

Proiect baze de date

---

# BIBLIOTECA FACULTĂȚII

---

**Lungu Laura-Vanesa**

Anul I, Grupa 141

Facultatea de Matematică și Informatică

Universitatea din București

IUNIE 2023

# Cuprins

<b>1</b>	<b>Descriere și reguli de funcționare</b>	<b>3</b>
<b>2</b>	<b>Constrângeri</b>	<b>3</b>
<b>3</b>	<b>Entități și chei primare</b>	<b>4</b>
<b>4</b>	<b>Relații și cardinalități</b>	<b>5</b>
<b>5</b>	<b>Atribute</b>	<b>6</b>
<b>6</b>	<b>Diagrama Entitate-Relație</b>	<b>10</b>
<b>7</b>	<b>Diagrama Conceptuală</b>	<b>11</b>
<b>8</b>	<b>Scheme Relaționale</b>	<b>12</b>
<b>9</b>	<b>Normalizare</b>	<b>13</b>
9.1	Prima formă normală - FN1	13
9.2	A doua formă normală - FN2	14
9.3	A treia formă normală - FN3	15
<b>10</b>	<b>Crearea secvențelor</b>	<b>16</b>
<b>11</b>	<b>Crearea tabelelor</b>	<b>17</b>
11.1	Creare și inserare în tabelul <i>AUTOR</i>	17
11.2	Creare și inserare în tabelul <i>EDITURĂ</i>	19
11.3	Creare și inserare în tabelul <i>CARTE</i>	19
11.4	Creare și inserare în tabelul <i>SCRIERE</i>	22
11.5	Creare și inserare în tabelul <i>CITITOR</i>	24
11.6	Creare și inserare în tabelul <i>BIBLIOTECAR</i>	25
11.7	Creare și inserare în tabelul <i>SECȚIUNE</i>	27
11.8	Creare și inserare în tabelul <i>CATEGORIE</i>	28
11.9	Creare și inserare în tabelul <i>REZERVARE</i>	29
11.10	Creare și inserare în tabelul <i>TAXĂ</i>	31
11.11	Creare și inserare în tabelul <i>EXEMPLAR</i>	32
11.12	Creare și inserare în tabelul <i>ÎMPRUMUT</i>	35
<b>12</b>	<b>Cereri SQL</b>	<b>37</b>
12.1	Cererea I	37
12.2	Cererea II	38
12.3	Cererea III	39
12.4	Cererea IV	40
12.5	Cererea V	41
<b>13</b>	<b>Operații de actualizare și suprimare a datelor</b>	<b>42</b>
13.1	Operația 1	42
13.2	Operația 2	44
13.3	Operația 3	46
<b>14</b>	<b>Crearea unei vizualizări complexe</b>	<b>47</b>
<b>15</b>	<b>Cereri SQL</b>	<b>48</b>
15.1	Operația outer-join	48

15.2 Operația division . . . . .	49
15.3 Analiza top-n . . . . .	50
<b>16 Optimizarea unei cereri . . . . .</b>	<b>51</b>
16.1 Cererea SQL . . . . .	51
16.2 Optimizare . . . . .	51
16.3 Expresii algebrice . . . . .	51
16.4 Arbore algebric . . . . .	52
<b>17 Normalizarea BCNF, FN4, FN5 . . . . .</b>	<b>52</b>
17.1 Forma Normală Boyce-Codd . . . . .	52
17.2 FN4 . . . . .	53
17.3 FN5 . . . . .	53
17.4 Aplicarea denormalizării . . . . .	54
<b>18 Optimizarea cererilor prin indexare . . . . .</b>	<b>55</b>
18.1 Cererea I . . . . .	55
18.2 Cererea II . . . . .	55

# 1 Descriere și reguli de funcționare

În biblioteca Facultății de Matematică și Informatică din cadrul Universității din București se găsesc mai multe categorii de cărți. Pentru a facilita accesul studenților, al profesorilor și al personalului bibliotecii la materialele bibliotecii, conducerea acesteia decide să implementeze o bază de date pentru o platformă online care să îi ajute pe utilizatori să găsească toate informațiile necesare despre cărți, autori, categorii sau secțiunile din bibliotecă în care se găsesc cărțile.

Conform modelului din noua bază de date, utilizatorii bibliotecii pot fi identificați cu un ID unic, iar cu ajutorul acestuia pot împrumuta o carte, sau pot face rezervare în cazul în care niciun exemplar al cărții dorite nu este disponibil în bibliotecă.

Mai mult de atât, din responsabilitate pentru cărțile lor, administratorii bibliotecii au hotărât să impună o taxă de întârziere pentru cititorii care depășesc data de restituire a cărților. Mai exact pentru fiecare zi de întârziere, utilizatorii vor fi taxați cu câte 1leu, termenul de restituire fiind același pentru toți cititorii, și anume 30 de zile de la momentul împrumutului.

