

# **SISTEM DE GESTIUNE A BAZEI DE DATE A UNOR BIBLIOTECI**

*PROIECT – BAZE DE DATE*

*ANUL I*

Fota Ștefania-Flavia

Grupa 131

semestrul 2, 2022-2023

## Cuprins:

1. Descrierea modelului real.
  - 1.1 Descrierea utilității acestuia.
  - 1.2 Descrierea regulilor de funcționare.
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.
3. Descrierea entităților, incluzând precizarea cheii primare.
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
6. Diagrama entitate-relație.
7. Diagrama conceptuală.
8. Schemele relaționale corespunzătoare diagramei conceptuale proiectate.
9. Normalizarea până la forma normală 3 (FN1 – FN3).
  - 9.1 Forma normală 1
  - 9.2 Forma normală 2
  - 9.3 Forma normală 3
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabel.
11. Implementarea bazei de date în Oracle.
  - 11.1 Crearea tabelelor și a constrângerilor.
  - 11.2 Inserarea datelor coerente în tabele (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).
12. Crearea a 5 cereri complexe în SQL.
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri
- ~~14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.~~
15. Crearea a 3 cereri în SQL.
  - 15.1 Cerere ce utilizează operația outer-join pe minimum 4 tabele.
  - 15.2 Cerere ce utilizează operația division.
  - 15.3 Cerere care implementează analiza top-n.
- ~~16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale.~~
17. Normalizări BCNF, FN4, FN5
  - 17.1 Realizarea normalizării BCNF, FN4, FN5
    - 17.1.1 Forma normală Boyce-Codd
    - 17.1.2 Forma normală 4
    - 17.1.3 Forma normală 5
  - 17.2 Aplicarea denormalizării.

## 1. Descrierea modelului real.

### 1.1 Descrierea utilității acestuia.

În cadrul acestui proiect, tema aleasă este "Baza de date pentru gestionarea bibliotecilor". Scopul acesteia este de a facilita administrarea eficientă a bibliotecilor și resurselor lor.

### 1.2 Descrierea regulilor de funcționare.

- Biblioteca se află într-o anumită locație.
- Într-o locație se pot afla o singură bibliotecă.
- În bibliotecă lucrează bibliotecari.
- bibliotecă poate avea mai mulți membrii (persoane cu permis de acces în bibliotecă).
- Un membru poate avea permis de acces la mai multe biblioteci.
- Bibliotecarii pot vinde sau împrumuta cărți membrilor bibliotecii.
- Cărțile din bibliotecă aparțin unei edituri.
- editură poate publica una sau mai multe cărți.
- Autorii pot scrie mai multe cărți.
- Autorii scriu cărți de un anumit gen.
- Cărțile pot fi împărțite după genul lor.
- Ele sunt scrise de cel puțin un autor.

## 2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

O carte poate să fie de mai multe genuri și trebuie să fie măcar de un anumit gen. Pot exista mai multe cărți cu același gen, dar nu trebuie să existe neapărat o carte de un anumit gen pentru a exista acel gen.

O editură poate publica mai multe cărți ale mai multor autori, dar trebuie să publice măcar una. O carte este scrisă de unul sau mai mulți autori. Un autor poate scrie mai multe cărți, dar trebuie să scrie măcar una.

O carte este vândută/împrumutată unui membru al bibliotecii de către un bibliotecar. Un bibliotecar poate vinde/împrumuta mai multe cărți mai multor membrii, dar nu trebuie neapărat să vândă/împrumute vreo carte vreunui client. La fel, un client poate să cumpere/împrumute mai multe cărți de la mai mulți bibliotecari, dar nu este necesar să facă asta. O carte poate fi

cumpărată/împrumutată de mai mulți membrii de la mai mulți bibliotecari, dar nu trebuie să fie neapărat cumpărată/împrumutată.

Într-o bibliotecă pot lucra mai mulți bibliotecari și trebuie să lucreze măcar unul pentru a putea funcționa. Un bibliotecar poate lucra într-o bibliotecă și numai în una.

O bibliotecă trebuie să se afle la o locație și se poate afla doar la una. Într-o locație se poate găsi doar o bibliotecă, dar nu trebuie să fie neapărat una. Nu poate exista o bibliotecă fără locație.

### 3. Descrierea entităților, incluzând precizarea cheii primare.

Pentru modelul de date al unor biblioteci, structurile GEN, AUTOR, CARTE, BIBLIOTECAR, LOCATIE, BIBLIOTECA, MEMBRU, EDITURA reprezintă entități.

- *BIBLIOTECA* = instituție autorizată și stabilă în care membrii pot cumpăra sau împrumuta cărți de la bibliotecari. Biblioteca este deschisă publicului și cărțile sunt expuse pentru ca persoanele să poată alege oricâte și oricare doresc. Cheia primară a entității este *id\_biblioteca*.
- *GEN* = entitate care reprezintă o categorie literară în care se încadrează cărțile în funcție de trăsături comune, tematică sau stil. Cheia primară a entității este *id\_gen*.
- *AUTOR* = persoană care scrie cărți. Cheia primară a entității este *id\_autor*.
- *CARTE* = lucrare scrisă care poate fi cumpărată/împrumutată din bibliotecă. Fiecare carte are un titlu unic și poate fi asociată cu un autor, un gen literar, o editură și alte informații relevante, precum anul publicării, prețul și numărul de pagini. Cheia primară a entității este *id\_carte*.
- *EDITURA* = companie specializată în publicarea și distribuirea cărților. Cheia primară a entității este *id\_editura*.
- *BIBLIOTECAR* = persoană angajată într-o bibliotecă și se ocupă cu procesarea achizițiilor și împrumuturilor de cărți membrilor. Cheia primară a entității este *id\_bibliotecar*.
- *LOCATIE* = entitate care reține adresa unei biblioteci. Include informații despre strada, numărul, orașul și alte detalii relevante pentru a localiza precis biblioteca. Cheia primară a entității este *id\_locatie*.
- *MEMBRU* = persoană înregistrată într-o bibliotecă care are dreptul de a împrumuta și cumpăra cărți. Cheia primară a entității este *id\_membru*.

## 4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

EDITURA\_publica\_CARTE\_scrisa\_de\_AUTOR = relație de tipul 3 care leagă entitățile EDITURA, CARTE și AUTOR, reflectând cine este editura, ce carte a publicat și care este autorul cărții respective. Această relație va fi denumită PUBLICA.

MEMBRU\_împrumută\_CARTE\_de\_la\_MEMBRU = relație de tipul 3 care leagă entitățile MEMBRU, CARTE și BIBLIOTECAR, reflectând cine este membrul bibliotecii, ce carte împrumută și de la ce bibliotecar împrumută cartea respectivă. Această relație va fi denumită IMPRUMUTA.

MEMBRU\_cumpără\_CARTE\_de\_la\_MEMBRU = relație de tipul 3 care leagă entitățile MEMBRU, CARTE și BIBLIOTECAR, reflectând cine este membrul bibliotecii, ce carte cumpără și de la ce bibliotecar cumpără cartea respectivă. Această relație va fi denumită CUMPARA.

CARTE\_are\_GEN = relație care leagă entitățile CARTE și GEN, reflectând legătura dintre acestea (de ce gen este o anumită carte). Ea are cardinalitatea minimă 0:1 (o carte trebuie să aibă un anumit gen, dar un gen nu trebuie să aibă o carte asociată) și cardinalitatea maximă m:m (o carte poate fi încadrată în mai multe genuri și mai multe cărți pot fi scrise de același gen)

MEMBRU\_apartine\_BIBLIOTECA = relație care leagă entitățile MEMBRU și BIBLIOTECA, reflectând legătura dintre acestea (la ce bibliotecă are permis un anumit membru). Ea are cardinalitatea minimă 0:1 (un membru trebuie să aibă asociată cel puțin o bibliotecă, dar o bibliotecă nu trebuie să aibă neapărat membrii) și cardinalitatea maximă m:m (un membru poate să aibă permis la mai multe biblioteci și o bibliotecă poate avea mai mulți membrii).

BIBLIOTECAR\_lucraza\_BIBLIOTECA = relație care leagă entitățile BIBLIOTECAR și BIBLIOTECA, reflectând legătura dintre acestea (bibliotecarul lucrează într-o anumită bibliotecă). Ea are cardinalitatea minimă 1:1 (un bibliotecar trebuie să lucreze în cel puțin o bibliotecă și într-o bibliotecă trebuie să lucreze cel puțin un bibliotecar) și cardinalitatea maximă m:1 (un bibliotecar poate să lucreze în cel mult o bibliotecă și într-o bibliotecă pot lucra mai mulți bibliotecari).

BIBLIOTECA\_se\_afla\_LOCATIE = relație care leagă entitățile BIBLIOTECA și LOCATIE, reflectând legătura dintre acestea (biblioteca se află într-un anumit loc). Ea are cardinalitatea minimă 0:1 (o bibliotecă trebuie să se afle într-o anumită locație, dar într-o locație nu trebuie să se găsească neapărat o bibliotecă) și cardinalitatea maximă 1:1 (o bibliotecă se poate afla în cel mult o locație și la o locație se poate afla cel mult o bibliotecă).

## 5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

- Entitatea GEN va avea următoarele atribute: id-ul și numele genului respectiv. Tipul de date pentru id\_gen va fi number(10), iar pentru nume\_gen va fi varchar2(20).  
Cheia primară va fi id\_gen.  
Constrângere pentru numele genului: Unique.
- Entitatea EDITURA va avea următoarele atribute: id-ul, numele, email-ul, numărul de telefon și anul înființării editurii. Tipul de date pentru id\_editură va fi number(10), nume\_editură varchar2(25), email varchar2(30), nr\_telefon varchar2(10), an\_înființare number(4).  
Cheia primară va fi id\_editură.  
Constrângere pentru numele editurii: Unique.
- Entitatea AUTOR va avea următoarele atribute: id-ul, numele, naționalitatea, sexul, data de naștere și data de deces ale autorului respectiv. Tipul de date pentru id\_autor va fi number(10), nume\_autor varchar2(40), naționalitate varchar2(20), sex char(1), dată\_naștere date, dată\_deces date.  
Cheia primară va fi id\_autor.  
Constrângere pentru numele autorului: Unique.  
Atributul dată\_naștere va avea ca default sysdate (data curentă).  
Constrângere pentru data nașterii și data decesului: check data\_naștere < dată\_deces.
- Entitatea LOCATIE va avea următoarele atribute: id-ul locației, orașul, strada și codul poștal al bibliotecii respective. Tipul de date pentru id\_locație va fi number(10), oraș varchar2(20), stradă varchar2(50), cod\_poștal varchar2(15).  
Cheia primară va fi id\_locație.  
Constrângere pentru codul poștal: Unique.

- Entitatea BIBLIOTECA va avea următoarele atribute: id-ul, numele, website-ul și data înființării bibliotecii dar și id-ul locației la care se află respectiva bibliotecă.

Tipul de date pentru id\_bibliotecă va fi number(10), nume\_bibliotecă varchar2(20), website varchar2(50), data\_înființare date.

Cheia primară va fi id\_bibliotecă.

Cheia externă va fi id\_locație (din tabelul LOCATIE), cu tipul de date number(10).

Constrângere pentru numele bibliotecii și website: Unique.

Constrângere pentru data de înființare: NOT NULL.
- Entitatea BIBLIOTECAR va avea următoarele atribute: id-ul, numele, data angajării și salariul bibliotecarului, dar și id-ul bibliotecii la care lucrează respectivul bibliotecar.

Tipul de date pentru id\_bibliotecar va fi number(10), nume\_bibliotecar varchar2(20), dată\_angajare date, salariu number(10).

Cheia primară va fi id\_bibliotecar.

Cheia externă va fi id\_bibliotecă (din tabelul BIBLIOTECA), cu tipul de date number(10).

Atributul dată\_angajare va avea ca default sysdate (data curentă).
- Entitatea MEMBRU va avea următoarele atribute: id-ul, numele, prenumele, data nașterii și numărul de telefon ale unui membru al bibliotecii.

Tipul de date pentru id\_membru va fi number(10), nume varchar2(30), prenume varchar2(30), dată\_naștere date, nr\_telefon varchar2(10).

Cheia primară va fi id\_membru.
- Entitatea CARTE va avea următoarele atribute: id-ul, titlul, numărul de pagini și limba în care e scrisă cartea respectivă.

Tipul de date pentru id\_carte va fi number(10), titlu varchar2(30), nr\_pagini number(5), limba varchar2(30).

Cheia primară va fi id\_carte.

Constrângere pentru titlul cărții: Unique.
- Tabelul asociativ PUBLICA va avea următoarele atribute: id-ul cărții, id-ul autorului, id-ul editurii și data publicării.

Cheia primară, deoarece avem o relație de tip 3 între CARTE, AUTOR și EDITURA, este formată din id-ul cărții (FK din tabelul CARTE), id-ul autorului (FK din tabelul AUTOR) și id-ul editurii (FK din tabelul EDITURA).

Tipul de date pentru id\_carte va fi number(10), id\_autor number(10), id\_editură number(10) și dată\_publicare date.

- Tabelul asociativ IMPRUMUTA va avea următoarele attribute: id-ul cărții, id-ul bibliotecarului, id-ul membrului, data împrumutului și data restituirii.

