

Facultatea de Matematică și Informatică

Universitatea din București

Temă de casă
Proiectarea Bazelor de Date

Student, Marin Constantin – Cătălin

Grupa 354

Cuprins:

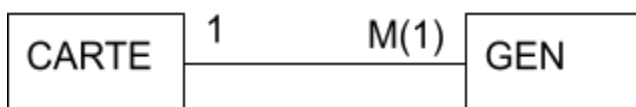
1. (1,5p)
 - a) Să se dea un exemplu de atribut repetitiv (multivaloare) al unei entități în modelul entitate-legătură.
 - b) Să se dea un exemplu de atribut repetitiv (multivaloare) al unei relații mulți-la-mulți în modelul entitate-legătură.
 - c) Să se arate cum se transformă atributele de mai sus la crearea design-ului logic al unei baze de date relaționale.
2. (1p)
 - a) Să se dea un exemplu de relație de tip 3 (între mai mult de două entități) în modelul entitate-legătură.
 - b) Să se dea un exemplu de trei sau mai multe entități care nu formează o relație de tip 3, ci, relația aparentă de tip 3, „se sparge” de fapt în relații mulți-la-mulți (între câte două entități).
3. (1p)
 - a) Să se dea un exemplu de tabel relațional care este în FN1, dar nu în FN2. Să se aducă tabelul în FN2.
 - b) Să se dea un exemplu de tabel relațional care este în FN2, dar nu în FN3. Să se aducă tabelul în FN3.
4. (0,5p) Să se dea un exemplu de tabel relațional în care există o dependență multivaloare (multidependență) între atributele (coloanele) sale, care nu este dependență funcțională.
5. (1p)
 - a) Să se ilustreze printr-un exemplu structura unui index de tip arbore B* și modul în care o interogare SQL folosește acest index.
 - b) Să se ilustreze printr-un exemplu structura unui index de tip bitmap și modul în care o interogare SQL folosește acest index.
6. (0,5p) Să se ilustreze printr-un exemplu modul în care o vedere (vizualizare) poate fi folosită pentru a asigura securitatea într-o bază de date.
7. (1p) Să se arate printr-un exemplu în ce condiții este posibil ca două select-uri identice consecutive (fără nici o altă comandă între ele), efectuate în aceeași sesiune de lucru, pe același tabel, pot produce rezultate diferite.
8. (0,5p) Să se dea un exemplu care să ilustreze interblocarea.
9. (1p) Să se ilustreze printr-un exemplu utilizarea unui trigger pentru a realiza o constrangere de integritate care nu ar putea fi implementată folosind un constraint din definiția unui tabel.
10. (1p) Să se ilustreze printr-un exemplu de program PL/SQL multi-bloc modul de propagare a excepțiilor. Vor fi ilustrate cel puțin situațiile în care o excepție este tratată sau nu în blocul curent și în care controlul programului va fi transmis blocului următor din secvență sau blocului exterior.
11. (1p) Să se ilustreze prin exemple folosirea instrucțiunii RAISE pentru a ridica atât o excepție predefinită cât și o excepție definită de utilizator. În cazul excepțiilor predefinite, să se explice cum anume folosirea instrucțiunii RAISE schimbă funcționalitatea programului (față de cazul când această instrucțiune nu există).

Rezolvări

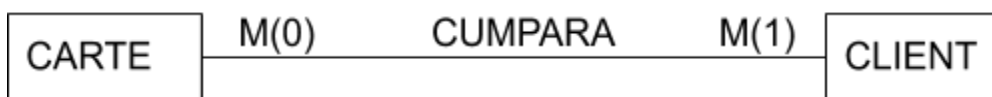
1. a) Un exemplu de atribut repetitive/multivaloare, este **GENUL** unei **CARTI** deoarece, o carte poate apartine mai multor genuri.

b) In cadrul relatiei N:M “CUMPARA”, dintre entitatile CARTE si CLIENT, putem considera ca atribut repetitiv COMANDA, deoarece un client poate da un numar de comenzi.

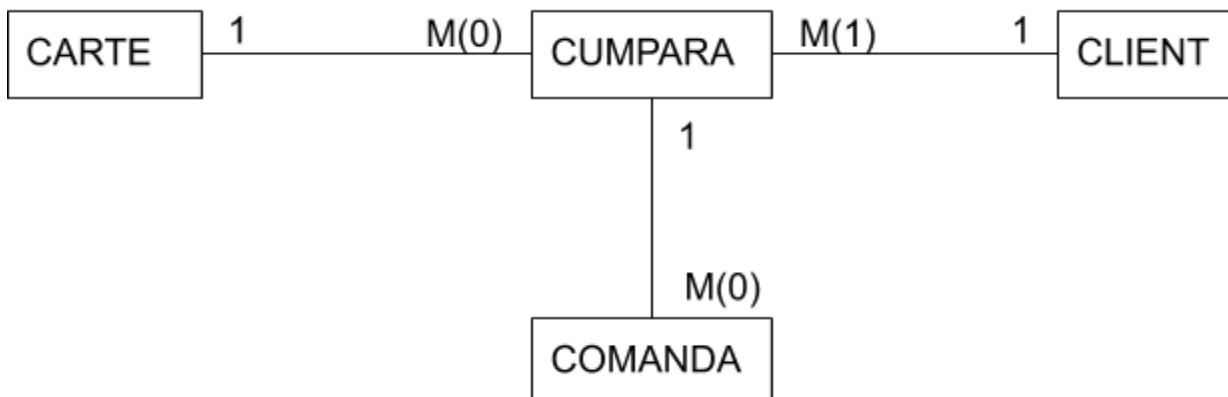
c) Pentru cazul de la subpunctul a)