## 2 Constrângeri

- O carte poate fi scrisă de unul sau mai mulți autori.
- Un autor poate scrie una sau mai multe cărți.
- O carte poate fi publicată de o singură editură.
- O editură poate publica una sau mai multe cărți.
- O carte poate avea unul sau mai multe exemplare.
- Un exemplar aparține unei singure cărți.
- Un exemplar al unei cărți trebuie să se regăsească într-o singură categorie.
- O categorie poate să cuprindă mai multe exemplare de cărți sau niciunul.
- O categorie de cărți se regăsește într-o singură secțiune a bibliotecii.
- O secțiune a bibliotecii poate conține mai multe categorii de cărți.
- Pentru fiecare secțiune există un singur bibliotecar responsabil.
- Un bibliotecar poate fi responsabil de una sau mai multe secțiuni.
- Un cititor poate împrumuta mai multe exemplare ale unei cărți sau niciunul.
- Un exemplar poate fi împrumutat de mai mulți cititori sau niciunul.

- Un cititor poate rezerva pentru a împrumuta mai târziu mai multe cărți sau niciuna.
- O carte poate fi rezervată de mai mulți cititori sau de niciunul.
- Cititorilor li se poate impune o taxă de întârziere dacă nu returnează cărțile împrumutate în termen de 30 de zile, iar taxa se calculează în funcție de numărul de zile depășit pentru fiecare exemplar al cărților împrumutate.

### 3 Entități și chei primare

ENTITATE	DESCRIERE	CHEIE PRIMARĂ
CARTE	entitate ce reține informații despre cărțile care se află în bibliotecă	#id_carte
AUTOR	entitate ce reține date despre persoanele care au publicat cel puțin o carte dintre cele care se găsesc în bibliotecă	#id_autor
EDITURĂ	entitate care reține informații despre editurile care au publicat cărți care se găsesc în incinta bibliotecii	#id_editură
EXEMPLAR	entitate care reține informații despre fiecare copie a cărților pe care le deține biblioteca	#id_exemplar
CATEGORIE	entitate care reține date despre domeniile după care sunt clasificate cărțile din bibliotecă	#id_categorie
SECȚIUNE	entitate care reține informații despre locul exact în care se află categoriile cărților din bibliotecă	#id_secțiune
BIBLIOTECAR	entitate în care se rețin informații despre persoanele care se ocupă de secțiunile bibliotecii	#id_bibliotecar
CITITOR	entitate în care se stochează date despre fiecare cititor al bibliotecii care împrumută/rezervă cărți	#id_cititor
TAXĂ	entitate în care se rețin informații despre taxele plătite de cititorii care au depășit termenele de restituire ale exemplarelor împrumutate	#id_taxă

## 4 Relații și cardinalități

RELATIE	DESCRIERE	CARDINALITATE MINIMĂ	CARDINALITATE MAXIMĂ
<b>AUTOR</b> _scrie_ <b>CARTE</b>	relație care leagă entitățile AUTOR și CARTE	1:1 (un autor scrie cel puțin o carte, iar o carte trebuie scrisă de cel puțin un autor)	M:N (un autor poate scrie mai multe cărți, iar o carte poate fi scrisă de mai mulți autori)
<b>EDITURA</b> _publică_ <b>CARTE</b>	relație care leagă entitățile EDITURĂ și CARTE	1:1 (o editură publică cel puțin o carte, iar o carte este publicată de o singură editură)	1:N (o editură poate să publice mai multe cărți, dar o carte trebuie să fie publicată de o singură editură)
<b>CARTE</b> _are_ <b>EXEMPLAR</b>	relație care leagă entitățile CARTE și EXEMPLAR	1:1 (o carte are cel puțin un exemplar, iar un exemplar este al unei singure cărți)	1:N (o carte poate avea mai multe exemplare, dar un exemplar este al unei singure cărți)
<b>CATEGORIE</b> _aparțin_ <b>EXEMPLARE</b>	relație care leagă entitățile CATEGORIE și EXEMPLAR	1:0 (unei categorii poate să nu îi aparțină niciun exemplar, un exemplar aparține unei singure categorii)	1:N (unei categorii pot să îi aparțină mai multe exemplare, dar un exemplar aparține unei singure categorii)
<b>SECȚIUNE</b> _conține_ <b>CATEGORIE</b>	relație care leagă entitățile SECȚIUNE și CATEGORIE	1:1 (o secțiune conține cel puțin o categorie de cărți, iar o categorie se regăsește într-o singură secțiune)	1:N (o secțiune poate conține mai multe categorii de cărți, iar o categorie se regăsește într-o singură secțiune)
<b>BIBLIOTECAR</b> _se ocupă de_ <b>SECȚIUNE</b>	relație care leagă entitățile BIBLIOTECAR și SECȚIUNE	1:1 (un bibliotecar se ocupă de cel puțin o secțiune, iar de o secțiune se ocupă un singur bibliotecar)	1:N (un bibliotecar se poate ocupa de mai multe secțiuni, dar de o secțiune se ocupă un singur bibliotecar)
<b>CITITOR</b> _rezervă_ <b>CARTE</b>	relație care leagă entitățile CITITOR și CARTE	1:0 (un cititor poate să nu rezerve nicio carte, iar o carte poate să nu aibă nicio rezervare)	M:N (un cititor poate să rezerve mai multe cărți, iar o carte poate să fie rezervată de mai mulți cititori)
<b>CITITOR</b> _plătește_ <b>TAXĂ</b> _pentru_ <b>EXEMPLAR</b>	relație de tip 3 care leagă entitățile CITITOR, EXEMPLAR, TAXĂ	1:0:0 (un cititor poate să nu împrumute niciun exemplar, și să nu plătească nicio taxă, dar o taxă aparține unui singur cititor)	M:M:M (un cititor poate împrumuta mai multe exemplare, iar un exemplar poate fi împrumutat de mai mulți cititori)