Cheia primară, deoarece avem o relație de tip 3 între CARTE, BIBLIOTECAR și MEMBRU, este formată din id-ul cărții (FK din tabelul CARTE), id-ul bibliotecarului (FK din tabelul BIBLIOTECAR) și id-ul membrului (FK din tabelul MEMBRU).

Tipul de date pentru id\_carte va fi number(10), id\_bibliotecar number(10), id\_membru number(10), dată\_împrumut date și dată\_restituire date.
- Tabelul asociativ CUMPARA va avea următoarele attribute: id-ul cărții, id-ul bibliotecarului, id-ul membrului, data achiziției și prețul.

Cheia primară, deoarece avem o relație de tip 3 între CARTE, BIBLIOTECAR și MEMBRU, este formată din id-ul cărții (FK din tabelul CARTE), id-ul bibliotecarului (FK din tabelul BIBLIOTECAR) și id-ul membrului (FK din tabelul MEMBRU).

Tipul de date pentru id\_carte va fi number(10), id\_bibliotecar number(10), id\_membru number(10), dată\_achiziție date și preț number(10).
- Tabelul asociativ ARE va avea următoarele attribute: id-ul cărții și id-ul genului.

Cheia primară este compusă din id-ul cărții (FK din tabelul CARTE) și id-ul genului (FK din tabelul GEN).

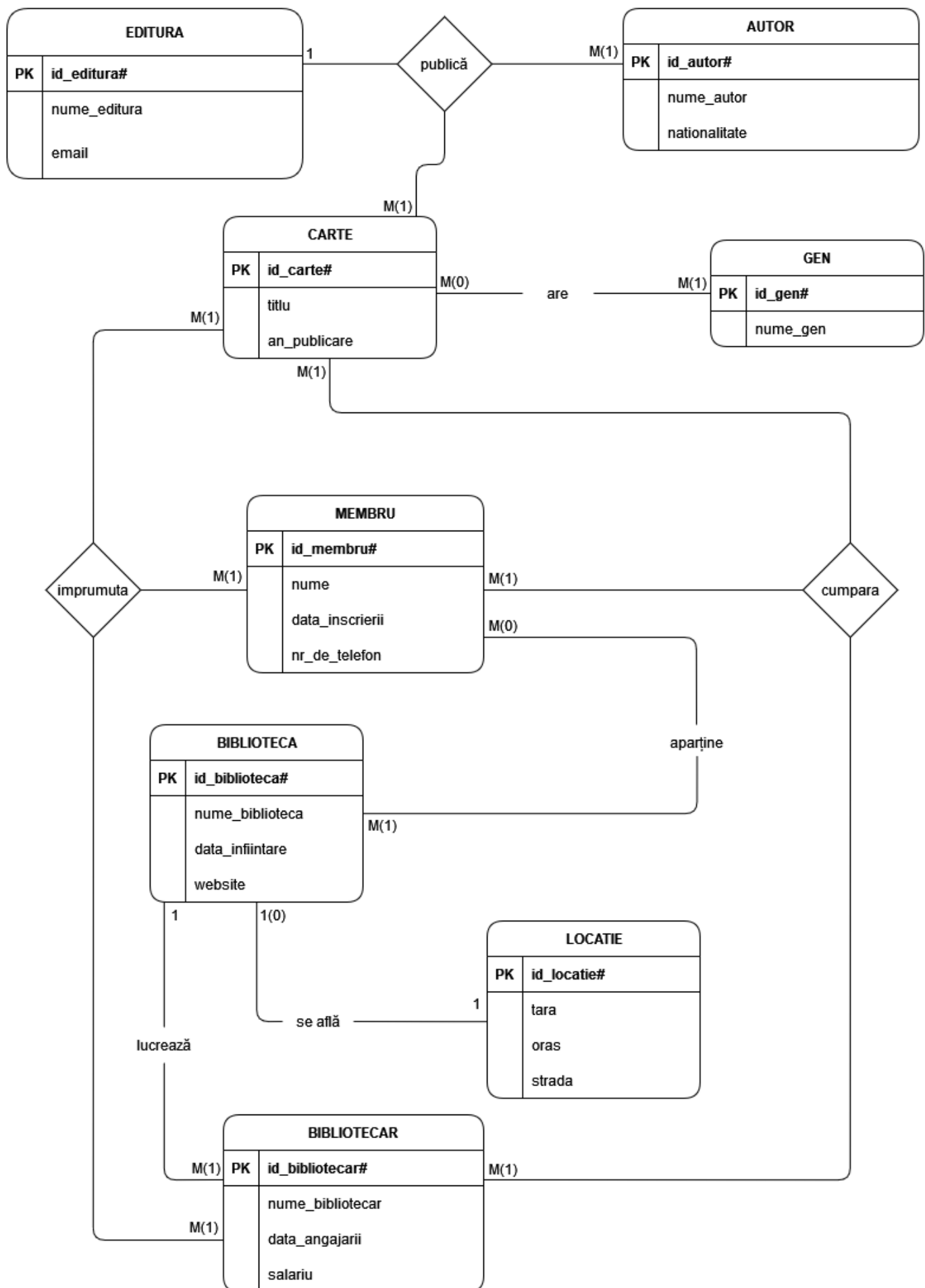
Tipul de date pentru id\_carte va fi number(10), id\_gen number(10).
- Tabelul asociativ APARTINE va avea următoarele attribute: id-ul membrului și id-ul bibliotecii.

Cheia primară este compusă din id-ul membrului (FK din tabelul MEMBRU) și id-ul bibliotecii (FK din tabelul BIBLIOTECA).

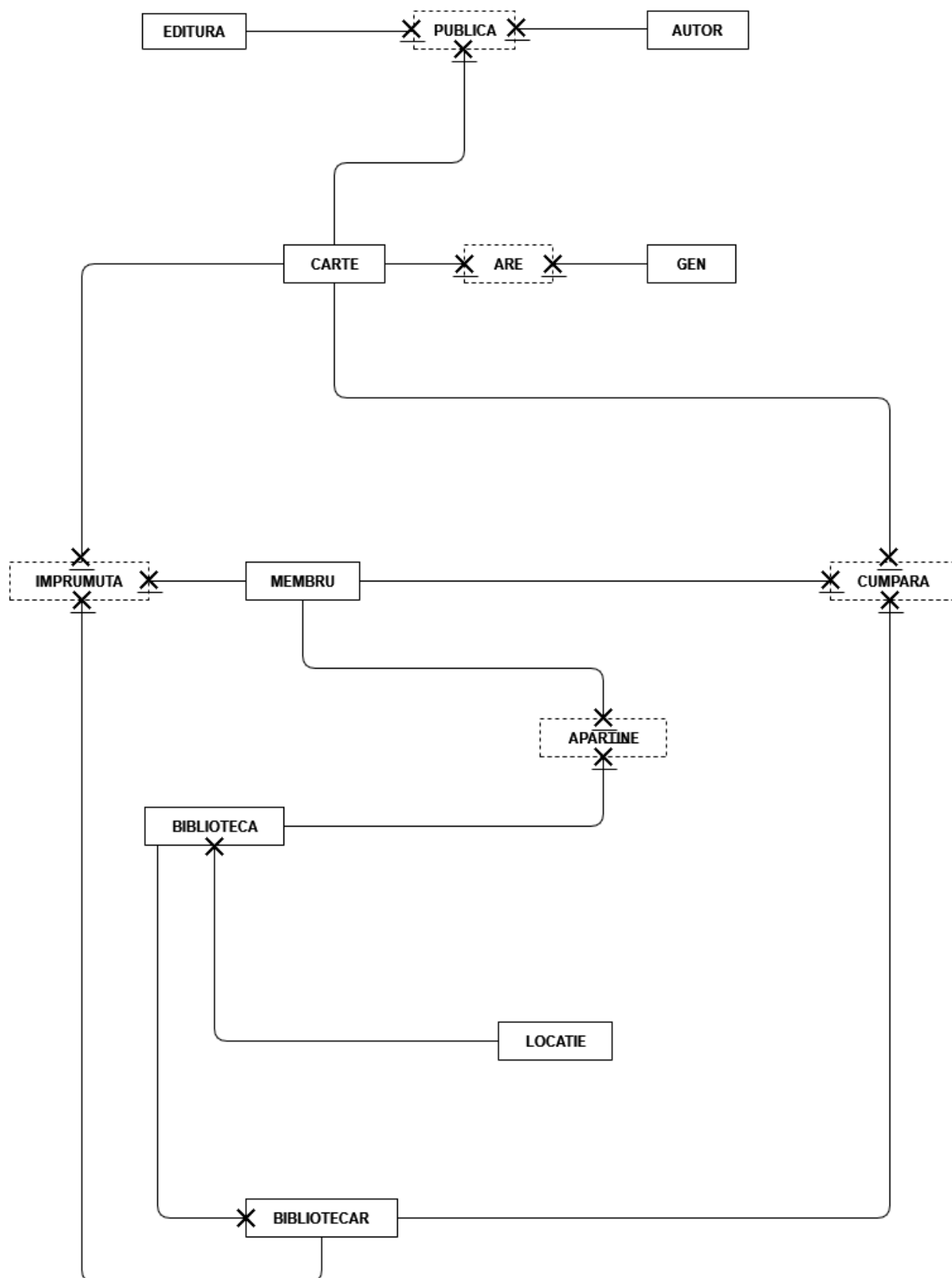
Tipul de date pentru id\_membru va fi number(10), id\_biblioteca number(10).



## 6. Diagrama entitate-relație.



## 7. Diagrama conceptuală.



## 8. Schemele relaționale corespunzătoare diagramei conceptuale proiectate.

GEN (ID\_GEN#, nume\_gen)

EDITURA (ID\_EDITURA#, nume\_editura, email, nr\_telefon, an\_infiintare)

AUTOR (ID\_AUTOR#, nume\_autor, nationalitate, sex, data\_nastere, data\_deces)

LOCATIE (ID\_LOCATIE#, oras, strada, cod\_postal)

BIBLIOTECA (ID\_BIBLIOTECA#, id\_locatie, nume\_biblioteca, website, data\_infiintare)

BIBLIOTECAR (ID\_BIBLIOTECAR#, id\_biblioteca, nume\_bibliotecar, data\_angajare, salariu)

MEMBRU (ID\_MEMBRU#, nume, prenume, data\_nastere, nr\_telefon)

CARTE (ID\_CARTTE#, titlu, nr\_pagini, limba)

PUBLICA (ID\_CARTE#, ID\_AUTOR#, ID\_EDITURA#, data\_publicare)

IMPRUMUTA (ID\_CARTE#, ID\_BIBLIOTECAR#, ID\_MEMBRU#, data\_imprumut, data\_restituire)

CUMPARA (ID\_CARTE#, ID\_BIBLIOTECAR#, ID\_MEMBRU#, data\_achizitie, pret)

ARE (ID\_CARTE#, ID\_GEN#)

APARTINE (ID\_MEMBRU#, ID\_BIBLIOTECA#)

## 9. Normalizarea până la forma normală 3 (FN1 – FN3).

### 9.1 Forma normală 1

Pentru ca o relație să fie în FN1, trebuie ca fiecare atribut din care e compusă să cuprindă o singură valoare atomică. De asemenea, trebuie și ca fiecare înregistrare să fie definită astfel încât să poată fi identificată în mod unic prin intermediul PK-ului.

Un exemplu de încălcare a formei normale 1 ar fi dacă entitatea CARTE ar conține un atribut genuri, deoarece o carte poate avea mai multe genuri:

CARTE (ID\_CARTE#, titlu, nr\_pagini, limba, **genuri**)

## 9.2 Forma normală 2

Pentru ca o relație să fie în FN2, trebuie să satisfacă atât regulile pentru FN1, cât și să aibă fiecare atribut dependent funcțional de totalitatea cheii primare. Dacă unele atribute depind funcțional doar de o parte a cheii primare, trebuie să le separăm în tabele diferite.

Un exemplu de încălcare a formei normale 2 ar fi dacă am scoate entitatea AUTOR și pentru entitatea CARTE am avea schema următoare:

CARTE (ID\_CARTE#, ID AUTOR#, titlu, nr\_pagini, limba, nume autor, nationalitate)

În acest caz, atributele titlu, nr\_pagini și limba depind doar de id-ul cărții, nu și de cel al autorului, iar nume\_autor și nationalitate depind doar de id-ul autorului, nu și de cel al cărții.

## 9.3 Forma normală 3

Pentru ca o relație să fie în FN2, trebuie să satisfacă regulile pentru FN2, dar și ca fiecare atribut care are o dependență tranzitivă (depinde de un alt atribut non-cheie).

