Tema proiectului

Sa se creeze o baza de date care tine evidenta activitatii unei biblioteci. Biblioteca are

mai multe sucursale, fiecare avand nume, numar unic si adresa. Fiecare carte are un numar unic ,

titlu, autori si editura unde a fost publicata. Fiecare sucursala are mai multe copii ale unei carti.

Cititorii pot imprumuta mai multe carti de la diferite sucursale ale bibliotecii, pe o perioada

limitata. Cititorii sunt identificati prin numar de fisa, nume, adresa si telefon. Se va reţine de

asemenea data la care o carte a fost împrumutată, data returnarii ei, iar in cazul in care nu a fost

returnata la timp, cel care a imprumutat-o va fi trecut in tabelul de client nedorit.

## Cerinte:

1. Să se realizeze modelul entitate-legătură

2. Să se facă trecerea la modelul relaţional (să se detalieze operaţiile effectuate pentru

fiecare entitate în parte)

3. Să se creeze o bază de date care să conţină:

a. Legături între tabele;

b. Integritatea referenţială pentru a efectua corect operaţiile de actualizare (adăugare,

ştergere, modificare);

c. Chei primare şi chei externe;

d. Reguli de ştergere şi modificare în cascadă pentru entităţile cu chei externe.

4. Să se verifice calitatea proiectării bazei de date cu ajutorul formelor normale (se va

normaliza baza de date până la FNBC - inclusiv).

5. Să se indexeze baza de date creată (crearea minim a unui index).

6. Să se facă testarea bazei de date prin efectuarea de diferite operaţii asupra ei. Trebuie

să existe cel puţin o interogare din fiecare tip (în total minim 15 interogări), 2 funcţii

diferite (care să nu facă un simplu select), 2 proceduri stocate, şi 1 trigger.

\* Baza de date trebuie să conţină minim 10 tabele

\*\* Se va adăuga în tabele un volum mare de informaţii reale – minim 20 înregistrări în

fiecare tabel

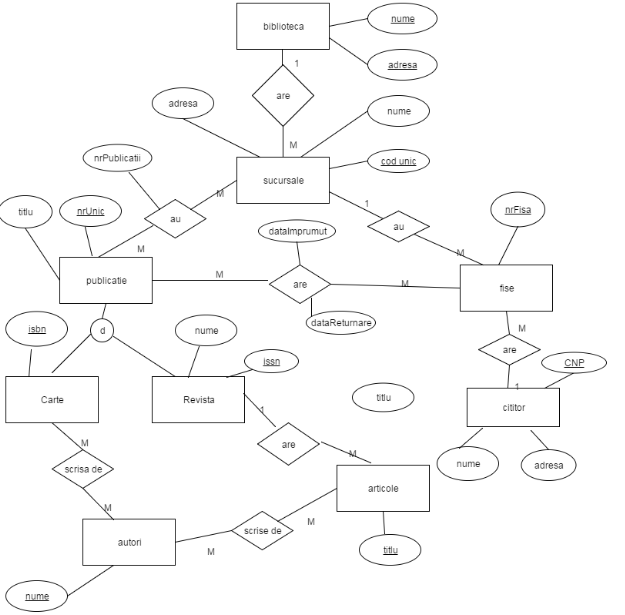
\*\*\* Baza de date trebuie să fie realizată în Oracle

\*\*\*\* Se vor folosi programele Visio sau Embarcadero pentru crearea modelului entitatelegătură

