate fi null. Atributul trebuie să fie unic.

parolă = variabilă de tip caracter, de lungime maximă 25, VARCHAR2(25), care reprezintă numele contului utilizatorului. Nu poate fi null.

data_nașterii = variabilă de tip dată calendaristică, DATE, care reprezintă data nașterii utilizatorului. Poate fi null.

email = variabilă de tip caracter, de lungime maximă 50, VARCHAR2(50), care reprezintă adresa de email a utilizatorului. Nu poate fi null.

nr_telefon = variabilă de tip caracter, de lungime maximă 15, VARCHAR2(15), care reprezintă numărul de telefon al utilizatorului. Poate fi null.

Entitatea independentă GRUP are ca attribute:

id_grup = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unui grup. Cheia primară a entității.

nume = variabilă de tip caracter, de lungime maximă 50, VARCHAR2(50), care reprezintă numele grupului. Nu poate fi null.

nr_membrii = variabilă de tip întreg, de lungime maximă 7, NUMBER(7), care reprezintă numărul de membrii din grup. Nu poate fi null.

Entitatea dependentă POSTARE are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul utilizatorului care a scris postarea Atributul trebuie să corespundă la o valoare a cheii primare din tabelul UTILIZATOR. Parte din cheia primară a entității.

id_postare = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unei postări. Parte din cheia primară a entității.

continut = variabilă de tip caracter, de lungime maximă 4000, VARCHAR2(4000), care reprezintă conținutul postării. Nu poate fi null.

nr_aprecieri = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă numărul de aprecieri ale postării. Nu poate fi null.

nr_comentarii = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă numărul de comentarii ale postării. Care va avea valoarea implicită 0.

tip_postare = variabilă de tip caracter, de lungime maximă 8, VARCHAR2(8), care reprezintă tipul postării. Acest atribut poate avea fie valoarea „POSTARE”, fie valoarea „RECENZIE”, în funcție de tipul de postare. Nu poate fi null.

Subentitatea RECENZIE are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul utilizatorului care a scris postarea Perechea (id_utilizator, id_postare) trebuie să corespundă la o valoare a cheii primare compuse din tabelul POSTARE. Parte din cheia primară a subentității.

id_postare = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unei postări. Perechea (id_utilizator, id_postare) trebuie să corespundă la o valoare a cheii primare compuse din tabelul POSTARE. Parte din cheia primară a subentității.

id_carte = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unei cărți. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CARTE. Parte din cheia primară a subentității.

nr_stele = variabilă de tip real, de lungime maximă 1, cu o singură zecimală, NUMBER(1,1), care reprezintă numărul de stele pe care utilizatorul îl acordă cărții. Acesta poate fi un număr întreg de la 1 la 5, sau un număr real din mulțimea {0.5, 1.5, 2.5, 3.5, 4.5}. Nu poate fi null.

Entitatea dependentă COMENTARIU are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul utilizatorului care a scris postarea. Perechea (id_utilizator, id_postare) trebuie să corespundă la o valoare a cheii primare compuse din tabelul POSTARE. Parte din cheia primară a entității.

id_postare = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unei postări. Perechea (id_utilizator, id_postare) trebuie să corespundă la o valoare a cheii primare compuse din tabelul POSTARE. Parte din cheia primară a entității.

id_comentariu = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul comentariului. Parte din cheia primară a entității.

continut = variabilă de tip caracter, de lungime maximă 4000, VARCHAR2(4000), care reprezintă conținutul comentariului. Nu poate fi null.

id_utilizator_comentariu = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul utilizatorului care scrie comentariul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul UTILIZATOR.

Relația CARTE_apartine_CATEGORIE are ca attribute:

id_carte = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unei cărți. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CARTE. Parte din cheia primară a entității.

id_categorie = variabilă de tip întreg, de lungime maximă 3, NUMBER(3), care reprezintă id-ul unei categorii. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CATEGORIE. Parte din cheia primară a entității.

Relația EDITURA_publica_CARTE are ca attribute:

id_editura = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unei edituri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul EDITURA. Parte din cheia primară a entității.

id_carte = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unei cărți. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CARTE. Parte din cheia primară a entității.

nr_exemplare = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă numărul de exemplare ale cărții respective pe care le-a publicat editura. Poate fi null.

Relația UTILIZATOR_citeste_CARTE are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul UTILIZATOR. Parte din cheia primară a entității.

id_carte = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unei cărți. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CARTE. Parte din cheia primară a entității.

data_inceput = variabilă de tip dată calendaristică, DATE, care reprezintă data la care utilizatorul a început să citească cartea respectivă. Nu poate fi null.

data_sfarsit = variabilă de tip dată calendaristică, DATE, care reprezintă data la care utilizatorul a terminat lectura cărții respective. Nu poate fi null. Trebuie să fie mai mare decât data_inceput.

Relația UTILIZATOR_apartine_GRUP are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul UTILIZATOR. Parte din cheia primară a entității.

id_grup = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unui grup. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul GRUP. Parte din cheia primară a entității.

data_intrare = variabilă de tip dată calendaristică, DATE, care reprezintă data la care utilizatorul a intrat în grupul respectiv. Nu poate fi null.

Relația AUTOR_lucreaza_cu_EDITOR_si_scrie_mai_multe_CARTE are ca attribute:

id_autor = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui autor. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul AUTOR. Parte din cheia primară a entității.

id_editor = variabilă de tip întreg, de lungime maximă 9, NUMBER(9), care reprezintă id-ul unui editor. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul EDITOR. Parte din cheia primară a entității.

id_carte = variabilă de tip întreg, de lungime maximă 18, NUMBER(18), care reprezintă id-ul unei cărți. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CARTE. Parte din cheia primară a entității.

pret = variabilă de tip întreg, de lungime maximă 6, NUMBER(6), care reprezintă pretul, în euro, editării cărții de către acest editor. Poate fi null.

Diagrama Entitate-Relație

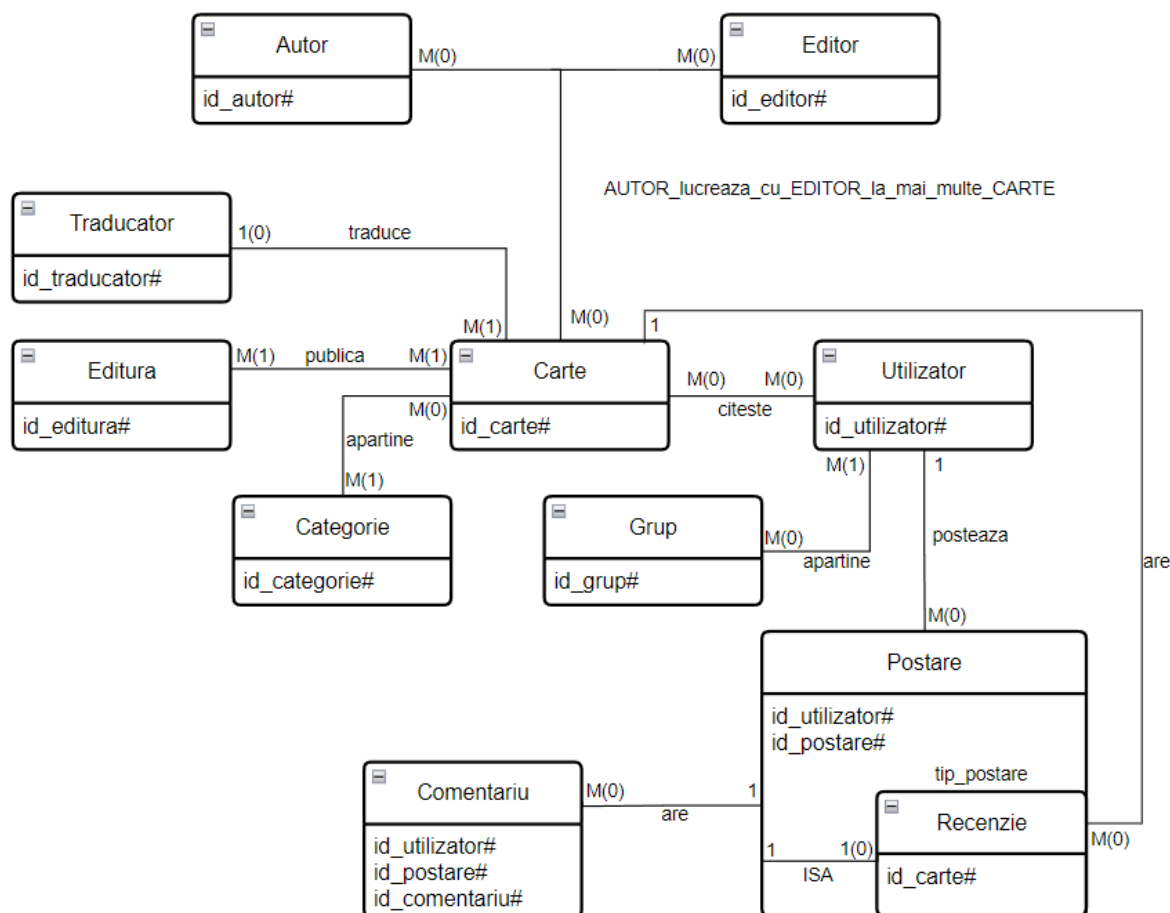
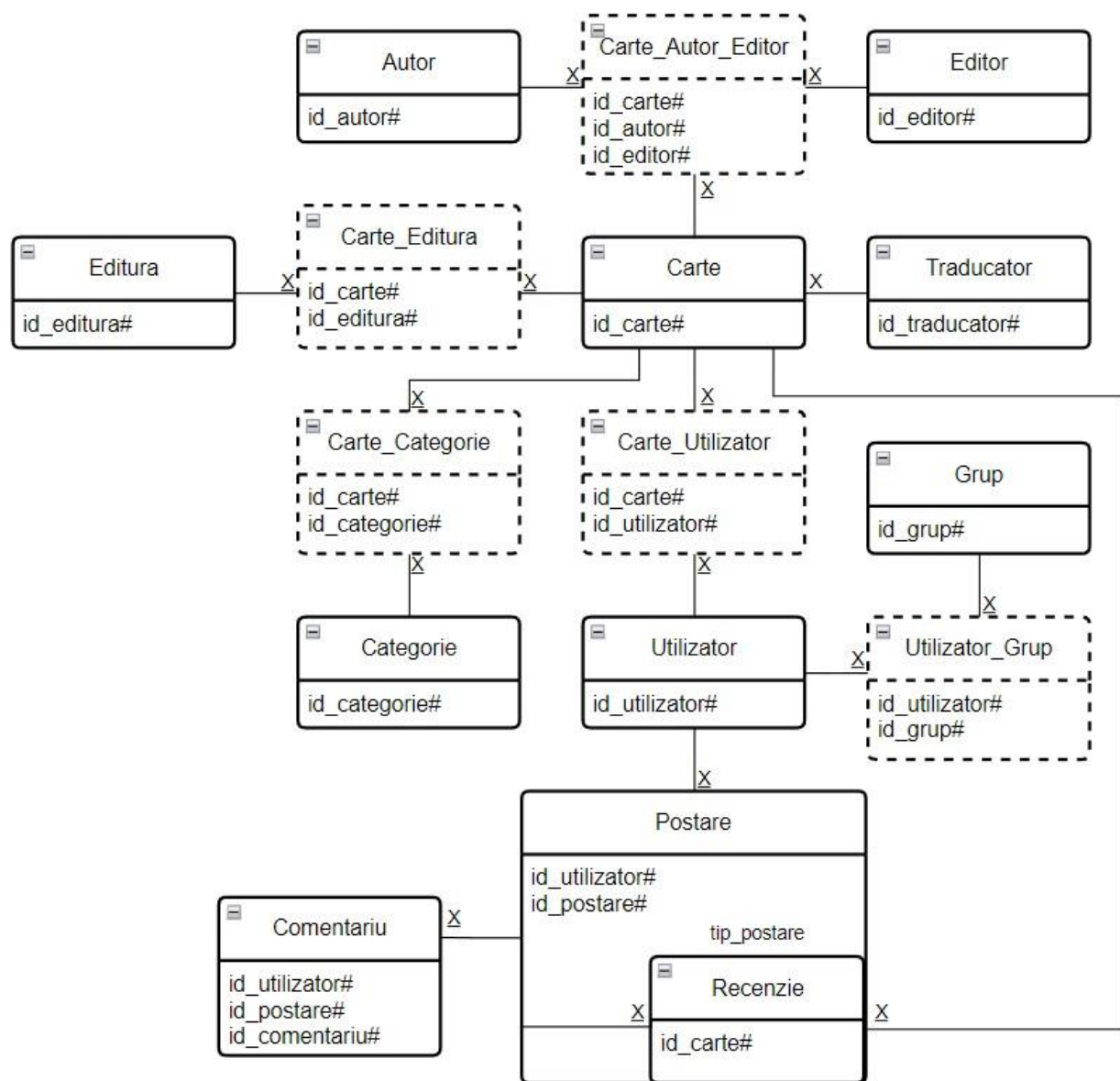


Diagrama Conceptuală



Scheme relaționale

corespunzătoare diagramei conceptuale

- CATEGORIE(id_categorie#, nume)
- CARTE_CATEGORIE(id_carte#, id_categorie#)
- EDITURA(id_editura#, nume, data_infiintare)
- CARTE_EDITURA(id_editura#, id_carte#, nr_exemplare)
- TRADUCATOR(id_traducator#, nume, prenume)
- AUTOR(id_ador#, nume, prenume, data_debut)
- EDITOR(id_editor#, nume, prenume, data_debut)

- CARTE_AUTOR_EDITOR(id_autor#, id_editor#, id_carte#, pret)
- CARTE(id_carte#, nume, nr_capitole, nr_pagini, isbn, cost_productie, pret, id_traducator)
- UTILIZATOR(id_utilizator#, nume, prenume, data_nasterii, email, nr_telefon)
- CARTE_UTILIZATOR(id_utilizator#, id_carte#, data_inceput, data_sfarsit)
- GRUP(id_grup#, nume, nr_membrii)
- UTILIZATOR_GRUP(id_utilizator#, id_grup#, data_intrare)
- POSTARE(id_utilizator#, id_postare#, continut, nr_aprecieri, nr_comentarii)
- RECENZIE(id_utilizator#, id_postare#, id_carte#, nr_stele)
- COMENTARIU(id_utilizator#, id_postare#, id_comentariu#, continut)

Realizarea normalizării până la forma normală 3

Datorită modului în care este proiectată diagrama conceptuală, ea se află deja în forma normală 3, astfel încât, în continuare, voi prezenta doar câte un exemplu de non-FN1, non-FN2, non-FN3 și transformarea fiecăruia în forma normală corespunzătoare.