Un exemplu de încălcare a formei normale 3 ar fi dacă am adăuga un atribut adresa în tabela BIBLIOTECA, deoarece acesta ar depinde de id\_locatie:

BIBLIOTECA (ID\_BIBLIOTECA#, id locatie, nume\_biblioteca, website, data\_infiintare, adresa)

## 10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabel +

## 11. Implementarea bazei de date în Oracle

### 11.1 Crearea tabelelor și a constrângerilor

-- LOCATIE

CREATE TABLE locatie (

id\_locatie NUMBER(10) constraint pk\_locatie primary key,

oras VARCHAR2(20),

strada VARCHAR2(50),

```
-- LOCATIE
CREATE TABLE locatie (
  id_locatie NUMBER(10) constraint pk_locatie primary key,
  oras VARCHAR2(20),
  strada VARCHAR2(50),
  cod_postal VARCHAR2(15) UNIQUE
);
```

Table LOCATIE created.

```
cod_postal VARCHAR2(15) UNIQUE

);
```

PROJECTBD~1.sql   PROJECTBD   LOCATIE x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID LOCATIE	NUMBER (10, 0)	No	(null)	1	(null)
2	ORAS	VARCHAR2 (20 BYTE)	Yes	(null)	2	(null)
3	STRADA	VARCHAR2 (50 BYTE)	Yes	(null)	3	(null)
4	COD POSTAL	VARCHAR2 (15 BYTE)	Yes	(null)	4	(null)

```
-- BIBLIOTECA
```

```
CREATE TABLE biblioteca (
```

```
    id_biblioteca NUMBER(10) constraint pk_biblioteca primary key,
```

```
    id_locatie    NUMBER(10)    constraint    fk_biblioteca_locatie    references
locatie(id_locatie),
```

```
    nume_biblioteca VARCHAR2(20) UNIQUE, --modificat ulterior in VARCHAR2(50)
```

```
    website VARCHAR2(50) UNIQUE,
```

```
    data_infiintare DATE constraint data_infiintare not null
```

```
);

-- BIBLIOTECA

CREATE TABLE biblioteca (
    id_biblioteca NUMBER(10) constraint pk_biblioteca primary key,
    id_locatie    NUMBER(10) constraint fk_biblioteca_locatie references locatie(id_locatie)
    nume_biblioteca VARCHAR2(20) UNIQUE,
    website VARCHAR2(50) UNIQUE,
    data_infiintare DATE constraint data_infiintare not null
);
```

Script Output x

Task completed in 0,062 seconds

Table BIBLIOTECA created.

PROJECTBD~1.sql   PROJECTBD   BIBLIOTECA x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID BIBLIOTECA	NUMBER (10, 0)	No	(null)	1	(null)
2	ID LOCATIE	NUMBER (10, 0)	Yes	(null)	2	(null)
3	NUME BIBLIOTECA	VARCHAR2 (20 BYTE)	Yes	(null)	3	(null)
4	WEBSITE	VARCHAR2 (50 BYTE)	Yes	(null)	4	(null)
5	DATA INFIINTARE	DATE	No	(null)	5	(null)

-- BIBLIOTECAR

CREATE TABLE bibliotecar (

id\_bibliotecar NUMBER(10) constraint pk\_bibliotecar primary key,

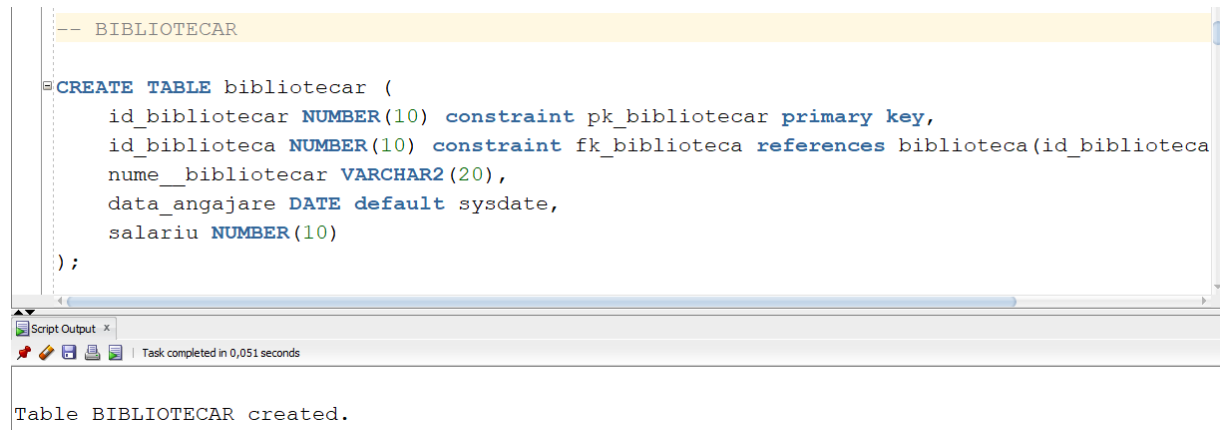
id\_biblioteca NUMBER(10) constraint fk\_biblioteca references  
biblioteca(id\_biblioteca),

nume\_\_bibliotecar VARCHAR2(20),

data\_angajare DATE default sysdate,

salariu NUMBER(10)

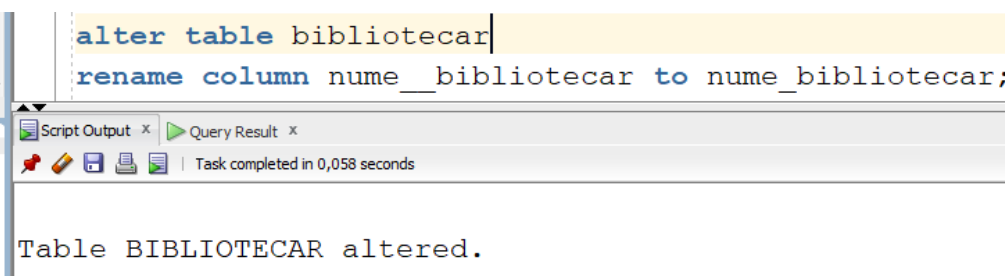
);



PROJECTBD~1.sql   PROJECTBD   BIBLIOTECAR					
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL					
Actions...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID BIBLIOTECAR	NUMBER(10,0)	No	(null)	1 (null)	
2 ID BIBLIOTECA	NUMBER(10,0)	Yes	(null)	2 (null)	
3 NUME BIBLIOTECAR	VARCHAR2(20 BYTE)	Yes	(null)	3 (null)	
4 DATA ANGAJARE	DATE	Yes	sysdate	4 (null)	
5 SALARIU	NUMBER(10,0)	Yes	(null)	5 (null)	

alter table bibliotecar

rename column nume\_\_bibliotecar to nume\_bibliotecar; --am modificat numele coloanei



-- MEMBRU

CREATE TABLE membru (

id\_membru NUMBER(10) constraint pk\_membru primary key,

nume VARCHAR2(20),

prenume VARCHAR2(20),

data\_nastere DATE,

nr\_telefon VARCHAR2(10)

-- MEMBRU

```
CREATE TABLE membru (  
    id_membru NUMBER(10) constraint pk_membru primary key,  
    nume VARCHAR2(20),  
    prenume VARCHAR2(20),  
    data_nastere DATE,  
    nr_telefon VARCHAR2(10)  
);
```

PROJECTBD~1.sql   PROJECTBD   MEMBRU x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID MEMBRU	NUMBER(10,0)	No	(null)	1	(null)
2	NUME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	PRENUME	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	DATA NASTERE	DATE	Yes	(null)	4	(null)
5	NR TELEFON	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)

);

-- GEN

CREATE TABLE gen (

id\_gen NUMBER(10) constraint pk\_gen primary key,

nume\_gen VARCHAR2(20)

-- GEN

```
CREATE TABLE gen (  
    id_gen NUMBER(10) constraint pk_gen primary key,  
    nume_gen VARCHAR2(20)  
);
```

Script Output x
Task completed in 0,058 seconds

Table GEN created.

);

PROJECTBD~1.sql   PROJECTBD   GEN x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID GEN	NUMBER(10,0)	No	(null)	1	(null)
2	NUME GEN	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)

-- EDITURA

CREATE TABLE editura (

id\_editura NUMBER(10) constraint pk\_editura primary key,

```

    nume_editura VARCHAR2(25)
    UNIQUE,

    email VARCHAR2(30),

    nr_telefon VARCHAR2(10),

    an_infiintare NUMBER(4)

);

```

```

-- EDITURA

CREATE TABLE editura (
    id_editura NUMBER(10) constraint pk_editura primary key,
    nume_editura VARCHAR2(25) UNIQUE,
    email VARCHAR2(30),
    nr_telefon VARCHAR2(10),
    an_infiintare NUMBER(4)
);

```

Script Output  
Task completed in 0,057 seconds

Table EDITURA created.

PROJECTBD~1.sql   PROJECTBD   EDITURA x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID EDITURA	NUMBER(10,0)	No	(null)	1	(null)
2	NUME EDITURA	VARCHAR2(25 BYTE)	Yes	(null)	2	(null)
3	EMAIL	VARCHAR2(30 BYTE)	Yes	(null)	3	(null)
4	NR TELEFON	VARCHAR2(10 BYTE)	Yes	(null)	4	(null)
5	AN INFIINTARE	NUMBER(4,0)	Yes	(null)	5	(null)

-- AUTOR

CREATE TABLE autor (

id\_autor NUMBER(10) constraint pk\_autor primary key,

nume\_autor VARCHAR2(40)  
UNIQUE,

nationalitate VARCHAR2(20),

sex CHAR(1),

data\_nastere DATE default  
sysdate,

data\_deces DATE default null,

CONSTRAINT chk\_date CHECK (data\_nastere < data\_deces)

);

```

-- AUTOR

CREATE TABLE autor (
    id_autor NUMBER(10) constraint pk_autor primary key,
    nume_autor VARCHAR2(40) UNIQUE,
    nationalitate VARCHAR2(20),
    sex CHAR(1),
    data_nastere DATE default sysdate,
    data_deces DATE default null,
    CONSTRAINT chk_date CHECK (data_nastere < data_deces)
);

```

Script Output  
Task completed in 0,059 seconds

Table AUTOR created.

PROJECTBD~1.sql   PROJECTBD   AUTOR x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID AUTOR	NUMBER(10,0)	No	(null)	1	(null)
2	NUME AUTOR	VARCHAR2(40 BYTE)	Yes	(null)	2	(null)
3	NATIONALITATE	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	SEX	CHAR(1 BYTE)	Yes	(null)	4	(null)
5	DATA NASTERE	DATE	Yes	sysdate	5	(null)
6	DATA DECES	DATE	Yes	null	6	(null)



-- CARTE

```
CREATE TABLE carte (  
    id_carte NUMBER(10) constraint  
pk_carte primary key,  
    titlu VARCHAR2(30) UNIQUE,  
    nr_pagini NUMBER(5),  
    limba VARCHAR2(30)  
);
```

```
-- CARTE  
  
CREATE TABLE carte (  
    id_carte NUMBER(10) constraint pk_carte primary key,  
    titlu VARCHAR2(30) UNIQUE,  
    nr_pagini NUMBER(5),  
    limba VARCHAR2(30)  
);  
  
Script Output  
Task completed in 0,051 seconds  
  
Table CARTE created.
```

PROIECTBD~1.sql   PROIECTBD   CARTE x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID CARTE	NUMBER(10,0)	No	(null)	1	(null)
2	TITLU	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)
3	NR PAGINI	NUMBER(5,0)	Yes	(null)	3	(null)
4	LIMBA	VARCHAR2(30 BYTE)	Yes	(null)	4	(null)

-- tabelul asociativ PUBLICA

```
CREATE TABLE publica (  
    id_carte NUMBER(10),  
    id_autor NUMBER(10),  
    id_editura NUMBER(10),  
    data_publicare DATE default  
sysdate,  
    constraint fk_publica_autor foreign key(id_autor) references autor(id_autor),  
    constraint fk_publica_carte foreign key(id_carte) references carte(id_carte),  
    constraint fk_publica_editura foreign key(id_editura) references editura(id_editura),  
    constraint pk_compus_publica primary key(id_autor, id_carte, id_editura)  
);
```

```
-- tabelul asociativ PUBLICA  
  
CREATE TABLE publica (  
    id_carte NUMBER(10),  
    id_autor NUMBER(10),  
    id_editura NUMBER(10),  
    data_publicare DATE default sysdate,  
    constraint fk_publica_autor foreign key(id_autor) references autor(id_autor),  
    constraint fk_publica_carte foreign key(id_carte) references carte(id_carte),  
    constraint fk_publica_editura foreign key(id_editura) references editura(id_editura),  
    constraint pk_compus_publica primary key(id_autor, id_carte, id_editura)  
);  
  
Script Output  
Task completed in 0,052 seconds  
  
Table PUBLICA created.
```

PROIECTBD~1.sql   PROIECTBD   PUBLICA x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID CARTE	NUMBER(10,0)	No	(null)	1	(null)
2	ID AUTOR	NUMBER(10,0)	No	(null)	2	(null)
3	ID EDITURA	NUMBER(10,0)	No	(null)	3	(null)
4	DATA PUBLICARE	DATE	Yes	sysdate	4	(null)

-- tabelul asociativ IMPRUMUTA

CREATE TABLE imprumuta (

id\_carte NUMBER(10),

id\_membru NUMBER(10),

id\_bibliotecar NUMBER(10),

data\_imprumut DATE default sysdate,

data\_restituire DATE default null,

constraint fk\_imprumuta\_carte foreign key(id\_carte) references carte(id\_carte),

constraint fk\_imprumuta\_membru foreign key(id\_membru) references membru(id\_membru),

constraint fk\_imprumuta\_bibliotecar foreign key(id\_bibliotecar) references bibliotecar(id\_bibliotecar),

constraint pk\_compus\_imprumuta primary key(id\_carte,id\_membru,id\_bibliotecar)

);

```
-- tabelul asociativ IMPRUMUTA

CREATE TABLE imprumuta (
  id_carte NUMBER(10),
  id_membru NUMBER(10),
  id_bibliotecar NUMBER(10),
  data_imprumut DATE default sysdate,
  data_restituire DATE default null,
  constraint fk_imprumuta_carte foreign key(id_carte) references carte(id_carte),
  constraint fk_imprumuta_membru foreign key(id_membru) references membru(id_membru),
  constraint fk_imprumuta_bibliotecar foreign key(id_bibliotecar) references bibliotecar(id_bibliotecar),
  constraint pk_compus_imprumuta primary key(id_carte,id_membru,id_bibliotecar)
);
```

Table IMPRUMUTA created.