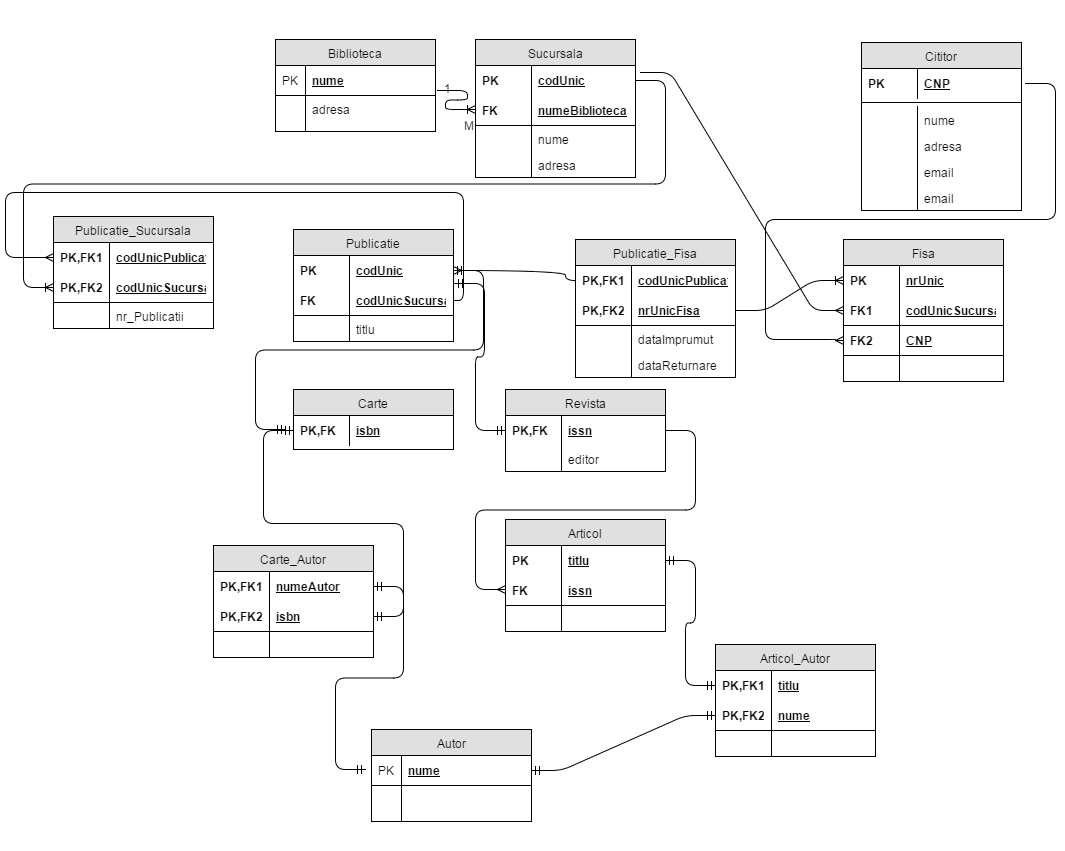
(listarea pentru dosar a modelului Entitate Legătură)

\*\*\*\*\* Fiecare cerinţă are punctaj separat

## Modelul entitate-legatura



## Modelul relational



## Crearea bazei de date

CREATE TABLE "REVISTA"

( "ISSN" VARCHAR2(50) NOT NULL ENABLE,

"EDITOR" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "REVISTA\_PK" PRIMARY KEY ("ISSN") ENABLE,

CONSTRAINT "REVISTA\_FK" FOREIGN KEY ("ISSN")

REFERENCES "PUBLICATIE" ("CODUNIC") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "ARTICOL"

( "TITLU" VARCHAR2(50) NOT NULL ENABLE,

"ISSN" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "ARTICOL\_PK" PRIMARY KEY ("TITLU") ENABLE,

CONSTRAINT "ARTICOL\_FK" FOREIGN KEY ("ISSN")

REFERENCES "REVISTA" ("ISSN") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "AUTOR"

( "NUME" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "AUTOR\_PK" PRIMARY KEY ("NUME") ENABLE

)

/

CREATE TABLE "ARTICOL\_AUTOR"

( "TITLU" VARCHAR2(50) NOT NULL ENABLE,

"NUME" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "ARTICOL\_AUTOR\_PK" PRIMARY KEY ("TITLU", "NUME") ENABLE,

CONSTRAINT "ARTICOL\_AUTOR\_FK" FOREIGN KEY ("TITLU")

REFERENCES "ARTICOL" ("TITLU") ON DELETE CASCADE ENABLE,

CONSTRAINT "ARTICOL\_AUTOR\_FK2" FOREIGN KEY ("NUME")

REFERENCES "AUTOR" ("NUME") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "BIBLIOTECA"

( "NUME" VARCHAR2(50),

"ADRESA" VARCHAR2(50),

CONSTRAINT "BIBLIOTECA\_PK" PRIMARY KEY ("NUME") ENABLE

)

/

CREATE TABLE "PUBLICATIE"

( "CODUNIC" VARCHAR2(50) NOT NULL ENABLE,

"TITLU" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "PUBLICATIE\_PK" PRIMARY KEY ("CODUNIC") ENABLE

)

/

CREATE TABLE "CARTE"

( "ISBN" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "CARTE\_PK" PRIMARY KEY ("ISBN") ENABLE,

CONSTRAINT "CARTE\_FK" FOREIGN KEY ("ISBN")

REFERENCES "PUBLICATIE" ("CODUNIC") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "CARTE\_AUTOR"

( "NUME" VARCHAR2(50) NOT NULL ENABLE,

"ISBN" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "CARTE\_AUTOR\_PK" PRIMARY KEY ("NUME", "ISBN") ENABLE,

CONSTRAINT "CARTE\_AUTOR\_FK" FOREIGN KEY ("NUME")

REFERENCES "AUTOR" ("NUME") ON DELETE CASCADE ENABLE,

CONSTRAINT "CARTE\_AUTOR\_FK2" FOREIGN KEY ("ISBN")

REFERENCES "CARTE" ("ISBN") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "CITITOR"

( "CNP" VARCHAR2(13) NOT NULL ENABLE,

"NUME" VARCHAR2(50) NOT NULL ENABLE,

"ADRESA" VARCHAR2(50) NOT NULL ENABLE,

"EMAIL" VARCHAR2(50) NOT NULL ENABLE,

"PAROLA" VARCHAR2(64),

CONSTRAINT "CITITOR\_PK" PRIMARY KEY ("CNP") ENABLE

)