Forma normală 1

O relație este în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă(atomică).

Exemplu:

TRADUCATOR(Non-FN1)

| id_traducator# | nume_si_prenume |
|----------------|-----------------|
| 1 | Neculae Andrei |
| 2 | Ilie Dumitru |
| 3 | Popescu Stefan |

Relația TRADUCATOR(id_traducator#, nume_si_prenume) nu este în forma normală 1 pentru că atributul nume_si_prenume nu este atomic. Pentru a normaliza relația, vom împărți atributul în două atribute: nume și prenume.

TRADUCATOR(FN1)

| id_traducator# | nume | prenume |
|----------------|---------|---------|
| 1 | Neculae | Andrei |
| 2 | Ilie | Dumitru |
| 3 | Popescu | Stefan |

Forma normală 2

O relație R este în a doua formă normală dacă și numai dacă este în FN1 și fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară. FN2 interzice manifestarea unor dependențe funcționale parțiale în cadrul relației R.

Exemplu:

CARTE_EDITURA(Non-FN2)

| id_editura# | id_carte# | nr_exemplare | nume_editura |
|-------------|-----------|--------------|--------------|
| 1 | 1 | 100 | Nemira |
| 1 | 2 | 120 | Nemira |
| 2 | 4 | 250 | Polirom |

Relația CARTE_EDITURA este în FN2 dacă și numai dacă:

- este în FN1, toate atributele fiind în formă atomică.
- Fiecare atribut care nu participă la cheia primară este dependent de întreaga cheie primară, în cazul de față, atributele nr_exemplare, cost_productie, nume_editura nu sunt chei și trebuie să depindă direct de întreaga cheie primară id_editura# și nr_proiect#. Aceste atribute nu depind direct de întreaga cheie primară, deoarece se observă dependența directă dintre id_editura# și nume_editura, însemnând că nume_editura depinde direct doar de o parte a cheii primare, și anume doar de id_editura.

Deci, relația CARTE_EDITURA nu se află în FN2.

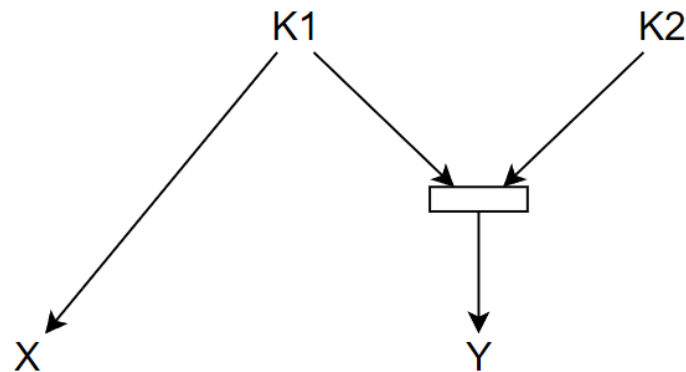
Avem următoarele relații de dependență:

- {id_editura#} => {nume_editura} - id_editura determina functional nume_editura
- {id_editura#, id_carte#} => {nr_exemplare}

Aplicarea regulii Casey-Delobel pentru FN2:

Fie relația R(K1, K2, X, Y), unde K1 și K2 definesc cheia primară, iar X și Y sunt mulțimi de atribute, astfel încât K1 => X. Din cauza dependenței

funcționale $K1 \Rightarrow X$, care arată că R nu este în FN2, se înlocuiește R (fără pierdere de informație) cu două proiecții: $R1(K1, K2, Y)$ și $R2(K1, X)$.



În acest caz:

- $R = \text{CARTE_EDITURA}$
- $K1 = \text{id_editura}$
- $K2 = \text{id_carte}$
- $X = \{\text{nume_editura}\}$
- $Y = \{\text{nr_exemplare}\}$

Transformarea în FN2:

CARTE_EDITURA

| id_editura# | id_carte# | nr_exemplare |
|-------------|-----------|--------------|
| 1 | 1 | 100 |
| 1 | 2 | 120 |
| 2 | 4 | 250 |

EDITURA

| id_editura# | nume_editura |
|-------------|--------------|
| 1 | Nemira |
| 2 | Polirom |

În acest moment, în ambele relații attributele sunt atomice și toate attributele depind de întreaga cheie primară din respectiva relație. Deci, ele se află în FN2.

Forma normală 3

O relație R este în a treia formă normală dacă și numai dacă este în FN2 și fiecare atribut care nu este cheie (nu participă la cheia primară) depinde direct de cheia primară.

Cu alte cuvinte, o relație este în FN3 dacă și numai dacă fiecare atribut care nu este cheie, depinde de cheie, de întreaga cheie și numai de cheie.

Exemplu:

Pentru a nu se încărca exemplul, luăm doar aceste atribute din entitatea CARTE.

CARTE(Non-FN3)

| id_carte# | id_traducator | nume_traducator |
|-----------|---------------|-----------------|
| 1 | 1 | Neculae |
| 2 | 2 | Rusu |
| 3 | 3 | Popescu |

Pentru a aduce relația CARTE în FN3 se aplică regula Casey-Delobel. Relația se descompune, prin eliminarea dependențelor funcționale tranzitive, în proiecțiile:

CARTE_A(id_carte#, id_traducator)

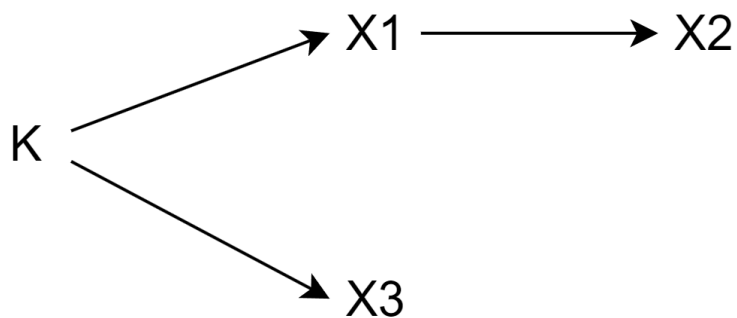
CARTE_B(id_traducator#, nume_traducator)

Aplicarea regulii Casey-Delobel pentru FN3:

Fie relația $R(K, X1, X2, X3)$, unde $X2$ depinde tranzitiv de K , iar K este cheia primară a lui R . Presupunem că $K \Rightarrow X1 \Rightarrow X2$. Din cauza dependenței funcționale $X1 \Rightarrow X2$ care arată că R nu este în FN3, se înlocuiește R (fără pierdere de informație) prin două proiecții $R1(K, X1, X3)$ și $R2(X1, X2)$.

În acest caz:

- $R = \text{CARTE}$
- $K = \{\text{id_carte}\# \}$
- $X1 = \{\text{id_traducator}\}$
- $X2 = \{\text{nume_traducator}\}$
- $X3 = \{ \}$



Transformarea în FN3:

CARTE_A

| id_carte# | id_traducator |
|-----------|---------------|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

CARTE_B

| id_traducator# | nume_traducator |
|----------------|-----------------|
| 1 | Neculae |
| 2 | Rusu |
| 3 | Popescu |

În acest moment, ambele relații sunt în FN2 și nu există nicio dependență indirectă a vreunui atribut față de cheia primară.

În diagrama conceptuală, entitatea CARTE_A este entitatea CARTE, iar entitatea CARTE_B este entitatea TRADUCATOR. Deci, diagrama conceptuală este în FN3.

Crearea de secvențe pentru popularea înregistrărilor în tabele

CREATE SEQUENCE secventa_categorie

INCREMENT BY 1

START WITH 1

MAXVALUE 999

NOCYCLE;

```
CREATE SEQUENCE secventa_editura  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 999999999  
NOCYCLE;
```

```
CREATE SEQUENCE secventa_traducator  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 999999999  
NOCYCLE;
```

```
CREATE SEQUENCE secventa_autor  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 999999999  
NOCYCLE;
```

```
CREATE SEQUENCE secventa_editor  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 999999999  
NOCYCLE;
```

```
CREATE SEQUENCE secventa_carte  
INCREMENT BY 1
```

START WITH 1
MAXVALUE 999999999999999999
NOCYCLE;

CREATE SEQUENCE secventa_utilizator
INCREMENT BY 1
START WITH 1
MAXVALUE 999999999
NOCYCLE;

CREATE SEQUENCE secventa_grup
INCREMENT BY 1
START WITH 1
MAXVALUE 999999999999999999
NOCYCLE;

CREATE SEQUENCE secventa_postare
INCREMENT BY 1
START WITH 1
MAXVALUE 999999999
NOCYCLE;

```
Worksheet | Query Builder
1 CREATE SEQUENCE secventa_categorie
2 INCREMENT BY 1
3 START WITH 1
4 MAXVALUE 999
5 NOCYCLE;
6
7 CREATE SEQUENCE secventa_editura
8 INCREMENT BY 1
9 START WITH 1
10 MAXVALUE 999999999
11 NOCYCLE;
12
13 CREATE SEQUENCE secventa_traducator
14 INCREMENT BY 1
15 START WITH 1
16 MAXVALUE 999999999
17 NOCYCLE;
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

Script Output x | Task completed in 0.036 seconds

Sequence SECVENTA_CATEGORIE created.

Sequence SECVENTA_EDITURA created.

```
Worksheet | Query Builder
13 CREATE SEQUENCE secventa_traducator
14 INCREMENT BY 1
15 START WITH 1
16 MAXVALUE 999999999
17 NOCYCLE;
18
19
20 CREATE SEQUENCE secventa_autor
21 INCREMENT BY 1
22 START WITH 1
23 MAXVALUE 999999999
24 NOCYCLE;
25
26 CREATE SEQUENCE secventa_editor
27 INCREMENT BY 1
28 START WITH 1
29 MAXVALUE 999999999
30 NOCYCLE;
31
```

Script Output x | Task completed in 0.047 seconds

Sequence SECVENTA_TRADUCATOR created.

Sequence SECVENTA_AUTOR created.

Worksheet Query Builder

```
26 CREATE SEQUENCE secventa_editor
27 INCREMENT BY 1
28 START WITH 1
29 MAXVALUE 999999999
30 NOCYCLE;
31
32 CREATE SEQUENCE secventa_carte
33 INCREMENT BY 1
34 START WITH 1
35 MAXVALUE 999999999999999999
36 NOCYCLE;
37
38 CREATE SEQUENCE secventa_utilizator
39 INCREMENT BY 1
40 START WITH 1
41 MAXVALUE 999999999
42 NOCYCLE;
43
44 CREATE SEQUENCE secventa_grup
```

Script Output x

Task completed in 0.042 seconds

Sequence SECVENTA_EDITOR created.

Sequence SECVENTA_CARTE created.

Worksheet Query Builder

```
38 CREATE SEQUENCE secventa_utilizator
39 INCREMENT BY 1
40 START WITH 1
41 MAXVALUE 999999999
42 NOCYCLE;
43
44 CREATE SEQUENCE secventa_grup
45 INCREMENT BY 1
46 START WITH 1
47 MAXVALUE 999999999999999999
48 NOCYCLE;
49
50 CREATE SEQUENCE secventa_postare
51 INCREMENT BY 1
52 START WITH 1
53 MAXVALUE 999999999
54 NOCYCLE;
```

Script Output x Query Result x

Task completed in 0.103 seconds

Sequence SECVENTA_UTILIZATOR created.

Sequence SECVENTA_GRUP created.

Sequence SECVENTA_POSTARE created.

Crearea tabelelor și inserarea datelor

CREATE TABLE CATEGORIE

```
(  
    id_categorie NUMBER(3) PRIMARY KEY,  
    nume VARCHAR2(25) NOT NULL  
);
```

CREATE TABLE EDITURA

```
(  
    id_editura NUMBER(9) PRIMARY KEY,  
    nume VARCHAR2(25) NOT NULL,  
    data_infiintare DATE NOT NULL  
);
```

CREATE TABLE TRADUCATOR

```
(  
    id_traducator NUMBER(9) PRIMARY KEY,  
    nume VARCHAR2(25) NOT NULL,  
    prenume VARCHAR2(25) NOT NULL  
);
```

CREATE TABLE AUTOR