## 5 Atribute

- Entitatea **CARTE** are ca atribute:

- *id\_carte* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei cărți.
- *titlu\_carte* = variabilă de tip șir de caractere, de lungime maximă 40, care reprezintă numele unei cărți, și care nu poate fi nulă.
- *id\_editură* = variabilă de tip întreg, care reprezintă id-ul editurii care a publicat cartea, și care nu poate fi nulă. Atributul trebuie să corespundă unei valori a cheii primare din tabelul EDITURĂ.
- *an\_publicare* = variabilă de tip întreg, care reprezintă anul în care a fost publicată cartea.
- *număr\_pagini* = variabilă de tip întreg, care reprezintă numărul de pagini al cărții.

- Entitatea **AUTOR** are ca atribute:

- *id\_autor* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui autor.
- *nume* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele unui autor, și care nu poate fi nulă.
- *prenume* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă prenumele unui autor, și care nu poate fi nulă.
- *data\_naștere* = variabilă de tip dată calendaristică, care reprezintă data nașterii autorului.

- Entitatea **EDITURĂ** are ca atribute:

- *id\_editură* = variabilă de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei edituri.
- *nume\_editură* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele unei edituri, și care nu poate fi nulă.
- *email* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă adresa de email a editurii, și care nu poate fi nulă.
- *nr\_telefon* = variabilă de tip șir de caractere, de lungime 10, care reprezintă numărul de telefon prin care se poate contacta editura.

- Entitatea **EXEMPLAR** are ca attribute:

- *id\_exemplar* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul exemplarului.
- *id\_carte* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul cărții căreia aparține exemplarul. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CARTE.
- *id\_categorie* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul categoriei căreia îi aparține exemplarul. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CATEGORIE.
- *status* = variabilă de tip șir de caractere, de lungime 10, care reprezintă statusul exemplarului (posibile valori: disponibil, împrumutat).

- Entitatea **CATEGORIE** are ca attribute:

- *id\_categorie* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul categoriei.
- *nume\_categorie* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele unei categorii, și care nu poate fi nulă.
- *id\_sectiune* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul secțiunii din bibliotecă în care se găsește categoria. Atributul trebuie să corespundă unei valori a cheii primare din tabelul SECȚIUNE.

- Entitatea **SECȚIUNE** are ca attribute:

- *id\_sectiune* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei secțiuni a bibliotecii.
- *nume\_sectiune* = variabilă de tip șir de caractere, de lungime maximă 10, care reprezintă numele secțiunii, și care nu poate fi nulă.
- *nr\_raft* = variabilă de tip întreg, care nu poate fi nulă și care reprezintă numărul raftului în care se găsește o anumită secțiune.
- *id\_bibliotecar* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul bibliotecarului care se ocupă de acea secțiune. Atributul trebuie să corespundă unei valori a cheii primare din tabelul BIBLIOTECAR.

- Entitatea **BIBLIOTECAR** are ca attribute:

- *id\_bibliotecar* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui bibliotecar.
- *nume* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele unui bibliotecar, și care nu poate fi nulă.
- *prenume* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă prenumele unui bibliotecar, și care nu poate fi nulă.



- *email* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă adresa de email a bibliotecarului, și care nu poate fi nulă.
- *nr\_telefon* = variabilă de tip șir de caractere, de lungime 10, care reprezintă numărul de telefon al bibliotecarului.
- *salariu* = variabilă de tip întreg care reprezintă salariul bibliotecarului.
- *data\_angajării* = variabilă de tip dată calendaristică, care reprezintă data la care a fost angajat bibliotecarul.