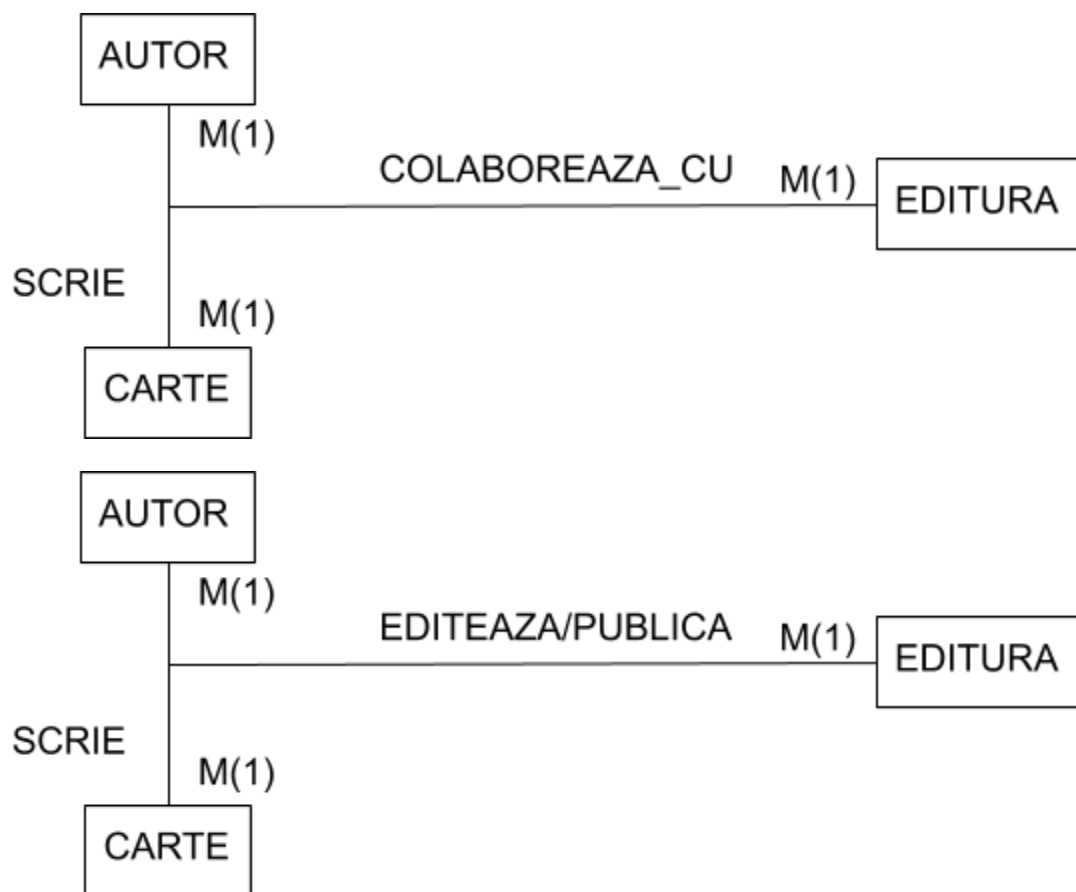
Pentru cazul de la subpunctul b)



Dupa transformare se obtine:

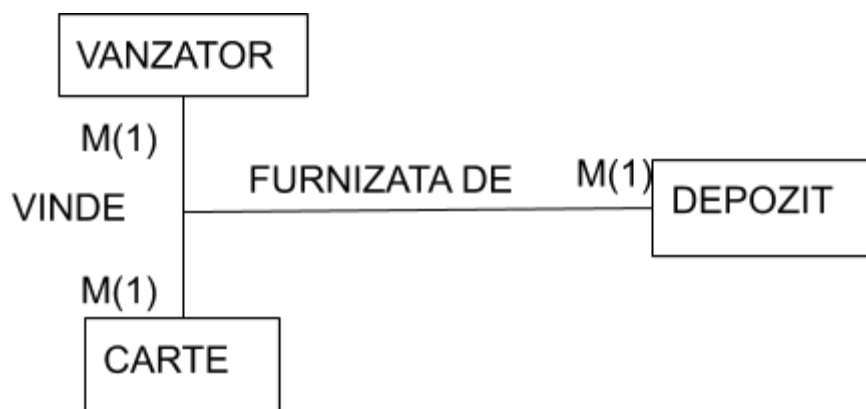


2. a) Pentru a ilustra o relatie de tip 3 am ales urmatorul exemplu:



Am considerat ca un autor scrie mai multe carti si ca fie acesta colaboreaza cu una sau mai multe edituri, fie ca aceste carti sunt editate/publicate de una sau mai multe edituri.

b) Relatia de tip 3 ar fi fost:



dar aceasta a fost “sparta” in relatiile N:M:



3. O să presupunem entitatea PARTENERIAT(id_autor, nume, prenume, email, id_carte, titlu, data_publicarii, pret, id_editura, oras, cantitate, telefon). De asemenea vom respecta regulile urmatoare:

- I. Autorul are un singur email;
- II. Editura are un singur numar de telefon si(optional, are un singur oras)

a) Astfel, tabelul este in FN1. Vom alege PK: {id_autor, id_editura}

Avem urmatoarele dependente de PK:

- A. {id_autor} -> {nume, prenume, email, id_carte, titlu, data_publicarii, pret}
- B. {id_editura} -> {oras, cantitate, telefon}