```
(  
    id_autor NUMBER(9) PRIMARY KEY,  
    nume VARCHAR2(25) NOT NULL,  
    prenume VARCHAR2(25) NOT NULL,  
    data_debut DATE NOT NULL
```


);

CREATE TABLE EDITOR

(
 id_editor NUMBER(9) PRIMARY KEY,
 nume VARCHAR2(25) NOT NULL,
 prenume VARCHAR2(25) NOT NULL,
 data_debut DATE NOT NULL
);

CREATE TABLE CARTE

(
 id_carte NUMBER(18) PRIMARY KEY,
 nume VARCHAR2(50) NOT NULL,
 nr_capitole NUMBER(3) NOT NULL,
 nr_pagini NUMBER(4) NOT NULL,
 isbn VARCHAR2(13) NOT NULL CHECK(LENGTH(isbn)=10 or
LENGTH(isbn)=13),
 cost_productie NUMBER(2),
 pret NUMBER(3) NOT NULL,
 limba VARCHAR2(15) NOT NULL,
 id_traducator NUMBER(3) REFERENCES TRADUCATOR(id_traducator)
);

CREATE TABLE UTILIZATOR

(
 id_utilizator NUMBER(9) PRIMARY KEY,
 nume VARCHAR2(25) NOT NULL,

```
    prenume VARCHAR2(25) NOT NULL,  
    nume_cont VARCHAR2(25) UNIQUE NOT NULL,  
    parola VARCHAR2(25) NOT NULL,  
    data_nasterii DATE,  
    email VARCHAR2(50) NOT NULL,  
    nr_telefon VARCHAR2(15)  
);
```

CREATE TABLE GRUP

```
(  
    id_grup NUMBER(18) PRIMARY KEY,  
    nume VARCHAR2(25) NOT NULL,  
    nr_membrii NUMBER(7) NOT NULL  
);
```

CREATE TABLE POSTARE

```
(  
    id_utilizator NUMBER(9) REFERENCES UTILIZATOR(id_utilizator) NOT  
NULL,  
    id_postare NUMBER(9) NOT NULL,  
    continut VARCHAR2(4000) NOT NULL,  
    nr_aprecieri NUMBER(9) NOT NULL,  
    nr_comentarii NUMBER(18) DEFAULT 0,  
    tip_postare VARCHAR2(8) NOT NULL CHECK(tip_postare='POSTARE'  
or tip_postare='RECENZIE'),  
    PRIMARY KEY (id_utilizator, id_postare)  
);
```

CREATE TABLE RECENZIE

```
(
    id_utilizator NUMBER(9) NOT NULL,
    id_postare NUMBER(9) NOT NULL,
    id_carte NUMBER(9) REFERENCES CARTE(id_carte) NOT NULL,
    nr_stele NUMBER(2,1) NOT NULL CHECK(nr_stele IN (0.5, 1, 1.5, 2, 2.5,
3, 3.5, 4, 4.5, 5)),
    PRIMARY KEY (id_utilizator, id_postare, id_carte),
    FOREIGN KEY (id_utilizator, id_postare) REFERENCES
POSTARE(id_utilizator,id_postare)
);
```

CREATE TABLE COMENTARIU

```
(
    id_utilizator NUMBER(9) NOT NULL,
    id_postare NUMBER(9) NOT NULL,
    id_comentariu NUMBER(18) NOT NULL,
    continut VARCHAR2(4000) NOT NULL,
    id_utilizator_comentariu REFERENCES UTILIZATOR(id_utilizator) NOT
NULL,
    PRIMARY KEY (id_utilizator, id_postare, id_comentariu),
    FOREIGN KEY (id_utilizator, id_postare) REFERENCES
POSTARE(id_utilizator,id_postare)
);
```

CREATE TABLE CARTE_CATEGORIE

```
(
    id_carte NUMBER(18) REFERENCES CARTE(id_carte) NOT NULL,
```

```

    id_categorie NUMBER(3) REFERENCES CATEGORIE(id_categorie) NOT
NULL,
    PRIMARY KEY (id_carte, id_categorie)
);

```

```

CREATE TABLE CARTE_EDITURA
(
    id_editura NUMBER(9) REFERENCES EDITURA(id_editura) NOT NULL,
    id_carte NUMBER(18) REFERENCES CARTE(id_carte) NOT NULL,
    nr_exemplare NUMBER(9),
    PRIMARY KEY (id_editura, id_carte)
);

```

```

CREATE TABLE CARTE_UTILIZATOR
(
    id_utilizator NUMBER(9) REFERENCES UTILIZATOR(id_utilizator) NOT
NULL,
    id_carte NUMBER(18) REFERENCES CARTE(id_carte) NOT NULL,
    data_inceput DATE NOT NULL,
    data_sfarsit DATE NOT NULL,
    CONSTRAINT check_date_valide CHECK(data_inceput<=data_sfarsit)
);

```

```

CREATE TABLE UTILIZATOR_GRUP
(
    id_utilizator NUMBER(9) REFERENCES UTILIZATOR(id_utilizator) NOT
NULL,
    id_grup NUMBER(18) REFERENCES GRUP(id_grup) NOT NULL,

```

```
data_intrare DATE NOT NULL,  
PRIMARY KEY (id_utilizator, id_grup)  
);
```

```
CREATE TABLE CARTE_AUTOR_EDITOR  
(  
    id_autor NUMBER(9) REFERENCES AUTOR(id_autor) NOT NULL,  
    id_editor NUMBER(9) REFERENCES EDITOR(id_editor) NOT NULL,  
    id_carte NUMBER(18) REFERENCES CARTE(id_carte) NOT NULL,  
    pret NUMBER(6),  
    PRIMARY KEY (id_autor, id_editor, id_carte)  
);
```

```
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'Fantastic');  
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'SF');  
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'Thriller');  
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'Romanta');  
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'Groaza');  
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'Motivational');  
INSERT INTO CATEGORIE(id_categorie, nume)  
VALUES (secventa_categorie.nextval,'Politica');
```

```

INSERT INTO EDITURA(id_editura, nume, data_infiintare)
VALUES (secventa_editura.nextval,'Nemira',TO_DATE('01-01-1991','dd-mm-
yyyy'));

INSERT INTO EDITURA(id_editura, nume, data_infiintare)
VALUES (secventa_editura.nextval,'Polirom',TO_DATE('24-06-1999','dd-mm-
yyyy'));

INSERT INTO EDITURA(id_editura, nume, data_infiintare)
VALUES (secventa_editura.nextval,'Paladin',TO_DATE('19-09-2008','dd-mm-
yyyy'));

INSERT INTO EDITURA(id_editura, nume, data_infiintare)
VALUES (secventa_editura.nextval,'Gollancz',TO_DATE('23-05-1998','dd-
mm-yyyy'));

INSERT INTO EDITURA(id_editura, nume, data_infiintare)
VALUES (secventa_editura.nextval,'Trei',TO_DATE('25-08-1994','dd-mm-
yyyy'));

INSERT INTO EDITURA(id_editura, nume, data_infiintare)
VALUES (secventa_editura.nextval,'Tor Books',TO_DATE('02-04-1980','dd-
mm-yyyy'));


INSERT INTO TRADUCATOR(id_traducator, nume, prenume)
VALUES (secventa_traducator.nextval,'Popescu','Alexandru');

INSERT INTO TRADUCATOR(id_traducator, nume, prenume)
VALUES (secventa_traducator.nextval,'Gheorghe','Stefan');

INSERT INTO TRADUCATOR(id_traducator, nume, prenume)
VALUES (secventa_traducator.nextval,'Brincus','Mihai');

INSERT INTO TRADUCATOR(id_traducator, nume, prenume)
VALUES (secventa_traducator.nextval,'Crainicul','George');

INSERT INTO TRADUCATOR(id_traducator, nume, prenume)

```

```
VALUES (secventa_traducator.nextval,'Rusu','Florin');
```

```
INSERT INTO AUTOR(id_autor, nume, prenume,data_debut)
```

```
VALUES (secventa_autor.nextval,'Haig','Matt',TO_DATE('04-02-2002','dd-mm-yyyy'));
```

```
INSERT INTO AUTOR(id_autor, nume, prenume, data_debut)
```

```
VALUES (secventa_autor.nextval,'King','Stephen',TO_DATE('06-06-1970','dd-mm-yyyy'));
```

```
INSERT INTO AUTOR(id_autor, nume, prenume, data_debut)
```

```
VALUES (secventa_autor.nextval,'Sanderson','Brandon',TO_DATE('21-04-2005','dd-mm-yyyy'));
```

```
INSERT INTO AUTOR(id_autor, nume, prenume, data_debut)
```

```
VALUES (secventa_autor.nextval,'Ferriss','Tim',TO_DATE('01-08-2007','dd-mm-yyyy'));
```

```
INSERT INTO AUTOR(id_autor, nume, prenume, data_debut)
```

```
VALUES (secventa_autor.nextval,'Brown','Pierce',TO_DATE('28-01-2014','dd-mm-yyyy'));
```

```
INSERT INTO AUTOR(id_autor, nume, prenume, data_debut)
```

```
VALUES (secventa_autor.nextval,'Niven','Jennifer',TO_DATE('06-01-2015','dd-mm-yyyy'));
```

```
INSERT INTO EDITOR(id_editor, nume, prenume, data_debut)
```

```
VALUES (secventa_editor.nextval,'Olsen','Clark',TO_DATE('20-11-2020','dd-mm-yyyy'));
```

```
INSERT INTO EDITOR(id_editor, nume, prenume, data_debut)
```

```
VALUES (secventa_editor.nextval,'Adams','Purple',TO_DATE('10-12-2019','dd-mm-yyyy'));
```

```
INSERT INTO EDITOR(id_editor, nume, prenume, data_debut)
```

```
VALUES (secventa_editor.nextval,'Driver','Abraham',TO_DATE('04-05-1988','dd-mm-yyyy'));
```

```
INSERT INTO EDITOR(id_editor, nume, prenume, data_debut)
VALUES (secventa_editor.nextval,'Park','Mark',TO_DATE('09-03-1997','dd-
mm-yyyy'));
```

```
INSERT INTO EDITOR(id_editor, nume, prenume, data_debut)
VALUES (secventa_editor.nextval,'Bryant','Arthur',TO_DATE('10-08-
2003','dd-mm-yyyy'));
```

```
INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES
(secventa_carte.nextval,'Misery',20,356,'1235654213233',5,12,'romana',1);
```

```
INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'Cateva motive sa iubesti
viata',26,240,'1233334456433',4,11,'romana',2);
```

```
INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'Furia
rosie',44,506,'2342812937123',6,13,'romana',3);
```

```
INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'JFK
22.11.63',60,720,'9786064309075',8,15,'romana',1);
```

```
INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'4 ore-saptamana de
lucru',23,496,'9786069130193',6,13,'romana',4);
```

```
INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'Toate acele locuri
minunate',31,401,'9786067194098',5,12,'romana',5);
```



```

INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'Ultimul
Imperiu',45,816,'9789737078230',7,14,'romana',2);

INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba, id_traducator)
VALUES (secventa_carte.nextval,'Mr.
Mercedes',39,488,'9786067581799',5,12,'romana',1);

INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba)
VALUES (secventa_carte.nextval,'The Final
Empire',45,537,'1222234568',7,14,'engleza');

INSERT INTO CARTE(id_carte, nume, nr_capitole, nr_pagini, isbn,
cost_productie, pret, limba)
VALUES
(secventa_carte.nextval,'Calamity',51,421,'1034985610113',6,13,'engleza');

```

```

INSERT INTO UTILIZATOR(id_utilizator, nume, prenume, nume_cont,
parola, data_nasterii, email, nr_telefon)
VALUES
(secventa_utilizator.nextval,'Buzatu','Giulian','Giulian617','ImiPlaceMatematica'
,TO_DATE('16-07-2003','dd-mm-
yyyy'),'buzatugiulian@yahoo.com','0754641082');

INSERT INTO UTILIZATOR(id_utilizator, nume, prenume, nume_cont,
parola, data_nasterii, email, nr_telefon)
VALUES
(secventa_utilizator.nextval,'Neculae','Andrei','Andrei12','ImiPlaceSaMananc','T
O_DATE('7-10-2003','dd-mm-
yyyy'),'neculaefabian@gmail.com','0722227890');

INSERT INTO UTILIZATOR(id_utilizator, nume, prenume, nume_cont,
parola, data_nasterii, email, nr_telefon)
VALUES
(secventa_utilizator.nextval,'Ilie','Dumitru','TheWinner','TheWinnerNeverLoses'

```

```
,TO_DATE('05-05-1998','dd-mm-  
yyyy'),'thewinner@gmail.com','0788899911');
```

```
INSERT INTO UTILIZATOR(id_utilizator, nume, prenume, nume_cont,  
parola, data_nasterii, email)
```

```
VALUES
```

```
(secventa_utilizator.nextval,'Preda','Maria','MariaMP','AmMedia10HAHA',TO_  
DATE('19-03-2004','dd-mm-yyyy'),'mariamp@gmail.com');
```

```
INSERT INTO UTILIZATOR(id_utilizator, nume, prenume, nume_cont,  
parola, email, nr_telefon)
```

```
VALUES
```

```
(secventa_utilizator.nextval,'Grigore','Mihai','MihaiSuntSmecher','SuntDinGalat  
iBoss','seful@gmail.com','0777777777');
```