• Entitatea **CITITOR** are ca atribute:

- *id\_cititor* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui cititor.
- *nume* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele unui cititor, și care nu poate fi nulă.
- *prenume* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă prenumele unui cititor, și care nu poate fi nulă.
- *email* = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă adresa de email a cititorului, și care nu poate fi nulă.
- *nr\_telefon* = variabilă de tip șir de caractere, de lungime 10, care reprezintă numărul de telefon al cititorului.
- *data\_înscrisoare* = variabilă de tip dată calendaristică, care reprezintă data la care a fost înscris cititorul pe platforma bibliotecii.

• Entitatea **TAXĂ** are ca atribute:

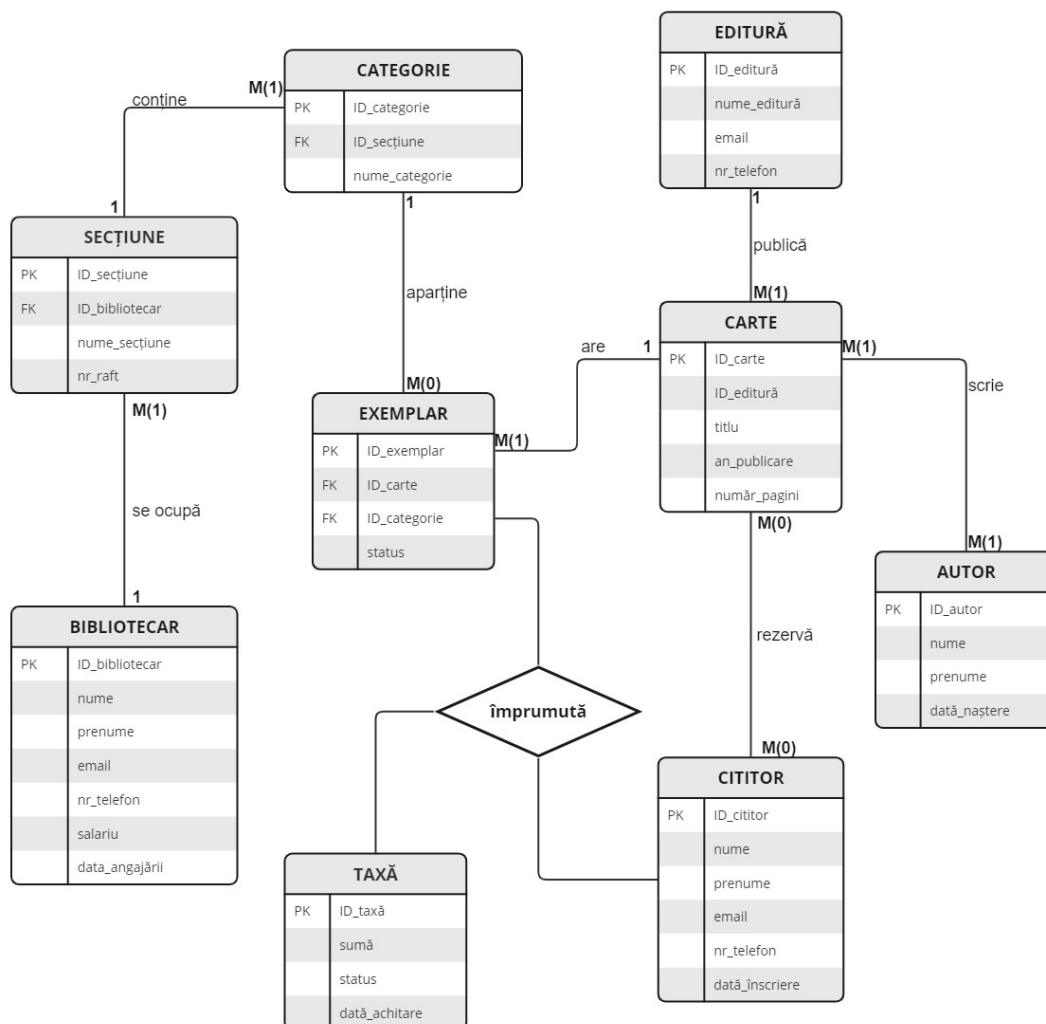
- *id\_taxă* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei taxe.
- *sumă* = variabilă de tip întreg care reprezintă suma ce trebuie achitată de cititor.
- *status* = variabilă de tip șir de caractere, de lungime 10, care reprezintă statusul taxei (posibile valori: achitată, neachitată).
- *data\_achitare* = variabilă de tip dată calendaristică, care reprezintă data la care a fost achitată taxa.

• Relația **AUTOR\_scrie\_CARTE** numită **SCRIERE** are ca atribute:

- *id\_carte* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei cărți. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CARTE.
- *id\_autor* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui autor. Atributul trebuie să corespundă unei valori a cheii primare din tabelul AUTOR.