Eliminam dependenta A:

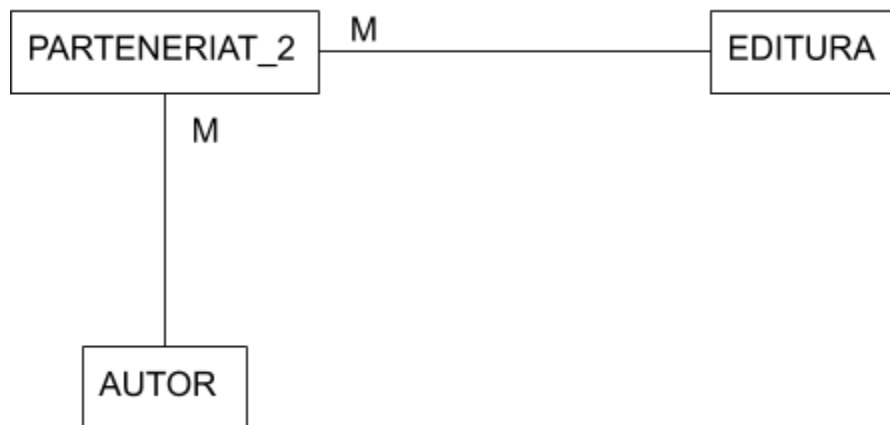
PARTENERIAT -> PARTENERIAT_1 = {id_autor, id_editura, oras, cantitate, telefon}

AUTOR = {id_autor, nume, prenume, email, id_carte, titlu, data_publicarii, pret}

Eliminam dependenta B:

PARTENERIAT_1 -> PARTENERIAT_2 = {id_autor, id_editura}

EDITURA = {id_editura, oras, cantitate, telefon}



Tabelele obtinute sunt in FN2.

b) Tabelul AUTOR de la subpunctul anterior se afla in FN2, dar nu este in FN3. De asemenea, celelalte tabele, PARTENERIAT_2 si EDITURA sunt in FN3.

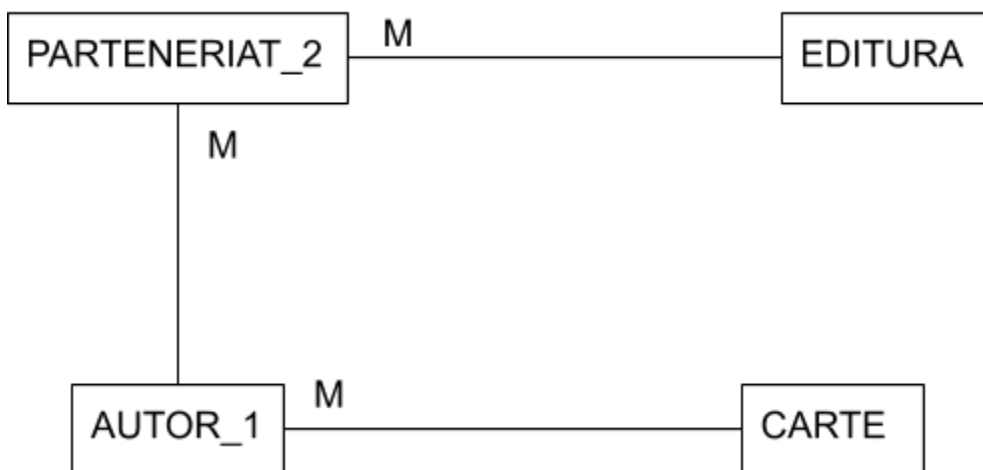
AUTOR = {id_autor, nume, prenume, email, id_carte, titlu, data_publicarii, pret}

Acest tabel nu este in FN3 deoarece avem dependenta tranzitiva:

$\{id_autor\} \rightarrow \{id_carte\} \rightarrow \{titlu, data_publicarii, pret\}$

AUTOR \rightarrow AUTOR_1 = $\{id_autor, nume, prenume, email, id_carte\}$

CARTE = $\{id_carte, titlu, data_publicarii, pret\}$



Tabelele obtinute sunt in FN3.

4. Vom presupunem urmatorul tabel numit CARTE(titlu, gen, editura) in BCNF:

Titlu	Gen	Editura
"The Witcher"	Fantasy	Nemira
"The Witcher"	Drama	Nemira
"The Witcher"	Fantasy	Gollancz
"The Witcher"	Drama	Gollancz
"Lord of the rings"	Fantasy	Del Rey
"Lord of the rings"	Adventure	Del Rey
"Lord of the rings"	Fantasy	Allen&Unwin
"Lord of the rings"	Adventure	Allen&Unwin

O carte poate avea mai multe genuri și poate fi publicată de mai multe edituri. Din cauza dependentelor multivaloare această tabelă nu este în FN4. Astfel, această redundanță a datelor este cauzată de relațiile N:M independente:



Vom elimina aceste dependente ($\{titlu\} \rightarrow \{gen\}$ și $\{titlu\} \rightarrow \{editura\}$) și obținem:

CARTE \rightarrow CARTE_1 $\{\underline{titlu}, \underline{gen}\}$

CARTE_2 $\{\underline{titlu}, \underline{editura}\}$

Titlu	Gen
“The Witcher”	Fantasy
“The Witcher”	Drama
“Lord of the rings”	Fantasy
“Lord of the rings”	Adventure

Titlu	Editura
“The Witcher”	Nemira
“The Witcher”	Gollancz
“Lord of the rings”	Del Rey
“Lord of the rings”	Allen&Unwin

