

BAZĂ DE DATE PENTRU GESTIUNEA UNEI BIBLIOTECI

ROGOZA RALUCA-IOANA

GRUPA 134

ANUL 1

CUPRINS

1. Descrierea modelului real, a utilității și a regulilor de funcționare
2. Prezentarea constrângerilor impuse asupra modelului
3. Descrierea entităților
4. Descrierea relațiilor
5. Descrierea atributelor
6. Diagrama entitate-relație
7. Diagrama conceptuală
8. Enumerarea schemelor relaționale
9. Realizarea normalizărilor până la FN3
10. Crearea secvenței
11. Crearea tabelelor în SQL
12. Cereri SQL
13. Operații de actualizare și suprimare a datelor
14. Cereri SQL: outer-join și division
15. Optimizarea cererii
16. Realizarea normalizării BCNF, FN4, FN5
17. Denormalizarea

1.DESCRIEREA MODELULUI REAL, A UTILITĂȚII ACESTUIA ȘI A REGULILOR DE FUNCȚIONARE

Acesta este o bază de date pentru o bibliotecă. Ea este utilă, deoarece permite stocarea informațiilor despre bibliotecari, cărți și copiile lor, autorii cărților, editurile de la care se găsesc publicații în bibliotecă și genurile în care se integrează cărțile, cititori, abonamentele posibile pe care le pot deține cititorii, cărțile împrumutate și toate tipurile de penalizări pe care le pot obține cititorii.

Biblioteca permite tuturor oamenilor să împrumute cărți. Bibliotecarii au grijă de cărțile care sunt variate și aparțin diferitor domenii. Cărțile pot fi unicate sau pot avea mai multe copii. În bibliotecă se găsesc cărți de la mai multe edituri. Cititorii își pot face un abonament care are un preț pe lună diferit în funcție de tipul lui. Cu timpul, tipul de abonament al cititorului poate să difere față de cele pe care le-a avut deja, dar cititorul nu poate să dețină mai multe abonamente în același timp. Cititorii pot primi penalizări pentru nerespectarea regulilor impuse de bibliotecari.

2. PREZENTAREA CONSTRÂNGERILOR IMPUSE ASUPRA MODELULUI

- Într-o bibliotecă există mai mulți bibliotecari și, prin urmare, mai mulți bibliotecari pot avea grijă de cărți, iar de o carte pot avea grijă mai mulți bibliotecari. Trebuie ca măcar un bibliotecar să aibă grijă de o carte, iar de o carte trebuie să aibă grijă măcar un bibliotecar.
- O carte poate avea unul sau mai mulți autori, iar fiecare autor poate să fi scris una sau mai multe cărți. Trebuie ca fiecare autor să fi scris măcar o carte, iar fiecare carte trebuie să fie scrisă de cel puțin un autor.
- O carte din bibliotecă poate să aparțină mai multor genuri și mai multe cărți pot avea același gen. Poate exista un gen de care să nu aparțină nicio carte din bibliotecă, dar o carte trebuie să se încadreze măcar într-un gen.
- O carte poate fi publicată de o singură editură, iar la o editură pot fi publicate mai multe cărți. O carte trebuie să fie publicată de o singură editură, iar o editură trebuie să publice măcar o carte.
- O carte poate să aibă mai multe copii, dar o copie poate să aparțină unei singure cărți. O carte trebuie să aibă cel puțin o copie(cartea însăși), iar o copie trebuie să aparțină unei singure cărți.
- O copie de carte se poate găsi pe un anumit raft, iar pe un raft se pot afla mai multe copii ale unei cărți. O copie de carte trebuie să se găsească pe un singur raft, iar pe un raft trebuie să se găsească cel puțin o copie de carte.
- Un cititor poate deține unul sau mai multe abonamente în funcție de data când îl achiziționează și numărul de luni cât va fi disponibil, iar un abonament identic poate fi deținut de unul sau mai mulți cititori. Pot exista abonamente care să nu fie deținute de niciun cititor, dar un cititor trebuie să dețină cel puțin un abonament.
- Un cititor poate obține una sau mai multe penalizări dacă nu respectă regulile bibliotecii, iar aceeași penalizare poate fi primită de mai mulți cititori. Un cititor nu trebuie neapărat să primească o penalizare, dar o penalizare trebuie să fie primită de cel puțin un cititor.
- Un cititor poate împrumuta una sau mai multe copii de cărți de la mai mulți autori. Un cititor trebuie să împrumute cel puțin o copie de carte scrisă de cel puțin un autor.

3. DESCRIEREA ENTITĂȚILOR

BIBLIOTECAR

Aici se memorează toți bibliotecarii și datele lor, ei fiind identificați după un cheia primară id_bibliotecar.

CARTE

Aici se memorează informații despre toate cărțile din bibliotecă, ele fiind identificate după cheia primară id_carte, copiile unei cărți regăsindu-se la același id_carte.

COPIE_CARTE

Aici se memorează pentru fiecare carte toate copiile ei, iar ele sunt identificate după cheia primară compusă formată din id_copie și id_carte, deoarece pentru fiecare carte există un anumit număr de copii și pot fi identificare în mod unic doar împreună.

CITITOR

Aici sunt memorate informațiile despre toți cititorii care împrumută cărți de la bibliotecă și sunt identificați după cheia primară id_cititor.

AUTOR

Aici sunt memorate informații despre toți autorii ale căror cărți se găsesc în bibliotecă și sunt identificați după cheia primară id_autor.

ABONAMENT

Aici sunt memorate informații despre toate abonamentele pe care și le poate face un client și sunt identificate după cheia primară id_abonament.

PENALIZARE

Aici sunt memorate informații despre penalizările pe care le pot primi cititorii din varii motive, cum ar fi: deteriorarea cărților, restituirea după mai mult timp decât ar fi trebuit. Ele sunt identificate după cheia primară id_penalizare care reprezintă numărul penalizării.

GEN

Aici sunt memorate informații despre toate genurile de cărți disponibile în bibliotecă, iar ele sunt identificate după cheia primară `id_gen`.

EDITURA

Aici sunt memorate informații despre toate editurile de la care se găsesc cărți în bibliotecă, iar ele sunt identificate după cheia primară `id_editura`.

RAFT

Aici sunt memorate informații despre rafturile din bibliotecă, iar ele sunt identificate după cheia primară `id_raft`.

4. DESCRIEREA RELAȚIILOR

BIBLIOTECAR are_grija_de CARTE

Relația are cardinalitatea maximă many-many (n:m).

Relația are cardinalitatea minimă one-one (1:1).

CARTE scrisa_de AUTOR

Relația are cardinalitatea maximă many-many (n:m).

Relația are cardinalitatea minimă one-one (1:1).

CARTE apartine GEN

Relația are cardinalitatea maximă many-many (n:m).

Relația are cardinalitatea minimă one-zero (1:0).

CARTE publicata_de EDITURA

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-one (1:1).

CARTE are COPIE_CARTE

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-one (1:1).

COPIE_CARTE se_gaseste_pe RAFT

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-one (1:1).

CITITOR detine ABONAMENT

Relația are cardinalitatea maximă many-many (n:m).

Relația are cardinalitatea minimă one-zero (1:0).

CITITOR primește PENALIZARE

Relația are cardinalitatea maximă many-many (n:m).

Relația are cardinalitatea minimă one-zero (1:0).

COPIE_CARTE

CITITOR

carte_imprumutata

AUTOR

Relația are cardinalitatea maximă many-many-many (n:m:n).

Relația are cardinalitatea minimă one-one-one (1:1:1).

5. DESCRIEREA ATRIBUTELOR

5.1. DESCRIEREA ATRIBUTELOR ENTITĂȚILOR

BIBLIOTECAR

- id_bibliotecar
 - un număr unic în funcție de care este identificat fiecare bibliotecar
 - tip de date: number
 - nu poate fi null
- nume
 - numele bibliotecarului
 - tip de date: varchar
 - nu poate fi null
- prenume
 - prenumele bibliotecarului
 - tip de date: varchar
 - nu poate fi null

- varsta
 - varsta bibliotecarului
 - tip de date: number
 - nu poate fi null
- salariu
 - salariul bibliotecarului
 - tip de date: number
 - nu poate fi null
- numar_etaj
 - numărul etajului la care lucrează
 - tip de date: number
 - nu poate fi null

CARTE

- id_carte
 - un număr unic în funcție de care este identificată fiecare carte
 - tip de date: number
 - nu poate fi null
- denumire
 - denumirea cărții
 - tip de date: varchar
 - nu poate fi null
- an_publicatie
 - anul în care a fost publicată cartea
 - tip de date: number
 - nu poate fi null
- numar_pagini
 - numărul de pagini ale cărții
 - tip de date: number
 - nu poate fi null
- numar_copii
 - numărul de copii al fiecărei cărți
 - tip de date: number
 - default: 1
- id_editura
 - id-ul editurii care a publicat cartea
 - este cheie externă(referențiază tabela EDITURA)
 - tip de date: number

COPIE_CARTE

- id_copie
 - a câta copie a unei cărți este
 - tip de date: number
 - nu poate fi null
- id_carte
 - numărul în funcție de care este identificată unic o carte
 - tip de date: number
 - nu poate fi null
- id_raft
 - numărul raftului pe care se găsește cartea
 - este cheie externă(referențiază tabela RAFT)
 - tip de date: number

RAFT

- id_raft
 - al câtelea raft este
 - tip de date: number
 - nu poate fi null
- etaj
 - etajul la care se află raftul
 - tip de date: number
 - nu poate fi null

GEN

- id_gen
 - un număr unic în funcție de care este identificat fiecare gen de carte
 - tip de date: number
 - nu poate fi null
- denumire
 - denumirea genului
 - tip de date: varchar
 - nu poate fi null

EDITURA

- id_editura

- un număr unic în funcție de care este identificată fiecare editură
 - tip de date: number
 - nu poate fi null
- denumire
 - denumirea editurii
 - tip de date: varchar
 - nu poate fi null
- locatie
 - adresa editurii
 - tip de date: varchar
 - nu poate fi null