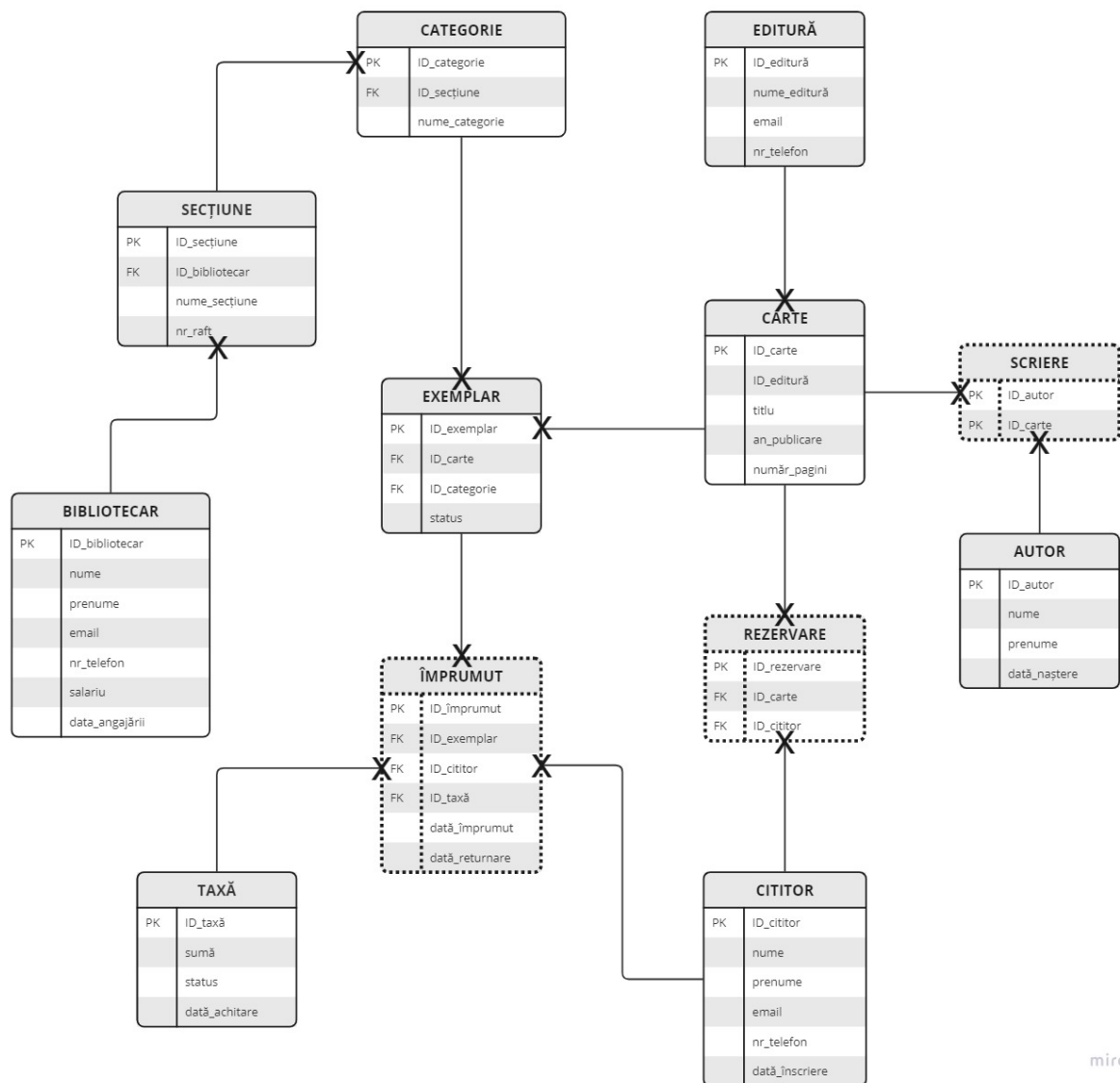
- Relația **CITITOR\_rezervă\_CARTE** numită **REZERVARE** are ca attribute:
  - *id\_rezervare* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei rezervări.
  - *id\_carte* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei cărți. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CARTE.
  - *id\_cititor* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui cititor. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CITITOR.
  
- Relația **CITITOR\_plătește\_TAXĂ\_pentru\_EXEMPLAR** numită **ÎMPRUMUT** are ca attribute:
  - *id\_împrumut* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui împrumut.
  - *id\_exemplar* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui exemplar. Atributul trebuie să corespundă unei valori a cheii primare din tabelul EXEMPLAR.
  - *id\_cititor* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unui cititor. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CITITOR.
  - *id\_taxă* = variabila de tip întreg, care nu poate fi nulă, și care reprezintă id-ul unei taxe. Atributul trebuie să corespundă unei valori a cheii primare din tabelul TAXĂ.
  - *data\_împrumut* = variabilă de tip dată calendaristică, care reprezintă data la care a fost împrumutat un anumit exemplar.
  - *data\_returnare* = variabilă de tip dată calendaristică, care reprezintă data la care a fost returnat un anumit exemplar.