/

CREATE TABLE "SUCURSALA"

( "CODUNIC" VARCHAR2(50) NOT NULL ENABLE,

"NUMEBIBLIOTECA" VARCHAR2(50) NOT NULL ENABLE,

"NUME" VARCHAR2(50) NOT NULL ENABLE,

"ADRESA" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "SUCURSALA\_PK" PRIMARY KEY ("CODUNIC") ENABLE,

CONSTRAINT "SUCURSALA\_FK" FOREIGN KEY ("NUMEBIBLIOTECA")

REFERENCES "BIBLIOTECA" ("NUME") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "FISA"

( "NRUNIC" VARCHAR2(50) NOT NULL ENABLE,

"CODUNICSUCURSALA" VARCHAR2(50) NOT NULL ENABLE,

"CNP" VARCHAR2(13) NOT NULL ENABLE,

CONSTRAINT "FISA\_PK" PRIMARY KEY ("NRUNIC") ENABLE,

CONSTRAINT "FISA\_FK" FOREIGN KEY ("CODUNICSUCURSALA")

REFERENCES "SUCURSALA" ("CODUNIC") ON DELETE CASCADE ENABLE,

CONSTRAINT "FISA\_FK2" FOREIGN KEY ("CNP")

REFERENCES "CITITOR" ("CNP") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "PUBLICATIE\_FISA"

( "CODUNICPUBLICATIE" VARCHAR2(50) NOT NULL ENABLE,

"NRUNICFISA" VARCHAR2(50) NOT NULL ENABLE,

"DATAIMPRUMUT" DATE,

"DATARETURNARE" DATE,

CONSTRAINT "PUBLICATIE\_FISA\_PK" PRIMARY KEY ("CODUNICPUBLICATIE", "NRUNICFISA") ENABLE,

CONSTRAINT "PUBLICATIE\_FISA\_FK" FOREIGN KEY ("CODUNICPUBLICATIE")

REFERENCES "PUBLICATIE" ("CODUNIC") ON DELETE CASCADE ENABLE,

CONSTRAINT "PUBLICATIE\_FISA\_FK2" FOREIGN KEY ("NRUNICFISA")

REFERENCES "FISA" ("NRUNIC") ON DELETE CASCADE ENABLE

)

/

CREATE TABLE "PUBLICATIE\_SUCURSALA"

( "CODUNICPUBLICATIE" VARCHAR2(50) NOT NULL ENABLE,

"CODUNICSUCURSALA" VARCHAR2(50) NOT NULL ENABLE,

CONSTRAINT "PUBLICATIE\_SUCURSALA\_PK" PRIMARY KEY ("CODUNICPUBLICATIE", "CODUNICSUCURSALA") ENABLE,

CONSTRAINT "PUBLICATIE\_SUCURSALA\_FK" FOREIGN KEY ("CODUNICPUBLICATIE")

REFERENCES "PUBLICATIE" ("CODUNIC") ON DELETE CASCADE ENABLE,

CONSTRAINT "PUBLICATIE\_SUCURSALA\_FK2" FOREIGN KEY ("CODUNICSUCURSALA")

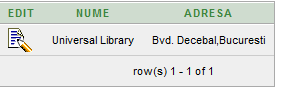
REFERENCES "SUCURSALA" ("CODUNIC") ON DELETE CASCADE ENABLE

)

/

Tabelele rezultate in urma operatiilor de inserare sunt:

Tabelul “Biblioteca”



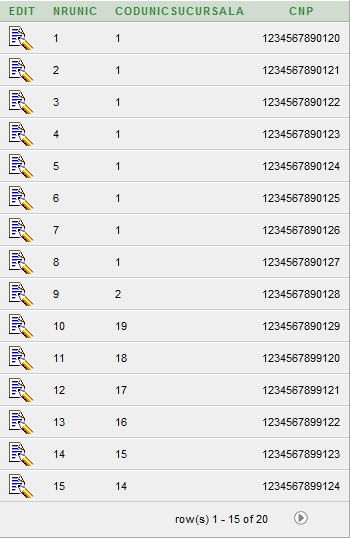
Tabelul “Sucursala”