```
INSERT INTO GRUP(id_grup, nume, nr_membrii)
```

```
VALUES (secventa_grup.nextval,'Despre carti',3);
```

```
INSERT INTO GRUP(id_grup, nume, nr_membrii)
```

```
VALUES (secventa_grup.nextval,'Comunitatea SF',2);
```

```
INSERT INTO GRUP(id_grup, nume, nr_membrii)
```

```
VALUES (secventa_grup.nextval,'Books for boys',4);
```

```
INSERT INTO GRUP(id_grup, nume, nr_membrii)
```

```
VALUES (secventa_grup.nextval,'Romania pe Goodreads',5);
```

```
INSERT INTO GRUP(id_grup, nume, nr_membrii)
```

```
VALUES (secventa_grup.nextval,'FMI',5);
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri,  
tip_postare)
```

```
VALUES (1,secventa_postare.nextval,'Ce parere aveti despre atacul asupra  
literaturii SF?',1,'POSTARE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri,  
nr_comentarii, tip_postare)
```

```
VALUES (4,secventa_postare.nextval,'Top 10 carti pe care le-ati citit?',2,2,'POSTARE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri, nr_comentarii, tip_postare)
```

```
VALUES (2,secventa_postare.nextval,'Am de gand sa citesc ceva din categoria fantastic. Recomandari?',2,1,'POSTARE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri, tip_postare)
```

```
VALUES (1,secventa_postare.nextval,'Mi s-a parut foarte bine structurata si cu idei misto.',3,'RECENZIE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri, nr_comentarii, tip_postare)
```

```
VALUES (1,secventa_postare.nextval,'Foarte dezamagitoare.',2,1,'RECENZIE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri, nr_comentarii, tip_postare)
```

```
VALUES (3,secventa_postare.nextval,'Cea mai faina carte citita de mine.',3,1,'RECENZIE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri, tip_postare)
```

```
VALUES (2,secventa_postare.nextval,'A rupt broooo',1,'RECENZIE');
```

```
INSERT INTO POSTARE(id_utilizator, id_postare, continut, nr_aprecieri, tip_postare)
```

```
VALUES (5,secventa_postare.nextval,'Nota 9/10. Se putea mai bine. Mi-am irosit timpul.',4,'RECENZIE');
```

```
INSERT INTO RECENZIE(id_utilizator, id_postare, id_carte, nr_stele)
```

```
VALUES (1,4,1,4);
```

```
INSERT INTO RECENZIE(id_utilizator, id_postare, id_carte, nr_stele)
```

```
VALUES (1,5,2,1.5);
```

```
INSERT INTO RECENZIE(id_utilizator, id_postare, id_carte, nr_stele)
```

```

VALUES (3,6,9,5);
INSERT INTO RECENZIE(id_utilizator, id_postare, id_carte, nr_stele)
VALUES (2,7,4,5);
INSERT INTO RECENZIE(id_utilizator, id_postare, id_carte, nr_stele)
VALUES (5,8,2,4.5);

INSERT INTO COMENTARIU(id_utilizator, id_postare, id_comentariu,
continut, id_utilizator_comentariu)
VALUES (4,2,1,'Mi-este greu sa alcatuiesc un top 10... Vrei sa iesim zilele
viitoare si sa discutam despre carti?',1);
INSERT INTO COMENTARIU(id_utilizator, id_postare, id_comentariu,
continut, id_utilizator_comentariu)
VALUES (4,2,2,'Da, suna foarte bine. (Spoiler alert:nu o sa vin)',4);
INSERT INTO COMENTARIU(id_utilizator, id_postare, id_comentariu,
continut, id_utilizator_comentariu)
VALUES (2,3,1,'The Final Empire e cartea potrivita.',3);
INSERT INTO COMENTARIU(id_utilizator, id_postare, id_comentariu,
continut, id_utilizator_comentariu)
VALUES (1,5,1,'Imi pare rau sa aud asta bro=(',2);
INSERT INTO COMENTARIU(id_utilizator, id_postare, id_comentariu,
continut, id_utilizator_comentariu)
VALUES (3,6,1,'Wow pare ca ti-a placut foarte mult. Cred ca o sa o incerc si
eu.',2);

INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (1,3);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (1,5);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (2,6);

```

```
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (3,1);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (3,2);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (4,1);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (4,2);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (4,3);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (4,4);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (5,6);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (6,4);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (7,1);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (8,3);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (8,5);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (9,1);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
VALUES (10,1);
INSERT INTO CARTE_CATEGORIE(id_carte, id_categorie)
```

VALUES (10,2);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (1,1,2000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (1,2,3000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (3,3,2500);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (1,4,2500);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (2,5,800);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (5,6,5000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (5,7,1500);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (1,8,1000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (4,9,400000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (6,9,800000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (4,10,100000);

INSERT INTO CARTE_EDITURA(id_editura,id_carte,nr_exemplare)
VALUES (6,10,300000);

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (1,1,TO_DATE('03-03-2020','dd-mm-yyyy'),TO_DATE('07-03-2020','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (1,2,TO_DATE('03-06-2019','dd-mm-yyyy'),TO_DATE('08-06-2019','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (1,7,TO_DATE('07-11-2022','dd-mm-yyyy'),TO_DATE('17-11-2022','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (2,4,TO_DATE('03-05-2020','dd-mm-yyyy'),TO_DATE('17-05-2020','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (2,5,TO_DATE('01-07-2021','dd-mm-yyyy'),TO_DATE('08-08-2021','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (3,9,TO_DATE('22-03-2022','dd-mm-yyyy'),TO_DATE('07-04-2022','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (3,10,TO_DATE('10-09-2020','dd-mm-yyyy'),TO_DATE('14-09-2020','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (4,3,TO_DATE('13-10-2018','dd-mm-yyyy'),TO_DATE('23-10-2018','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (4,6,TO_DATE('03-12-2022','dd-mm-yyyy'),TO_DATE('18-12-
2022','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (4,8,TO_DATE('04-03-2023','dd-mm-yyyy'),TO_DATE('10-03-
2023','dd-mm-yyyy'));
```

```
INSERT INTO
CARTE_UTILIZATOR(id_utilizator,id_carte,data_inceput,data_sfarsit)
VALUES (5,2,TO_DATE('03-03-2019','dd-mm-yyyy'),TO_DATE('07-03-
2019','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (1,1,TO_DATE('01-01-2019','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (3,1,TO_DATE('02-01-2019','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (4,1,TO_DATE('02-01-2019','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (1,2,TO_DATE('11-11-2020','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (2,2,TO_DATE('21-11-2020','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (1,3,TO_DATE('10-12-2020','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (2,3,TO_DATE('11-12-2020','dd-mm-yyyy'));
```

```
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (3,3,TO_DATE('12-01-2021','dd-mm-yyyy'));
```



```

INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (5,3,TO_DATE('13-01-2021','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (1,4,TO_DATE('15-03-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (2,4,TO_DATE('16-03-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (3,4,TO_DATE('17-03-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (4,4,TO_DATE('19-03-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (5,4,TO_DATE('30-05-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (1,5,TO_DATE('10-11-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (2,5,TO_DATE('13-11-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (3,5,TO_DATE('13-11-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (4,5,TO_DATE('15-11-2022','dd-mm-yyyy'));
INSERT INTO UTILIZATOR_GRUP(id_utilizator,id_grup,data_intrare)
VALUES (5,5,TO_DATE('24-12-2022','dd-mm-yyyy'));

INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte,pret)
VALUES (2,1,1,10000);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte,pret)
VALUES (1,2,2,15000);

```

```

INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte,pret)
VALUES (5,3,3,20000);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte,pret)
VALUES (2,2,4,8000);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte,pret)
VALUES (4,4,5,20000);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte,pret)
VALUES (6,5,6,16000);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte)
VALUES (3,3,7);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte)
VALUES (2,3,8);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte)
VALUES (3,2,9);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte)
VALUES (3,1,10);
INSERT INTO CARTE_AUTOR_EDITOR(id_autor,id_editor,id_carte)
VALUES (2,2,10);

```

Worksheet Query Builder

```
1 select * from traducator;
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.01 seconds

| | ID_TRADUCATOR | NUME | PRENUME |
|---|---------------|-----------|-----------|
| 1 | 1 | Popescu | Alexandru |
| 2 | 2 | Gheorghe | Stefan |
| 3 | 3 | Brincus | Mihai |
| 4 | 4 | Crainicul | George |
| 5 | 5 | Rusu | Florin |

Worksheet Query Builder

```
1 select * from autor;
```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0.001 seconds

| | ID_AUTOR | NUME | PRENUME | DATA_DEBUT |
|---|----------|-----------|----------|------------|
| 1 | 1 | Haiq | Matt | 04-FEB-02 |
| 2 | 2 | King | Stephen | 06-JUN-70 |
| 3 | 3 | Sanderson | Brandon | 21-APR-05 |
| 4 | 4 | Ferriss | Tim | 01-AUG-07 |
| 5 | 5 | Brown | Pierce | 28-JAN-14 |
| 6 | 6 | Niven | Jennifer | 06-JAN-15 |

Worksheet Query Builder

```
1 select * from editor;
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.002 seconds

| | ID_EDITOR | NUME | PRENUME | DATA_DEBUT |
|---|-----------|--------|---------|------------|
| 1 | 1 | Olsen | Clark | 20-NOV-20 |
| 2 | 2 | Adams | Purple | 10-DEC-19 |
| 3 | 3 | Driver | Abraham | 04-MAY-88 |
| 4 | 4 | Park | Mark | 09-MAR-97 |
| 5 | 5 | Bryant | Arthur | 10-AUG-03 |

Worksheet

Query Builder

1

select * from postare;

Script Output x

Query Result x

SQL

All Rows Fetched: 8 in 0.007 seconds

| | ID_UTILIZATOR | ID_POSTARE | CONTINUT | NR_APPRECIERI | NR_COMENTARII | TIP_POSTARE |
|---|---------------|------------|---|---------------|---------------|-------------|
| 1 | 1 | 1 | Ce parere aveti despre atacul asupra literaturii SF? | 1 | 0 | POSTARE |
| 2 | 4 | 2 | Top 10 carti pe care le-ati citit? | 2 | 2 | POSTARE |
| 3 | 2 | 3 | Am de gand sa citesc ceva din categoria fantastic. Recomandari? | 2 | 1 | POSTARE |
| 4 | 1 | 4 | Mi s-a parut foarte bine structurata si cu idee misto. | 3 | 0 | RECENZIE |
| 5 | 1 | 5 | Foarte dezamagitoare. | 2 | 1 | RECENZIE |
| 6 | 3 | 6 | Cea mai faina carte citita de mine. | 3 | 1 | RECENZIE |
| 7 | 2 | 7 | A rupt broooo | 1 | 0 | RECENZIE |
| 8 | 5 | 8 | Nota 9/10. Se putea mai bine. Mi-am irosit timpul. | 4 | 0 | RECENZIE |

Worksheet

Query Builder

1

select * from recenzie;

Script Output x

Query Result x

SQL

All Rows Fetched: 5 in 0.002 seconds

| | ID_UTILIZATOR | ID_POSTARE | ID_CARTE | NR_STELE |
|---|---------------|------------|----------|----------|
| 1 | 1 | 4 | 1 | 4 |
| 2 | 1 | 5 | 2 | 1.5 |
| 3 | 3 | 6 | 9 | 5 |
| 4 | 2 | 7 | 4 | 5 |
| 5 | 5 | 8 | 2 | 4.5 |

Worksheet Query Builder

```
1 select * from comentariu;
```

Script Output x Query Result x

SQL All Rows Fetched: 5 in 0.002 seconds

| | ID_UTILIZATOR | ID_POSTARE | ID_COMENTARIU | CONTINUT | ID_UTILIZATOR_COMENTARIU |
|---|---------------|------------|---------------|---|--------------------------|
| 1 | 4 | 2 | | 1Mi-este greu sa alcatuiesc un top 10... Vrei sa iesim zilele viitoare... | 1 |
| 2 | 4 | 2 | | 2Da, suna foarte bine. (Spoiler alert:nu o sa vin) | 4 |
| 3 | 2 | 3 | | 1The Final Empire e cartea potrivita. | 3 |
| 4 | 1 | 5 | | 1Imi pare rau sa aud asta bro=(| 2 |
| 5 | 3 | 6 | | 1Wow pare ca ti-a placut foarte mult. Cred ca o sa o incerc si eu. | 2 |

Worksheet Query Builder

```
1 select * from carte_categorie;
```

Script Output x Query Result x

SQL All Rows Fetched: 17 in 0.002 seconds

| | ID_CARTE | ID_CATEGORIE |
|----|----------|--------------|
| 1 | 1 | 3 |
| 2 | 1 | 5 |
| 3 | 2 | 6 |
| 4 | 3 | 1 |
| 5 | 3 | 2 |
| 6 | 4 | 1 |
| 7 | 4 | 2 |
| 8 | 4 | 3 |
| 9 | 4 | 4 |
| 10 | 5 | 6 |
| 11 | 6 | 4 |
| 12 | 7 | 1 |
| 13 | 8 | 3 |
| 14 | 8 | 5 |
| 15 | 9 | 1 |
| 16 | 10 | 1 |
| 17 | 10 | 2 |

Worksheet Query Builder

```
1 select * from carte_editura;
```

Script Output x Query Result x

SQL All Rows Fetched: 12 in 0.002 seconds

| | ID_EDITURA | ID_CARTE | NR_EXEMPLARE |
|----|------------|----------|--------------|
| 1 | 1 | 1 | 2000 |
| 2 | 1 | 2 | 3000 |
| 3 | 3 | 3 | 2500 |
| 4 | 1 | 4 | 2500 |
| 5 | 2 | 5 | 800 |
| 6 | 5 | 6 | 5000 |
| 7 | 5 | 7 | 1500 |
| 8 | 1 | 8 | 1000 |
| 9 | 4 | 9 | 400000 |
| 10 | 6 | 9 | 800000 |
| 11 | 4 | 10 | 100000 |
| 12 | 6 | 10 | 300000 |