1	ID_CARTE	NUMBER(10,0)	No	(null)	1	(null)
2	ID_MEMBRU	NUMBER(10,0)	No	(null)	2	(null)
3	ID_BIBLIOTECAR	NUMBER(10,0)	No	(null)	3	(null)
4	DATA IMPRUMUT	DATE	Yes	sysdate	4	(null)
5	DATA RESTITUIRE	DATE	Yes	null	5	(null)

-- tabelul asociativ CUMPARA

CREATE TABLE cumpara (

id\_carte NUMBER(10),

id\_membru NUMBER(10),

id\_bibliotecar NUMBER(10),

data\_achizitie DATE default sysdate,

pret NUMBER(10),

constraint fk\_cumpara\_carte foreign key(id\_carte) references carte(id\_carte),

constraint fk\_cumpara\_membru foreign key(id\_membru) references membru(id\_membru),

```
-- tabelul asociativ CUMPARA

CREATE TABLE cumpara (
  id_carte NUMBER(10),
  id_membru NUMBER(10),
  id_bibliotecar NUMBER(10),
  data_achizitie DATE default sysdate,
  pret NUMBER(10),
  constraint fk_cumpara_carte foreign key(id_carte) references carte(id_carte),
  constraint fk_cumpara_membru foreign key(id_membru) references membru(id_membru),
  constraint fk_cumpara_bibliotecar foreign key(id_bibliotecar) references bibliotecar(id_bibliotecar),
  constraint pk_compus_cumpara primary key(id_carte,id_membru,id_bibliotecar)
);
```

Table CUMPARA created.

```

constraint fk_cumpara_bibliotecar foreign key(id_bibliotecar) references
bibliotecar(id_bibliotecar),

constraint pk_compus_cumpara primary key(id_carte,id_membru,id_bibliotecar)

);

```

PROJECTBD~1.sql   PROJECTBD   CUMPARA						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID CARTE	NUMBER(10,0)	No	(null)	1	(null)
2	ID MEMBRU	NUMBER(10,0)	No	(null)	2	(null)
3	ID BIBLIOTECAR	NUMBER(10,0)	No	(null)	3	(null)
4	DATA ACHIZITIE	DATE	Yes	sysdate	4	(null)
5	PRET	NUMBER(10,0)	Yes	(null)	5	(null)

-- tabelul asociativ ARE

CREATE TABLE are (

id\_carte NUMBER(10),

id\_gen NUMBER(10),

constraint fk\_are\_carte foreign key(id\_carte) references carte(id\_carte),

constraint fk\_are\_gen foreign key(id\_gen) references gen(id\_gen),

constraint pk\_compus\_are primary key(id\_carte,id\_gen)

);

```

-- tabelul asociativ ARE

CREATE TABLE are (
  id_carte NUMBER(10),
  id_gen NUMBER(10),
  constraint fk_are_carte foreign key(id_carte) references carte(id_carte),
  constraint fk_are_gen foreign key(id_gen) references gen(id_gen),
  constraint pk_compus_are primary key(id_carte,id_gen)
);

```

Script Output: Task completed in 0,042 seconds

Table ARE created.

PROJECTBD~1.sql   PROJECTBD   ARE						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID CARTE	NUMBER(10,0)	No	(null)	1	(null)
2	ID GEN	NUMBER(10,0)	No	(null)	2	(null)

-- tabelul asociativ APARTINE

CREATE TABLE apartine (

id\_membru NUMBER(10),

id\_biblioteca NUMBER(10),

```

constraint fk_apartine_membru foreign key(id_membru) references
membru(id_membru),

```

```

-- tabelul asociativ APARTINE

CREATE TABLE apartine (
  id_membru NUMBER(10),
  id_biblioteca NUMBER(10),
  constraint fk_apartine_membru foreign key(id_membru) references membru(id_membru),
  constraint fk_apartine_biblioteca foreign key(id_biblioteca) references biblioteca(id_biblioteca)
);

```

Script Output: Task completed in 0,041 seconds

Table APARTINE created.

```

constraint fk_apartine_biblioteca foreign key(id_biblioteca) references
biblioteca(id_biblioteca)

);

```

PROJECTBD~1.sql   PROJECTBD   APARTINE x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID MEMBRU	NUMBER(10,0)	Yes	(null)	1	(null)
2	ID BIBLIOTECA	NUMBER(10,0)	Yes	(null)	2	(null)

## 11.2 Inserarea datelor coerente în tabele (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

--SECVENTE SI INSERARI TABELE

--secventa + inserari LOCATIE

create sequence  
id\_locatie

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

select * from locatie;			
cript Output x Query Result x			
SQL   All Rows Fetched: 7 in 0 seconds			
ID_LOCATIE	ORAS	STRADA	COD_POSTAL
1	1 Bucuresti	Str. Boteanu	010027
2	2 Cluj-Napoca	Calea Dorobantilor	400394
3	3 Brasov	Bulevardul Eroilor	500036
4	4 Craiova	Str. Mihail Kogalniceanu	200390
5	5 Iasi	Strada Pacurari	700511
6	6 Timisoara	Piata Libertatii	300077
7	7 Cluj-Napoca	Str. Mihail Kogalniceanu	400084

insert into locatie values(id\_locatie.nextval, 'Bucuresti', 'Str. Boteanu', '010027');

insert into locatie values(id\_locatie.nextval, 'Cluj-Napoca', 'Calea Dorobantilor', '400394');

insert into locatie values(id\_locatie.nextval, 'Brasov', 'Bulevardul Eroilor', '500036');

insert into locatie values(id\_locatie.nextval, 'Craiova', 'Str. Mihail Kogalniceanu', '200390');

```
insert into locatie values(id_locatie.nextval, 'Iasi', 'Strada Pacurari', '700511');

insert into locatie values(id_locatie.nextval, 'Timisoara', 'Piata Libertatii', '300077');

insert into locatie values(id_locatie.nextval, 'Cluj-Napoca', 'Str. Mihail Kogalniceanu',
'400084');
```

--secventa + inserari BIBLIOTECA

```
create sequence id_biblioteca
```

```
start with 1
```

```
increment by 1
```

```
minvalue 0
```

```
maxvalue 9999
```

```
nocycle;
```

```
alter table biblioteca
```

```
modify(ume_biblioteca varchar2(50)); --am modificat tipul de date de la varchar2(20) la
varchar2(50)
```

```
alter table biblioteca
modify(ume_biblioteca varchar2(50));
```

Script Output x

Task completed in 0,083 seconds

Table BIBLIOTECA altered.

```
insert into biblioteca values (id_biblioteca.nextval, 4, 'Biblioteca Judeteana "Alexandru si
Aristia Aman"', 'www.aman.ro', TO_DATE('21-12-1908', 'DD-MM-YYYY'));
```

```
insert into biblioteca values (id_biblioteca.nextval, 2, 'Biblioteca Judeteana "Octavian Goga"',
'www.new.bjc.ro', TO_DATE('15-10-1921', 'DD-MM-YYYY'));
```

```
insert into biblioteca values (id_biblioteca.nextval, 5, 'Biblioteca Centrala Universitara
"Mihai Eminescu', 'www.bcu-iasi.ro', TO_DATE('05-07-1835', 'DD-MM-YYYY'));
```

```
insert into biblioteca values (id_biblioteca.nextval, 1, 'Biblioteca Centrala Universitara "Carol
I"', 'www.bcub.ro', TO_DATE('12-07-1948', 'DD-MM-YYYY'));
```

```
insert into biblioteca values (id_biblioteca.nextval, 3, 'Biblioteca Judeteana "Gheorghe
Baritui"', 'www.bjbv.ro', TO_DATE('30-03-1928', 'DD-MM-YYYY'));
```

```
insert into biblioteca values (id_biblioteca.nextval, 7, 'Biblioteca de Fizica', 'www.bcucluj.ro',
TO_DATE('20-04-1922', 'DD-MM-YYYY'));
```

```
select * from biblioteca;
```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0,001 seconds

ID_BIBLIOTECA	ID_LOCATIE	NUME_BIBLIOTECA	WEBSITE	DATA_INFINTARE
1	6	7 Biblioteca de Fizica	www.bcucluj.ro	20-04-1922
2	1	4 Biblioteca Judeteana "Alexandru si Aristia Aman"	www.aman.ro	21-12-1908
3	2	2 Biblioteca Judeteana "Octavian Goga"	www.new.bjc.ro	15-10-1921
4	3	5 Biblioteca Centrala Universitara "Mihai Eminescu	www.bcu-iasi.ro	05-07-1835
5	4	1 Biblioteca Centrala Universitara "Carol I"	www.bcub.ro	12-07-1948
6	5	3 Biblioteca Judeteana "Gheorghe Baritui"	www.bjbv.ro	30-03-1928

--secventa + inserari BIBLIOTECAR

create sequence  
id\_bibliotecar

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;



The screenshot shows a SQL query window with the query `select * from bibliotecar;` and its results. The results are displayed in a table with 5 columns: ID\_BIBLIOTECAR, ID\_BIBLIOTECA, NUME\_BIBLIOTECAR, DATA\_ANGAJARE, and SALARIU. The table contains 11 rows of data.

ID_BIBLIOTECAR	ID_BIBLIOTECA	NUME_BIBLIOTECAR	DATA_ANGAJARE	SALARIU
1	1	2 Andrei Vasile	04-05-2010	3200
2	2	5 Pop Maria	26-01-2011	3000
3	3	3 Buraga Stefania	03-06-2020	2300
4	4	2 Huica Ruxandra	26-11-2018	2500
5	5	1 Pricop Elena	07-11-2009	4000
6	6	1 Simion Andreea	15-04-2015	2800
7	7	4 Dodan Felicia	21-11-2020	2200
8	8	4 Popa Andreea	10-07-2007	4100
9	9	5 Ionescu Mihai	12-05-2012	3800
10	10	5 Popovici David	18-08-2017	2600
11	11	6 Bontea Bogdan	17-05-2015	2800

insert into bibliotecar values (id\_bibliotecar.nextval, 2, 'Andrei Vasile', TO\_DATE('04-05-2010', 'DD-MM-YYYY'), 3200);

insert into bibliotecar values (id\_bibliotecar.nextval, 5, 'Pop Maria', TO\_DATE('26-01-2011', 'DD-MM-YYYY'), 3000);

insert into bibliotecar values (id\_bibliotecar.nextval, 3, 'Buraga Stefania', TO\_DATE('03-06-2020', 'DD-MM-YYYY'), 2300);

insert into bibliotecar values (id\_bibliotecar.nextval, 2, 'Huica Ruxandra', TO\_DATE('26-11-2018', 'DD-MM-YYYY'), 2500);

insert into bibliotecar values (id\_bibliotecar.nextval, 1, 'Pricop Elena', TO\_DATE('07-11-2009', 'DD-MM-YYYY'), 4000);

insert into bibliotecar values (id\_bibliotecar.nextval, 1, 'Simion Andreea', TO\_DATE('15-04-2015', 'DD-MM-YYYY'), 2800);

insert into bibliotecar values (id\_bibliotecar.nextval, 4, 'Dodan Felicia', TO\_DATE('21-11-2020', 'DD-MM-YYYY'), 2200);

insert into bibliotecar values (id\_bibliotecar.nextval, 4, 'Popa Andreea', TO\_DATE('10-07-2007', 'DD-MM-YYYY'), 4100);

insert into bibliotecar values (id\_bibliotecar.nextval, 5, 'Ionescu Mihai', TO\_DATE('12-05-2012', 'DD-MM-YYYY'), 3800);

insert into bibliotecar values (id\_bibliotecar.nextval, 5, 'Popovici David', TO\_DATE('18-08-2017', 'DD-MM-YYYY'), 2600);

insert into bibliotecar values (id\_bibliotecar.nextval, 6, 'Bontea Bogdan', TO\_DATE('17-05-2015', 'DD-MM-YYYY'), 2800);

--secventa + inserari MEMBRU

create sequence  
id\_membru

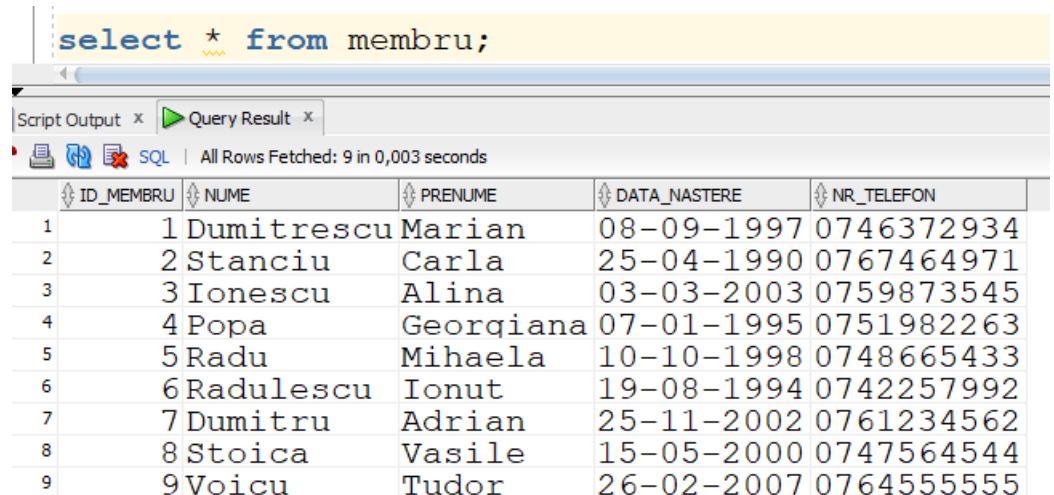
start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;



The screenshot shows a SQL query window with the query `select * from membru;` and its results. The results are displayed in a table with 6 columns: ID\_MEMBRU, NUME, PRENUME, DATA\_NASTERE, and NR\_TELEFON. There are 9 rows of data.