## 6 Diagrama Entitate-Relație



miro

## 7 Diagrama Conceptuală



miro

## 8 Scheme Relaționale

- CARTE (id\_carte#, id\_editură, titlu, an\_publicare, număr\_pagini)
- AUTOR (id\_autor#, nume, prenume, dată\_nastere)
- EDITURĂ (id\_editură#, nume\_editură, email, nr\_telefon)
- EXEMPLAR (id\_exemplar#, id\_carte, id\_categorie, status)
- CATEGORIE (id\_categorie#, id\_secțiune, nume\_categorie)
- SECȚIUNE (id\_secțiune#, id\_bibliotecar, nume\_secțiune, nr\_raft)
- BIBLIOTECAR (id\_bibliotecar#, nume, prenume, email, nr\_telefon, dată\_angajare)
- CITITOR (id\_cititor#, nume, prenume, email, nr\_telefon, dată înscriere)
- TAXĂ (id\_taxă#, sumă, status, dată\_achitare)
- SCRIERE (id\_autor#, id\_carte#)
- REZERVARE (id\_rezervare#, id\_carte, id\_cititor)
- ÎMPRUMUT (id împrumut#, id\_exemplar, id\_cititor, id\_taxă, dată împrumut, dată\_returnare)

## 9 Normalizare

### 9.1 Prima formă normală - FN1

O entitate se găsește în prima formă normală dacă și numai dacă:

- \* nu există attribute cu valori multiple
- \* nu există attribute sau grupuri de attribute care se repetă

Se poate observa cu ușurință că modelul implementat respectă condițiile primei forme normale întrucât toate înregistrările din tabel sunt identificate în mod unic prin intermediul cheii primare (fiecare entitate primește un ID unic). De asemenea, în baza de date implementată nu există grupuri repetitive deoarece fiecărui atribut îi corespunde o valoare unică indivizibilă. Pentru a exemplifica normalizarea, se consideră următorul exemplu fictiv non-FN1:

ID_SECTIUNE	NUME_CATEGORIE
A	LITERATURĂ_CLASICĂ, LITERATURĂ_CONTEMPORANĂ, LITERATURĂ_VECHE
B	FIZICĂ_MECANICĂ, FIZICĂ_OPTICĂ, FIZICĂ_CUANTICĂ
C	DEZVOLTARE_WEB, PROGRAMARE_COMPETITIVĂ, BAZE_DE_DATA

Astfel se poate observa că relația de mai sus nu este în prima formă normală deoarece atributului nume\_categorie nu îi corespunde o valoare indivizibilă. În urma normalizării relația va arăta așa:

ID_CATEGORIE	NUME_CATEGORIE	ID_SECTIUNE
100	LITERATURĂ_CLASICĂ	A
101	LITERATURĂ_CONTEMPORANĂ	A
102	LITERATURĂ_VECHE	A
103	FIZICĂ_MECANICĂ	B
104	FIZICĂ_CUANTICĂ	B
105	FIZICĂ_OPTICĂ	B
106	DEZVOLTARE_WEB	C
107	PROGRAMARE_COMPETITIVĂ	C
108	BAZE_DE_DATA	C

Se poate observa că după normalizare fiecărui atribut îi corespunde o valoare indivizibilă.

## 9.2 A doua formă normală - FN2

O entitate se găsește în a doua formă normală dacă și numai dacă:

- \* se găsește în prima formă normală
- \* orice atribut care nu face parte din UID (Unique Identifier) va depinde de întregul UID nu doar de o parte a acestuia

Observăm că modelul implementat respectă și cerințele celei de-a doua formă normală: toate relațiile respectă FN1, iar fiecare atribut care nu e UID este dependent de întreaga cheie primară. Pentru a exemplifica normalizarea, se consideră următorul tabel fictiv CARTE drept exemplu non-FN2:

ID_CARTE#	ID_AUTOR#	TITLU_CARTE	DATA_NAȘTERE	DATA_PUBLICARE
101	1001	Culegere probleme geometrie analitică	1970	2008
102	1002	Proiectarea bazelor de date	1985	2013
103	1003	Manual Fizică Mecanică	1974	2003

Pentru a fi în FN2 trebuie ca orice atribut care nu este cheie primară să fie dependent de întreaga cheie primară. În acest caz, attributele titlu\_carte, data\_naștere și dată\_publicare trebuie să depindă de întreaga cheie primară (id\_carte, id\_autor), dar se poate observa că acest lucru nu se întâmplă întrucât dată\_naștere depinde doar de id\_autor, iar dată\_publicare și titlu\_carte depind doar de id\_carte. În urma normalizării relația va arăta așa:

### TABELUL CARTE:

ID_CARTE	TITLU_CARTE	DATA_PUBLICARE
101	Culegere probleme geometrie analitică	2008
102	Proiectarea bazelor de date	2013
103	Manual Fizică Mecanică	2003

### TABELUL AUTOR:

ID_AUTOR	NUME	DATA_NAȘTERE
1001	Popescu	1970
1002	Ionescu	1985
1003	Alexandrescu	1974

### 9.3 A treia formă normală - FN3

O entitate se găsește în a treia formă normală dacă și numai dacă:

- \* se găsește în a doua formă normală
- \* niciun atribut care nu este parte a UID-ului nu depinde de un alt atribut non-UID

Pentru a exemplifica normalizarea, se consideră următorul tabel fictiv CARTE drept exemplu non-FN3:

ID_CARTE	TITLU	NUME_EDITURĂ	ADRESĂ_EDITURĂ
101	Culegere probleme geometrie analitică	Litera	Iași, str 1 Decembrie 1918, nr 99
102	Proiectarea bazelor de date	Corint	București, sector 3
103	Manual Fizică Mecanică	Aramis	Alexandria

Această relație este în FN1 pentru că există identificator unic pentru toate intrările în tabel (id\_carte) și pentru că valorile atributelor sunt atomice. De asemenea, relația este și în FN2 deoarece toate atributele care nu sunt parte din cheia primară sunt dependente de întreaga cheie primară. Cu toate acestea relația nu este în FN3 din cauza relațiilor de dependență întâlnite:

- {id\_carte} - {titlu, nume\_editură, adresă\_editură}
- {id\_carte} - {nume\_editură} - {adresă\_editură}

Pentru a aduce relația în FN3 se aplică regula Casey-Delobel care presupune descompunerea relației prin eliminarea dependenței funcționale tranzitive. În urma normalizării, relația va arăta astfel:

**TABELUL CARTE:**

ID_CARTE	TITLU	ID_EDITURĂ
101	Culegere probleme geometrie analitică	1
102	Proiectarea bazelor de date	2
103	Manual Fizică Mecanică	3

**TABELUL EDITURĂ:**

ID_EDITURĂ	NUME_EDITURĂ	ADRESĂ
1	Litera	Iași, str 1 Decembrie 1918, nr 99
2	Corint	București, sector 3
3	Aramis	Alexandria



## 10 Crearea secvențelor

```
CREATE SEQUENCE seq_autor
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2000
  NOCACHE
  NOCYCLE;
```

```
CREATE SEQUENCE seq_carte
  START WITH 100
  INCREMENT BY 1
  MAXVALUE 3000
  NOCYCLE;
```

```
CREATE SEQUENCE seq_cititor
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 9000
  NOCACHE
  NOCYCLE;
```

```
CREATE SEQUENCE seq_bibliotecar_id
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 100
  NOCYCLE;
```

```
CREATE SEQUENCE seq_rezervare
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 10000
  NOCACHE
  NOCYCLE;
```

```
CREATE SEQUENCE seq_taxa
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 10000
  NOCACHE
  NOCYCLE;
```

```
CREATE SEQUENCE seq_exemplar
  START WITH 1000
  INCREMENT BY 1
  MAXVALUE 9000
  NOCACHE
  NOCYCLE;
```

```

CREATE SEQUENCE seq_imprumut
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 10000
  NOCACHE
  NOCYCLE;

```

## 11 Crearea tabelelor

### 11.1 Creare și inserare în tabelul *AUTOR*

```

CREATE TABLE AUTOR (
  id_autor NUMBER(6) CONSTRAINT pk_autor PRIMARY KEY,
  nume VARCHAR2(25) NOT NULL,
  prenume VARCHAR2(25) NOT NULL,
  data_nastere DATE
);

```

```

CREATE SEQUENCE seq_autor
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2000
  NOCACHE
  NOCYCLE;

```

```

INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Feynman', 'Richard P.', TO_DATE('11-05-1918',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Watson', 'James D.', TO_DATE('06-04-1928',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Hawking', 'Stephen', TO_DATE('08-01-1942',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Darwin', 'Charles', TO_DATE('12-02-1809',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Kuhn', 'Thomas S.', TO_DATE('18-07-1922',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Austen', 'Jane', TO_DATE('16-12-1775',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Shakespeare', 'William', TO_DATE('26-04-1564',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)