AUTOR

- id_autor
 - un număr unic în funcție de care este identificat fiecare autor
 - tip de date: number
 - nu poate fi null
- nume
 - numele autorului
 - tip de date: varchar
 - nu poate fi null
- prenume
 - prenumele autorului
 - tip de date: varchar
 - nu poate fi null
- varsta
 - varsta autorului
 - tip de date: number
 - dacă e null înseamnă că autorul nu mai este în viață

ABONAMENT

- id_abonament
 - un număr unic în funcție de care este identificat fiecare abonament
 - tip de date: number
 - nu poate fi null
- pret_pe_luna
 - pretul abonamentului pe o luna

- tip de date: number
- nu poate fi null
- numar_carti_de_imprumutat
 - numarul de carti pe care le poate împrumuta un cititor care deține abonamentul respectiv
 - tip de date: number
 - nu poate fi null
- tip
 - tipul abonamentului care poate fi: elev, student, angajat, pensionar
 - tip de date: varchar
 - nu poate fi null

CITITOR

- id_cititor
 - un număr unic în funcție de care este identificat fiecare cititor
 - tip de date: number
 - nu poate fi null
- nume
 - numele cititorului
 - tip de date: varchar
 - nu poate fi null
- prenume
 - prenumele cititorului
 - tip de date: varchar
 - nu poate fi null
- varsta
 - varsta cititorului
 - tip de date: number
 - nu poate fi null
- telefon
 - numarul de telefon al cititorului
 - tip de date: varchar
 - nu poate fi null

PENALIZARE

- id_penalizare
 - un număr unic în funcție de care este identificată fiecare penalizare

- tip de date: number
 - nu poate fi null
- motiv
 - motivul pentru care se aplică penalizarea
 - tip de date: varchar
 - poate fi null
- pret
 - pretul platit pentru respectiva penalizare
 - tip de date: number
 - default: 0

5.2. DESCRIEREA ATRIBUTELOR RELAȚIILOR

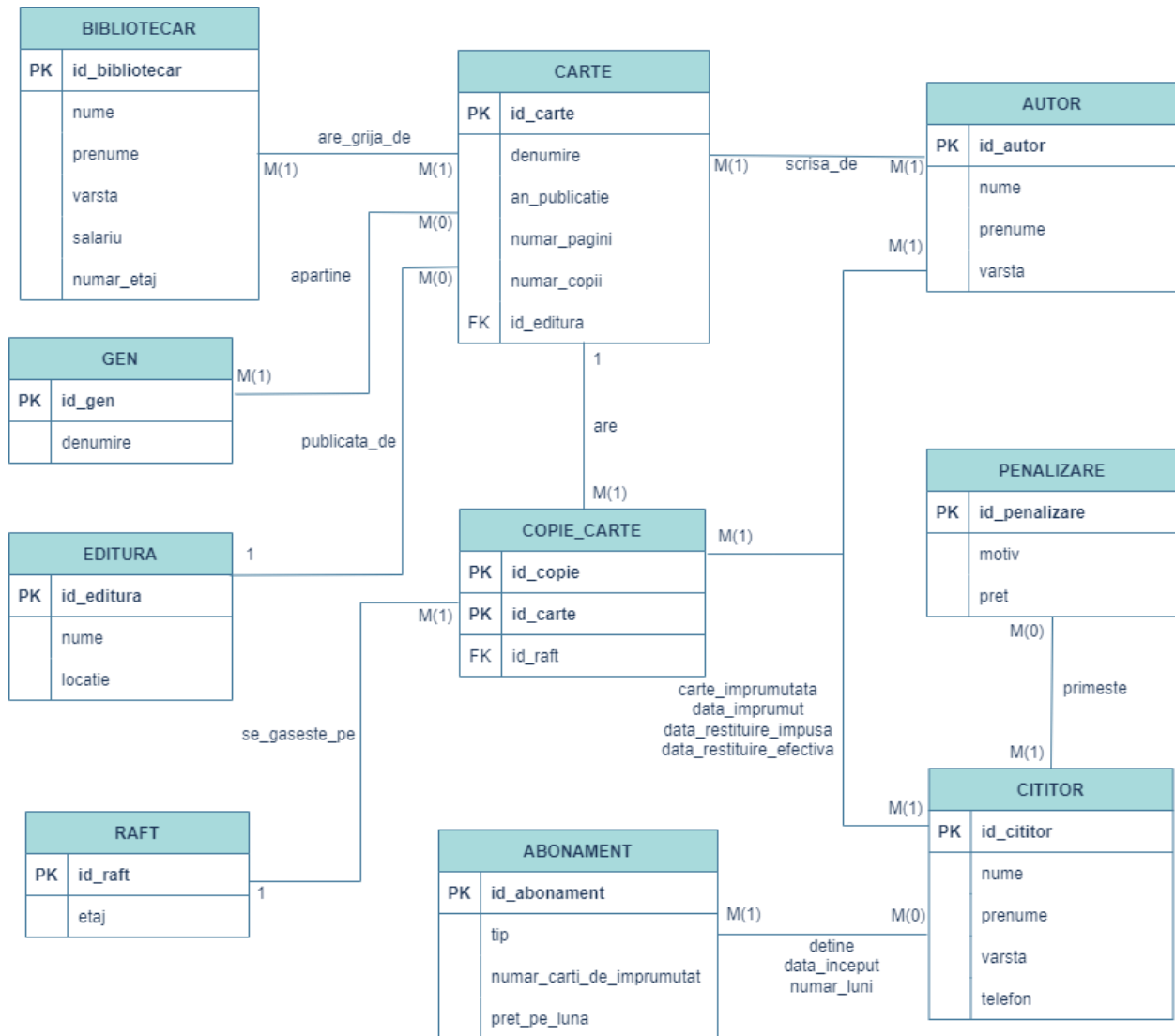
DETINE

- data_inceput
 - data la care un cititor și-a făcut abonamentul
 - tip de date: date
 - nu poate fi null
- numar_luni
 - numărul de luni pentru care cititorul își face un abonament
 - tip de date: date
 - nu poate fi null

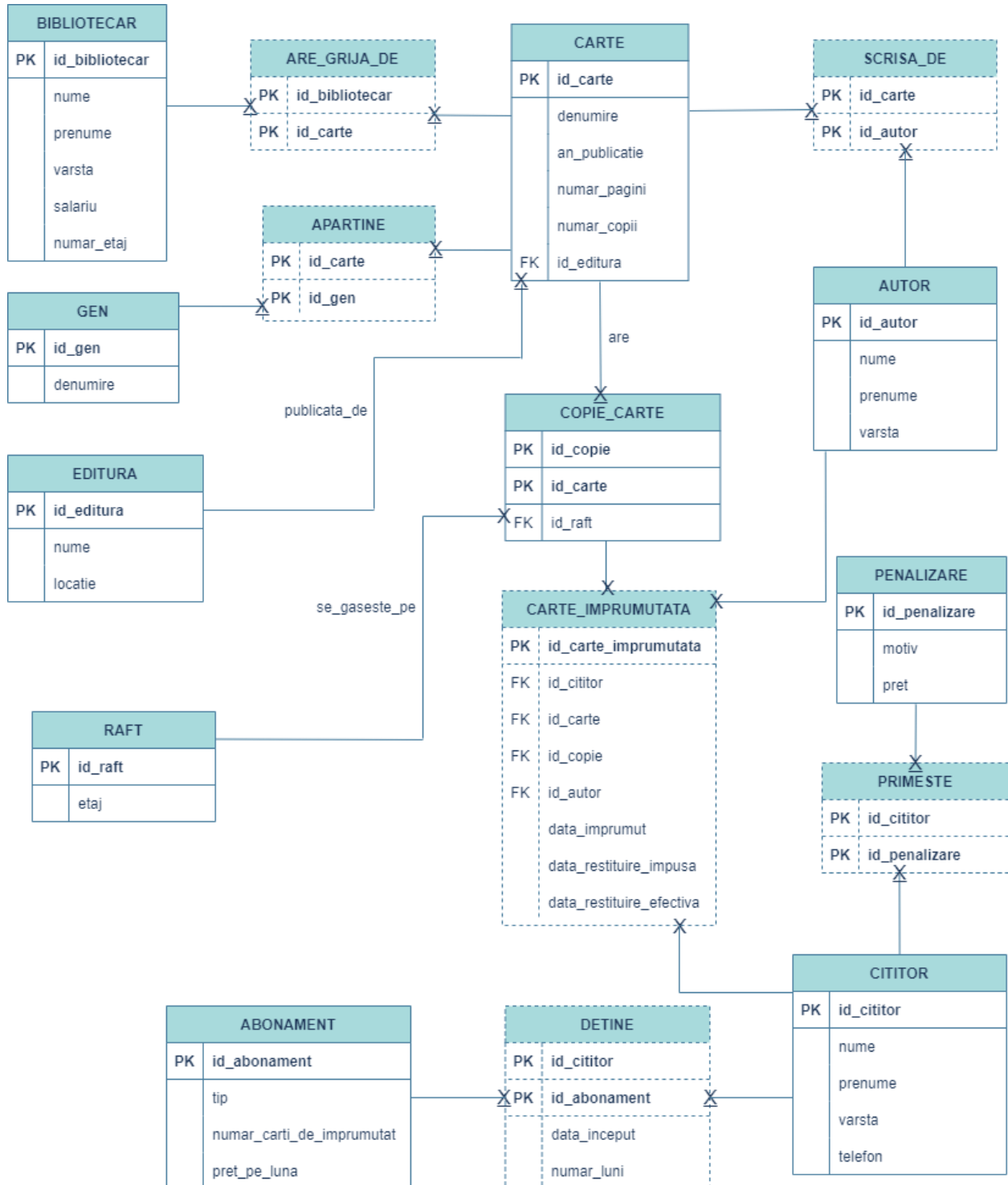
CARTE_IMPRUMUTATA

- data_imprumut
 - data la care s-a realizat împrumutul unei anumite cărți de către un anumit cititor
 - tip de date: date
 - nu poate fi null
- data_restituire_impusa
 - data la care cititorul care a împrumutat cartea ar trebui să o returneze
 - tip de date: date
 - nu poate fi null
- data_restituire_efectiva
 - data la care cititorul returnează cartea
 - tip de date: date
 - poate fi null (înseamnă că nu a fost returnată cartea)