ID_MEMBRU	NUME	PRENUME	DATA_NASTERE	NR_TELEFON
1	Dumitrescu	Marian	08-09-1997	0746372934
2	Stanciu	Carla	25-04-1990	0767464971
3	Ionescu	Alina	03-03-2003	0759873545
4	Popa	Georgiana	07-01-1995	0751982263
5	Radu	Mihaela	10-10-1998	0748665433
6	Radulescu	Ionut	19-08-1994	0742257992
7	Dumitru	Adrian	25-11-2002	0761234562
8	Stoica	Vasile	15-05-2000	0747564544
9	Voicu	Tudor	26-02-2007	0764555555

insert into membru values (id\_membru.nextval, 'Dumitrescu', 'Marian', TO\_DATE('08-09-1997','DD-MM-YYYY'), '0746372934');

insert into membru values (id\_membru.nextval, 'Stanciu', 'Carla', TO\_DATE('25-04-1990','DD-MM-YYYY'), '0767464971');

insert into membru values (id\_membru.nextval, 'Ionescu', 'Alina', TO\_DATE('03-03-2003','DD-MM-YYYY'), '0759873545');

insert into membru values (id\_membru.nextval, 'Popa', 'Georgiana', TO\_DATE('07-01-1995','DD-MM-YYYY'), '0751982263');

insert into membru values (id\_membru.nextval, 'Radu', 'Mihaela', TO\_DATE('10-10-1998','DD-MM-YYYY'), '0748665433');

insert into membru values (id\_membru.nextval, 'Radulescu', 'Ionut', TO\_DATE('19-08-1994','DD-MM-YYYY'), '0742257992');

insert into membru values (id\_membru.nextval, 'Dumitru', 'Adrian', TO\_DATE('25-11-2002','DD-MM-YYYY'), '0761234562');

insert into membru values (id\_membru.nextval, 'Stoica', 'Vasile', TO\_DATE('15-05-2000','DD-MM-YYYY'), '0747564544');

insert into membru values (id\_membru.nextval, 'Voicu', 'Tudor', TO\_DATE('26-02-2007','DD-MM-YYYY'), '0764555555');

--secventa + inserari GEN

create sequence id\_gen

start with 1

increment by 1



minvalue 0

maxvalue 9999

nocycle;

insert into gen values(id\_gen.nextval, 'Fictiune');

insert into gen values(id\_gen.nextval, 'Poezie');

insert into gen values(id\_gen.nextval, 'Romantic');

insert into gen values(id\_gen.nextval, 'Biografie');

insert into gen values(id\_gen.nextval, 'Mister');

insert into gen values(id\_gen.nextval, 'Dezvoltare personala');

--secventa + inserari EDITURA

create sequence id\_editura

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into editura values(id\_editura.nextval, 'Humanitas', 'www.humanitas.ro',  
'0763254758', 1990);

insert into editura values(id\_editura.nextval, 'Corint', 'www.corint.ro', '0214088350', 1994);

insert into editura values(id\_editura.nextval, 'Litera', 'www.litera.ro', '074567465', 1992);

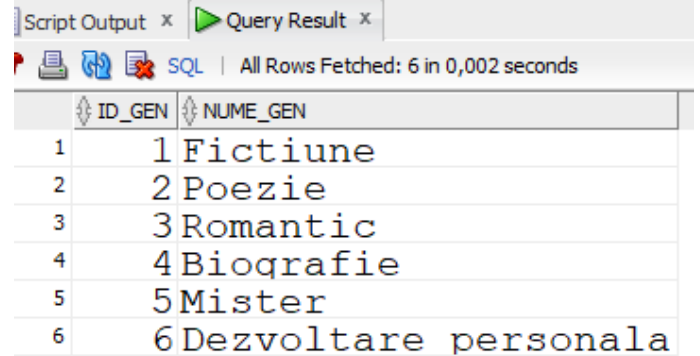
insert into editura values(id\_editura.nextval, 'Nemira', 'www.nemira.ro', '0721747464', 1991);

insert into editura values(id\_editura.nextval, 'Berkley Books', 'www.berkley.ro',  
'0216356565', 1985);

insert into editura values(id\_editura.nextval, 'Trei', 'www.trei.ro', '0213006090', 1995);

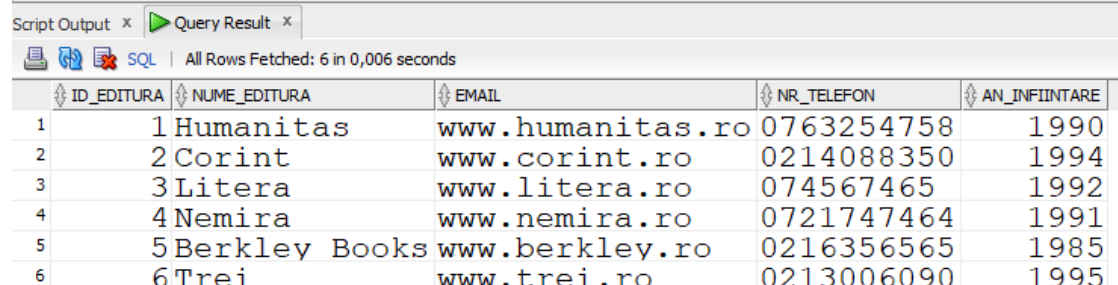
--secventa + inserari AUTOR

```
select * from gen;
```



ID_GEN	NUME_GEN
1	1 Fictiune
2	2 Poezie
3	3 Romantic
4	4 Biografie
5	5 Mister
6	6 Dezvoltare personala

```
select * from editura;
```



ID_EDITURA	NUME_EDITURA	EMAIL	NR_TELEFON	AN_INFIINTARE
1	1 Humanitas	www.humanitas.ro	0763254758	1990
2	2 Corint	www.corint.ro	0214088350	1994
3	3 Litera	www.litera.ro	074567465	1992
4	4 Nemira	www.nemira.ro	0721747464	1991
5	5 Berkley Books	www.berkley.ro	0216356565	1985
6	6 Trei	www.trei.ro	0213006090	1995



create sequence id\_autor

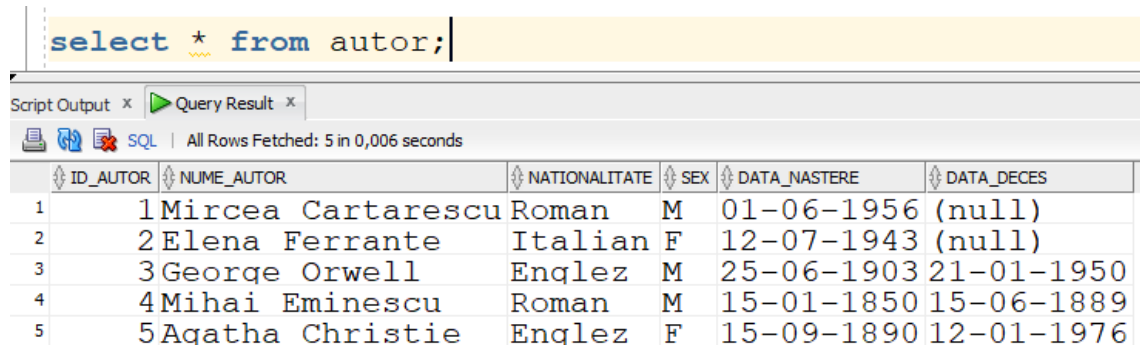
start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;



ID_AUTOR	NUME_AUTOR	NATIONALITATE	SEX	DATA_NASTERE	DATA_DECES
1	Mircea Cartarescu	Roman	M	01-06-1956	(null)
2	Elena Ferrante	Italian	F	12-07-1943	(null)
3	George Orwell	Englez	M	25-06-1903	21-01-1950
4	Mihai Eminescu	Roman	M	15-01-1850	15-06-1889
5	Agatha Christie	Englez	F	15-09-1890	12-01-1976

insert into autor values (id\_autor.nextval, 'Mircea Cartarescu', 'Roman', 'M', TO\_DATE('01-06-1956', 'DD-MM-YYYY'), NULL);

insert into autor values (id\_autor.nextval, 'Elena Ferrante', 'Italian', 'F', TO\_DATE('12-07-1943', 'DD-MM-YYYY'), NULL);

insert into autor values (id\_autor.nextval, 'George Orwell', 'Englez', 'M', TO\_DATE('25-6-1903', 'DD-MM-YYYY'), TO\_DATE('21-01-1950', 'DD-MM-YYYY'));

insert into autor values (id\_autor.nextval, 'Mihai Eminescu', 'Roman', 'M', TO\_DATE('15-01-1850', 'DD-MM-YYYY'), TO\_DATE('15-06-1889', 'DD-MM-YYYY'));

insert into autor values (id\_autor.nextval, 'Agatha Christie', 'Englez', 'F', TO\_DATE('15-09-1890', 'DD-MM-YYYY'), TO\_DATE('12-01-1976', 'DD-MM-YYYY'));

--secventa + inserari CARTE

create sequence id\_carte

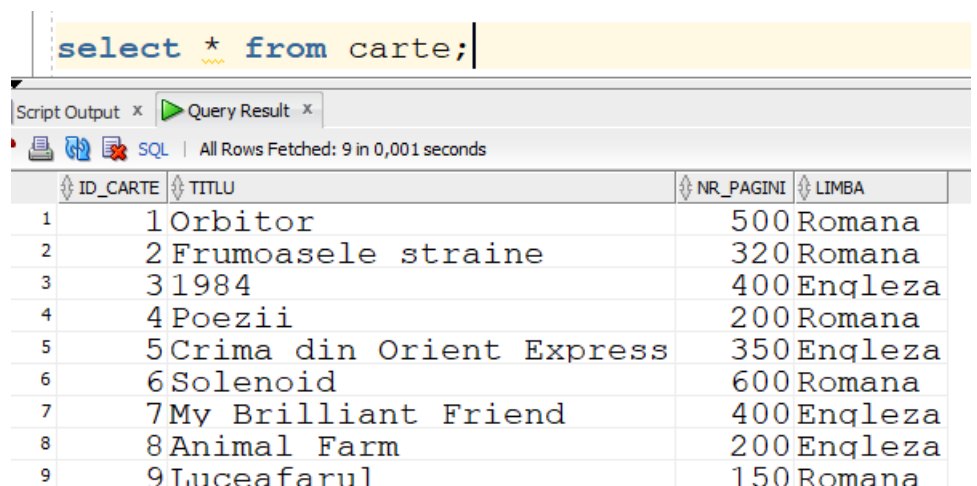
start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;