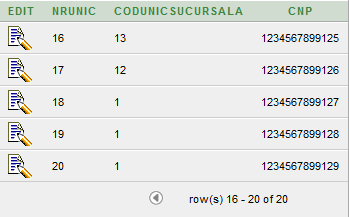




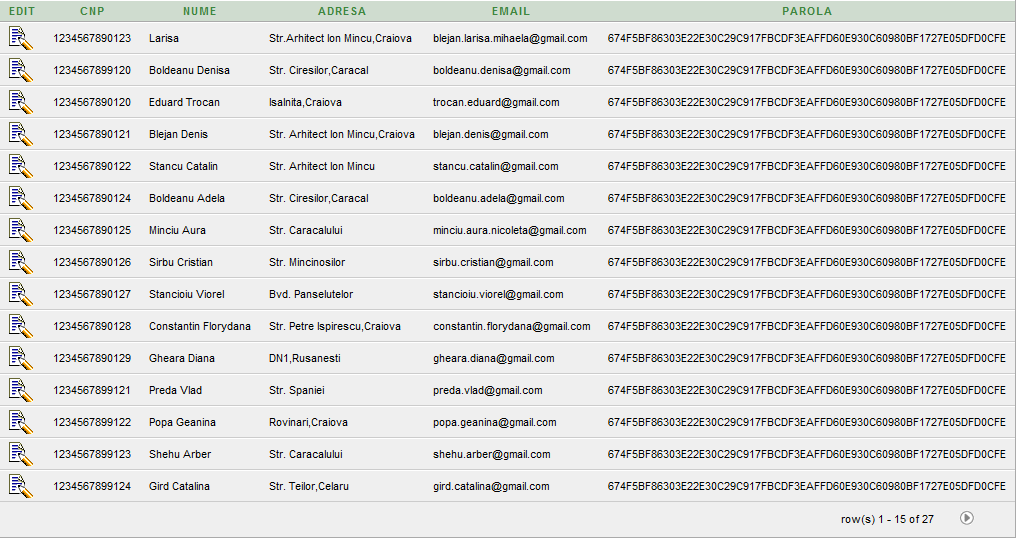


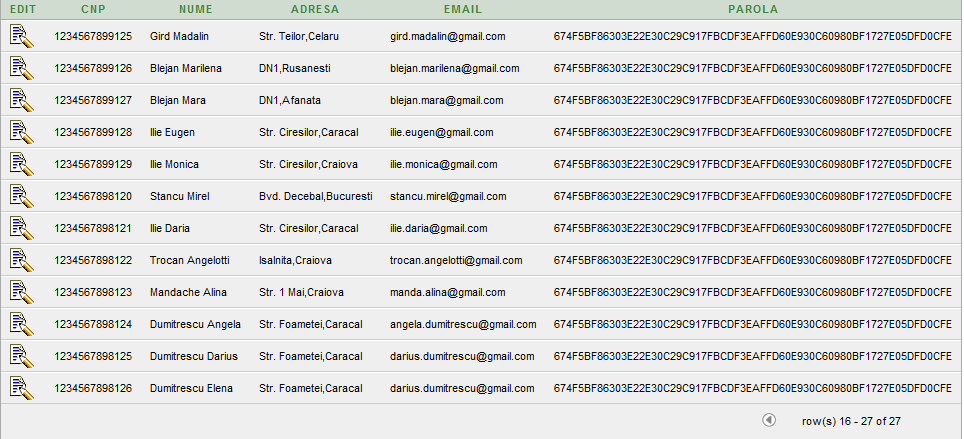
Tabelul “Fisa”



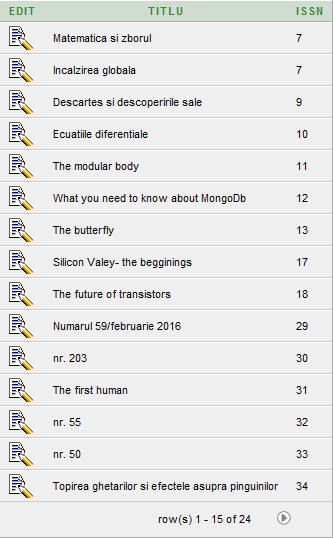


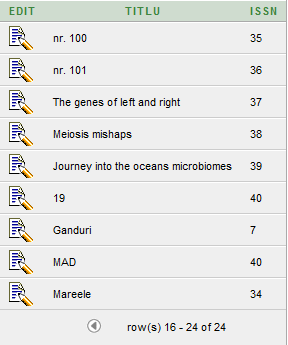
Tabelul “Cititor”



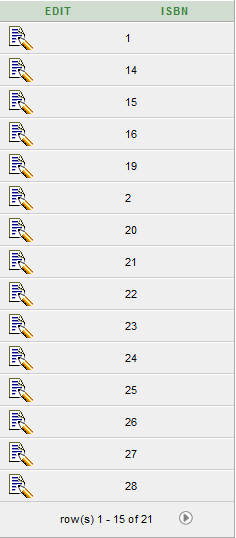


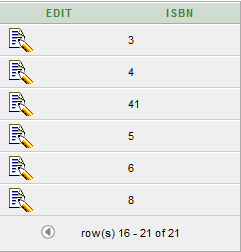
Tabelul “Publicatie”





Tabelul “Carte”





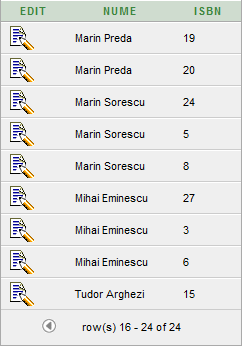
Tabelul “Autor”