Pentru următoarele exercitii ne vom folosi de modelul următor:

Diagrama Entitate-Relatie:

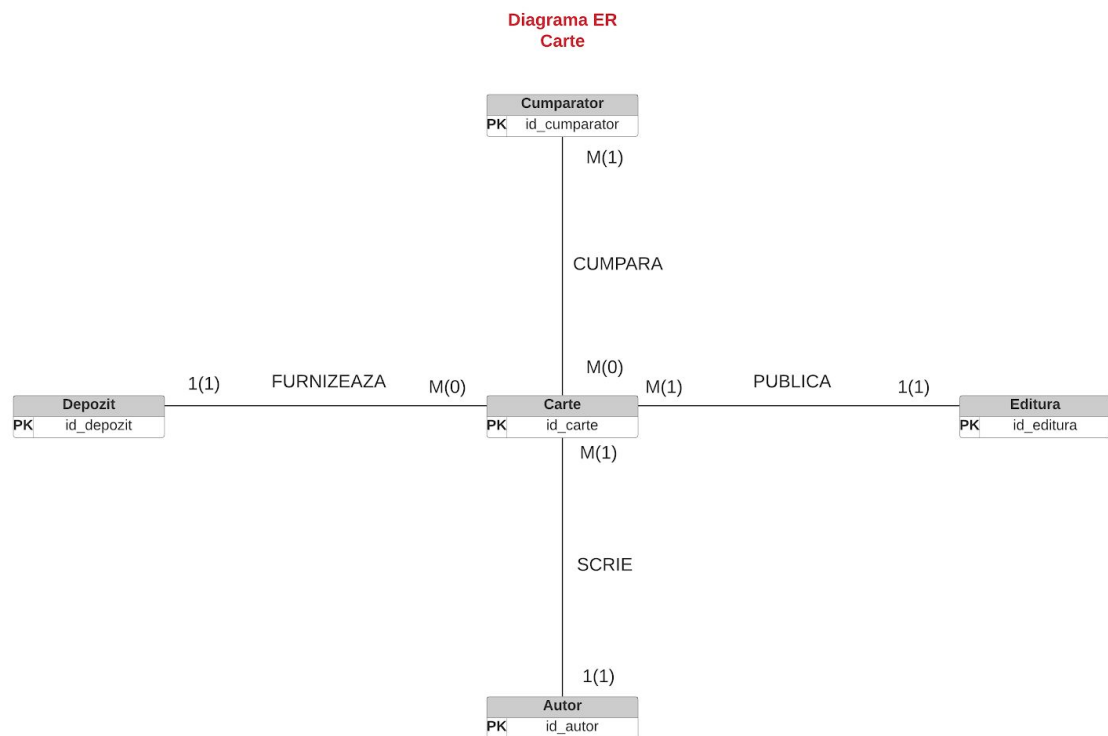
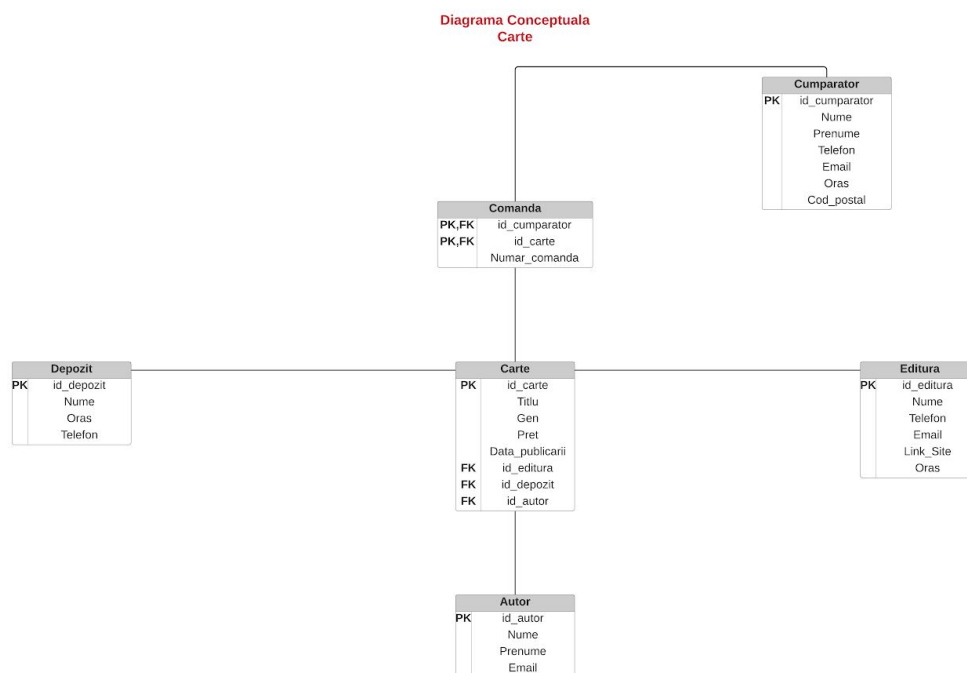


Diagrama Conceptuala:



Am aplicat urmatoarele reguli:

1. O carte poate fi publicata de o singura editura si poate avea un singur autor.
2. Aceasta are un singur gen si este furnizata de un singur depozit.
3. Autorul si clientul au un singur prenume.
4. Depozitul, clientul si editura au un singur numar de telefon.
5. Autorul si editura au un singur email.
6. Optional: un cumparator poate comanda o singura data aceeasi carte.

```
--Create tabelle
```