Worksheet Query Builder

```
1 select * from carte_utilizator;
```

Script Output x Query Result x

SQL All Rows Fetched: 11 in 0.002 seconds

| | ID_UTILIZATOR | ID_CARTE | DATA_INCEPUT | DATA_SFARSIT |
|----|---------------|----------|--------------|--------------|
| 1 | 1 | 1 | 103-MAR-20 | 07-MAR-20 |
| 2 | 1 | 1 | 203-JUN-19 | 08-JUN-19 |
| 3 | 1 | 1 | 707-NOV-22 | 17-NOV-22 |
| 4 | 2 | 4 | 403-MAY-20 | 17-MAY-20 |
| 5 | 2 | 5 | 501-JUL-21 | 08-AUG-21 |
| 6 | 3 | 9 | 22-MAR-22 | 07-APR-22 |
| 7 | 3 | 10 | 10-SEP-20 | 14-SEP-20 |
| 8 | 4 | 3 | 13-OCT-18 | 23-OCT-18 |
| 9 | 4 | 6 | 03-DEC-22 | 18-DEC-22 |
| 10 | 4 | 8 | 04-MAR-23 | 10-MAR-23 |
| 11 | 5 | 2 | 03-MAR-19 | 07-MAR-19 |

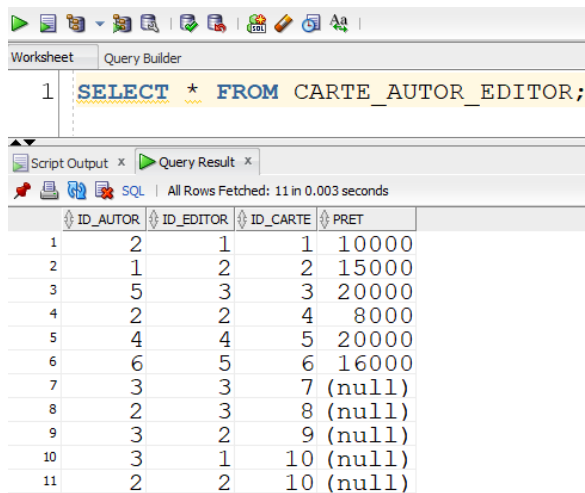
Worksheet Query Builder

```
1 select * from utilizator_grup;
```

Script Output x Query Result x

SQL All Rows Fetched: 19 in 0.003 seconds

| | ID_UTILIZATOR | ID_GRUP | DATA_INTRARE |
|----|---------------|---------|--------------|
| 1 | 1 | 1 | 101-JAN-19 |
| 2 | 3 | 1 | 102-JAN-19 |
| 3 | 4 | 1 | 102-JAN-19 |
| 4 | 1 | 2 | 211-NOV-20 |
| 5 | 2 | 2 | 221-NOV-20 |
| 6 | 1 | 3 | 310-DEC-20 |
| 7 | 2 | 3 | 311-DEC-20 |
| 8 | 3 | 3 | 312-JAN-21 |
| 9 | 5 | 3 | 313-JAN-21 |
| 10 | 1 | 4 | 415-MAR-22 |
| 11 | 2 | 4 | 416-MAR-22 |
| 12 | 3 | 4 | 417-MAR-22 |
| 13 | 4 | 4 | 419-MAR-22 |
| 14 | 5 | 4 | 430-MAY-22 |
| 15 | 1 | 5 | 510-NOV-22 |
| 16 | 2 | 5 | 513-NOV-22 |
| 17 | 3 | 5 | 513-NOV-22 |
| 18 | 4 | 5 | 515-NOV-22 |
| 19 | 5 | 5 | 524-DEC-22 |



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons. Below it, a 'Worksheet' tab is active, displaying a SQL query: `SELECT * FROM CARTE_AUTOR_EDITOR;`. Below the query, there's a 'Query Result' tab showing the results of the query. The results are displayed in a table with 5 columns: ID_AUTOR, ID_EDITOR, ID_CARTE, and PRET. The table contains 11 rows of data.

| ID_AUTOR | ID_EDITOR | ID_CARTE | PRET |
|----------|-----------|----------|--------|
| 1 | 2 | 1 | 10000 |
| 2 | 1 | 2 | 15000 |
| 3 | 5 | 3 | 20000 |
| 4 | 2 | 2 | 8000 |
| 5 | 4 | 4 | 20000 |
| 6 | 6 | 5 | 16000 |
| 7 | 3 | 3 | (null) |
| 8 | 2 | 3 | (null) |
| 9 | 3 | 2 | (null) |
| 10 | 3 | 1 | (null) |
| 11 | 2 | 2 | (null) |

Cereri SQL complexe

Cerința: Să se afișeze numele editurilor care au publicat cel puțin o carte care are exact doi autori.

Pentru rezolvare se îndeplinește cerința: subcereri sincronizate în care intervin cel puțin 3 tabele.

SELECT e.num

FROM editura e

WHERE 2=ANY(SELECT COUNT(cae.id_carte)

FROM carte_editura cee JOIN carte c ON (cee.id_carte=c.id_carte)

JOIN carte_autor_editor cae ON(c.id_carte=cae.id_carte)

WHERE cee.id_editura=e.id_editura

GROUP BY(cae.id_carte)

HAVING COUNT(cae.id_carte)>=2

);

| Worksheet | Query Builder |
|-----------|---|
| 1 | <code>SELECT e.ume</code> |
| 2 | <code>FROM editura e</code> |
| 3 | <code>WHERE 2=ANY(SELECT COUNT(cae.id_carte)</code> |
| 4 | <code>FROM carte_editura cee JOIN carte c ON (cee.id_carte=c.id_carte)</code> |
| 5 | <code>JOIN carte_autor_editor cae ON(c.id_carte=cae.id_carte)</code> |
| 6 | <code>WHERE cee.id_editura=e.id_editura</code> |
| 7 | <code>GROUP BY(cae.id_carte)</code> |
| 8 | <code>HAVING COUNT(cae.id_carte)>=2</code> |
| 9 | <code>);</code> |

| Query Result |
|--------------------------------------|
| All Rows Fetched: 2 in 0.014 seconds |
| NUME |
| 1 Gollancz |
| 2 Tor Books |

Cerința: Să se afișeze numele grupurilor care au cel puțin un utilizator care are în numele contului subsecvența 'er'.

Pentru rezolvare se îndeplinesc cerințele: subcereri sincronizate în clauza FROM și utilizarea a 2 funcții pe șiruri de caractere. Se vor elimina duplicatele.

SELECT DISTINCT g.ume

FROM grup g JOIN utilizator_grup ug on (g.id_grup=ug.id_grup)

JOIN (SELECT id_utilizator

FROM utilizator

WHERE INSTR(LOWER(utilizator.ume_cont),'er')!=0) semiu on (seui.id_utilizator=ug.id_utilizator);

| Worksheet | Query Builder |
|-----------|---|
| 1 | <code>SELECT DISTINCT g.ume</code> |
| 2 | <code>FROM grup g JOIN utilizator_grup ug on (g.id_grup=ug.id_grup)</code> |
| 3 | <code>JOIN (SELECT id_utilizator</code> |
| 4 | <code>FROM utilizator</code> |
| 5 | <code>WHERE INSTR(LOWER(utilizator.ume_cont),'er')!=0) semiu on (seui.id_utilizator=ug.id_utilizator);</code> |

| Query Result |
|--------------------------------------|
| All Rows Fetched: 4 in 0.014 seconds |
| NUME |
| 1 Books for boys |
| 2 Romania pe Goodreads |
| 3 FMI |
| 4 Despre carti |

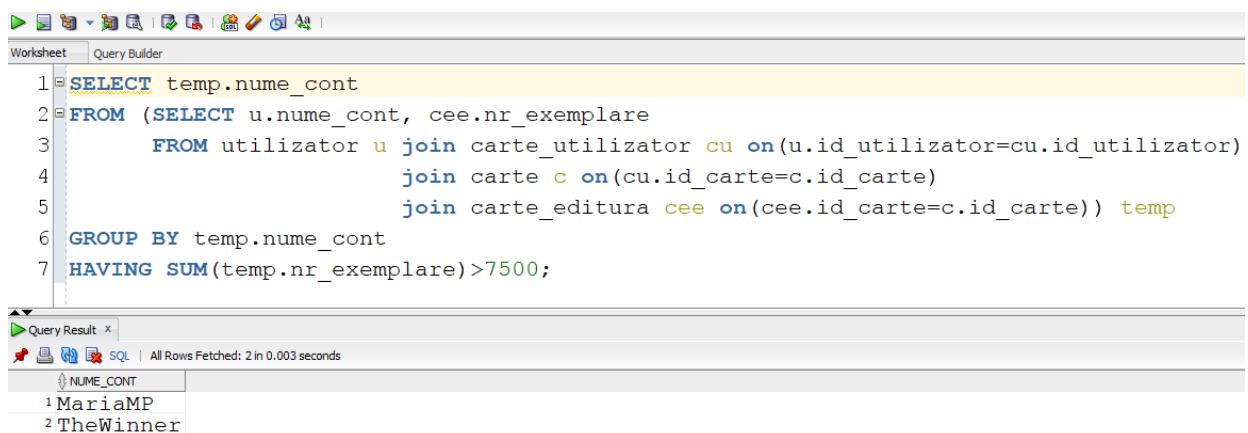
Cerința: Să se afișeze numele contului fiecărui utilizator care a citit cel puțin o carte care a avut un numar de exemplare mai mare de 7500.

Pentru rezolvare se îndeplinește cerința: grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri.


```

SELECT temp.ume_cont
FROM (SELECT u.ume_cont, cee.nr_exemplare
      FROM utilizator u join carte_utilizator cu on(u.id_utilizator=cu.id_utilizator)
           join carte c on(cu.id_carte=c.id_carte)
           join carte_editura cee on(cee.id_carte=c.id_carte)) temp
GROUP BY temp.ume_cont
HAVING SUM(temp.nr_exemplare)>7500;

```



Cerința: Să se afișeze numele și prenumele autorilor, ordonate crescător după nume și apoi după prenume, împreună cu următoarele mesaje: dacă autorul nu are nicio carte pentru care se știe prețul plătit editorului, atunci se va afișa „Autorul este misterios”, dacă există vreun preț de cel puțin 15000, se va afișa „Autorul este foarte generos”, altfel se va afișa „Autorul nu este prea generos”.

Pentru rezolvare se îndeplinesc cerințele: ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri), utilizarea a cel puțin o expresie CASE și utilizarea clauzei WITH.

```

WITH valoare_maxima AS (SELECT id_autor, MAX(NVL(pret,-1000))
pret_maxim
                        FROM carte_autor_editor
                        GROUP BY id_autor
                        )
SELECT nume, prenume, DECODE(

```

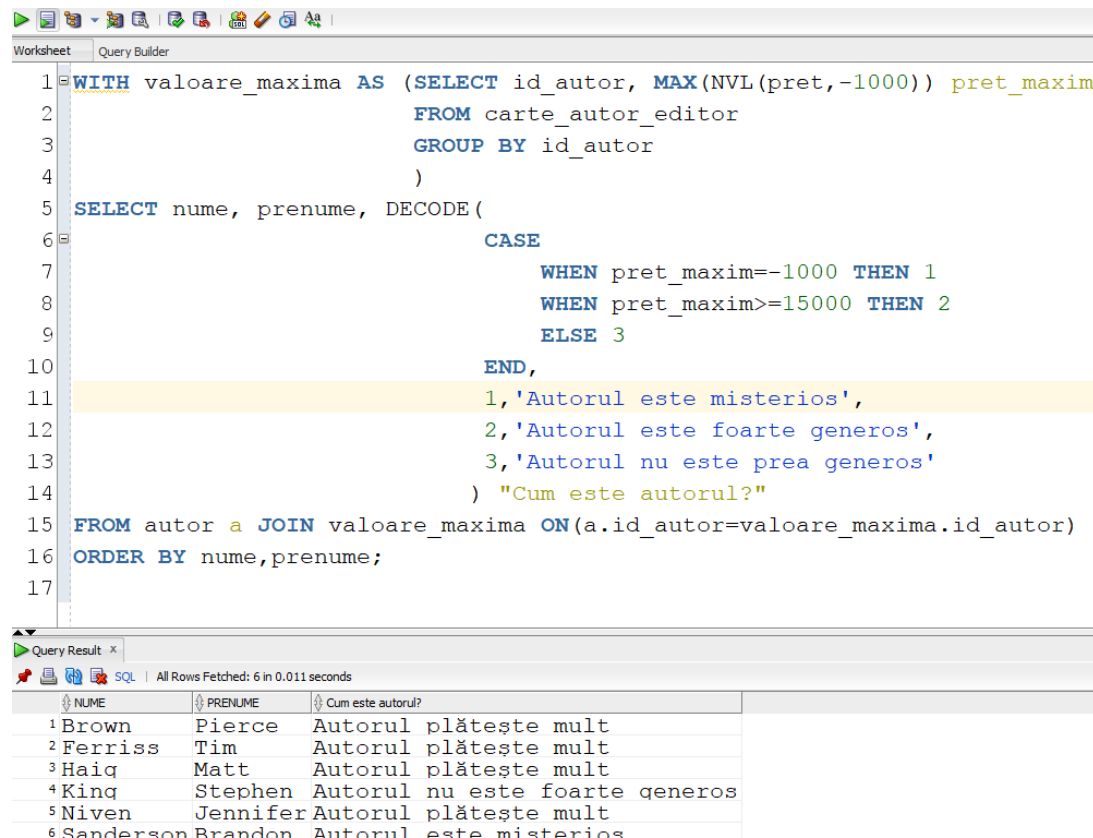
```

CASE
    WHEN pret_maxim=-1000 THEN 1
    WHEN pret_maxim>=15000 THEN 2
    ELSE 3
END,
1,'Autorul este misterios',
2,'Autorul este foarte generos',
3,'Autorul nu este prea generos'
) "Cum este autorul?"