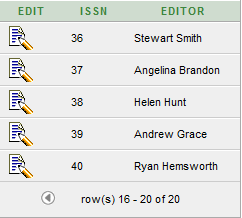
Tabelul "Carte\_Autor"



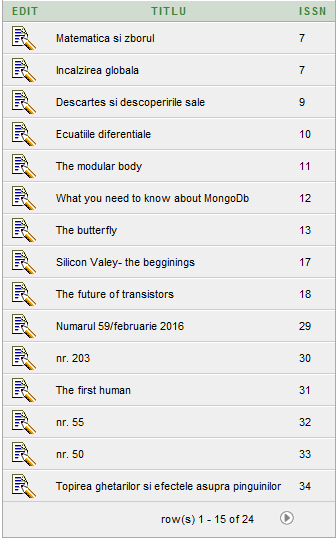


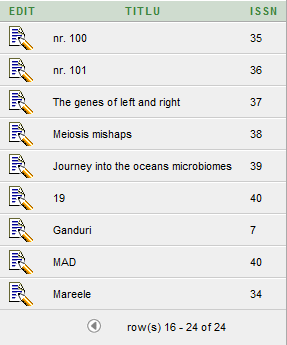
Tabelul "Revista"



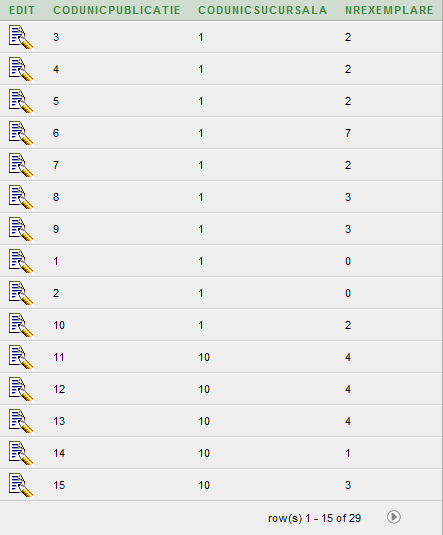


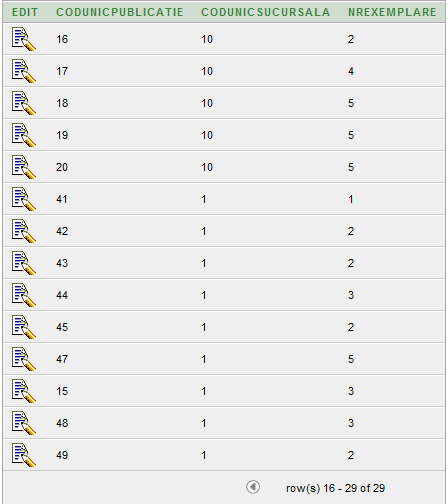
Tabelul "Articol"



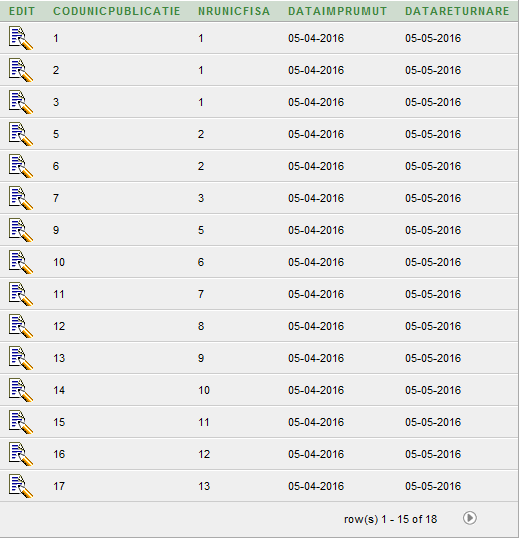


Tabelul “Publicatie\_Sucursala”



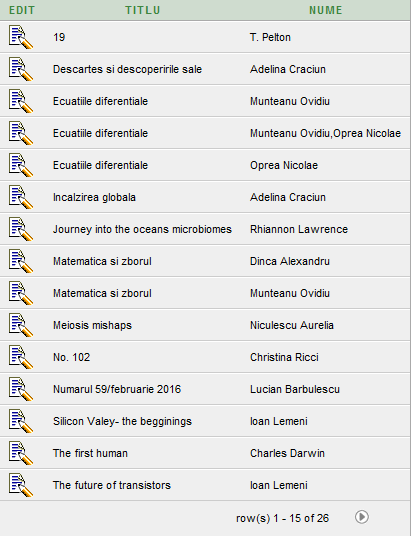


Tabelul “Publicatie\_Fisa”





Tabelul “Articol\_Autor”





## Normalizarea relatiilor

Baza de date creata este normalizata. Fiecare din cele 13 tabele create se afla in forma normala 1 deoarece fiecare atribut al fiecarei relatii are un domeniu de valori atomice.

Fiecare tabel se afla in FN2, deoarece orice atribut nonprim din R este complet dependent functional de toate cheile candidat din R.