6. DIAGRAMA ENTITATE-RELAȚIE



7. DIAGRAMA CONCEPTUALĂ



8. ENUMERAREA SCHEMELOR RELAȚIONALE

BIBLIOTECAR(id_bibliotecar#, nume, prenume, varsta, salariu, numar_etaj)

ARE_GRIJA_DE(id_bibliotecar#, id_carte#)

CARTE(id_carte#, denumire, an_publicatie, numar_copii, id_editura)

AUTOR(id_autor#, nume, prenume, varsta)

SCRISA_DE(id_carte#, id_autor#)

GEN(id_gen#, denumire)

APARTINE(id_carte#, id_gen#)

EDITURA(id_editura#, nume, locatie)

COPIE_CARTE(id_copie#, id_carte#, id_raft)

RAFT(id_raft#, etaj)

ABONAMENT(id_abonament#, tip, numar_carti_de_imprumutat, pret)

CITITOR(id_cititor#, nume, prenume, varsta, telefon)

DETINE(id_cititor#, id_abonament#, data_inceput, data_sfarsit)

PENALIZARE(id_penalizare#, motiv, pret)

PRIMESTE(id_cititor#, id_penalizare#)

CARTE_IMPRUMUTATA(id_cititor#, id_carte#, id_copie#, id_autor#, data_imprumut, data_restituire_impusa, data_restituire_efectiva)

9. FORMELE NORMALE FN1, FN2, FN3

FN1

Modelul proiectat se află în forma normală 1.

Exemplu de relație care nu se află în forma normală 1:

id_editura#	id_carte
E1	C1, C4, C8
E2	C3, C5
E3	C6

Aducere în FN1:

id_editura#	id_carte#
E1	C1
E1	C4
E1	C8
E2	C3
E2	C5
E3	C6

FN2

Modelul proiectat se află în forma normală 2.

Exemplu de relație care nu se află în forma normală 2:

CARTE_COPIE (id_carte#, id_copie#, denumire, an_publicatie, numar_pagini, numar_copii)

CARTE_COPIE

id_carte#	id_copie#	denumire	an_publicatie	numar_pagini	numar_copii
1	2	Ion	1920	256	5
2	1	Hotul	2003	532	3
2	3	Hotul	2003	532	3
3	1	Baltagul	1930	123	10

id_carte -> denumire, an_publicatie, numar_pagini, numar_copii

Acestea trebuie despărțite în:

CARTE (id_carte#, denumire, an_publicatie, numar_pagini, numar_copii)

COPIE_CARTE (id_copie#, id_carte#)

CARTE

id_carte#	denumire	an_publicatie	numar_pagini	numar_copii
1	Ion	1920	256	5
2	Hotul	2003	532	3
3	Baltagul	1930	123	10

COPIE_CARTE

id_copie#	id_carte#
1	2
1	3
2	1
3	2

FN3

Modelul proiectat se află în forma normală 3.

Exemplu de relație care nu se află în forma normală 3:

CITITOR_ABONAMENT (id_cititor#, nume, prenume, id_abonament, pret_pe_luna)

id_cititor -> nume, prenume, id_abonament

id_abonament -> pret_pe_luna

CITITOR_ABONAMENT

id_cititor#	nume	prenume	id_abonament	pret_pe_luna
1	Rogoza	Raluca	1	20
2	Stoica	Elena	2	35
3	Cojoc	Georgiana	1	20

CITITOR (id_cititor#, nume, prenume, id_abonament)

ABONAMENT (id_abonament#, pret_pe_luna)

CITITOR

id_cititor	nume	prenume	id_abonament
1	Rogoza	Raluca	1
2	Stoica	Elena	2
3	Cojoc	Georgiana	1

ABONAMENT

id_abonament	pret_pe_luna
1	20
2	35

10. CREAREA UNEI SECVENȚE PENTRU INSERAREA ÎN TABELE

```
create sequence seq_bibliotecari -- setarea denumirii secvenței
start with 1 -- valoarea de la care pornește secvența
increment by 1 -- valoarea care este incrementată la fiecare pas
minvalue 0 -- valoarea minimă la care poate ajunge secvența
maxvalue 100 -- valoarea maximă la care poate ajunge secvența
nocycle; --când ajunge la maxvalue nu o mai ia de la capăt, ci va arunca o excepție
```

11. INSERAREA ÎN TABELE

BIBLIOTECARI

```
create table bibliotecari
(id_bibliotecar number(3) constraint pk_bibliotecari primary key,
 nume varchar2(20) constraint null_bibliotecar_nume not null,
 prenume varchar2(20) constraint null_bibliotecar_prenume not null,
 varsta number(2) constraint null_bibliotecar_varsta not null,
 salariu number(8,2) constraint null_bibliotecar_salariu not null,
 numar_etaj number(2) constraint null_numar_etaj not null,
 constraint unq_bibliotecar_nume_prenume unique(nume, prenume));

create sequence seq_bibliotecari
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;

insert into bibliotecari values(seq_bibliotecari.nextval, 'Teleaga', 'Mihaela', 57, 2300, 1);
insert into bibliotecari values(seq_bibliotecari.nextval, 'Popa', 'Mariana', 30, 2100, 2);
insert into bibliotecari values(seq_bibliotecari.nextval, 'Boghiu', 'Emilia', 60, 2500, 3);
insert into bibliotecari values(seq_bibliotecari.nextval, 'Fornica', 'Gabriel', 57, 2400, 1);
insert into bibliotecari values(seq_bibliotecari.nextval, 'Botezatu', 'Raluca', 46, 2100, 2);
insert into bibliotecari values(seq_bibliotecari.nextval, 'Tudor', 'Lavinia', 35, 2300, 1);
insert into bibliotecari values(seq_bibliotecari.nextval, 'Panfil', 'Otilia', 57, 2700, 3);
```

insert into bibliotecari values(seq_bibliotecari.nextval, 'Rafaila', 'Camelia', 30, 1900, 4);

ID_BIBLIOTECAR	NUME	PRENUME	VARSTA	SALARIU	NUMAR_ETAJ
1	1 Teleaga	Mihaela	57	2300	1
2	2 Popa	Mariana	30	2100	2
3	3 Boghiu	Emilia	60	2500	3
4	4 Fornica	Gabriel	57	2400	1
5	5 Botezatu	Raluca	46	2100	2
6	6 Tudor	Lavinia	35	2300	1
7	7 Panfil	Otilia	57	2700	3
8	8 Rafaila	Camelia	30	1900	4

CARTI

create table carti

(id_carte number(3) constraint pk_carti primary key,
denumire varchar2(30) constraint null_carte_denumire not null,
an_publicatie number(4) constraint null_carte_an_publicatie not null,
numar_pagini number(4) constraint null_carte_numar_pagini not null,
numar_copii number(2) default 1,
id_editura number(2) constraint fk_carte_editura references edituri(id_editura));

create sequence seq_carti

start with 1

increment by 1

minvalue 0

maxvalue 100

nocycle;

insert into carti values(seq_carti.nextval, 'Harry Potter', 2005, 566, 4, 1);

insert into carti values(seq_carti.nextval, 'Hotul', 2012, 600, 6, 2);

insert into carti values(seq_carti.nextval, 'Baltagul', 1930, 123, 3, 2);

insert into carti values(seq_carti.nextval, 'Ion', 1920, 289, 5, 3);

insert into carti values(seq_carti.nextval, 'Enigma Otiliei', 1938, 400, 1, 4);

ID_CARTE	DENUMIRE	AN_PUBLICATIE	NUMAR_PAGINI	NUMAR_COPII	ID_EDITURA
1	1 Harry Potter	2005	566	4	1
2	2 Hotul	2012	600	6	2
3	3 Baltagul	1930	123	3	2
4	4 Ion	1920	289	5	3
5	5 Enigma Otiliei	1938	400	1	4

CITITORI

create table cititori

```
(id_cititor number(3) constraint pk_cititor primary key,  
  nume varchar2(20) constraint null_cititor_nume not null,  
  prenume varchar2(20) constraint null_cititor_prenume not null,  
  varsta number(2) constraint null_cititor_varsta not null,  
  telefon varchar2(12) constraint null_cititor_telefon not null,  
  constraint unq_cititor_nume_prenume unique(nume, prenume));
```

create sequence seq_cititori

start with 1

increment by 1

minvalue 0

maxvalue 100

nocycle;

insert into cititori values(seq_cititori.nextval, 'Rogoza', 'Raluca', 20, '0741837264');

insert into cititori values(seq_cititori.nextval, 'Stoica', 'Elena', 14, '0778347287');

insert into cititori values(seq_cititori.nextval, 'Udrea', 'Robert', 19, '0756123656');

insert into cititori values(seq_cititori.nextval, 'Obreja', 'Carina', 40, '0723457846');

insert into cititori values(seq_cititori.nextval, 'Talpalariu', 'Iulia', 65, '0789648247');

ID_CITITOR	NUME	PRENUME	VARSTA	TELEFON
1	Rogoza	Raluca	20	0741837264
2	Stoica	Elena	14	0778347287
3	Udrea	Robert	19	0756123656
4	Obreja	Carina	40	0723457846
5	Talpalariu	Iulia	65	0789648247