ID_CARTE	TITLU	NR_PAGINI	LIMBA
1	Orbitor	500	Romana
2	Frumoasele straine	320	Romana
3	1984	400	Engleza
4	Poezii	200	Romana
5	Crima din Orient Express	350	Engleza
6	Solenoid	600	Romana
7	My Brilliant Friend	400	Engleza
8	Animal Farm	200	Engleza
9	Luceafarul	150	Romana

insert into carte values(id\_carte.nextval, 'Orbitor', 500, 'Romana');

insert into carte values(id\_carte.nextval, 'Frumoasele straine', 320, 'Romana');

insert into carte values(id\_carte.nextval, '1984', 400, 'Engleza');

insert into carte values(id\_carte.nextval, 'Poezii', 200, 'Romana');

```

insert into carte values(id_carte.nextval, 'Crima din Orient Express', 350, 'Engleza');
insert into carte values(id_carte.nextval, 'Solenoid', 600, 'Romana');
insert into carte values(id_carte.nextval, 'My Brilliant Friend', 400, 'Engleza');
insert into carte values(id_carte.nextval, 'Animal Farm', 200, 'Engleza');
insert into carte values(id_carte.nextval, 'Luceafarul', 150, 'Romana');

--inserari PUBLICA

-- Orbitor, Mircea Cartarescu, Humanitas
insert into publica values (1, 1, 1, TO_DATE('05-09-2002', 'DD-MM-YYYY'));

-- Frumoasele străine, Mircea Cartarescu, Humanitas
insert into publica values(2, 1, 1, TO_DATE('21-12-2004', 'DD-MM-YYYY'));

-- 1984, George Orwell, Corint
insert into publica values(3, 3, 2, TO_DATE('10-04-1949', 'DD-MM-YYYY'));

-- Poezii, Mihai Eminescu, Litera
insert into publica values(4, 4, 3, TO_DATE('19-07-1883', 'DD-MM-YYYY'));

-- Crima din Orient Express, Agatha Christie, Berkley Books
insert into publica values(5, 5, 5, TO_DATE('16-02-1934', 'DD-MM-YYYY'));

-- Solenoid, Mircea Cartarescu, Humanitas
insert into publica values(6, 1, 1, TO_DATE('29-08-2005', 'DD-MM-YYYY'));

-- My Brilliant Friend, Elena Ferrante, Corint
insert into publica values(7, 2, 2, TO_DATE('08-06-2011', 'DD-MM-YYYY'));

-- Animal Farm, George Orwell, Corint
insert into publica values(8, 3, 2, TO_DATE('14-11-1945', 'DD-MM-YYYY'));

-- Luceafărul, Mihai Eminescu, Litera
insert into publica values(9, 4, 3, TO_DATE('07-09-1883', 'DD-MM-YYYY'));

```

```
select * from publica;
```

ID_CARTE	ID_AUTOR	ID_EDITURA	DATA_PUBLICARE
1	9	1	1 05-09-2002
2	9	3	2 10-04-1949
3	1	1	1 05-09-2002
4	2	1	1 21-12-2004
5	3	3	2 10-04-1949
6	4	4	3 19-07-1883
7	5	5	5 16-02-1934
8	6	1	1 29-08-2005
9	7	2	2 08-06-2011
10	8	3	2 14-11-1945
11	9	4	3 07-09-1883

--inserari IMPRUMUTA

```
insert into imprumuta values (5, 4, 1, TO_DATE('01-01-2023', 'DD-MM-YYYY'),  
TO_DATE('15-01-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (3, 2, 2, TO_DATE('05-02-2023', 'DD-MM-YYYY'),  
TO_DATE('28-02-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (6, 5, 3, TO_DATE('10-03-2023', 'DD-MM-YYYY'),  
TO_DATE('30-03-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (9, 3, 4, TO_DATE('15-04-2023', 'DD-MM-YYYY'),  
TO_DATE('15-05-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (2, 4, 5, TO_DATE('20-05-2023', 'DD-MM-YYYY'),  
TO_DATE('10-06-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (5, 1, 6, TO_DATE('25-06-2023', 'DD-MM-YYYY'),  
TO_DATE('20-07-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (9, 1, 7, TO_DATE('01-08-2023', 'DD-MM-YYYY'),  
TO_DATE('31-08-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (2, 4, 8, TO_DATE('05-09-2023', 'DD-MM-YYYY'),  
TO_DATE('30-09-2023', 'DD-MM-YYYY'));
```

```
insert into imprumuta values (3, 5, 2, TO_DATE('10-10-2023', 'DD-MM-YYYY'),  
TO_DATE('05-11-2023', 'DD-MM-YYYY'));
```

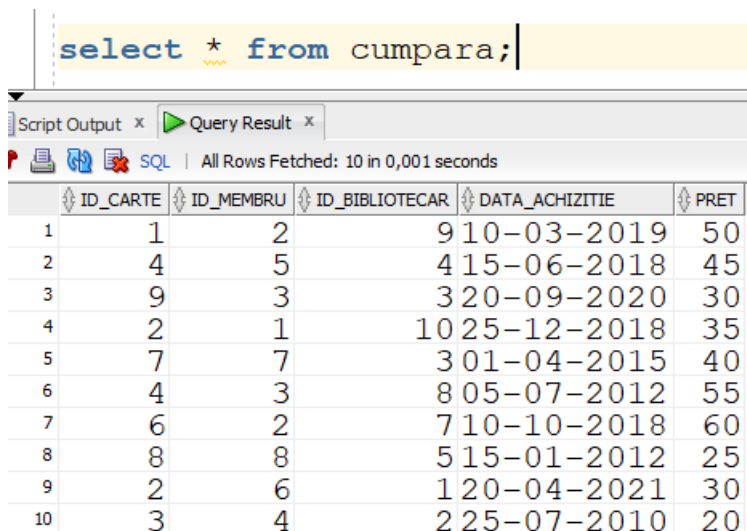
```
insert into imprumuta values (8, 3, 3, TO_DATE('15-11-2023', 'DD-MM-YYYY'),  
TO_DATE('10-12-2023', 'DD-MM-YYYY'));
```

```
select * from imprumuta;
```

ID_CARTE	ID_MEMBRU	ID_BIBLIOTECAR	DATA_IMPRUMUT	DATA_RESTITUIRE
1	5	4	1 01-01-2023	15-01-2023
2	3	2	2 05-02-2023	28-02-2023
3	6	5	3 10-03-2023	30-03-2023
4	9	3	4 15-04-2023	15-05-2023
5	2	4	5 20-05-2023	10-06-2023
6	5	1	6 25-06-2023	20-07-2023
7	9	1	7 01-08-2023	31-08-2023
8	2	4	8 05-09-2023	30-09-2023
9	3	5	2 10-10-2023	05-11-2023
10	8	3	3 15-11-2023	10-12-2023

--inserari CUMPARA

```
insert into cumpara values(1, 2, 9, TO_DATE('10-03-2019', 'DD-MM-YYYY'), 50);
insert into cumpara values(4, 5, 4, TO_DATE('15-06-2018', 'DD-MM-YYYY'), 45);
insert into cumpara values(9, 3, 3, TO_DATE('20-09-2020', 'DD-MM-YYYY'), 30);
insert into cumpara values(2, 1, 10, TO_DATE('25-12-2018', 'DD-MM-YYYY'), 35);
insert into cumpara values(7, 7, 3, TO_DATE('01-04-2015', 'DD-MM-YYYY'), 40);
insert into cumpara values(4, 3, 8, TO_DATE('05-07-2012', 'DD-MM-YYYY'), 55);
insert into cumpara values(6, 2, 7, TO_DATE('10-10-2018', 'DD-MM-YYYY'), 60);
insert into cumpara values(8, 8, 5, TO_DATE('15-01-2012', 'DD-MM-YYYY'), 25);
insert into cumpara values(2, 6, 1, TO_DATE('20-04-2021', 'DD-MM-YYYY'), 30);
insert into cumpara values(3, 4, 2, TO_DATE('25-07-2010', 'DD-MM-YYYY'), 20);
```



```
select * from cumpara;
```

ID_CARTE	ID_MEMBRU	ID_BIBLIOTECAR	DATA_ACHIZITIE	PRET
1	1	2	9 10-03-2019	50
2	4	5	4 15-06-2018	45
3	9	3	3 20-09-2020	30
4	2	1	10 25-12-2018	35
5	7	7	3 01-04-2015	40
6	4	3	8 05-07-2012	55
7	6	2	7 10-10-2018	60
8	8	8	5 15-01-2012	25
9	2	6	1 20-04-2021	30
10	3	4	2 25-07-2010	20

--inserari ARE

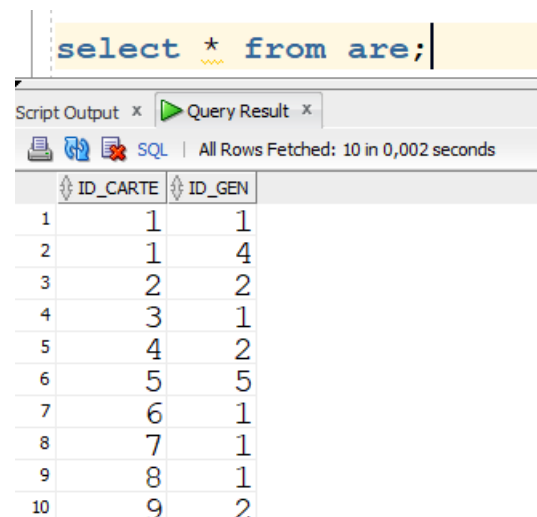
--Orbitor, fictiune

insert into are values(1, 1);

--Frumoasele straine, poezie

insert into are values(2, 2);

--1984, fictiune



```
select * from are;
```

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

```

insert into are values(3, 1);

--Poezii, poezie

insert into are values(4, 2);

--Crima din Orient Express, Mister

insert into are values(5, 5);

--Solenoid, fictiune

insert into are values(6, 1);

--My Brilliant Friend, fictiune

insert into are values(7, 1);

--Animal Farm, fictiune

insert into are values(8, 1);

--Luceafarul, poezie

insert into are values(9, 2);

--Orbitor, biografie

insert into are values(1,4);

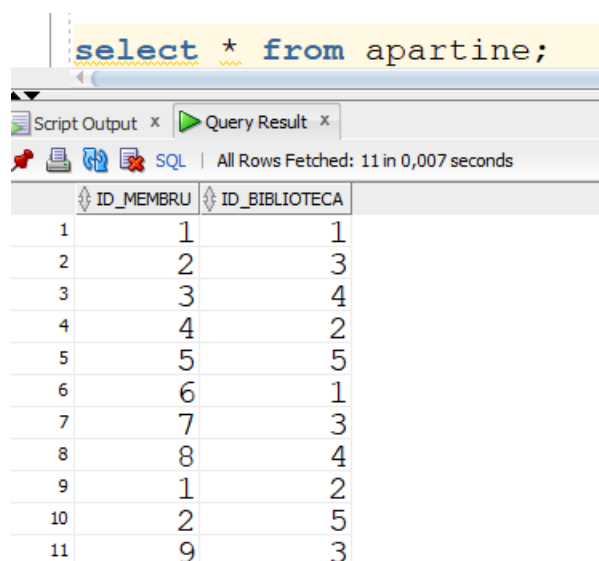
```

```
--inserari APARTINE
```

```

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

```



The screenshot shows a SQL query result in a web interface. The query is `select * from apartine;`. The result is displayed in a table with two columns: `ID_MEMBRU` and `ID_BIBLIOTECA`. The table contains 11 rows of data. The interface also shows a status bar indicating 'All Rows Fetched: 11 in 0,007 seconds'.

	ID_MEMBRU	ID_BIBLIOTECA
1	1	1
2	2	3
3	3	4
4	4	2
5	5	5
6	6	1
7	7	3
8	8	4
9	1	2
10	2	5
11	9	3

## 12. Crearea a 5 cereri complexe în SQL

--cerinta 12

--1. Sa se afiseze ID-ul si numele bibliotecii care are salariu mediu

--pentru bibliotecari mai mic decat salariul mediu maxim din

--bibliotecile din Cluj-Napoca.

--grupari de date cu subcereri nesincronizate in care

--intervin cel putin 3 tabele, functii grup,

--filtrare la nivel de grupuri (in cadrul aceleiasi cereri)

select b.id\_biblioteca, b.numa\_biblioteca, round(avg(bi.salariu))

from biblioteca b, bibliotecar bi

where b.id\_biblioteca = bi.id\_biblioteca

group by b.id\_biblioteca, b.numa\_biblioteca

having round(avg(bi.salariu)) < any

(select max(round(avg(salariu)))

from biblioteca b, bibliotecar bi, locatie l

where b.id\_biblioteca = bi.id\_bibliotecar

and b.id\_locatie = l.id\_locatie

and lower(l.oras) = lower('Cluj-Napoca')

group by b.id\_biblioteca);