Deoarece nu avem dependente tranzitive, nu mai este necesara verificarea bazei de date pentru FN3.

## Indexarea bazei de date

Sa se indexeze tabelul “Carte” dupa campul “nume” .

CREATE INDEX "CARTEINDEX" ON "CITITOR" ("NUME")

Sa se indexeze tabelul “Fisa” dupa campul “CNP” .

CREATE INDEX "FISAINDEX" ON "FISA" ("CNP")

## Interogari

1.Afisati autorul si titlul fiecarui articol + numele si ISSN-ul revistei de care apartine

ordonand datele obtinute dupa numele revistei si al articolului.

SELECT Publicatie.titlu AS Revista,Articol.titlu AS Articol,Autor.nume AS autor,Articol.ISSN FROM Articol,Revista,Publicatie,Autor,Articol\_Autor

WHERE Articol.ISSN=Revista.ISSN

and Revista.ISSN=Publicatie.codUnic

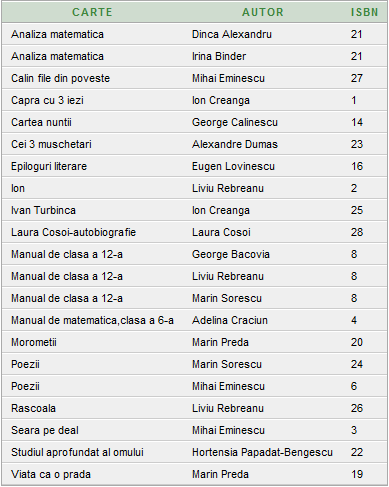
and Articol.titlu=Articol\_Autor.titlu

and Articol\_Autor.nume=Autor.nume

order by Revista,Articol



2.Afisati titlul,autorul si ISBN-ul fiecarei carti ordonand datele dupa titlul cartii si numele autorului.



3.Afisati toate cartile de aflate in sucursala dintr-un anumit oras si locatia exacta a sucursalei.

SELECT Publicatie.titlu AS Carte,Autor.nume AS autor,Carte.ISBN,Sucursala.adresa as AdresaSucursala

FROM Carte,Publicatie,Autor,Carte\_Autor,Sucursala,Publicatie\_Sucursala

WHERE Carte.ISBN=Publicatie.codUnic

and Carte\_Autor.ISBN=Carte.ISBN

and Carte\_Autor.nume=Autor.nume

and Sucursala.codUnic=Publicatie\_Sucursala.codUnicSucursala

and Publicatie\_Sucursala.codUnicPublicatie=Publicatie.codUnic

and Sucursala.adresa like '%Craiova%'

order by Carte,Autor



4. Sa se afiseze pentru un anumit cititor imprumuturile facute.

SELECT Cititor.nume,Publicatie.titlu,Fisa.nrUnic as Fisa from Cititor,Publicatie,Publicatie\_Fisa,Fisa

WHERE Cititor.CNP=Fisa.CNP

AND Fisa.nrUnic=Publicatie\_Fisa.nrUnicFisa

AND Publicatie\_Fisa.codUnicPublicatie=Publicatie.codUnic

AND Cititor.nume='Eduard Trocan'



5.Sa se afiseze numele si adresa de email ale cititorilor care locuiesc in Bucuresti.

SELECT nume,email

FROM Cititor

WHERE adresa LIKE '%Bucuresti%'

C:\Users\larisa\Pictures\Untitled.png

6.Sa se afiseze toate Articolele ce apartin de o anumita revista.

SELECT Publicatie.titlu AS Revista,Articol.titlu AS Articol,Autor.nume AS autor,Articol.ISSN FROM Articol,Revista,Publicatie,Autor,Articol\_Autor

WHERE Articol.ISSN=Revista.ISSN

and Revista.ISSN=Publicatie.codUnic

and Articol.titlu=Articol\_Autor.titlu

and Articol\_Autor.nume=Autor.nume

and Publicatie.titlu like '%Pi%'

order by Articol

C:\Users\larisa\Pictures\Untitled.png

7.Sa se afiseze titlul cartilor care nu sunt imprumutate de nimeni.

SELECT Publicatie.titlu,Publicatie.codUnic from Publicatie

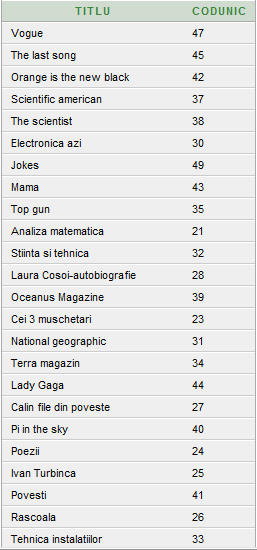
WHERE Publicatie.codUnic NOT IN

(SELECT Publicatie.codUnic from Cititor,Publicatie,Publicatie\_Fisa,Fisa

WHERE Cititor.CNP=Fisa.CNP

AND Fisa.nrUnic=Publicatie\_Fisa.nrUnicFisa

AND Publicatie\_Fisa.codUnicPublicatie=Publicatie.codUnic)



8.Sa se calculeze cate carti are fiecare Cititor imprumutate sortand datele dupa numele acestuia.