```

FROM autor a JOIN valoare_maxima ON(a.id_autor= valoare_maxima.id_autor)

ORDER BY nume,prenume;



```

1 WITH valoare_maxima AS (SELECT id_autor, MAX(NVL(pret,-1000)) pret_maxim
2 FROM carte_autor_editor
3 GROUP BY id_autor
4 )
5 SELECT nume, prenume, DECODE(
6 CASE
7 WHEN pret_maxim=-1000 THEN 1
8 WHEN pret_maxim>=15000 THEN 2
9 ELSE 3
10 END,
11 1,'Autorul este misterios',
12 2,'Autorul este foarte generos',
13 3,'Autorul nu este prea generos'
14 ) "Cum este autorul?"
15 FROM autor a JOIN valoare_maxima ON(a.id_autor=valoare_maxima.id_autor)
16 ORDER BY nume,prenume;
17

```

| NUME | PRENUME | Cum este autorul? |
|-------------|----------|--------------------------------|
| 1 Brown | Pierce | Autorul plătește mult |
| 2 Ferriss | Tim | Autorul plătește mult |
| 3 Haiq | Matt | Autorul plătește mult |
| 4 King | Stephen | Autorul nu este foarte generos |
| 5 Niven | Jennifer | Autorul plătește mult |
| 6 Sanderson | Brandon | Autorul este misterios |

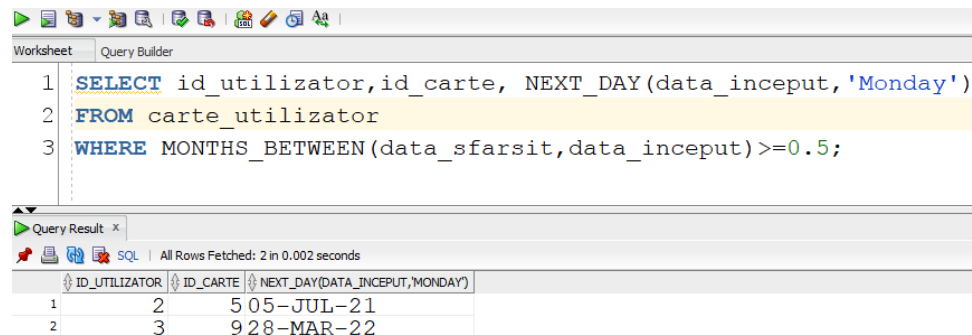
Cerința: Să se afișeze următoarea zi de luni pentru fiecare dată de început a citirii unei cărți de către un utilizator doar pentru cazurile în care diferența dintre data de început și cea de sfârșit este de cel puțin jumătate de lună.

Pentru rezolvare se îndeplinește cerința: utilizarea a cel puțin 2 funcții pe date calendaristice.

```
SELECT id_utilizator, id_carte, NEXT_DAY(data_inceput, 'Monday')
```

```
FROM carte_utilizator
```

```
WHERE MONTHS_BETWEEN(data_sfarsit, data_inceput) >= 0.5;
```



The screenshot shows a SQL query editor with the following query:

```
1 SELECT id_utilizator, id_carte, NEXT_DAY(data_inceput, 'Monday')
2 FROM carte_utilizator
3 WHERE MONTHS_BETWEEN(data_sfarsit, data_inceput) >= 0.5;
```

Below the query, the 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 2 in 0.002 seconds'.

| ID_UTILIZATOR | ID_CARTE | NEXT_DAY(DATA_INCEPUT, MONDAY) |
|---------------|----------|--------------------------------|
| 1 | 2 | 5 05-JUL-21 |
| 2 | 3 | 9 28-MAR-22 |

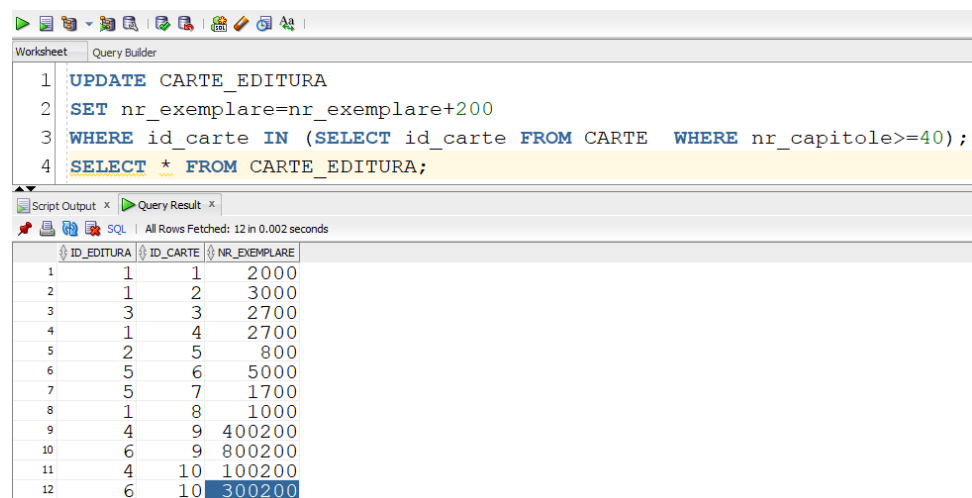
Operații de actualizare și de suprimare

Cerința: Să se actualizeze numărul de exemplare pentru cărțile care au cel puțin 40 de capitole. Pentru fiecare carte să se adauge 200 de exemplare.

```
UPDATE CARTE_EDITURA
```

```
SET nr_exemplare=nr_exemplare+200
```

```
WHERE id_carte IN (SELECT id_carte FROM CARTE WHERE nr_capitole >= 40);
```



The screenshot shows a SQL query editor with the following query:

```
1 UPDATE CARTE_EDITURA
2 SET nr_exemplare=nr_exemplare+200
3 WHERE id_carte IN (SELECT id_carte FROM CARTE WHERE nr_capitole >= 40);
4 SELECT * FROM CARTE_EDITURA;
```

Below the query, the 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 12 in 0.002 seconds'.

| ID_EDITURA | ID_CARTE | NR_EXEMPLARE |
|------------|----------|--------------|
| 1 | 1 | 2000 |
| 2 | 1 | 3000 |
| 3 | 3 | 2700 |
| 4 | 1 | 2700 |
| 5 | 2 | 800 |
| 6 | 5 | 5000 |
| 7 | 5 | 1700 |
| 8 | 1 | 1000 |
| 9 | 4 | 9 400200 |
| 10 | 6 | 9 800200 |
| 11 | 4 | 10 100200 |
| 12 | 6 | 10 300200 |

Cerința: Să se actualizeze numele și prenumele traducătorului cu id-ul egal cu 1 numai dacă a tradus o carte pentru care Stephen King este autor. Noul nume să fie „Toma”, iar prenumele „Ruxandra”.

UPDATE TRADUCATOR

SET nume='Toma', prenume='Ruxandra'

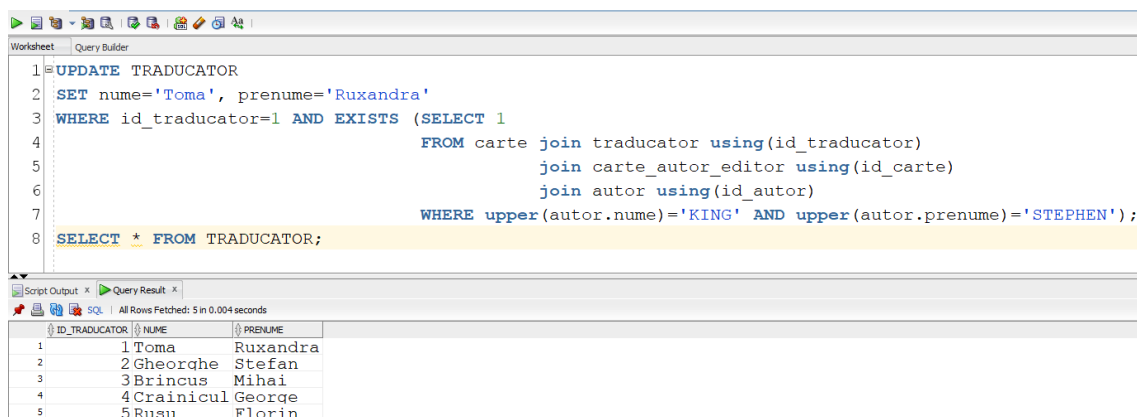
WHERE id_traducator=1 AND EXISTS (SELECT 1

FROM carte join traducator using(id_traducator)

join carte_autor_editor using(id_carte)

join autor using(id_autor)

WHERE upper(autor.nume)='KING' AND
upper(autor.prenume)='STEPHEN');



The screenshot shows a SQL query builder interface. The query is as follows:

```
1 UPDATE TRADUCATOR
2 SET nume='Toma', prenume='Ruxandra'
3 WHERE id_traducator=1 AND EXISTS (SELECT 1
4                                   FROM carte join traducator using(id_traducator)
5                                   join carte_autor_editor using(id_carte)
6                                   join autor using(id_autor)
7                                   WHERE upper(autor.nume)='KING' AND upper(autor.prenume)='STEPHEN');
8 SELECT * FROM TRADUCATOR;
```

Below the query, the results of the SELECT statement are displayed in a table:

| ID_TRADUCATOR | NUME | PRENUME |
|---------------|-----------|----------|
| 1 | Toma | Ruxandra |
| 2 | Gheorghe | Stefan |
| 3 | Brincus | Mihai |
| 4 | Crainicul | George |
| 5 | Rusu | Florin |

Cerința: Să se actualizeze prețul în tabelul CARTE_AUTOR_EDITOR. Toate prețurile mai mari sau egale ca 10000 ale autorului Stephen King vor crește cu 0.5, iar restul vor rămâne la fel.

UPDATE CARTE_AUTOR_EDITOR

SET pret=pret*1.5

WHERE pret>=10000 AND id_autor=(SELECT id_autor

FROM autor

WHERE upper(autor.nume)='KING' AND
upper(autor.prenume)='STEPHEN');

| | |
|---|---|
| 1 | UPDATE CARTE_AUTOR_EDITOR |
| 2 | SET pret=pret*1.5 |
| 3 | WHERE pret>=10000 AND id_autor=(SELECT id_autor |
| 4 | FROM autor |
| 5 | WHERE upper(autor.nume)='KING' AND upper(autor.prenume)='STEPHEN'); |
| 6 | |
| 7 | SELECT * FROM CARTE_AUTOR_EDITOR; |

| ID_AUTOR | ID_EDITOR | ID_CARTE | PRET |
|----------|-----------|----------|-----------|
| 1 | 2 | 1 | 15000 |
| 2 | 1 | 2 | 15000 |
| 3 | 5 | 3 | 20000 |
| 4 | 2 | 2 | 8000 |
| 5 | 4 | 4 | 20000 |
| 6 | 6 | 5 | 16000 |
| 7 | 3 | 3 | 7 (null) |
| 8 | 2 | 3 | 8 (null) |
| 9 | 3 | 2 | 9 (null) |
| 10 | 3 | 1 | 10 (null) |
| 11 | 2 | 2 | 10 (null) |

Cerința: Să se șteargă intrările din tabelul CARTE_UTILIZATOR pentru care data inceperii cărții este în luna decembrie și au cel puțin 400 de pagini.

```
DELETE FROM carte_utilizator
WHERE TO_CHAR(data_inceput,'MON')='DEC' AND
400< (SELECT nr_pagini
FROM carte
WHERE carte_utilizator.id_carte=carte.id_carte);
```

| | |
|---|--|
| 1 | DELETE FROM carte_utilizator |
| 2 | WHERE TO_CHAR(data_inceput, 'MON')='DEC' AND 400<(SELECT nr_pagini |
| 3 | FROM carte |
| 4 | WHERE carte_utilizator.id_carte=carte.id_carte); |
| 5 | SELECT * FROM carte_utilizator; |