--Autor

```
create table
```

Author (

```
id autor number(3) constraint autor pk primary key,
```

```
Nume varchar2(25) constraint ver num aut not null,
```

```
Prenume varchar2(45) constraint ver pren aut not null,
```

```
Email varchar2(35) constraint unic_email aut unique not null
```

) ;

```
--Depozit
create table
Depozit(
id_depozit number(3) constraint dep_pk primary key,
Nume varchar2(25) constraint ver_num_dep not null,
Oras varchar2(25) constraint verif_or_dep not null,
Telefon varchar2(20) constraint unic_tel_dep unique not null
);
```

```
--Editura
create table
Editura(
id_editura number(3) constraint edi_pk primary key,
Nume varchar2(25) constraint ver_num_edi not null,
Telefon varchar2(20) constraint unic_tel_edi unique not null,
Email varchar2(35) constraint unic_email_edi unique not null,
LinkSite varchar2(30) constraint unic_ls_edi unique not null,
Oras varchar2(25) constraint verif_or_edi not null
);
```

```
--Carte
create table
Carte(
id_carte number(3) constraint car_pk primary key,
Titlu varchar2(30) constraint unic_tit_car unique not null,
Gen varchar2(15) constraint ver_gen_car not null,
Pret number(7,2) constraint ver_pret_car not null,
Data_publicarii Date constraint verif_dp_car not null,
```

```
id_editura      constraint      fk_id_editura      references
Editura(id_editura),

id_depozit      constraint      fk_id_depozit      references
Depozit(id_depozit),

id_autor constraint fk_id_autor references Autor(id_autor)
);

--

alter table Carte
drop constraint fk_id_editura;

alter table Carte
add constraint fk_id_editura foreign key(id_editura)
references Editura(id_editura) on delete cascade;
--

alter table Carte
drop constraint fk_id_depozit;

alter table Carte
add constraint fk_id_depozit foreign key(id_depozit)
references Depozit(id_depozit) on delete cascade;
--

alter table Carte
drop constraint fk_id_autor;

alter table Carte
add constraint fk_id_autor foreign key(id_autor)
references Autor(id_autor) on delete cascade;

--Cumparator
```

```
create table
Cumparator(
id_cumparator number(3) constraint cump_pk primary key,
Nume varchar2(25) constraint ver_num_cump not null,
Prenume varchar2(45) constraint ver_pren_cump not null,
Telefon varchar2(20) constraint unic_tel_cump unique not null,
Email varchar2(35) constraint unic_email_cump unique not null,
Oras varchar2(25) constraint verif_or_cump not null,
Cod_postal char(6) constraint verif_cp_cump not null
);

--Comanda
create table
Comanda(
id_cumparator number(3),
id_carte number(3),
Numar_comanda number(4) constraint ver_nrc_com not null unique,

constraint fk_id_com_cump foreign key (id_cumparator) references
Cumparator(id_cumparator),

constraint fk_id_com_car foreign key (id_carte) references
Carte(id_carte),

constraint com_pk primary key (id_cumparator, id_carte)
);

--
alter table Comanda
drop constraint fk_id_com_cump;

alter table Comanda
```

```
add constraint fk_id_com_cump foreign key (id_cumparator)
references Cumparator(id_cumparator) on delete cascade;
--
alter table Comanda
drop constraint fk_id_com_car;

alter table Comanda
add constraint fk_id_com_car foreign key (id_carte)
references Carte(id_carte) on delete cascade;

--Inserare date

--Autor
insert                                into                                Autor
values(1,'Andrzej','Sapkowski','and_sap@gmail.com');
insert into Autor values(2,'George','Martin','geo_mr@gmail.com');
insert                                into                                Autor
values(3,'Constantin','Chirita','ctin@gmail.com');
insert into Autor values(4,'John','Tolkien','jo.tolk@gmail.com');

--Depozit
insert into Depozit values(1,'Damila','Bucuresti','0745221939');
insert into Depozit values(2,'Depo','Varsovia','48225082000');
insert                                into                                Depozit
values(3,'Booksforall','Berlin','493022385762');
insert into Depozit values(4,'BookKing','Cluj','0768245399');

--Editura
insert                                into                                Editura
values(1,'Nemira','0772543995','nemi@gmail.com','https://nemira.ro/', 'Bucuresti');
```

```
insert                                into                                Editura
values(2,'Gollancz','48815280500','gol.cz@yahoo.com','https://gol
lancz.com/','Lublin');
```

```
insert                                into                                Editura
values(3,'DelRey','351215862607','dr@yahoo.com','https://delrey.c
om/','Lisabona');
```

```
insert                                into                                Editura
values(4,'AllenUnwin','442079300381','allu@gmail.com','https://al
lenunwin.com/','Londra');
```