RAFTURI

create table rafturi

```
(id_raft number(3) constraint pk_raft primary key,  
  etaj number(1) constraint null_raft_etaj not null);
```

create sequence seq_rafturi

start with 1

increment by 1

minvalue 0

maxvalue 100

nocycle;

```

insert into rafturi values(seq_rafturi.nextval, 3);
insert into rafturi values(seq_rafturi.nextval, 2);
insert into rafturi values(seq_rafturi.nextval, 1);
insert into rafturi values(seq_rafturi.nextval, 1);
insert into rafturi values(seq_rafturi.nextval, 2);

```

	ID_RAFT	ETAJ
1	1	3
2	2	2
3	3	1
4	4	1
5	5	2

AUTORI

create table autori

```

(id_autor number(3) constraint pk_autor primary key,
nume varchar2(20) constraint null_autor_nume not null,
prenume varchar2(20) constraint null_autor_prenume not null,
varsta number(2),
constraint unq_autor_nume_prenume unique(nume, prenume));

```

create sequence seq_autori

start with 1

increment by 1

minvalue 0

maxvalue 100

nocycle;

```

insert into autori values(seq_autori.nextval, 'Rowling', 'Joanne', 56);
insert into autori(id_autor, nume, prenume) values(seq_autori.nextval, 'Rebreanu', 'Liviu');
insert into autori values(seq_autori.nextval, 'Zusak', 'Markus', 43);
insert into autori(id_autor, nume, prenume) values(seq_autori.nextval, 'Calinescu', 'George');
insert into autori(id_autor, nume, prenume) values(seq_autori.nextval, 'Sadoveanu', 'Mihail');

```

	ID_AUTOR	NUME	PRENUME	VARSTA
1	1	Rowling	Joanne	56
2	2	Rebreanu	Liviu	(null)
3	3	Zusak	Markus	43
4	4	Calinescu	George	(null)
5	5	Sadoveanu	Mihail	(null)

GENURI

```
create table genuri
(id_gen number(3) constraint pk_gen primary key,
denumire varchar(20) constraint null_gen_denumire not null);
```

```
create sequence seq_genuri
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
insert into genuri values(seq_genuri.nextval, 'Aventura');
insert into genuri values(seq_genuri.nextval, 'Mister');
insert into genuri values(seq_genuri.nextval, 'Fictiune');
insert into genuri values(seq_genuri.nextval, 'Psihologie');
insert into genuri values(seq_genuri.nextval, 'Dragoste');
```

ID_GEN	DENUMIRE
1	Aventura
2	Mister
3	Fictiune
4	Psihologie
5	Dragoste

EDITURI

```
create table edituri
(id_editura number(3) constraint pk_editura primary key,
nume varchar(20) constraint null_editura_nume not null,
locatie varchar(40) constraint null_editura_locatie not null);
```

```
create sequence seq_edituri
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
insert into edituri values(seq_edituri.nextval, 'Humanitas', 'Sector 1 Bucuresti');
insert into edituri values(seq_edituri.nextval, 'Trei', 'Sector 3 Bucuresti');
```

```

insert into edituri values(seq_edituri.nextval, 'Polirom', 'Sector 1 Bucuresti');
insert into edituri values(seq_edituri.nextval, 'Arthur', 'Sector 4 Bucuresti');
insert into edituri values(seq_edituri.nextval, 'YoungArt', 'Sector 1 Bucuresti');

```

ID_EDITURA	NUME	LOCATIE
1	1 Humanitas	Sector 1 Bucuresti
2	2 Trei	Sector 3 Bucuresti
3	3 Polirom	Sector 1 Bucuresti
4	4 Arthur	Sector 4 Bucuresti
5	5 YoungArt	Sector 1 Bucuresti

PENALIZARI

```
create table penalizari
```

```

(id_penalizare number(3) constraint pk_penalizare primary key,
motiv varchar2(40),
pret number(4, 2) default 0);

```

```
create sequence seq_penalizari
```

```
start with 1
```

```
increment by 1
```

```
minvalue 0
```

```
maxvalue 100
```

```
nocycle;
```

```

insert into penalizari values(seq_penalizari.nextval, 'nereturnarea cartii', 30);
insert into penalizari values(seq_penalizari.nextval, 'deteriorarea cartii', 15);
insert into penalizari values(seq_penalizari.nextval, 'comportament neregulamentar', 5);
insert into penalizari values(seq_penalizari.nextval, 'perturbarea linistii', 3);
insert into penalizari(id_penalizare) values(seq_penalizari.nextval);

```

ID_PENALIZARE	MOTIV	PRET
1	3 nereturnarea cartii	30
2	4 deteriorarea cartii	15
3	5 comportament neregulamentar	5
4	6 perturbarea linistii	3
5	7 (null)	0

ABONAMENTE

create table abonamente

```
(id_abonament number(3) constraint pk_abonament primary key,  
tip varchar2(10) constraint null_abonament_tip not null,  
numar_carti_de_imprumutat number(2) constraint null_abonament_numar_carti not null,  
pret_pe_luna number(5, 2) constraint null_abonament_pret not null);
```

create sequence seq_abonamente

start with 1

increment by 1

minvalue 0

maxvalue 100

nocycle;

insert into abonamente values(seq_abonamente.nextval, 'student', 15, 50);

insert into abonamente values(seq_abonamente.nextval, 'elev', 12, 40);

insert into abonamente values(seq_abonamente.nextval, 'adult', 18, 80);

insert into abonamente values(seq_abonamente.nextval, 'pensionar', 16, 50);

insert into abonamente values(seq_abonamente.nextval, 'student', 8, 25);

insert into abonamente values(seq_abonamente.nextval, 'premium', 15, 45);

	ID_ABONAMENT	TIP	NUMAR_CARTI_DE_IMPRUMUTAT	PRET_PE_LUNA
1	1	student	15	50
2	2	elev	12	40
3	3	adult	18	80
4	4	pensionar	16	50
5	5	student	8	25
6	6	premium	15	45

ARE_GRIJA_DE

create table are_grija_de

```
(id_bibliotecar number(3),
```

```
id_carte number(3),
```

```
constraint pk_bibliotecar_carte primary key(id_bibliotecar, id_carte));
```

insert into are_grija_de values(1, 1);

insert into are_grija_de values(2, 1);

insert into are_grija_de values(5, 3);

insert into are_grija_de values(4, 2);

insert into are_grija_de values(3, 1);


```

insert into are_grija_de values(2, 3);
insert into are_grija_de values(3, 2);
insert into are_grija_de values(1, 2);
insert into are_grija_de values(1, 5);
insert into are_grija_de values(4, 1);

```

	ID_BIBLIOTECAR	ID_CARTE
1	1	1
2	2	1
3	5	3
4	4	2
5	3	1
6	2	3
7	3	2
8	1	2
9	1	5
10	4	1

APARTINE

```

create table apartine
(id_carte number(3),
 id_gen number(3),
 constraint pk_gen_carte primary key(id_carte, id_gen));

```

```

insert into apartine values(4, 1);
insert into apartine values(1, 3);
insert into apartine values(2, 3);
insert into apartine values(3, 4);
insert into apartine values(1, 1);
insert into apartine values(1, 5);
insert into apartine values(3, 2);
insert into apartine values(4, 3);
insert into apartine values(5, 2);
insert into apartine values(3, 5);

```

	ID_CARTE	ID_GEN
1	4	1
2	1	3
3	2	3
4	3	4
5	1	1
6	1	5
7	3	2
8	4	3
9	5	2
10	3	5

COPII_CARTI

```
create table copii_carti
(id_copie number(3),
 id_carte number(3),
 id_raft number(3) constraint fk_copie_carte_raft references rafturi(id_raft),
 constraint pk_copie_carte primary key(id_copie, id_carte));
```

```
insert into copii_carti values(3, 1, 3);
insert into copii_carti values(1, 5, 2);
insert into copii_carti values(2, 1, 2);
insert into copii_carti values(5, 3, 4);
insert into copii_carti values(4, 2, 5);
insert into copii_carti values(2, 4, 1);
insert into copii_carti values(4, 1, 2);
insert into copii_carti values(1, 2, 5);
insert into copii_carti values(1, 3, 1);
insert into copii_carti values(5, 1, 2);
```

	ID_COPIE	ID_CARTE	ID_RAFT
1	3	1	3
2	1	5	2
3	2	1	2
4	5	3	4
5	4	2	5
6	2	4	1
7	4	1	2
8	1	2	5
9	1	3	1
10	5	1	2

SCRISA_DE

```
create table scrisa_de
(id_carte number(3),
 id_autor number(3),
 constraint pk_carte_autor primary key(id_carte, id_autor));
```

```
insert into scrisa_de values(3, 2);
insert into scrisa_de values(1, 2);
insert into scrisa_de values(5, 2);
insert into scrisa_de values(1, 1);
insert into scrisa_de values(4, 3);
insert into scrisa_de values(3, 5);
insert into scrisa_de values(1, 5);
insert into scrisa_de values(4, 2);
insert into scrisa_de values(5, 3);
insert into scrisa_de values(2, 1);
```

	ID_CARTE	ID_AUTOR
1	3	2
2	1	2
3	5	2
4	1	1
5	4	3
6	3	5
7	1	4
8	4	2
9	5	3
10	2	2

DETINE

```
create table detine
(id_cititor number(3),
 id_abonament number(3),
 data_inceput date constraint null_data_inceput not null,
 numar_luni number(1) constraint null_numar_luni not null,
 constraint pk_cititor_abonament primary key(id_cititor, id_abonament));
```

```
insert into detine values(2, 2, '02-MAY-22', 2);
insert into detine values(1, 1, '08-JAN-22', 1);
insert into detine values(3, 6, '12-MAR-22', 1);
```

```

insert into detine values(5, 4, '17-FEB-20', 4);
insert into detine values(2, 6, '02-JAN-21', 2);
insert into detine values(1, 5, '20-MAR-21', 1);
insert into detine values(4, 6, '22-SEP-21', 3);
insert into detine values(5, 6, '01-MAY-21', 2);
insert into detine values(3, 1, '15-MAY-22', 2);
insert into detine values(4, 3, '02-APR-22', 3);