xsneet Query Builder

Cerinta 12

```
--1. Sa se afiseze ID-ul si numele bibliotecii care are salariu mediu
--pentru bibliotecari mai mic decat salariul mediu maxim din
--bibliotecile din Cluj-Napoca.

--grupari de date cu subcereri nesincronizate in care
--intervin cel putin 3 tabele, functii grup,
--filtrare la nivel de grupuri (in cadrul aceleiasi cereri)
select b.id_biblioteca, b.numa_biblioteca, round(avg(bi.salariu))
from biblioteca b, bibliotecar bi
where b.id_biblioteca = bi.id_bibliotecar
group by b.id_biblioteca, b.numa_biblioteca
having round(avg(bi.salariu)) < any
    (select max(round(avg(salariu)))
     from biblioteca b, bibliotecar bi, locatie l
     where b.id_biblioteca = bi.id_bibliotecar
     and b.id_locatie = l.id_locatie
     and lower(l.oras) = lower('Cluj-Napoca'))
group by b.id_biblioteca);
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0,048 seconds

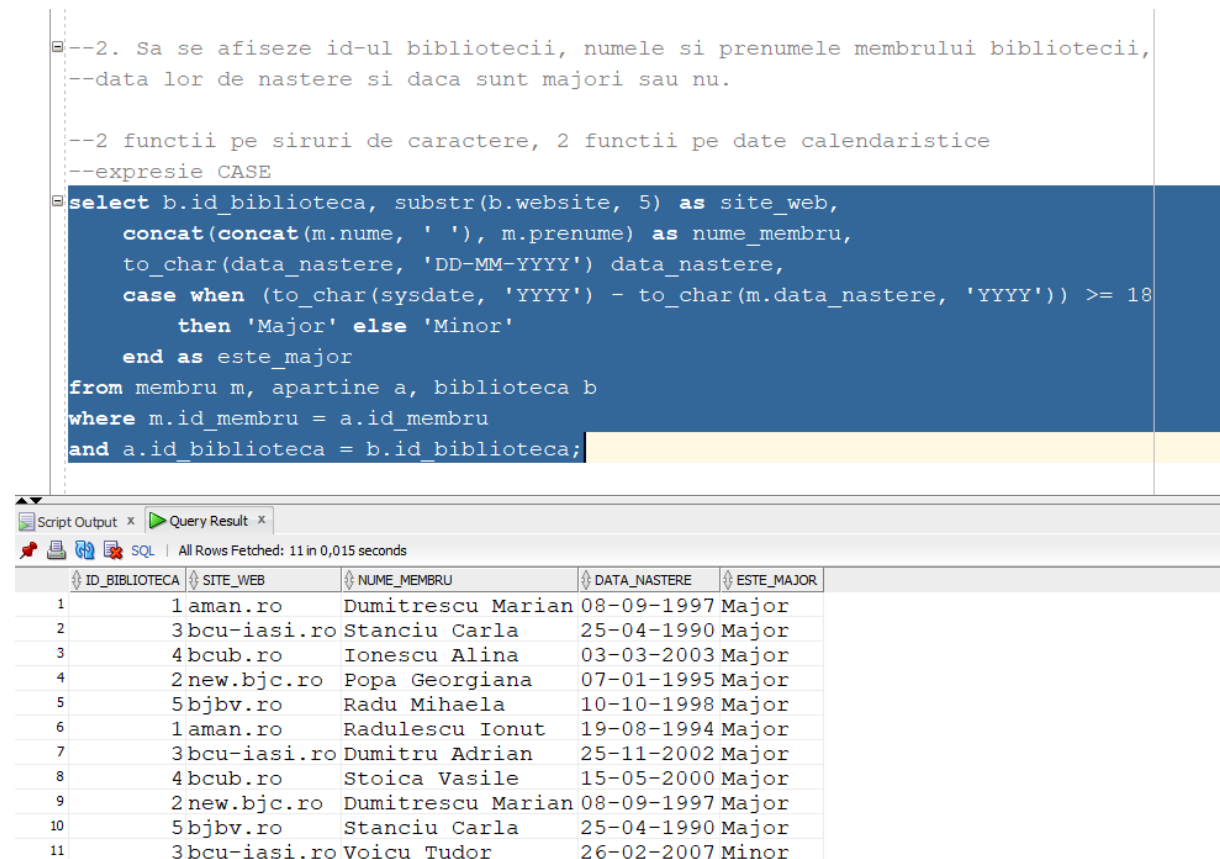
ID_BIBLIOTECA	NUMA_BIBLIOTECA	ROUND(AVG(BI.SALARIU))
1	2 Biblioteca Judeteana "Octavian Goga"	2850
2	6 Biblioteca de Fizica	2800
3	3 Biblioteca Centrala Universitara "Mihai Eminescu"	2300

--2. Sa se afiseze id-ul bibliotecii, numele si prenumele membrului bibliotecii,  
--data lor de nastere si daca sunt majori sau nu.

--2 functii pe siruri de caractere, 2 functii pe date calendaristice

--expresie CASE

```
select b.id_biblioteca, substr(b.website, 5) as site_web,  
       concat(concat(m.nume, ' '), m.prenume) as nume_membru,  
       to_char(data_nastere, 'DD-MM-YYYY') data_nastere,  
       case when (to_char(sysdate, 'YYYY') - to_char(m.data_nastere, 'YYYY')) >= 18  
             then 'Major' else 'Minor'  
       end as este_major  
from membru m, apartine a, biblioteca b  
where m.id_membru = a.id_membru  
and a.id_biblioteca = b.id_biblioteca;
```



The screenshot shows a SQL IDE with a query editor at the top and a results pane at the bottom. The query editor contains the same SQL code as the previous block. The results pane shows the output of the query, with columns ID\_BIBLIOTECA, SITE\_WEB, NUME\_MEMBRU, DATA\_NASTERE, and ESTE\_MAJOR. The results are displayed in a table with 11 rows.

ID_BIBLIOTECA	SITE_WEB	NUME_MEMBRU	DATA_NASTERE	ESTE_MAJOR
1	1aman.ro	Dumitrescu Marian	08-09-1997	Major
2	3bcu-iasi.ro	Stanciu Carla	25-04-1990	Major
3	4bcub.ro	Ionescu Alina	03-03-2003	Major
4	2new.bjc.ro	Popa Georgiana	07-01-1995	Major
5	5bjbv.ro	Radu Mihaela	10-10-1998	Major
6	1aman.ro	Radulescu Ionut	19-08-1994	Major
7	3bcu-iasi.ro	Dumitru Adrian	25-11-2002	Major
8	4bcub.ro	Stoica Vasile	15-05-2000	Major
9	2new.bjc.ro	Dumitrescu Marian	08-09-1997	Major
10	5bjbv.ro	Stanciu Carla	25-04-1990	Major
11	3bcu-iasi.ro	Voicu Tudor	26-02-2007	Minor

--3. Sa se afiseze pentru fiecare carte, id-ul si titlul cartii,  
 --id-ul membrului care a cumparat-o si numele bibliotecii  
 --de unde a luat-o.

--utilizarea a cel puțin 1 bloc de cerere(clauza WITH)

--subcereri nesincronizate în clauza FROM

--ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)

with carte1 as

(select distinct c.id\_carte, c.titlu, p.id\_membru, p.id\_bibliotecar

from carte c, cumpara p

where c.id\_carte = p.id\_carte(+))

select carte1.id\_carte, carte1.titlu, carte1.id\_membru,

decode(nvl(bib.id\_bibliotecar, 0), 0, 'Nu a fost cumparata', bib.id\_bibliotecar) vanzator,

bib.numa\_biblioteca

from carte1,

(select id\_bibliotecar, nume\_biblioteca

from biblioteca b, bibliotecar bi

where b.id\_biblioteca = bi.id\_biblioteca) bib

where carte1.id\_bibliotecar = bib.id\_bibliotecar(+)

order by 1;

<pre>--3. Sa se afiseze pentru fiecare carte, id-ul si titlul cartii, --id-ul membrului care a cumparat-o si numele bibliotecii --de unde a luat-o.  --utilizarea a cel puțin 1 bloc de cerere(clauza WITH) --subcereri nesincronizate în clauza FROM --ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)  with carte1 as   (select distinct c.id_carte, c.titlu, p.id_membru, p.id_bibliotecar    from carte c, cumpara p    where c.id_carte = p.id_carte(+)) select carte1.id_carte, carte1.titlu, carte1.id_membru,   decode(nvl(bib.id_bibliotecar, 0), 0, 'Nu a fost cumparata', bib.id_bibliotecar) vanzator,   bib.numa_biblioteca from carte1,   (select id_bibliotecar, nume_biblioteca    from biblioteca b, bibliotecar bi    where b.id_biblioteca = bi.id_biblioteca) bib where carte1.id_bibliotecar = bib.id_bibliotecar(+) order by 1;</pre>				
ID_CARTE	TITLU	ID_MEMBRU	VANZATOR	NUME_BIBLIOTECA
1	1Orbitor	29		Biblioteca Judeteana "Gheorghe Baritui"
2	2Frumoasele straine	61		Biblioteca Judeteana "Octavian Goga"
3	2Frumoasele straine	110		Biblioteca Judeteana "Gheorghe Baritui"
4	31984	42		Biblioteca Judeteana "Gheorghe Baritui"
5	4Poezii	54		Biblioteca Judeteana "Octavian Goga"
6	4Poezii	38		Biblioteca Centrala Universitara "Carol I"
7	5Crima din Orient Express	(null)	Nu a fost cumparata	(null)
8	6Solenoid	27		Biblioteca Centrala Universitara "Carol I"
9	7My Brilliant Friend	73		Biblioteca Centrala Universitara "Mihai Eminescu
10	8Animal Farm	85		Biblioteca Judeteana "Alexandru si Aristia Aman"
11	9Lucaafarul	33		Biblioteca Centrala Universitara "Mihai Eminescu



--4. Sa se afiseze id-ul si numele bibliotecii, numele si salariul  
--celor mai slabi platiti bibliotecari din fiecare biblioteca care  
--au efectuat cel putin un imprumut si o vanzare.

--subcerere sincronizata in care intervin cel putin 3 tabele

```
select b.id_biblioteca, b.numa_biblioteca, bi.numa_bibliotecar, bi.salariu
from biblioteca b, bibliotecar bi
where bi.id_biblioteca = b.id_biblioteca
and bi.salariu = (
    select min(bi2.salariu)
    from bibliotecar bi2, imprumuta i, cumpara c
    where bi2.id_bibliotecar = i.id_bibliotecar
    and c.id_bibliotecar = bi2.id_bibliotecar
    and bi2.id_biblioteca = b.id_biblioteca);
```

--4. Sa se afiseze id-ul si numele bibliotecii, numele si salariul  
--celor mai slabi platiti bibliotecari din fiecare biblioteca care  
--au efectuat cel putin un imprumut si o vanzare.

--subcerere sincronizata in care intervin cel putin 3 tabele

```
select b.id_biblioteca, b.numa_biblioteca, bi.numa_bibliotecar, bi.salariu
from biblioteca b, bibliotecar bi
where bi.id_biblioteca = b.id_biblioteca
and bi.salariu = (
    select min(bi2.salariu)
    from bibliotecar bi2, imprumuta i, cumpara c
    where bi2.id_bibliotecar = i.id_bibliotecar
    and c.id_bibliotecar = bi2.id_bibliotecar
    and bi2.id_biblioteca = b.id_biblioteca);
```

5. Sa se afiseze numele autorilor care au publicat cel putin o carte

ID_BIBLIOTECA	NUME_BIBLIOTECA	NUME_BIBLIOTECAR	SALARIU
1	4Biblioteca Centrala Universitara "Carol I"	Dodan Felicia	2200
2	5Biblioteca Judeteana "Gheorghe Baritui"	Pop Maria	3000
3	2Biblioteca Judeteana "Octavian Goga"	Huica Ruxandra	2500
4	3Biblioteca Centrala Universitara "Mihai Eminescu	Buraga Stefania	2300
5	1Biblioteca Judeteana "Alexandru si Aristia Aman"	Pricop Elena	4000

--5. Sa se afiseze numele autorilor care au publicat cel putin o carte

--inainte de anul 1946 si care nu au nicio carte publicata dupa anul 1950.

--utilizarea a cel puțin 1 bloc de cerere(clauza WITH)

with carti\_anterioare as

```
(select distinct a.nume_autor
from autor a, publica p
where a.id_autor = p.id_autor
and to_char(p.data_publicare, 'YYYY') < '1946'),
```

carti\_recente as

```
(select distinct a.nume_autor
from autor a, publica p
where a.id_autor = p.id_autor
and to_char(p.data_publicare, 'YYYY') >= '1950')
```

select c1.nume\_autor

from carti\_anterioare c1

where c1.nume\_autor not in

```
(select c2.nume_autor
from carti_recente c2);
```

```
--5. Sa se afiseze numele autorilor care au publicat cel putin o carte
--inainte de anul 1946 si care nu au nicio carte publicata dupa anul 1950.
```

```
--utilizarea a cel puțin 1 bloc de cerere(clauza WITH)
```

```
with carti_anterioare as
  (select distinct a.nume_autor
   from autor a, publica p
   where a.id_autor = p.id_autor
   and to_char(p.data_publicare, 'YYYY') < '1946'),
carti_recente as
  (select distinct a.nume_autor
   from autor a, publica p
   where a.id_autor = p.id_autor
   and to_char(p.data_publicare, 'YYYY') >= '1950')
select c1.nume_autor
from carti_anterioare c1
where c1.nume_autor not in
  (select c2.nume_autor
   from carti_recente c2);
```

Script Output x	Query Result x
SQL   All Rows Fetched: 3 in 0,005 seconds	
NUME_AUTOR	
1	George Orwell
2	Mihai Eminescu
3	Agatha Christie

### 13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

--cerinta 13

--1. operatie de actualizare

update membru

set nr\_telefon = null

where nume in (select m.nume from membru m where lower(m.nume) like '%radu%');

```
--cerinta 13

--1. operatie de actualizare

update membru
set nr_telefon = null
where nume in (select m.nume from membru m where lower(m.nume) like '%radu%');
```

Script Output x Query Result x  
Task completed in 0,045 seconds

2 rows updated.

--2. operatie de suprimare

delete from locatie

where oras = (select oras from locatie where lower(oras) like '%timisoara%');

```
--2. operatie de suprimare

delete from locatie
where oras = (select oras from locatie where lower(oras) like '%timisoara%');
```

Script Output x Query Result x  
Task completed in 0,055 seconds

1 row deleted.

--3. operatie de actualizare

update bibliotecar

set salariu = 4200

where data\_angajare in (select data\_angajare from bibliotecar where to\_char(data\_angajare, 'YYYY') <= '2010');

```
--3. operatie de actualizare

update bibliotecar
set salariu = 4200
where data_angajare in (select data_angajare from bibliotecar where to_char(data_angajare, 'YYYY') <= '2010');
```

Script Output x Query Result x  
Task completed in 0,058 seconds

3 rows updated.

## 15. Crearea a 3 cereri în SQL.

### 15.1 Cerere ce utilizează operația outer-join pe minimum 4 tabele.

--cerinta 15

--1. Sa se afiseze id-urile pentru locatie, biblioteca, membru si cartile

--cumparate de fiecare membru, inclusiv pt locatiile in care nu exista biblioteci

--si bibliotecile care nu au niciun membru si membrii care nu au cumparat nicio carte.

--outer join

```
SELECT l.ID_LOCATIE, b.ID_BIBLIOTECA, m.ID_MEMBRU, c.ID_CARTE
```

```
FROM LOCATIE l, BIBLIOTECA b, MEMBRU m, CUMPARA c, apartine ap
```

```
WHERE l.ID_LOCATIE = b.id_locatie (+)
```

```
and b.id_biblioteca = ap.id_biblioteca (+)
```

```
AND m.ID_MEMBRU = c.ID_MEMBRU (+)
```

```
and m.id_membru(+) = ap.id_membru;
```

--cerinta 15

--1. Sa se afiseze id-urile pentru locatie, biblioteca, membru si cartile  
--cumparate de fiecare membru, inclusiv pt locatiile in care nu exista biblioteci  
--si bibliotecile care nu au niciun membru si membrii care nu au cumparat nicio carte.