--Carte

```
insert                                into                                Carte                                values(1,'The
Witcher','Fantasy',150,TO_DATE('17/12/2008',
'DD/MM/YYYY'),1,2,1);
```

```
insert                                into                                Carte                                values(2,'Lord      of      the
rings','Fantasy',200,TO_DATE('18/06/2005', 'DD/MM/YYYY'),2,3,4);
```

```
insert                                into                                Carte                                values(3,'The
Hobbit','Fantasy',125,TO_DATE('27/08/2002', 'DD/MM/YYYY'),3,1,4);
```

```
insert                                into                                Carte                                values(4,'The      Dark
Tower','Adventure',100,TO_DATE('30/12/2010',
'DD/MM/YYYY'),1,4,3);
```

```
insert                                into                                Carte                                values(5,'Record      of
Ragnarok','Fantasy',175,TO_DATE('21/03/2016',
'DD/MM/YYYY'),4,3,2);
```

```
insert                                into                                Carte                                values(6,'Attack      on
Titan','Drama',225,TO_DATE('11/04/2015', 'DD/MM/YYYY'),4,4,2);
```

```
insert                                into                                Carte                                values(7,'Naruto','Adventure',190,TO_DATE('14/08/2004',
'DD/MM/YYYY'),2,1,3);
```

```
insert                                into                                Carte                                values(8,'One
piece','Adventure',240,TO_DATE('15/06/2000',
'DD/MM/YYYY'),1,4,3);
```

```
insert                                into                                Carte                                values(9,'A      song      of      ice      and
fire','Fantasy',250,TO_DATE('16/02/2014', 'DD/MM/YYYY'),3,4,2);
```

```
insert                                into                                Carte                                values(10,'Demon
Slayer','Drama',210,TO_DATE('11/09/2019', 'DD/MM/YYYY'),2,3,1);
```

--Cumparator

```
insert                                into                                Cumparator
values(1,'Kamado','Tanjiro','0768877562','ktaj@gmail.com','Bucure
sti','061113');
```

```
insert                                into                                Cumparator
values(2,'Ion','Mironel','442078395240','iomir@gmail.com','Manche
ster','332000');
```

```
insert                                into                                Cumparator
values(3,'William','Kaiser','493011285992','wilk@gmail.com','Berl
in','244058');
```

```
insert                                into                                Cumparator
values(4,'Vlaicu','Aurel','34914353110','vl.aur@gmail.com','Madri
d','300041');
```

--Comanda

--Cump1

```
insert into Comanda values(1,10,1);
insert into Comanda values(1,1,2);
insert into Comanda values(1,5,3);
insert into Comanda values(1,6,4);
```

--Cump2

```
insert into Comanda values(2,2,5);
insert into Comanda values(2,1,6);
insert into Comanda values(2,3,7);
insert into Comanda values(2,9,8);
insert into Comanda values(2,4,9);
```

--Cump3

```
insert into Comanda values(3,7,10);
insert into Comanda values(3,5,11);
insert into Comanda values(3,6,12);
```

```
insert into Comanda values(3,8,13);
```

```
--Cump4
```

```
insert into Comanda values(4,4,14);
```

```
insert into Comanda values(4,10,15);
```

```
insert into Comanda values(4,6,16);
```

```
insert into Comanda values(4,5,17);
```

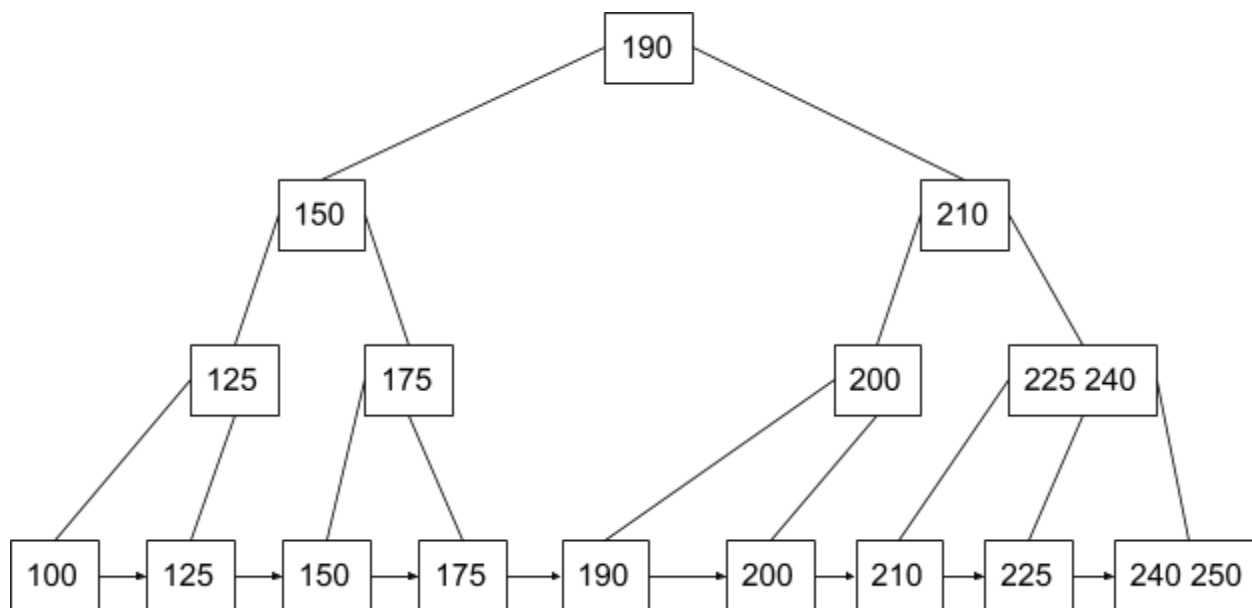
```
insert into Comanda values(4,1,18);
```

Tabelele sunt in FN3.