```

	ID_CITITOR	ID_ABONAMENT	DATA_INCEPUT	NUMAR_LUNI
1	2	2	02-MAY-22	2
2	1	1	08-JAN-22	1
3	3	6	12-MAR-22	1
4	5	4	17-FEB-20	4
5	2	6	02-JAN-21	2
6	1	5	20-MAR-21	1
7	4	6	22-SEP-21	3
8	5	6	01-MAY-21	2
9	3	1	15-MAY-22	2
10	4	3	02-APR-22	3

PRIMESTE

```

create table primeste
(id_cititor number(3),
 id_penalizare number(3),
 constraint pk_cititor_penalizare primary key(id_cititor, id_penalizare));

```

```

insert into primeste values(1, 2);
insert into primeste values(1, 5);
insert into primeste values(3, 4);
insert into primeste values(3, 1);
insert into primeste values(5, 2);
insert into primeste values(4, 1);
insert into primeste values(3, 2);
insert into primeste values(5, 3);
insert into primeste values(4, 2);
insert into primeste values(1, 1);

```

	ID_CITITOR	ID_PENALIZARE
1	1	2
2	1	5
3	3	4
4	3	1
5	5	2
6	4	1
7	3	2
8	5	3
9	4	2
10	1	1

CARTI_IMPRUMUTATE

create table carti_imprumutate

(id_carte_imprumutata number(3) constraint pk_carte_imprumutata primary key,
id_cititor number(3),
id_carte number(3),
id_copie number(3),
id_autor number(3),
data_imprumut date constraint null_data_imprumut not null,
data_restituire_impusa date constraint null_data_restituire_impusa not null,
data_restituire_efectiva date,
constraint unq_carte_imprumutata unique(id_cititor, id_carte, id_copie, id_autor));

create sequence seq_carte_imprumutata

start with 1

increment by 1

minvalue 0

maxvalue 100

nocycle;

insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 1, 1, 3, 1, '08-JAN-22',
'20-JAN-22', '30-JAN-22');

insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 1, 5, 1, 2, '21-MAR-22',
'30-MAR-22', '5-APR-22');

insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 2, 1, 2, 2, '03-JAN-22',
'13-JAN-22', '10-JAN-22');

insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 2, 3, 5, 5, '05-MAY-22',
'11-MAY-22', '10-MAY-22');

```

insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 3, 2, 4, 3, '14-MAR-22',
'25-MAR-22', '22-MAR-22');
insert into carti_imprumutate(id_carte_imprumutata, id_cititor, id_carte, id_copie, id_autor,
data_imprumut, data_restituire_impusa)
values(seq_carte_imprumutata.nextval, 3, 4, 2, 1, '20-MAY-22', '30-MAY-22');
insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 4, 2, 1, 2, '24-SEP-21',
'01-OCT-21', '06-OCT-22');
insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 4, 3, 1, 4, '01-NOV-22',
'14-NOV-22', '05-NOV-22');
insert into carti_imprumutate values(seq_carte_imprumutata.nextval, 5, 1, 5, 2, '05-MAY-21',
'10-MAY-21', '07-MAY-22');
insert into carti_imprumutate(id_carte_imprumutata, id_cititor, id_carte, id_copie, id_autor,
data_imprumut, data_restituire_impusa)
values(seq_carte_imprumutata.nextval, 5, 1, 4, 2, '19-FEB-20', '27-FEB-20');

```

	ID_CARTE_IMPRUMUTATA	ID_CITITOR	ID_CARTE	ID_COPIE	ID_AUTOR	DATA_IMPRUMUT	DATA_RESTITUIRE_IMPUSA	DATA_RESTITUIRE_EFECTIVA
1	1	1	1	3	1	08-JAN-22	20-JAN-22	30-JAN-22
2	2	1	5	1	2	21-MAR-22	30-MAR-22	05-APR-22
3	3	2	1	2	2	03-JAN-22	13-JAN-22	10-JAN-22
4	4	2	3	5	5	05-MAY-22	11-MAY-22	10-MAY-22
5	5	3	2	4	3	14-MAR-22	25-MAR-22	22-MAR-22
6	6	3	4	2	1	20-MAY-22	30-MAY-22	(null)
7	7	4	2	1	2	24-SEP-21	01-OCT-21	06-OCT-22
8	8	4	3	1	4	01-NOV-22	14-NOV-22	05-NOV-22
9	9	5	1	5	2	05-MAY-21	10-MAY-21	07-MAY-22
10	10	5	1	4	2	19-FEB-20	27-FEB-20	(null)

12. CERERI SQL

Cererea 1

Să se afișeze numărul total de abonamente pe care le dețin cititorii și câte dintre acestea s-au terminat/se vor termina pe parcursul anilor 2020, 2021, 2022.

```

select count(id_abonament) Total_abonamente,
       sum(decode(to_char(add_months(data_inceput,numar_luni),'yyy'), 2020, 1, 0)) "AN 2020",
       sum(decode(to_char(add_months(data_inceput,numar_luni),'yyy'), 2021, 1, 0)) "AN 2021",
       sum(decode(to_char(add_months(data_inceput,numar_luni),'yyy'), 2022, 1, 0)) "AN 2022"
from detine;

```

Worksheet Query Builder

```
--Să se afișeze numărul total de abonamente pe care le dețin cititorii
--și câte dintre acestea s-au terminat/se vor termina pe parcursul anilor 2020, 2021, 2022.

select count(id_abonament) Total_abonamente,
       sum(decode(to_char(add_months(data_inceput,numar_luni),'yyyy'), 2020, 1, 0)) "AN 2020",
       sum(decode(to_char(add_months(data_inceput,numar_luni),'yyyy'), 2021, 1, 0)) "AN 2021",
       sum(decode(to_char(add_months(data_inceput,numar_luni),'yyyy'), 2022, 1, 0)) "AN 2022"
from detine;

select *
from cititori;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.001 seconds

	TOTAL_ABONAMENTE	AN 2020	AN 2021	AN 2022
1	10	1	4	5

Cererea 2

Să se selecteze id-urile cărților împrumutate de cititori care au avut/au un abonament de tip student, al căror prenume începe cu 'R' și care au împrumutat cărți de care a avut grijă cel puțin un bibliotecar care se află printre primii 3 care câștigă cel mai mult.

```
with salarii3 as (select *
                  from (select distinct salariu
                        from bibliotecari
                        order by salariu desc) s
                  where rownum<=3)
select id_carte
from carti_imprumutate
where id_cititor in
      (select c.id_cititor
       from cititori c, abonamente a, detine d
       where a.id_abonament = d.id_abonament and c.id_cititor = d.id_cititor and a.tip =
'student'

                  and d.id_cititor in (select id_cititor
                                       from cititori
                                       where prenume like 'R%'))
and id_carte in (select id_carte
                 from are_grija_de
                 where id_bibliotecar in (select id_bibliotecar
```

```

from bibliotecari
where salariu in (select * from salarii3))
group by id_carte);

```

Worksheet Query Builder

--Să se selecteze id-urile cărților împrumutate de cititori care au avut/au un abonament de tip student,
 --al căror prenume începe cu 'R' și care au împrumutat cărți
 --de care a avut grijă cel puțin un bibliotecar care se află printre primii 3 care câștigă cel mai mult.

```

with salarii3 as (select *
                  from (select distinct salariu
                        from bibliotecari
                        order by salariu desc) s
                  where rownum<=3)

select id_carte
from carti_imprumutate
where id_cititor in
      (select c.id_cititor
       from cititori c, abonamente a, detine d
       where a.id_abonament = d.id_abonament and c.id_cititor = d.id_cititor and a.tip = 'student'
        and d.id_cititor in (select id_cititor
                             from cititori
                             where prenume like 'R%'))

and id_carte in (select id_carte
                 from are_grija_de
                 where id_bibliotecar in (select id_bibliotecar
                                           from bibliotecari
                                           where salariu in (select * from salarii3))

                 group by id_carte);

```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.027 seconds

ID_CARTE
1
2

Cererea 3

Să se afișeze vârsta, salariul total al bibliotecarilor cu acea vârstă și salariile bibliotecarilor care au acea vârstă în funcție de etajele la care lucrează pentru cei de la etajele 1, 2 și 3. Se selectează doar vârstele pe care le au cel puțin 2 bibliotecari.

```

select varsta, nvl(sum(salariu),0) Total, (select nvl(sum(salariu),0)
from bibliotecari
where numar_etaj = 1 and varsta = b.varsta) Etaj1,
(select NVL(sum(salariu),0)
from bibliotecari
where numar_etaj = 2 and varsta = b.varsta) Etaj2,
(select nvl(sum(salariu),0)
from bibliotecari
where numar_etaj = 3 and varsta = b.varsta) Etaj3

from bibliotecari b
group by varsta
having count(*)>1;

```


Worksheet Query Builder

```
--Să se afișeze vârsta, salariul total al bibliotecarilor cu acea vârstă și salariile bibliotecarilor
--care au acea vârstă în funcție de etajele la care lucrează pentru cei de la etajele 1, 2 și 3.
--Se selectează doar vârstele pe care le au cel puțin 2 bibliotecari.
select varsta, nvl(sum(salariu),0) Total, (select nvl(sum(salariu),0)
from bibliotecari
where numar_etaj = 1 and varsta = b.varsta) Eta1,
(select NVL(sum(salariu),0)
from bibliotecari
where numar_etaj = 2 and varsta = b.varsta) Eta2,
(select nvl(sum(salariu),0)
from bibliotecari
where numar_etaj = 3 and varsta = b.varsta) Eta3
from bibliotecari b
group by varsta
having count(*)>1;
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.015 seconds