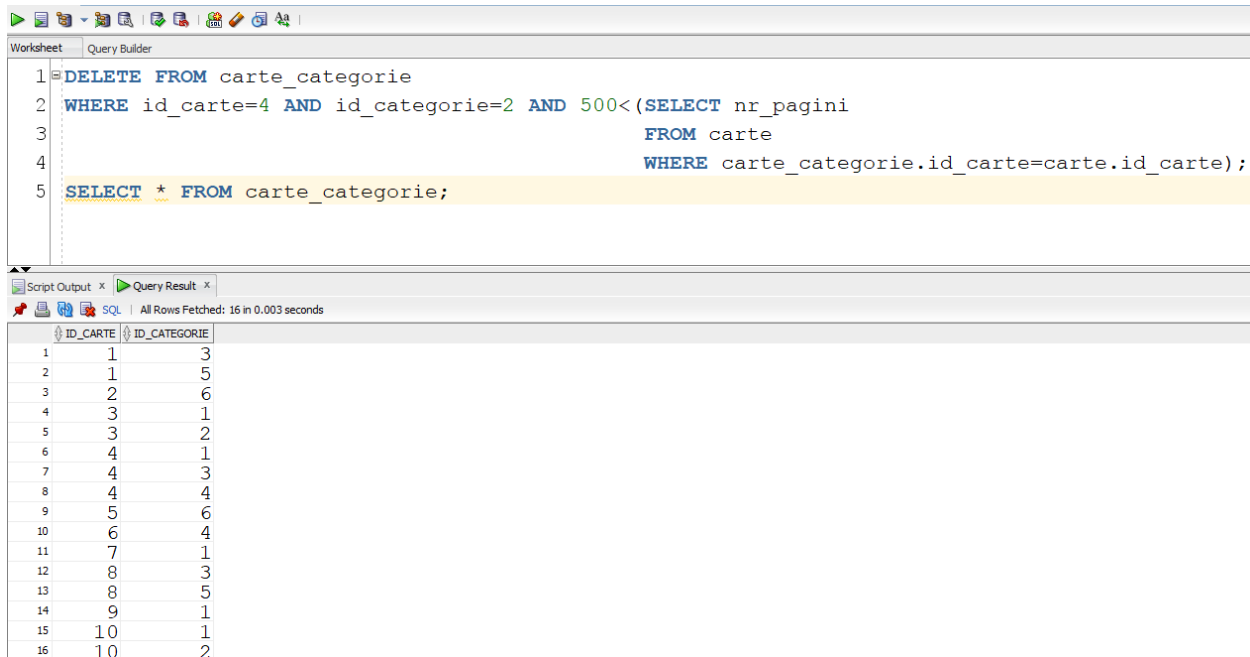
| ID_UTILIZATOR | ID_CARTE | DATA_INCEPUT | DATA_SFARSIT |
|---------------|----------|--------------|--------------|
| 1 | 1 | 1 03-MAR-20 | 07-MAR-20 |
| 2 | 1 | 2 03-JUN-19 | 08-JUN-19 |
| 3 | 1 | 7 07-NOV-22 | 17-NOV-22 |
| 4 | 2 | 4 03-MAY-20 | 17-MAY-20 |
| 5 | 2 | 5 01-JUL-21 | 08-AUG-21 |
| 6 | 3 | 9 22-MAR-22 | 07-APR-22 |
| 7 | 3 | 10 10-SEP-20 | 14-SEP-20 |
| 8 | 4 | 3 13-OCT-18 | 23-OCT-18 |
| 9 | 4 | 8 04-MAR-23 | 10-MAR-23 |
| 10 | 5 | 2 03-MAR-19 | 07-MAR-19 |

Cerința: Să se șteargă din tabelul CARTE_CATEGORIE cartea cu id-ul 4 categoria a 3-a, doar dacă aceasta are peste 500 de pagini.

```
DELETE FROM carte_categorie
WHERE id_carte=4 AND id_categorie=2 AND 500<(SELECT nr_pagini
```

FROM carte

WHERE carte_categorie.id_carte=carte.id_carte);



Worksheet Query Builder

```
1 DELETE FROM carte_categorie
2 WHERE id_carte=4 AND id_categorie=2 AND 500<(SELECT nr_pagini
3 FROM carte
4 WHERE carte_categorie.id_carte=carte.id_carte);
5 SELECT * FROM carte_categorie;
```

Script Output Query Result x All Rows Fetched: 16 in 0.003 seconds

| | ID_CARTE | ID_CATEGORIE |
|----|----------|--------------|
| 1 | 1 | 3 |
| 2 | 1 | 5 |
| 3 | 2 | 6 |
| 4 | 3 | 1 |
| 5 | 3 | 2 |
| 6 | 4 | 1 |
| 7 | 4 | 3 |
| 8 | 4 | 4 |
| 9 | 5 | 6 |
| 10 | 6 | 4 |
| 11 | 7 | 1 |
| 12 | 8 | 3 |
| 13 | 8 | 5 |
| 14 | 9 | 1 |
| 15 | 10 | 1 |
| 16 | 10 | 2 |

Cerința: Să se șteargă din tabelul CARTE_CATEGORIE cărți categoria a 4-a doar dacă ele au peste 47 de capitole.

DELETE FROM carte_categorie

WHERE id_categorie=4 AND 47<(SELECT nr_capitole

FROM carte

WHERE carte_categorie.id_carte=carte.id_carte);

| | |
|-----------|---|
| Worksheet | Query Builder |
| 1 | DELETE FROM carte_categorie |
| 2 | WHERE id_categorie=4 AND 47<(SELECT nr_capitole |
| 3 | FROM carte |
| 4 | WHERE carte_categorie.id_carte=carte.id_carte); |
| 5 | SELECT * FROM carte_categorie; |

| Script Output | Query Result |
|---|--------------|
| SQL All Rows Fetched: 15 in 0.001 seconds | |
| ID_CARTE | ID_CATEGORIE |
| 1 | 3 |
| 2 | 5 |
| 3 | 6 |
| 4 | 1 |
| 5 | 2 |
| 6 | 1 |
| 7 | 3 |
| 8 | 6 |
| 9 | 4 |
| 10 | 1 |
| 11 | 3 |
| 12 | 5 |
| 13 | 1 |
| 14 | 1 |
| 15 | 2 |

În continuare, cerințele vor avea rezultatele modificate de aceste actualizări și surprimări.

Cereri SQL outer-join, operația division și analiza top-n

Cerința: Să se afișeze numele categoriei împreună cu numele de utilizator pentru fiecare utilizator care a citit o carte din categoria respectivă. Categoriile care nu au nicio carte se vor afișa. Se vor elimina duplicatele.

Cerere care utilizează operația outer-join pe minimum 4 tabele.

SELECT DISTINCT cat.id_categorie,u.id_utilizator

FROM categorie cat LEFT JOIN carte_categorie cc

ON(cat.id_categorie=cc.id_categorie)

LEFT JOIN carte c ON(cc.id_carte=c.id_carte)

LEFT JOIN carte_utilizator cu ON(c.id_carte=cu.id_carte)

LEFT JOIN utilizator u ON(cu.id_utilizator=u.id_utilizator);

| | | |
|---|--|--|
| 1 | SELECT DISTINCT cat.id_categorie,u.id_utilizator | |
| 2 | FROM categorie cat LEFT JOIN carte_categorie cc ON(cat.id_categorie=cc.id_categorie) | |
| 3 | LEFT JOIN carte c ON(cc.id_carte=c.id_carte) | |
| 4 | LEFT JOIN carte_utilizator cu ON(c.id_carte=cu.id_carte) | |
| 5 | LEFT JOIN utilizator u ON(cu.id_utilizator=u.id_utilizator); | |

| | |
|---------------------------------------|---------------|
| Script Output | Query Result |
| All Rows Fetched: 16 in 0.004 seconds | |
| ID_CATEGORIE | ID_UTILIZATOR |
| 1 | 3 |
| 2 | 5 |
| 3 | 6 |
| 4 | 1 |
| 5 | 1 |
| 6 | 3 |
| 7 | 6 |
| 8 | 1 |
| 9 | 2 |
| 10 | 1 |
| 11 | 2 |
| 12 | 3 |
| 13 | 5 |
| 14 | 6 |
| 15 | 4 (null) |
| 16 | 7 (null) |

Cerința: Să se afișeze numele și prenumele tuturor editorilor care lucrează la cel puțin aceleași proiecte ca editorul cu id-ul 1.

Cerere care simulează operatorul DIVISION.

```

SELECT e.num, e.prenume
FROM editor e JOIN carte_ator_editor cae ON(e.id_editor=cae.id_editor)
WHERE id_ator IN (SELECT id_ator
                  FROM carte_ator_editor
                  WHERE id_editor=1) AND e.id_editor!=1
GROUP BY e.num, e.prenume, e.id_editor
HAVING COUNT(DISTINCT id_ator)=(SELECT COUNT(id_ator)
                                FROM carte_ator_editor
                                WHERE id_editor=1);

```


| | |
|-----------|---------------|
| Worksheet | Query Builder |
|-----------|---------------|

```

1 SELECT e.num, e.prenume
2 FROM editor e JOIN carte_autor_editor cae ON(e.id_editor=cae.id_editor)
3 WHERE id_autor IN (SELECT id_autor
4                     FROM carte_autor_editor
5                     WHERE id_editor=1) AND e.id_editor!=1
6 GROUP BY e.num, e.prenume, e.id_editor
7 HAVING COUNT(DISTINCT id_autor)=(SELECT COUNT(id_autor)
8                                 FROM carte_autor_editor
9                                 WHERE id_editor=1);

```

| | |
|---------------|--------------|
| Script Output | Query Result |
|---------------|--------------|

All Rows Fetched: 2 in 0.015 seconds

| | NUM | PRENUME |
|---|--------|---------|
| 1 | Adams | Purple |
| 2 | Driver | Abraham |

Cerința: Să se afișeze numele, prenumele și suma cheltuită a primilor 3 autori din topul autorilor care au plătit cel mai mult un editor pentru editarea unei cărți.

Cerere pentru analiza top-n.

SELECT a.num, a.prenume, top.suma, rownum

FROM autor a join(SELECT id_autor, NVL(pret,0) suma

FROM carte_autor_editor

ORDER BY 2 DESC

)top on(a.id_autor=top.id_autor)

WHERE rownum<=3

ORDER BY suma DESC;

| | |
|-----------|---------------|
| Worksheet | Query Builder |
|-----------|---------------|

```

1 SELECT a.num, a.prenume, top.suma, rownum
2 FROM autor a join(SELECT id_autor, NVL(pret,0) suma
3                   FROM carte_autor_editor
4                   ORDER BY 2 DESC
5                   )top on(a.id_autor=top.id_autor)
6 WHERE rownum<=3
7 ORDER BY suma DESC;

```

| | |
|---------------|--------------|
| Script Output | Query Result |
|---------------|--------------|

All Rows Fetched: 3 in 0.001 seconds

| | NUM | PRENUME | SUMA | ROWNUM |
|---|---------|----------|-------|--------|
| 1 | Ferriss | Tim | 20000 | 1 |
| 2 | Brown | Pierce | 20000 | 2 |
| 3 | Niven | Jennifer | 16000 | 3 |

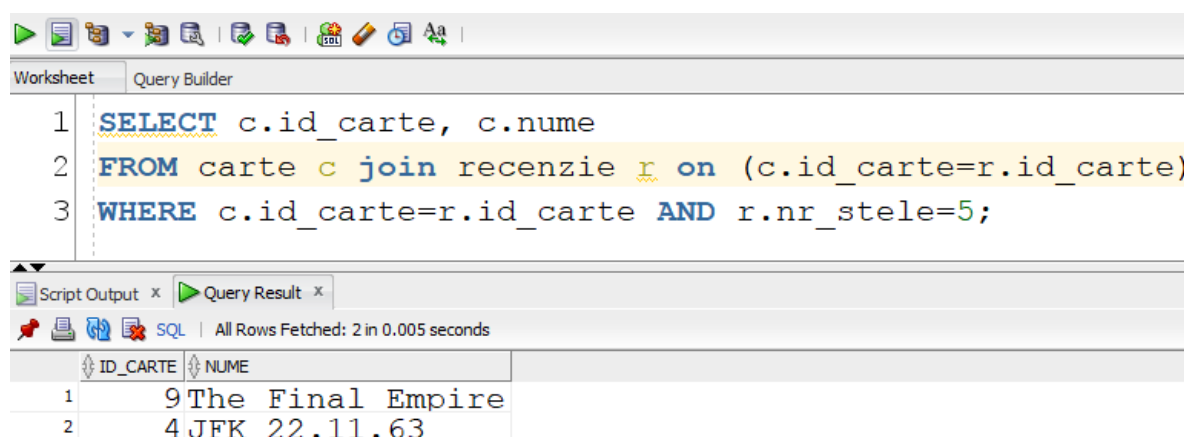
Optimizarea unei cereri

Cerința: Să se afișeze numele și id-ul cărților care au primit cel puțin o recenzie cu rating-ul de 5 stele. Nu se vor elimina duplicatele.