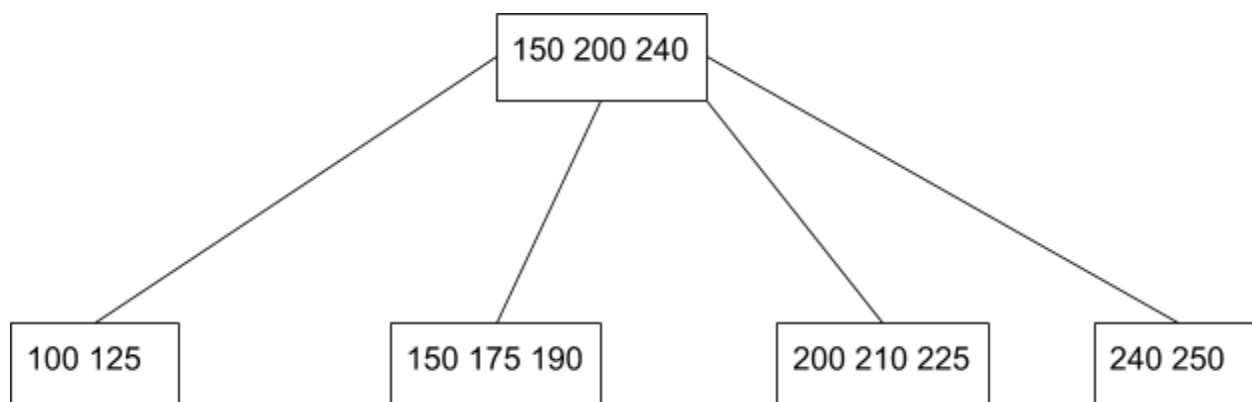
5. a) Pentru un index de tip arbore B* am ales ca si exemplu coloana Pret din entitatea Carte.

Am ales ca numarul maxim de elemente dintr-un nod, adica la care sa se faca separarea valorilor sa fie 4.

Daca consideram valorile pretului sortate crescator obtinem rezultatul:



Daca consideram valorile pretului asa cum apar in tabel, neordonate:



```
select *
from carte
where pret = 100;
```

```
create index pret_idx
on Carte(pret);
```

```
explain plan for
select *
from carte
where pret = 200;
```

```
select plan_table_output
from table(dbms_xplan.display());
```

Modul in care lucreaza acest index este asemanator cu cautarea binara, adica valoarea cautata se compara cu valorile superioare pana cand ajungem la frunza unde gasim valoarea respectiva.

b) Pentru a ilustra structura unui index de tip bitmap, am ales ca si exemplu coloana Gen din cadrul tablei Carte.

Gen = Fantasy	Gen = Adventure	Gen = Drama
1	0	0

1	0	0
1	0	0
0	1	0
1	0	0
0	0	1
0	1	0
0	1	0
1	0	0
0	0	1

```
select *
from carte
where lower(gen) = 'fantasy';
```

```
create bitmap index gen_idx
on Carte(gen);
```

```
explain plan for
select *
from carte
where lower(gen) = 'fantasy';
```

```
select plan_table_output
from table(dbms_xplan.display());
```

O interogare va numara valorile de 1 din bitmap-ul genului in cauza. Spre exemplu, ar fi echivalent cu cererea:

```
select count(*)
from carte
```

```
where lower(gen) = 'fantasy';
```

Deoarece, avem putine date, calea cea mai optima pentru acest exemplu este sa fie incarcata toata tabela. Un necesar de date ar fi fost un numar pentru care raportul dintre numarul de valori distincte si numarul de inregistrari ar fi fost mic. (Sursa de inspiratie pentru acest paragraf: “The advantages of using bitmap indexes are greatest for columns in which the ratio of the number of distinct values to the number of rows in the table is small. For example, on a table with one million rows, a column with 10,000 distinct values is a candidate for a bitmap index.”)

(<https://docs.oracle.com/database/121/DWHSG/schemas.htm#DWHSG9042>).

6. O vizualizare ca sa asigure securitatea bazei de date, poate contine mai putine campuri decat tabela dupa care este creata si astfel daca oferim unor utilizatori doar drepturi pe vizualizare si nu pe tabela in sine, nu vor putea vedea decat campurile vizualizarii. De asemenea, putem oferi drepturi pe acea vizualizare doar anumitor utilizatori. Acest lucru este realizabil prin: grant “drepturi” on nume_tabela/nume_view si revoke “drepturi” on nume_tabela/nume_view. Pentru a oferi un strat in plus de securitate putem folosi la crearea vizualizarilor constrangerile with check option sau with read only care limiteaza efectul operatiilor de manipulare a datelor efectuate asupra vizualizarilor pe tabelele de baza.

```
create or replace view cumparator_sec (prenume)
as
select prenume
from cumparator
with read only;
```

7. Pentru acest exercitiu am folosit codul:

```
select distinct to_char(CURRENT_TIMESTAMP,'hh24:mi:ss.ff')
from carte;
```

```
select distinct to_char(CURRENT_TIMESTAMP,'hh24:mi:ss.ff')
from carte;
```

--SAU

```
select distinct to_char(SYSTIMESTAMP,'hh24:mi:ss.ff')
from carte;
```

```
select distinct to_char(SYSTIMESTAMP,'hh24:mi:ss.ff')
from carte;
```

8. Pentru a sugera interblocarea am luat tabelele Editura si Autor pe care am aplicat doua tranzactii, T1 si T2 care contin doua update-uri. Pentru a realiza cele doua tranzactii am deschis doua sql developer-uri conectate cu aceeasi conexiune si vom rula codurile de mai jos in ordinea primul update din T1, apoi primul update din T2, apoi al doilea update din T1 urmat de cel de-al doilea update T2:

```
--Tranzactia T1  
update autor  
set email = 'andr.sapk@yahoo.com'  
where id_autor = 1;
```

```
update editura  
set nume = 'Humanitas'  
where id_editura = 1;
```

```
rollback;  
--Tranzactia T2  
update editura  
set email = 'hum@gmail.com'  
where id_editura = 1;
```

```
update autor  
set nume = 'Andrew'  
where id_autor = 1;
```

```
rollback;
```

Dupa executarea codului in ordinea mentionata vom primi o eroare de interblocare.