	VARSTA	TOTAL	ETA1	ETA2	ETA3
1	57	7400	4700	0	2700
2	30	4000	0	2100	0

Cererea 4

Să se afișeze pentru fiecare carte împrumutată denumirea ei, numele unuia dintre autorii care a scris-o(în caz că a fost scrisă de mai mulți), editura care a publicat-o (atât numele, cât și locația sediului acesteia prin concatenare), prenumele cititorului care a împrumutat-o și data la care a fost împrumutată. Cărțile vor fi afișate în ordine de la cea mai recenta carte împrumutată, la cea mai veche.

```
select ca.denumire as Denumire_Carte,
       a.numa as Nume_Autor,
       concat(concat(e.numa,' '), e.locatie) as Editura,
       c.prenume,
       ci.data_imprumut
from carti_imprumutate ci, cititori c, carti ca, autori a, edituri e
where ci.id_cititor = c.id_cititor and ci.id_carte = ca.id_carte and ci.id_autor = a.id_autor and
ca.id_editura = e.id_editura
order by 5 desc;
```

Worksheet Query Builder

--Să se afișeze pentru fiecare carte împrumutată denumirea ei,
--numele unuia dintre autorii care a scris-o (în caz că a fost scrisă de mai mulți),
--editura care a publicat-o (atât numele, cât și locația sediului acesteia prin concatenare),
--prenumele cititorului care a împrumutat-o și data la care a fost împrumutată.
--Cărțile vor fi afișate în ordine de la cea mai recenta carte împrumutată, la cea mai veche.

select ca.denumire as Denumire_Carte,
 a.numa as Nume_Autor,
 concat(concat(e.numa, ' '), e.locatie) as Editura,
 c.prenume,
 ci.data_imprumut
from carti_imprumutate ci, cititori c, carti ca, autori a, edituri e
where ci.id_cititor = c.id_cititor and ci.id_carte = ca.id_carte and ci.id_autor = a.id_autor and ca.id_editura = e.id_editura
order by 5 desc;

Script Output Query Result

All Rows Fetched: 10 in 0.014 seconds

DENUMIRE_CARTE	NUME_AUTOR	EDITURA	PRENUME	DATA_IMPRUMUT
1 Baltagul	Calinescu	Trei Sector 3 Bucuresti	Carina	01-NOV-22
2 Ion	Rowling	Polirom Sector 1 Bucuresti	Robert	20-MAY-22
3 Baltagul	Sadoveanu	Trei Sector 3 Bucuresti	Elena	05-MAY-22
4 Enigma Otiliei	Rebreanu	Arthur Sector 4 Bucuresti	Raluca	21-MAR-22
5 Hotul	Zusak	Trei Sector 3 Bucuresti	Robert	14-MAR-22
6 Harry Potter	Rowling	Humanitas Sector 1 Bucuresti	Raluca	08-JAN-22
7 Harry Potter	Rebreanu	Humanitas Sector 1 Bucuresti	Elena	03-JAN-22
8 Hotul	Rebreanu	Trei Sector 3 Bucuresti	Carina	24-SEP-21
9 Harry Potter	Rebreanu	Humanitas Sector 1 Bucuresti	Iulia	05-MAY-21
10 Harry Potter	Rebreanu	Humanitas Sector 1 Bucuresti	Iulia	19-FEB-20

Cererea 5

Să se stabilească pentru fiecare carte împrumutată dacă a fost restituită după maxim o săptămână de când a fost împrumutată și în, caz afirmativ, să se afișeze ziua din săptămână în care a fost restituită.

```
select id_carte_imprumutata, data_imprumut, data_restituire_efectiva,
       case
           when to_char(next_day(data_imprumut, 'Monday')) =
to_char(data_restituire_efectiva) then 'Monday'
           when to_char(next_day(data_imprumut, 'Tuesday')) =
to_char(data_restituire_efectiva) then 'Tuesday'
           when to_char(next_day(data_imprumut, 'Wednesday')) =
to_char(data_restituire_efectiva) then 'Wednesday'
           when to_char(next_day(data_imprumut, 'Thursday')) =
to_char(data_restituire_efectiva) then 'Thursday'
           when to_char(next_day(data_imprumut, 'Friday')) =
to_char(data_restituire_efectiva) then 'Friday'
           when to_char(next_day(data_imprumut, 'Saturday')) =
to_char(data_restituire_efectiva) then 'Saturday'
           when to_char(next_day(data_imprumut, 'Sunday')) =
to_char(data_restituire_efectiva) then 'Sunday'
           else 'A trecut mai mult de o saptamana'
```

```

end
as Zi_din_saptamana
from carti_imprumutate;

```

Worksheet Query Builder

```

--Să se stabilească pentru fiecare carte imprumutata dacă a fost restituita după maxim o săptămână de când a fost imprumutată
--și în, caz afirmativ, să se afișeze ziua din săptămână în care a fost restituită.

select id_carte_imprumutata, data_imprumut, data_restituire_efectiva,
       case
         when to_char(next_day(data_imprumut, 'Monday')) = to_char(data_restituire_efectiva) then 'Monday'
         when to_char(next_day(data_imprumut, 'Tuesday')) = to_char(data_restituire_efectiva) then 'Tuesday'
         when to_char(next_day(data_imprumut, 'Wednesday')) = to_char(data_restituire_efectiva) then 'Wednesday'
         when to_char(next_day(data_imprumut, 'Thursday')) = to_char(data_restituire_efectiva) then 'Thursday'
         when to_char(next_day(data_imprumut, 'Friday')) = to_char(data_restituire_efectiva) then 'Friday'
         when to_char(next_day(data_imprumut, 'Saturday')) = to_char(data_restituire_efectiva) then 'Saturday'
         when to_char(next_day(data_imprumut, 'Sunday')) = to_char(data_restituire_efectiva) then 'Sunday'
         else 'A trecut mai mult de o saptamana'
       end
       as Zi_din_saptamana
from carti_imprumutate;

```

Script Output x Query Result x

All Rows Fetched: 10 in 0.01 seconds

ID_CARTE_IMPRUMUTATA	DATA_IMPRUMUT	DATA_RESTITUIRE_EFECTIVA	ZI_DIN_SAPTAMANA
1	108-JAN-22	30-JAN-22	A trecut mai mult de o saptamana
2	221-MAR-22	05-APR-22	A trecut mai mult de o saptamana
3	303-JAN-22	10-JAN-22	Monday
4	405-MAY-22	10-MAY-22	Tuesday
5	514-MAR-22	22-MAR-22	A trecut mai mult de o saptamana
6	620-MAY-22	(null)	A trecut mai mult de o saptamana
7	724-SEP-21	06-OCT-22	A trecut mai mult de o saptamana
8	801-NOV-22	05-NOV-22	Saturday
9	905-MAY-21	07-MAY-22	A trecut mai mult de o saptamana
10	1019-FEB-20	(null)	A trecut mai mult de o saptamana

13. OPERAȚII DE ACTUALIZARE ȘI SUPRIMARE A DATELOR

OPERAȚIA 1

Să se crească cu 100 salariile bibliotecarilor care lucrează la etajul 1.

```

update bibliotecari
set salariu = salariu + 100
where id_bibliotecar in (select id_bibliotecar
                          from bibliotecari
                          where numar_etaj = 1);

```

Tabela BIBLIOTECARI înainte:

	ID_BIBLIOTECAR	NUME	PRENUME	VARSTA	SALARIU	NUMAR_ETAJ
1		1 Teleaga	Mihaela	57	2300	1
2		2 Popa	Mariana	30	2100	2
3		3 Boghiu	Emilia	60	2500	3
4		4 Fornica	Gabriel	57	2400	1
5		5 Botezatu	Raluca	46	2100	2
6		6 Tudor	Lavinia	35	2300	1
7		7 Panfil	Otilia	57	2700	3
8		8 Rafaila	Camelia	30	1900	4

Tabela BIBLIOTECARI după update:

3 rows updated.

The screenshot shows a database query editor with a 'Query Builder' tab. The SQL query is as follows:

```
update bibliotecari
set salariu = salariu + 100
where id_bibliotecar in (select id_bibliotecar
                        from bibliotecari
                        where numar_etaj = 1);

select * from bibliotecari;
```

Below the query, the 'Query Result' tab displays the updated data. The results show that the salary for the three librarians on the 1st floor (Teleaga, Fornica, and Tudor) has been increased by 100.

ID_BIBLIOTECAR	NUME	PRENUME	VARSTA	SALARIU	NUMAR_ETAJ
1	1 Teleaga	Mihaela	57	2400	1
2	2 Popa	Mariana	30	2100	2
3	3 Boghiu	Emilia	60	2500	3
4	4 Fornica	Gabriel	57	2500	1
5	5 Botezatu	Raluca	46	2100	2
6	6 Tudor	Lavinia	35	2400	1
7	7 Panfil	Otilia	57	2700	3
8	8 Rafaila	Camelia	30	1900	4

Operația 2

Să se șteargă din bibliotecă cărțile care au fost împrumutate de către cititori, dar nu au fost restituite.

```
delete from carti_imprumutate
where id_carte_imprumutata in (select id_carte imprumutata
                               from carti_imprumutate
                               where data_restituire_efectiva is null);
```

Tabela CARTI_IMPRUMUTATE înainte de update:

	ID_CARTE_IMPRUMUTATA	ID_CITITOR	ID_CARTE	ID_COPIE	ID_AUTOR	DATA_IMPRUMUT	DATA_RESTITUIRE_IMPUSA	DATA_RESTITUIRE_EFECTIVA
1	1	1	1	3	1	08-JAN-22	20-JAN-22	30-JAN-22
2	2	1	5	1	2	21-MAR-22	30-MAR-22	05-APR-22
3	3	2	1	2	2	03-JAN-22	13-JAN-22	10-JAN-22
4	4	2	3	5	5	05-MAY-22	11-MAY-22	10-MAY-22
5	5	3	2	4	3	14-MAR-22	25-MAR-22	22-MAR-22
6	6	3	4	2	1	20-MAY-22	30-MAY-22	(null)
7	7	4	2	1	2	24-SEP-21	01-OCT-21	06-OCT-22
8	8	4	3	1	4	01-NOV-22	14-NOV-22	05-NOV-22
9	9	5	1	5	2	05-MAY-21	10-MAY-21	07-MAY-22
10	10	5	1	4	2	19-FEB-20	27-FEB-20	(null)

Tabela CARTI_IMPRUMUTATE după update:

2 rows deleted.