Cererea neoptimizată

Cererea SQL:

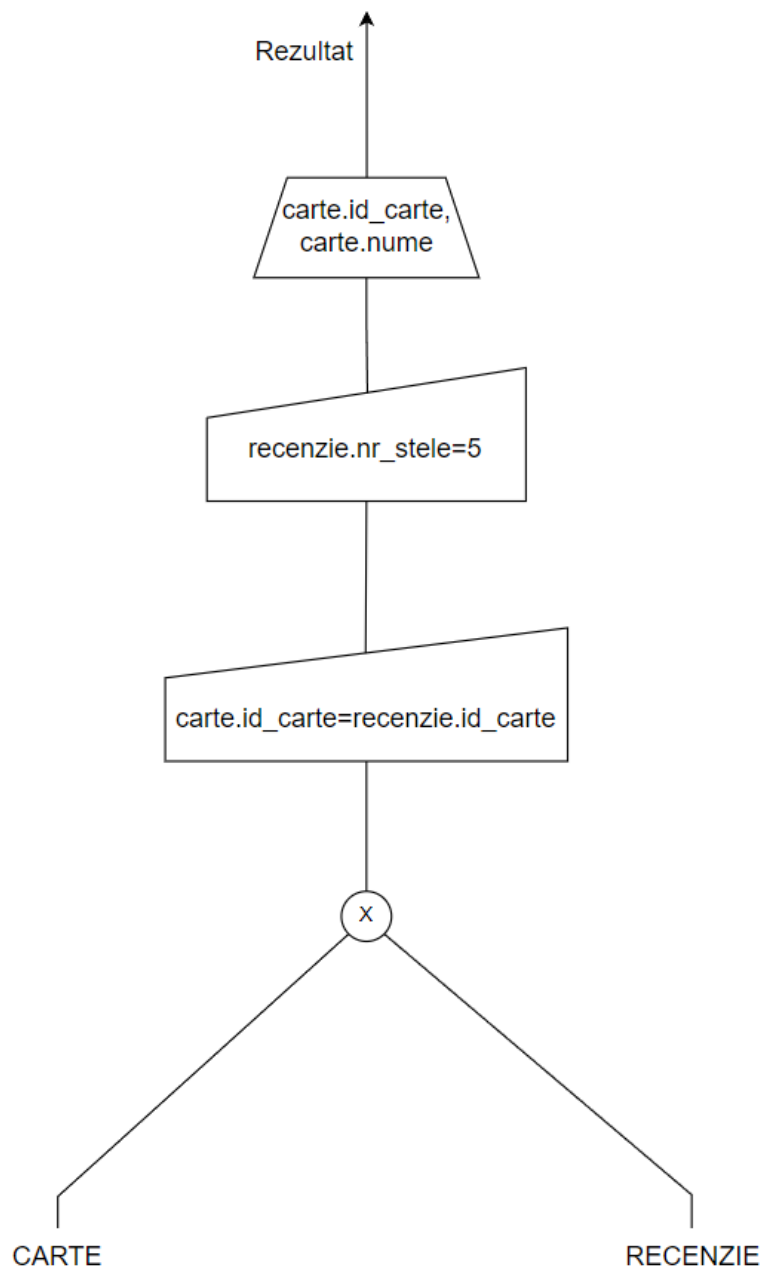
```
SELECT c.id_carte, c.num  
FROM carte c join recenzie r on (c.id_carte=r.id_carte)  
WHERE c.id_carte=r.id_carte AND r.nr_stele=5;
```



Expresie algebrică:

```
R1 = PRODUCT(CARTE, RECENZIE);  
R2 = SELECT(R1, carte.id_carte=recenzie.id_carte);  
R3 = SELECT(R2, recenzie.nr_stele=5);  
Rezultat = PROJECT(R3, carte.id_carte, carte.num);
```

Albore algebric:



Cererea optimizată

Cererea SQL:

```
SELECT c.id_carte, c.nume
FROM (SELECT id_carte, nume FROM carte) c
      join (SELECT id_carte, rating FROM recenzie
            WHERE nr_stele=5) r on (c.id_carte=r.id_carte);
```

| | |
|-----------|---|
| Worksheet | Query Builder |
| 1 | SELECT c.id_carte, c.num |
| 2 | FROM (SELECT id_carte, nume FROM carte) c join (SELECT id_carte, nr_stele |
| 3 | FROM recenzie |
| 4 | WHERE nr_stele=5) r on (c.id_carte=r.id_carte); |

| | |
|--------------------------------------|-------------------|
| Script Output x | Query Result x |
| All Rows Fetched: 2 in 0.006 seconds | |
| ID_CARTE | NUME |
| 1 | 9The Final Empire |
| 2 | 4JFK 22.11.63 |

Expresie algebrică:

$R1 = \text{PROJECT}(\text{CARTE}, \text{id_carte}, \text{nume});$

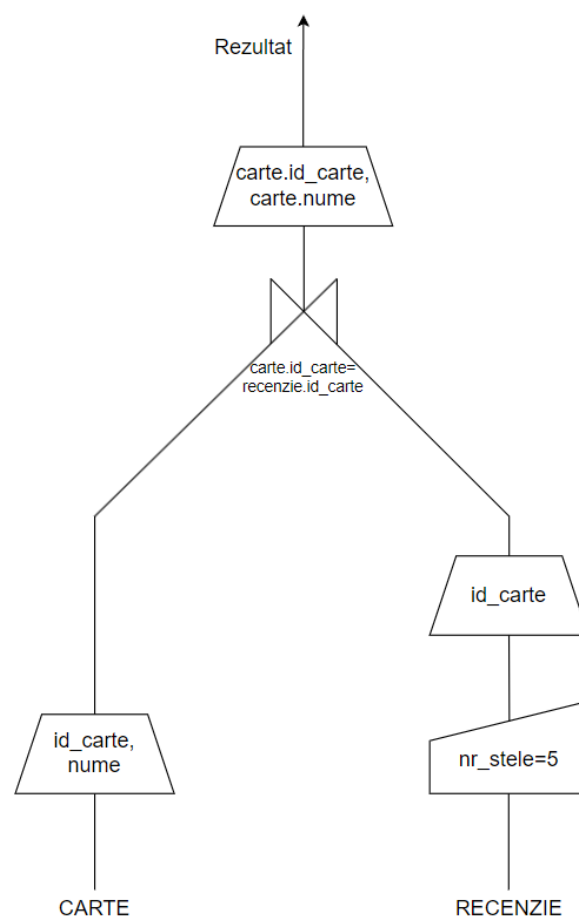
$R2 = \text{SELECT}(\text{RECENZIE}, \text{nr_stele}=5);$

$R3 = \text{PROJECT}(\text{RECENZIE}, \text{id_carte});$

$R4 = \text{JOIN}(R1, R3, \text{carte.id_carte}=\text{recenzie.id_carte});$

$\text{Rezultat} = \text{PROJECT}(R4, \text{carte.id_carte}, \text{carte.nume});$

Arbore algebric:



Explicație optimizare:

Pentru a eficientiza cererea trebuie să utilizăm mai multe reguli de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Dorim ca în momentul în care avem de făcut o operație JOIN să avem de combinat cât mai puține date. De aceea dorim să facem selecturile prima dată, reducând numărul de rânduri luate în considerare. După care facem proiecțiile ca să păstrăm doar coloanele care ne interesează la finalul cererii. În momentul în care trebuie să combinăm datele proiectate din cele 2 tabele este suficient să facem un JOIN, deoarece nu este nevoie să facem un produs cartezian.

Realizarea normalizării până la forma normală 5

BCNF

O relație R este în forma normală Boyce-Codd dacă și numai dacă fiecare determinant este o cheie candidat. O relație nu trebuie să fie în FN3 pentru a fi în BCNF. Cu toate acestea, BCNF este o versiune puțin mai restrictivă a lui FN3.

Exemplu:

CARTE(Non-BCNF)

| id_carte# | nume | isbn |
|-----------|--------------|---------------|
| 1 | Misery | 1042312312321 |
| 2 | JFK 22.11.63 | 1042312312322 |
| 3 | Mr.Mercedes | 1042312312323 |

Între attributele relației există dependențele:

$\{id_carte\} \Rightarrow \{nume, isbn\}$

$\{isbn\} \Rightarrow \{id_carte\}$

Deci isbn este o cheie candidat. Se aplică regula Casey-Delobel și se aduce relația în BCNF.

CARTE_A(isbn#,id_carte)

CARTE_B(id_carte#,nume)

Transformăm în BCNF:

CARTE_A

| isbn# | id_carte# |
|---------------|-----------|
| 1042312312321 | 1 |

| | |
|---------------|---|
| 1042312312322 | 2 |
| 1042312312323 | 3 |

CARTEA_B

| id_carte# | nume |
|-----------|--------------|
| 1 | Misery |
| 2 | JFK 22.11.63 |
| 3 | Mr.Mercedes |

FN4

FN4 elimină redundanțele datorate relațiilor m:n, adică datorate dependenței multiple. O relație R este în a patra formă normală dacă și numai dacă relația este în BCNF și nu conține relații m:n independente.

Exemplu:

CARTE_EDITURA(Non-FN4)

| id_editura# | id_carte# | id_tip_coperta# |
|-------------|-----------|-----------------|
| 1 | 1 | 1 |
| 1 | 2 | 1 |
| 2 | 1 | 2 |

Se poate observa că atributul id_carte creează multidependențele:

$id_carte\# \Rightarrow id_editura$

$id_carte\# \Rightarrow id_tip_coperta$

Relația CARTE_EDITURA este în BCNF, o putem aduce în FN4 împărțind-o în următoarele două relații:

CARTE_EDITURA_A(id_editura#,id_carte#)

CARTE_EDITURA_B(id_carte#,id_tip_coperta#)

Transformare în FN4:

CARTE_EDITURA_A(FN4)

| id_editura# | id_carte# |
|-------------|-----------|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |

CARTE_EDITURA_B(FN4)

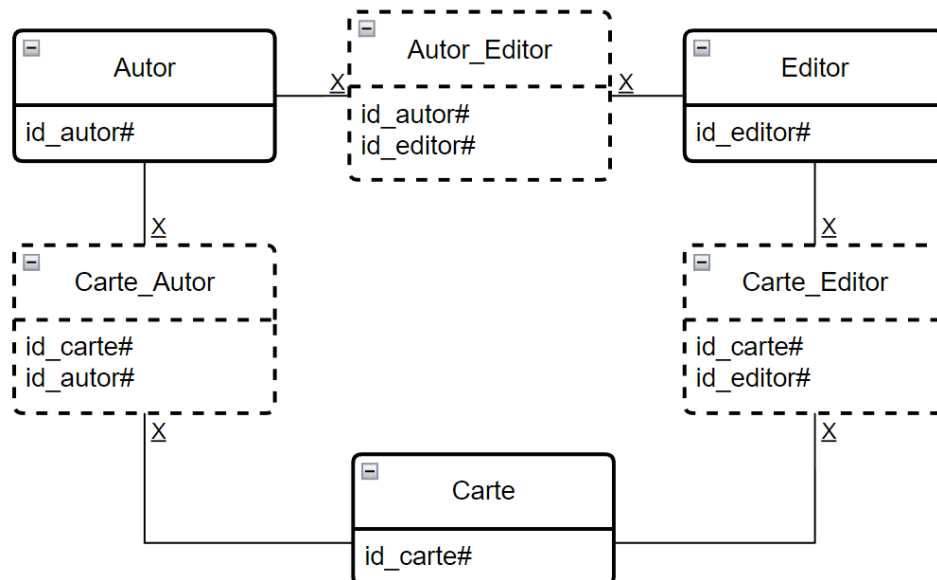
| id_carte# | id_tip_coperta# |
|-----------|-----------------|
| 1 | 1 |
| 2 | 1 |
| 1 | 2 |

FN5

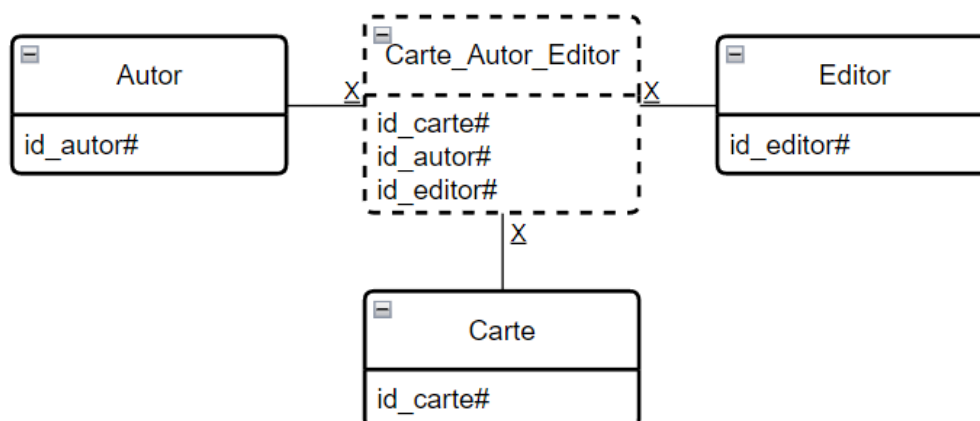
FN5 își propune eliminarea redundanțelor care apar în relațiile m:n dependente. O relație R este în a cincea formă normală dacă și numai dacă este în forma normală patru și nu conține dependențe ciclice.

Exemplu:

Diagramă non-FN5:



Diagramă FN5:



Prima diagramă conține 3 relații de tip 2 ciclice, deci nu este în FN5. A doua diagramă rezolvă această problemă prin crearea unui singur tabel asociativ pentru o relație de tip 3.

Denormalizarea

În urma aplicării tuturor formelor normale menționate mai sus putem ajunge să avem un număr mare de tabele. Astfel, numărul de operații JOIN pe care va trebui să le aplicăm pentru a afla informațiile de care avem nevoie este unul ridicat. Un exemplu de situație în care denormalizarea ar fi utilă este următoarea:

CARTE_EDITURA

| id_editura# | id_carte# | id_nr_exemplare |
|-------------|-----------|-----------------|
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 3 | 2 |

NR_EXEMPLARE

| id_nr_exemplare | nr_exemplare |
|-----------------|--------------|
| 1 | 1000 |
| 2 | 123200 |

Practic, păstrarea entității NR_EXEMPLARE nu ne ajută. Pierdem din punct de vedere al timpului din cauza operației de JOIN, dar și din punct de vedere al memoriei, pentru că în loc să păstrăm o simplă coloană în entitatea CARTE_EDITURA, noi păstrăm pe lângă respectiva coloană o nouă entitate cu două coloane. Deci, este mai inefficient din ambele puncte de vedere.

După aplicarea denormalizării obținem:

CARTE_EDITURA

| id_editura# | id_carte# | id_nr_exemplare |
|-------------|-----------|-----------------|
| 1 | 1 | 1000 |
| 2 | 2 | 1000 |
| 3 | 3 | 123200 |