```
SELECT l.ID_LOCATIE, b.ID_BIBLIOTECA, m.ID_MEMBRU, c.ID_CARTE
FROM LOCATIE l, BIBLIOTECA b, MEMBRU m, CUMPARA c, apartine ap
WHERE l.ID_LOCATIE = b.id_locatie (+)
and b.id_biblioteca = ap.id_biblioteca (+)
AND m.ID_MEMBRU = c.ID_MEMBRU (+)
and m.id_membru(+) = ap.id_membru;
```

Script Output x Query Result x

All Rows Fetched: 15 in 0,001 seconds

	ID_LOCATIE	ID_BIBLIOTECA	ID_MEMBRU	ID_CARTE
1	4	1	1	2
2	5	3	2	1
3	5	3	2	6
4	1	4	3	4
5	1	4	3	9
6	2	2	4	3
7	3	5	5	4
8	4	1	6	2
9	5	3	7	7
10	1	4	8	8
11	2	2	1	2
12	3	5	2	1
13	3	5	2	6
14	5	3	9	(null)
15	7	6	(null)	(null)

## 15.2 Cerere ce utilizează operația division.

--2. Sa se afiseze toti membrii care apartin tuturor bibliotecilor.

--division

insert into apartine values(1,3);

insert into apartine values(1,4);

insert into apartine values(1,5);

insert into apartine values(1,6);

select distinct m.id\_membru, m.nume

from membru m

where not exists

(select 1

from biblioteca b

where not exists

(select 1

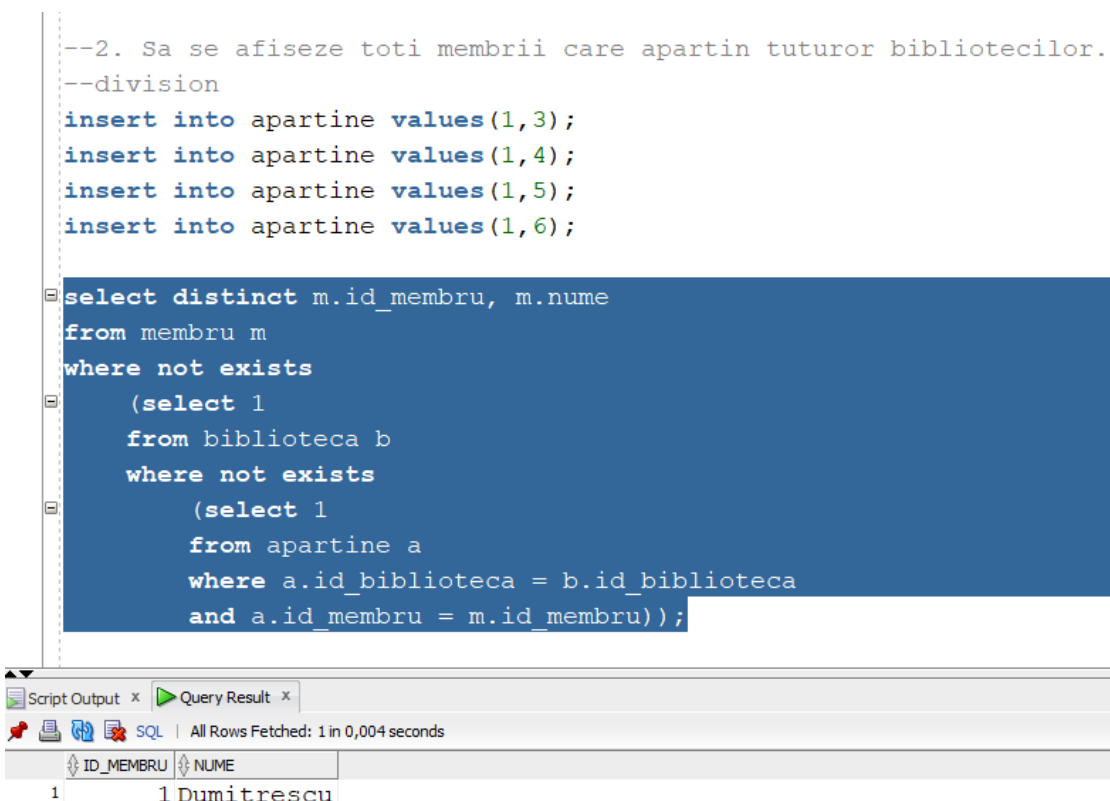
from apartine a

where a.id\_biblioteca = b.id\_biblioteca

and a.id\_membru = m.id\_membru));

```
--2. Sa se afiseze toti membrii care apartin tuturor bibliotecilor.
--division
insert into apartine values(1,3);
insert into apartine values(1,4);
insert into apartine values(1,5);
insert into apartine values(1,6);

select distinct m.id_membru, m.nume
from membru m
where not exists
    (select 1
     from biblioteca b
     where not exists
         (select 1
          from apartine a
          where a.id_biblioteca = b.id_biblioteca
               and a.id_membru = m.id_membru));
```



The screenshot shows a SQL IDE interface. The top part displays the SQL query being executed. Below the query, there is a blue box containing the same query text. At the bottom, the IDE's status bar shows 'Script Output' and 'Query Result' tabs. The 'Query Result' tab is active, displaying the results of the query. The results are shown in a table with two columns: 'ID\_MEMBRU' and 'NUME'. The first row shows the value '1' for 'ID\_MEMBRU' and '1 Dumitrescu' for 'NUME'.

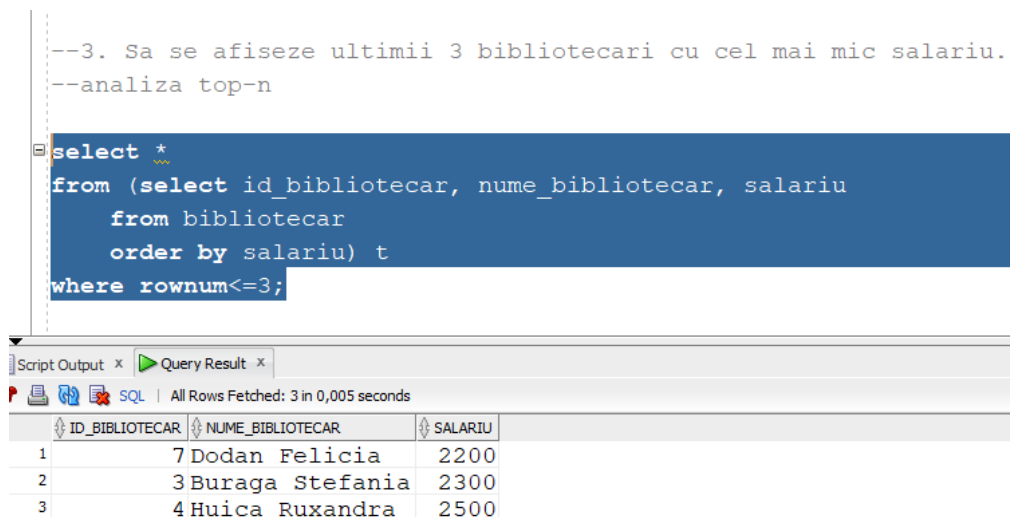
ID_MEMBRU	NUME
1	1 Dumitrescu

## 15.3 Cerere care implementează analiza top-n.

--3. Sa se afiseze ultimii 3 bibliotecari cu cel mai mic salariu.

--analiza top-n

```
select *  
from (select id_bibliotecar, nume_bibliotecar, salariu  
      from bibliotecar  
      order by salariu) t  
where rownum<=3;
```



The screenshot shows a SQL query editor with a blue background for the query text. The query is the same as the one in the previous block. Below the query, there is a 'Query Result' tab showing the results of the query. The results are displayed in a table with three columns: ID\_BIBLIOTECAR, NUME\_BIBLIOTECAR, and SALARIU. The results are ordered by salary in ascending order, and only the top 3 rows are shown.

ID_BIBLIOTECAR	NUME_BIBLIOTECAR	SALARIU
1	7 Dodan Felicia	2200
2	3 Buraga Stefania	2300
3	4 Huica Ruxandra	2500

## 17. Normalizări BCNF, FN4, FN5

### 17.1 Realizarea normalizării BCNF, FN4, FN5

#### 17.1.1 Forma normală Boyce-Codd

Pentru ca o relație să fie în FN1, trebuie să satisfacă atât regulile pentru FN3, cât și pentru oricare dependență funcțională netrivială  $A \rightarrow B$ , A trebuie să fie *supercheie*.

Schema este scoasă din BCNF deoarece în tabelul CUMPARA, id\_carte și id\_autor determină prețul. Dacă presupunem că id\_carte determină prețul și aplicăm normalizarea BCNF cu ajutorul regulii Casey-Delobel, tabelul CUMPARA se va împărți astfel:

CUMPARA1(ID\_CARTE#, pret)

CUMPARA2(ID\_CARTE#, ID\_AUTOR#, ID\_MEMBRU#)

### 17.1.2 Forma normală 4

Pentru ca o relație să fie în FN4, trebuie să satisfacă regulile pentru FN3 și să nu conțină dependențe multivaluate. Pentru ca un tabel să conțină dependențe multivaluate, trebuie să aibă minim 3 coloane(atribute), pentru o singură valoare a unui atribut A, trebuie să existe mai mult de o valoare diferită B, iar coloanele B și C trebuie să fie independente una față de cealaltă.

Schema este scoasă din FN4 deoarece în tabelul PUBLICA, id\_autor și id\_editură nu depind unul de celălalt, iar id\_carte multi-determină id\_autor, dar și id\_editură. Pentru a aduce schema în FN4, împărțim relația în două tabele.

PUBLICA1(ID\_CARTE#, ID\_AUTOR#)

PUBLICA2(ID\_CARTE#, ID\_EDITURA#, data\_publicare)

### 17.1.3 Forma normală 5

Pentru ca o relație să fie în FN5, trebuie să satisfacă regulile pentru FN4 și să nu conțină dependențe de uniune (join dependencies) care nu sunt implicate de o cheie.

În cadrul acestui context, o carte poate fi achiziționată de mai mulți membrii de la același bibliotecar sau poate fi achiziționată de la un bibliotecar la un preț diferit și la o dată diferită. De asemenea, același membru poate achiziționa mai multe cărți, fie la același preț și aceeași dată, fie la prețuri și date diferite.

Schema relațională inițială: CUMPARA (ID\_CARTE#, ID\_BIBLIOTECAR#, ID\_MEMBRU#, data\_achizitie, pret)

CUMPARA1(ID\_CARTE#, ID\_BIBLIOTECAR#, ID\_MEMBRU#)

CUMPARA2(ID\_CARTE#, pret, data\_achizitie)

CUMPARA3(ID\_BIBLIOTECAR#, ID\_MEMBRU#, pret, data\_achizitie)

Pentru a obține rezultatul din primul tabel, putem face un join între primele două tabele și apoi să facem un join între rezultatul obținut și a treia tabelă.

## 17.2 Aplicarea denormalizării.

Presupunem că în această schemă înlocuim atributul an\_infiintare din tabelul EDITURA cu un alt tabel separat numit INFO\_EDITURA care să conțină anul înființării fiecărei edituri, și o cheie primară ID\_INFO. De asemenea, vom adăuga în tabelul EDITURA un FK numit ID\_INFO.

Schemele vor arăta astfel:

EDITURA(ID\_EDITURA#, id\_info, nume\_editura, email, nr\_telefon)

INFO\_EDITURA (ID\_INFO#, an\_infiintare)

Prin realizarea denormalizării, eliminăm tabela INFO\_EDITURA și revenim la schema inițială. Aceasta este benefică deoarece ar permite accesarea atributului an\_infiintare mult mai eficient, fără a fi nevoie de a folosi tabelul suplimentar INFO\_EDITURA.

EDITURA(ID\_EDITURA#, nume\_editura, email, nr\_telefon, an\_infiintare)