Worksheet

Query Builder

```

select * from carti_imprumutate;
--Să se șteargă din bibliotecă cărțile care au fost împrumutate de către cititori, dar nu au fost restituite.
delete from carti_imprumutate
where id_carte_imprumutata in (select id_carte imprumutata
                                from carti_imprumutate
                                where data_restituire_efectiva is null);

```

Script Output x

Query Result x

SQL

All Rows Fetched: 8 in 0.001 seconds

ID_CARTE_IMPRUMUTATA	ID_CITITOR	ID_CARTE	ID_COPIE	ID_AUTOR	DATA_IMPRUMUT	DATA_RESTITUIRE_IMPUSA	DATA_RESTITUIRE_EFECTIVA
1	2	1	5	1	21-MAR-22	30-MAR-22	05-APR-22
2	3	2	1	2	03-JAN-22	13-JAN-22	10-JAN-22
3	5	3	2	4	14-MAR-22	25-MAR-22	22-MAR-22
4	6	3	4	2	20-MAY-22	30-MAY-22	(null)
5	7	4	2	1	24-SEP-21	01-OCT-21	06-OCT-22
6	8	4	3	1	01-NOV-22	14-NOV-22	05-NOV-22
7	9	5	1	5	05-MAY-21	10-MAY-21	07-MAY-22
8	10	5	1	4	19-FEB-20	27-FEB-20	(null)

Operația 3

Să se transforme data de început a abonamentelor cititorilor în data de final pentru cei cu abonament de tip premium (la data de început se adaugă numărul de luni cât durează abonamentul).

update detine

set data_inceput = add_months(data_inceput, numar_luni)

where id_abonament in (select id_abonament

from abonamente

where tip = 'premium');

Tabela DETINE înainte de update:

	ID_CITITOR	ID_ABONAMENT	DATA_INCEPUT	NUMAR_LUNI
1	2	2	02-MAY-22	2
2	1	1	08-JAN-22	1
3	3	6	12-MAR-22	1
4	5	4	17-FEB-20	4
5	2	6	02-JAN-21	2
6	1	5	20-MAR-21	1
7	4	6	22-SEP-21	3
8	5	6	01-MAY-21	2
9	3	1	15-MAY-22	2
10	4	3	02-APR-22	3

Tabela DETINE după update:

4 rows updated.

Worksheet

Query Builder

update detine

set data_inceput = add_months(data_inceput, numar_luni)

where id_abonament in (select id_abonament

from abonamente

where tip = 'premium');

Script Output

Query Result

SQL

All Rows Fetched: 10 in 0.002 seconds

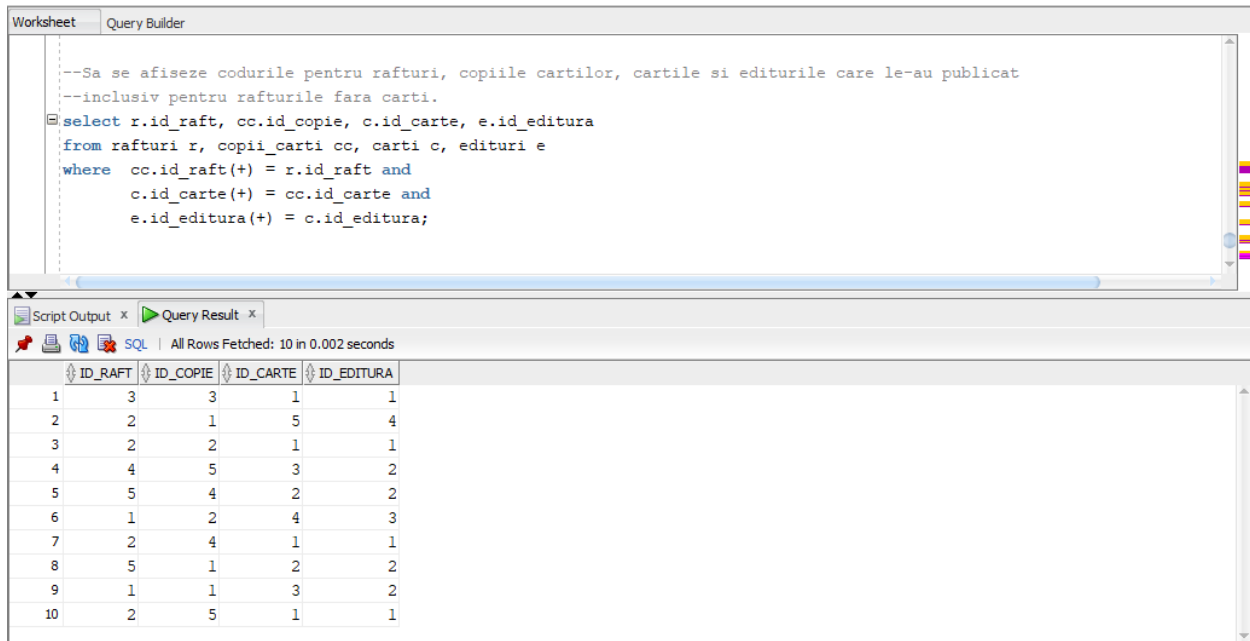
ID_CITITOR	ID_ABONAMENT	DATA_INCEPUT	NUMAR_LUNI
1	2	2 02-MAY-22	2
2	1	1 08-JAN-22	1
3	3	6 12-APR-22	1
4	5	4 17-FEB-20	4
5	2	6 02-MAR-21	2
6	1	5 20-MAR-21	1
7	4	6 22-DEC-21	3
8	5	6 01-JUL-21	2
9	3	1 15-MAY-22	2
10	4	3 02-APR-22	3

14. CERERI SQL: OUTER-JOIN ȘI DIVISION

OUTER-JOIN

Să se afișeze codurile pentru rafturi, copiile cărților, cărțile și editurile care le-au publicat, inclusiv pentru rafturile fără cărți.

```
select r.id_raft, cc.id_copie, c.id_carte, e.id_editura
from rafturi r, copii_carti cc, carti c, edituri e
where cc.id_raft(+) = r.id_raft and
      c.id_carte(+) = cc.id_carte and
      e.id_editura(+) = c.id_editura;
```



The screenshot shows a SQL query editor with a 'Query Builder' tab. The query is as follows:

```
--Sa se afiseze codurile pentru rafturi, copiile cartilor, cartile si editurile care le-au publicat
--inclusiv pentru rafturile fara carti.
select r.id_raft, cc.id_copie, c.id_carte, e.id_editura
from rafturi r, copii_carti cc, carti c, edituri e
where cc.id_raft(+) = r.id_raft and
      c.id_carte(+) = cc.id_carte and
      e.id_editura(+) = c.id_editura;
```

Below the query editor, the 'Query Result' tab is active, showing 10 rows of data. The columns are ID_RAFT, ID_COPIE, ID_CARTE, and ID_EDITURA.

ID_RAFT	ID_COPIE	ID_CARTE	ID_EDITURA
1	3	3	1
2	2	1	5
3	2	2	1
4	4	5	3
5	5	4	2
6	1	2	4
7	2	4	1
8	5	1	2
9	1	1	3
10	2	5	1

DIVISION 1

Să se obțină id-urile tuturor cititorilor care au deținut toate abonamentele ale căror prețuri se află între 40 și 50 de lei.

```
select distinct id_cititor
from detine d1
where not exists ((select id_abonament
                  from abonamente a
```