SELECT nume,count(Publicatie.codUnic) from Cititor,Publicatie,Publicatie\_Fisa,Fisa

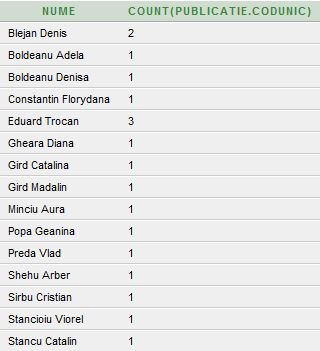
WHERE Cititor.CNP=Fisa.CNP

AND Fisa.nrUnic=Publicatie\_Fisa.nrUnicFisa

AND Publicatie\_Fisa.codUnicPublicatie=Publicatie.codUnic

group by nume

order by nume



9.Sa se afiseze cititorii care au trecut de termenul limita de returnare a cartilor.

SELECT nume,Publicatie.titlu,dataREturnare as "Data Returnare Obligatorie" from Cititor,Publicatie,Publicatie\_Fisa,Fisa

WHERE Cititor.CNP=Fisa.CNP

AND Fisa.nrUnic=Publicatie\_Fisa.nrUnicFisa

AND Publicatie\_Fisa.codUnicPublicatie=Publicatie.codUnic

AND Publicatie\_Fisa.dataReturnare < SYSDATE

order by nume



10.Sa se afiseze toate sucursalele si numarul de cititori pe care il au (0 daca nu au).

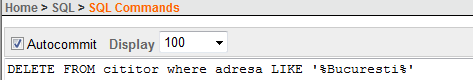
SELECT sucursala.nume,sucursala.adresa,count(cnp) FROM(sucursala LEFT OUTER JOIN Fisa ON sucursala.codUnic=Fisa.codUnicSucursala)

GROUP BY sucursala.nume,sucursala.adresa



11.Sa se stearga toti cititorii care traiesc in Bucuresti.

DELETE FROM cititor where adresa LIKE '%Bucuresti%'



12.Sa se afiseze numarul de carti imprumutate de la fiecare sucursala.

SELECT sucursala.nume,count(Publicatie\_Fisa.codUnicPublicatie) AS TOTAL\_CARTI from sucursala,Publicatie\_Fisa,Fisa

WHERE sucursala.codUnic=fisa.codUnicSucursala AND fisa.nrUnic=Publicatie\_Fisa.nrUnicFisa GROUP BY sucursala.nume



13.Sa se afiseze toate sucursalele dintr-un anumit oras.

SELECT sucursala.nume,sucursala.adresa from sucursala HAVING adresa LIKE '%Craiova%' GROUP BY sucursala.nume,sucursala.adresa

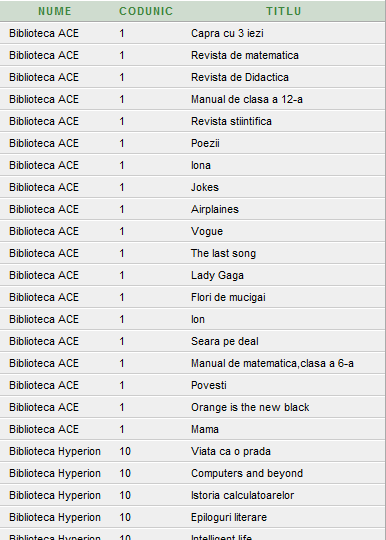


14.Sa se afiseze toate cartile de la fiecare sucursala,ordonand sucursalele dupa nume.

SELECT Sucursala.nume,Sucursala.codUnic,Publicatie.titlu FROM Sucursala,Publicatie,Publicatie\_Sucursala WHERE Sucursala.codUnic=Publicatie\_Sucursala.codUnicSucursala

AND Publicatie\_Sucursala.codUnicPublicatie=Publicatie.codUnic

ORDER BY Sucursala.nume



15.Sa se afiseze numele si adresele clientilor Sucursalei " Biblioteca Hyperion".

SELECT distinct Cititor.nume, Cititor.adresa

FROM Cititor, Fisa, Sucursala

WHERE Cititor.CNP = Fisa.CNP and

Sucursala.codUnic = Fisa.codUnicSucursala and

Sucursala.nume = 'Biblioteca Hyperion'

C:\Users\larisa\Pictures\Untitled.png

## Proceduri stocate

1.Sa se creeze o procedura care sa mute cartile de la o sucursala la alta si sa se insereze o noua carte in acea sucursala.