9. Pentru a simula aceasta constrangere de integritate fara check, am considerat ca implica mai multe tabele. Astfel, am presupus ca editura Nemira va avea carti cu preturi cuprinse intre minimul preturilor din tabela carti si maximul acestora. Poate exista problema cand preturile, min si max, sa fie egale si atunci am ales sa verificam acest lucru. O alta solutie, care s-ar putea folosi cand chiar dorim sa inseram/updatam inregistrările acestea, este sa ne alegem un range, spre exemplu $\text{min_pret} - 50$ si $\text{max_pret} + 50$.

```
set serveroutput on;

create or replace trigger limita_carte
for insert or update of pret on carte
compound trigger
    min_pr carte.pret%type;
    max_pr carte.pret%type;
    nr int;
    eroare_pret exception;
before statement is
begin
    select min(pret), max(pret), count(*)
    into min_pr, max_pr, nr
    from carte;
end before statement;

before each row is
begin
    if (nr>=2 and min_pr!=max_pr) and ((:old.id_editura = 1) or
(:new.id_editura = 1)) and (min_pr > :new.pret or max_pr <
:new.pret)
        then raise eroare_pret;
    end if;

exception
    when eroare_pret then
        DBMS_OUTPUT.PUT_LINE('Eroare, pretul introdus nu s-a
incadrat in interval');
        raise;
    when others then
```

```
        DBMS_OUTPUT.PUT_LINE('Alte erori');
end before each row;
end limita_carte;
/

update carte
set pret = 15
where id_carte = 1;

insert      into      carte      values(11,'Vrajitorul      din
Oz','Fantasy',1000,TO_DATE('13/06/2002', 'DD/MM/YYYY'),1,2,1);
rollback;

/*
create or replace trigger limita_carte_range
for insert or update of pret on carte
compound trigger
    min_pr carte.pret%type;
    max_pr carte.pret%type;
    nr int;
    eroare_pret exception;
before statement is
begin
    select min(pret), max(pret), count(*)
    into min_pr, max_pr, nr
    from carte;
end before statement;

before each row is
```

```
begin
    if (nr>=2 and min_pr!=max_pr) and ((:old.id_editura = 1) or
(:new.id_editura = 1)) and (min_pr - 50 > :new.pret or max_pr +
50 < :new.pret)
        then raise eroare_pret;
    end if;

exception
    when eroare_pret then
        DBMS_OUTPUT.PUT_LINE('Eroare, pretul introdus nu s-a
incadrat in interval');
        raise;
    when others then
        DBMS_OUTPUT.PUT_LINE('Alte erori');
end before each row;
end limita_carte_range;
/
*/
```

10. Am ales sa tratez exceptiile pe baza tabelului Carte:

```
declare

begin
    declare
        valoare carte.pret%type;
        eroare_pret exception;
    begin
        select min(pret)
        into valoare
        from carte;
```

```
        if (valoare < 110) then
            raise eroare_pret;
        end if;

    exception when eroare_pret then
        DBMS_OUTPUT.PUT_LINE('Eroare, pretul introdus n-a
fost mai mic ca ' || valoare);

    end;

declare
    valoare carte.pret%type;
    eroare_pret exception;
begin
    select min(pret)
    into valoare
    from carte;

    if (valoare < 150) then
        raise eroare_pret;
    end if;

    exception when eroare_pret then
        DBMS_OUTPUT.PUT_LINE('Eroare, pretul introdus n-a
fost mai mic ca ' || valoare);
        raise;

    end;

exception
```



```
        when others then
            DBMS_OUTPUT.PUT_LINE('Alte erori');

end;
/
```

Astfel, se poate observa ca blocul 1 imbricat prinde exceptia si o afiseaza, apoi blocul 2 va arunca o exceptie, si aceasta va fi prinsa de blocul exterior.

11. In cazul exceptiei predefinite:

```
declare
    valoare carte.pret%type;
begin
    select pret
    into valoare
    from carte
    where id_carte=11;
    --where pret>100;

    raise no_data_found;
    --raise too_many_rows;

exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Nu exista cartea cu id-ul
respectiv');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Prea multe randuri');
end;
/
```

In cazul exceptiei definite de utilizator:

```
declare
    valoare carte.pret%type;
    medie carte.pret%type;
    pret_mic exception;
begin
    select pret
    into valoare
    from carte
    where id_carte=3;

    select avg(pret)
    into medie
    from carte;

    if valoare < medie then
        raise pret_mic;
    end if;
exception
    when pret_mic then
        DBMS_OUTPUT.PUT_LINE('Pretul este mai mic decat media');
    --when too_many_rows then
        --DBMS_OUTPUT.PUT_LINE('Prea multe randuri');
end;
/
```

In cazul erorii predefinite, daca comentam raise-ul putem observa ca programul se executa in continuare deoarece erorile respective sunt tratate in exception, ca implica automat tratarea acelei erori in blocul exception si ca limiteaza un utilizator doar la detaliile/solutiile oferite in blocul exception:

```
declare
    valoare carte.pret%type;
```

```
begin
    select pret
    into valoare
    from carte
    where id_carte=11;
    --where pret>100;

    --raise no_data_found;
    --raise too_many_rows;

exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Nu exista cartea cu id-ul
respectiv');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Prea multe randuri');
end;
/
```

Astfel, in cazul in care nu folosim raise, programul ofera informatii mai explicite utilizatorului in legatura cu erorile.