Worksheet Query Builder

```
--Să se obțină id-urile tuturor cititorilor care au împrumutat toate cărțile ale căror autori au litera 'r' în nume.
select id_cititor
from carti_imprumutate
where id_autor in (select id_autor
                  from autori
                  where lower(nume) like '%r%')
group by id_cititor
having count(id_autor) = (select count(*)
                        from autori
                        where lower(nume) like '%r%');
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.002 seconds

ID_CITITOR	
1	1
2	5

15. OPTIMIZAREA CERERII

Să se obțină codurile editurilor și locațiile lor pentru cele care conțin litera 'r', denumirea cărților, numărul de pagini și anul de publicație pentru cărțile care au între 100 și 300 de pagini de la editurile respective.

CARTE(id_carte#, denumire, an_publicatie, numar_copii, id_editura)

EDITURA(id_editura#, nume, locatie)

R1 = PROJECT(CARTE, denumire, numar_pagini, an_publicatie)

R2 = SELECT(R1, numar_pagini > 100)

R3 = SELECT(R2, numar_pagini < 300)

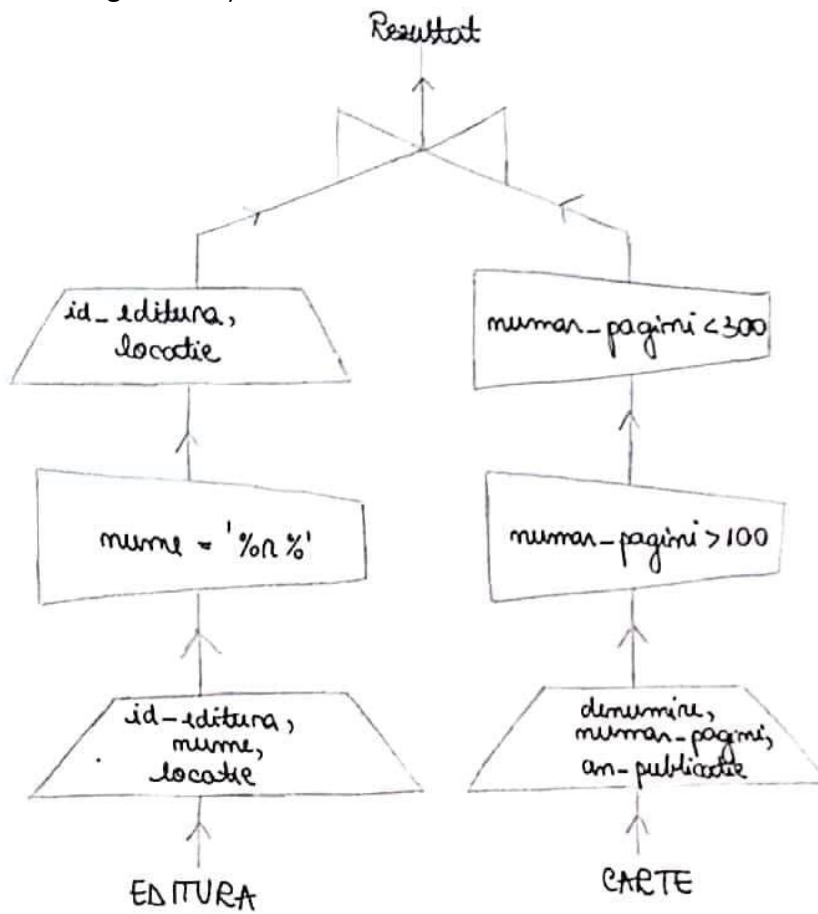
R4 = PROJECT(EDITURA, id_editura, nume, locatie)

R5 = SELECT(R4, nume like '%r%')

R6 = PROJECT(R5, id_editura, locatie)

Rezultat = JOIN(R3, R6)

Arborele algebric inițial:



Optimizarea cererii:

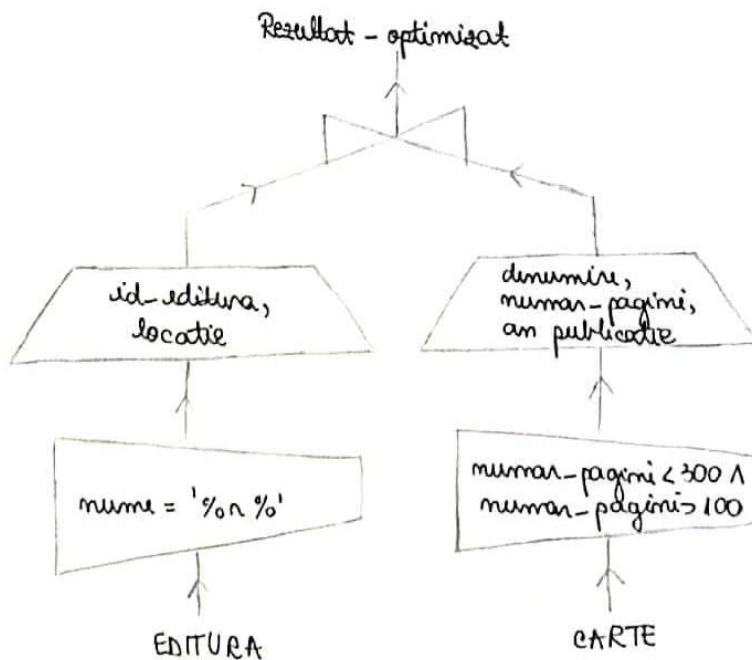
$$\text{Result} = \pi_{\text{id-edittura}, \text{locat}} \left(\sigma_{\text{nume} = 'e, n'} \left(\pi_{\text{id-edittura}, \text{nume}, \text{locat}} (\text{EDITTURA}) \right) \right) \bowtie \sigma_{\text{numan-pagini} < 300} \left(\sigma_{\text{numan-pagini} > 100} \left(\pi_{\text{denumire}, \text{numan-pagini}, \text{an-publicatie}} (\text{CARTE}) \right) \right)$$

$$\text{Resultat}^{\text{74}} = \pi_{\text{id-sedutara}, \text{locatie}} \left(\sqrt{\text{numeri} = \frac{1}{2} \cdot n\%} \left(\pi_{\text{id-sedutara}, \text{numeri}, \text{locatie}} (\text{EDUTURA}) \right) \right) \bowtie \sqrt{\text{numeri_pagini} < 300 \wedge \text{numeri_pagini} > 100} \left(\pi_{\text{denumire}, \text{numeri_pagini}, \text{an-publicatie}} (\text{CARTE}) \right)$$

$$Resultat = \pi_{id_editura, locatie} (\pi_{id_editura, (\sqrt{nume = \%n\%} (EDITURA)))} \bowtie \pi_{denumiri, nume_pagini, an_publicatie} (\sigma_{nume_pagini < 300, nume_pagini > 100} (CARTE)))$$

$$Rezultat = \pi_{id_editura, locatie} (\sigma_{nume = \%n\% (EDITURA)}) \bowtie \pi_{denumire, numar_pagini, an_publicatie} (\sigma_{numar_pagini < 300 \wedge (CARTE)}) = Rezultat_optimis$$

Arborele algebric rezultat:



```
select e.id_editura, e.locatie, c.denumire, c.numar_pagini, c.an_publicatie
from edituri e, carti c
where e.num like '%r%' and e.id_editura = c.id_editura and c.numar_pagini > 100 and
c.numar_pagini < 300;
```

Worksheet Query Builder

--Să se obțină codurile editurilor și locațiile lor pentru cele care conțin litera 'r', denumirea cărților,
--numărul de pagini și anul de publicație pentru cărțile care au între 100 și 300 de pagini de la editurile respective.

```
select e.id_editura, e.locatie, c.denumire, c.numar_pagini, c.an_publicatie
from edituri e, carti c
where e.num like '%r%' and e.id_editura = c.id_editura and c.numar_pagini > 100 and c.numar_pagini < 300;
```

Query Result x

All Rows Fetched: 2 in 0.005 seconds

ID_EDITURA	LOCATIE	DENUMIRE	NUMAR_PAGINI	AN_PUBLICATIE
1	2 Sector 3 Bucuresti Baltagul		123	1930
2	3 Sector 1 Bucuresti Ion		289	1920

16. REALIZAREA NORMALIZĂRII BCNF, FN4, FN5

BCNF

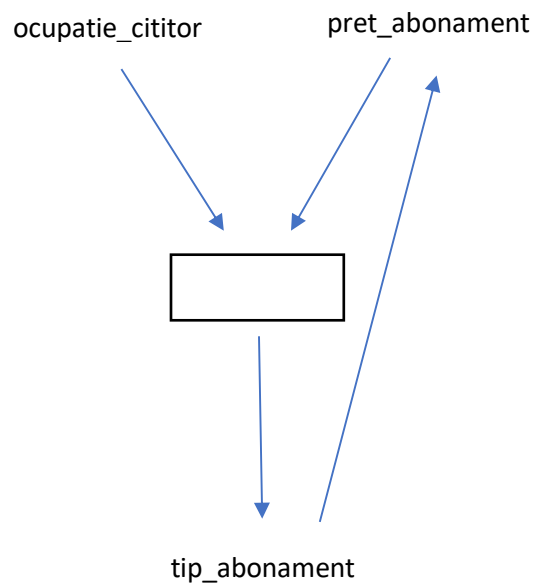
Modelul proiectat se află în forma normală Boyce-Codd.

Exemplu de relație care nu se află în forma normală Boyce-Codd:

DETINE (ocupatie_cititor#, pret_abonament#, tip_abonament)

{ocupatie_cititor#, pret_abonament#} -> {tip_abonament#}

{tip_abonament#} -> {pret_abonament#}



Se aplică regula Casey-Delobel și se aduce relația în BCNF.

DETINE1 (ocupatie_cititor#, tip_abonament)

DETINE2 (tip_abonament#, pret_abonament)

FN4

Relația CARTE_IMPRUMUTATA nu se afla in FN4, deoarece o carte poate avea mai multe copii și mai mulți autori.

id_carte ->> id_copie

id_carte ->> id_autor

CARTE_IMPRUMUTATA	
PK	id_carte_imprumutata
FK	id_cititor
FK	id_carte
FK	id_copie
FK	id_autor
	data_imprumut
	data_restituire_impusa
	data_restituire_efectiva

De exemplu, dacă a fost împrumutată cartea cu id-ul 1 care are 2 copii(cu id-urile 1 și 2) și 2 autori(cu id-urile 1 și 2), tabelul ar trebui să arate în felul următor:

id_carte	id_copie	id_autor
1	1	1
1	2	2
1	1	2
1	2	1

Pentru ca relația să fie în FN4 ar trebui să împărțim cele 3 attribute câte 2 în fiecare dintre cele 2 tabele în felul următor:

CARTE_IMPRUMUTATA1(id_carte_imprumutata#, id_cititor, id_carte, id_copie, data_imprumut, data_restituire_impusa, data_restituire_efectiva)

CARTE_IMPRUMUTATA2(id_carte_imprumutata#, id_cititor, id_carte, id_autor, data_imprumut, data_restituire_impusa, data_restituire_efectiva)

Acum relațiile se află în FN4.

FN5

Modelul proiectat se află în forma normală 5, deoarece nu conține join dependențe.

17. DENORMALIZAREA

Pentru relația DETINE cu cardinalitatea maximă m:n dintre CITITOR și ABONAMENT ar fi util să duplicăm numele cititorului și tipul abonamentului. Este important să avem în același loc aceste date împreună cu cele deja existente(id_cititor, id_abonament, data_inceput, numar_luni) pentru a nu mai fi nevoie de atât de multe join-uri pentru a afla anumite date. De exemplu, pentru a afla ce tip de abonamente a avut cititorul cu numele 'Rogoza' , care a fost data de început a lor și cât au durat nu mai este nevoie să se facă join-urile dintre DETINE cu CITITOR și DETINE cu ABONAMENT, ci toate datele se află în DETINE.

Cerere SQL înainte:

```
select c.id_cititor, a.id_abonament, a.tip, d.data_inceput, d.numar_luni
from cititori c, detine d, abonamente a
where d.id_cititor = c.id_cititor and a.id_abonament = d.id_abonament and c.num = 'Rogoza';
```

Cerere SQL după:

```
select id.cititor, id_abonament, tip, data_inceput, numar_luni
from detine
where nume='Rogoza';
```