CREATE OR REPLACE PROCEDURE carti( ISBNN IN Carte.ISBN%type , titluN IN Publicatie.titlu%type ,numeN IN Autor.nume%type , sucursala\_Initiala\_cod IN Publicatie\_Sucursala.codUnicSucursala % type , sucursala\_Finala\_cod IN Publicatie\_Sucursala.codUnicSucursala % type,nrExemplare IN Publicatie\_Sucursala.nrExemplare % type)

IS

BEGIN

UPDATE Publicatie\_Sucursala SET codUnicSucursala=sucursala\_Finala\_cod

WHERE codUnicSucursala=sucursala\_Initiala\_cod;

INSERT INTO Publicatie VALUES(ISBNN,titluN);

INSERT INTO Carte VALUES(ISBNN);

INSERT INTO Autor VALUES(numeN);

INSERT INTO Publicatie\_Sucursala VALUES(ISBNN,sucursala\_Initiala\_cod,nrExemplare);

INSERT INTO Carte\_Autor VALUES (numeN,ISBNN);

END carti ;

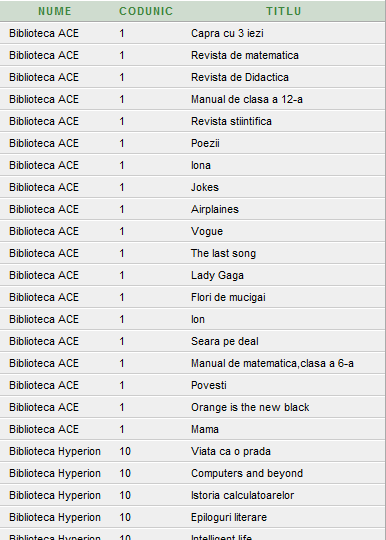
---Testare---

begin

carti('100','Carte noua','Autor\_Nou','2', '10',3);

end

---Inainte de apelare procedura---



---Dupa apelul procedurii---



2.Sa se creeze o procedura care sa aduca n exemplare in plus pentru o anumita carte.

## CREATE OR REPLACE PROCEDURE aducere\_exemplare (codUnic IN Publicatie.codUnic % type,

## extra IN Publicatie\_Sucursala.nrExemplare % type)

## IS

## BEGIN

## UPDATE Publicatie\_Sucursala SET nrExemplare = nrExemplare + extra WHERE Publicatie\_Sucursala.codUnicPublicatie = codUnic;

## END aducere\_exemplare;

---Testare---

BEGIN

aducere\_exemplare('3',2);

END;

---Inainte de apelare procedura---



---Dupa apelare---



## Functii

1.Sa se realizeze o functie ce va returna numarul de carti imprumutate intr-o perioada data.

CREATE OR REPLACE FUNCTION carti\_Imprumutate(data1 DATE,data2 DATE)

RETURN NUMBER

IS

CURSOR c\_total

IS

SELECT COUNT(Publicatie.ID\_Publicatie)

FROM Fisa,Client,Publicatie,Publicatie\_Fisa

WHERE Fisa.ID\_Client=Client.ID\_Client

AND Publicatie\_Fisa.ID\_Publicatie=Publicatie.ID\_Publicatie

AND Publicatie\_Fisa.Numar\_Fisa=Fisa.Numar\_Fisa;

total NUMBER;

BEGIN

OPEN c\_total;

FETCH c\_total INTO total;

CLOSE c\_total;

RETURN total;

END carti\_Imprumutate;

Executare functie:

begin

dbms\_output.put\_line('Numarul de carti imprumutate intre data 10-02-2016-10-06-2016 este '||carti\_Imprumutate('10-02-2016','10-06-2016'));

end;



2.Sa se creeze o functie pentru calcularea tuturor cartilor din toate sucursalele bibliotecii.

CREATE OR REPLACE FUNCTION Total\_Carti

return NUMBER

IS

total NUMBER;

BEGIN

SELECT SUM(nrExemplare) into total

FROM Publicatie\_Sucursala;

RETURN total;

END Total\_Carti;

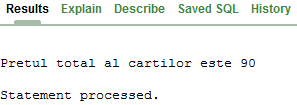
Executare Functie:

begin

## dbms\_output.put\_line('Pretul total al cartilor este '||Total\_Carti);

## end;

---Afisare---



## Trigger

Sa se creeze un trigger pentru modificarea „ISBN-ului” din tabelul „Carte”, si a „ID\_Publicatie” din „Sucursala\_Publicatie” atunci cand se modifica „ID\_Publicatie” in tabelul „Publicatie”.

CREATE OR REPLACE TRIGGER UP\_Publicatie

BEFORE UPDATE

ON PUBLICATIE

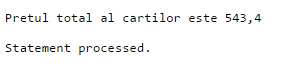
FOR EACH ROW

BEGIN

UPDATE Carte SET ISBN=:NEW.codUnic WHERE ISBN=:OLD.codUnic

UPDATE Publicatie\_Sucursala SET codUnicPublicatie=:NEW.codUnic WHERE codUnicPublicatie =:OLD.codUnic ;

DBMS\_OUTPUT.PUT\_LINE('S-a facut update in toate tabelele!');

END;