```

```

VALUES (seq_autor.NEXTVAL, 'Goethe', 'Johann Wolfgang', TO_DATE('28-08-1749',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Eminescu', 'Mihai', TO_DATE('15-01-1850',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Hugo', 'Victor', TO_DATE('26-02-1802',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Dostoevsky', 'Fyodor', TO_DATE('11-11-1821',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume)
VALUES (seq_autor.NEXTVAL, 'Popescu', 'Radu');
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Ionescu', 'Maria', TO_DATE('02-02-1980',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume)
VALUES (seq_autor.NEXTVAL, 'Radu', 'Andrei');
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Dumitrescu', 'Ana', TO_DATE('04-04-1970',
'DD-MM-YYYY'));
INSERT INTO AUTOR (id_autor, nume, prenume)
VALUES (seq_autor.NEXTVAL, 'Georgescu', 'Mihai');
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere)
VALUES (seq_autor.NEXTVAL, 'Constantin', 'Elena', TO_DATE('06-06-1995',
'DD-MM-YYYY'));

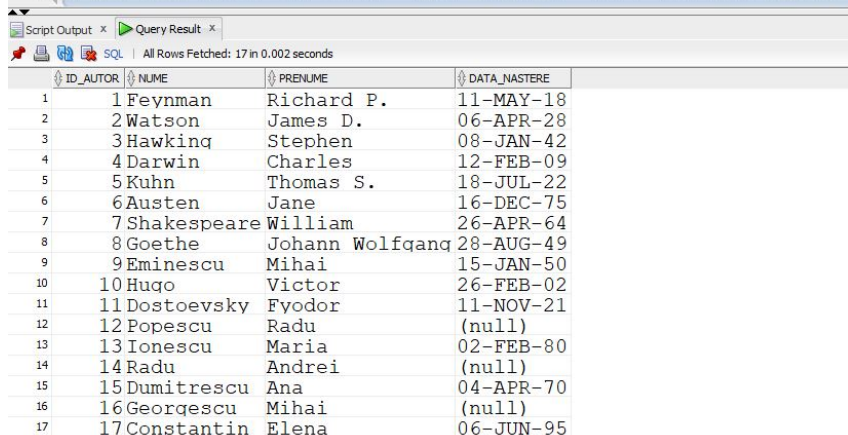
```

```
--INSERARE IN AUTOR
```

```

INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES
INSERT INTO AUTOR (id_autor, nume, prenume, data_nastere) VALUES

```

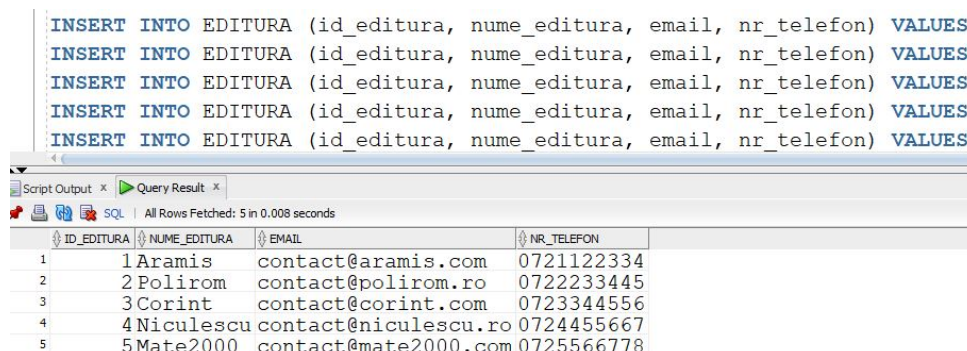


ID_AUTOR	NUME	PRENUME	DATA_NASTERE
1	Feynman	Richard P.	11-MAY-18
2	Watson	James D.	06-APR-28
3	Hawking	Stephen	08-JAN-42
4	Darwin	Charles	12-FEB-09
5	Kuhn	Thomas S.	18-JUL-22
6	Austen	Jane	16-DEC-75
7	Shakespeare	William	26-APR-64
8	Goethe	Johann Wolfgang	28-AUG-49
9	Eminescu	Mihai	15-JAN-50
10	Hugo	Victor	26-FEB-02
11	Dostoevsky	Fyodor	11-NOV-21
12	Popescu	Radu	(null)
13	Ionescu	Maria	02-FEB-80
14	Radu	Andrei	(null)
15	Dumitrescu	Ana	04-APR-70
16	Georgescu	Mihai	(null)
17	Constantin	Elena	06-JUN-95

## 11.2 Creare și inserare în tabelul *EDITURĂ*

```
CREATE TABLE EDITURA (  
    id_editura NUMBER(6) CONSTRAINT pk_editura PRIMARY KEY,  
    nume_editura VARCHAR2(25) NOT NULL,  
    email VARCHAR2(25) NOT NULL,  
    nr_telefon VARCHAR2(10)  
);
```

```
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
(1, 'Aramis', 'contact@aramis.com', '0721122334');  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
(2, 'Polirom', 'contact@polirom.ro', '0722233445');  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
(3, 'Corint', 'contact@corint.com', '0723344556');  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
(4, 'Niculescu', 'contact@niculescu.ro', '0724455667');  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
(5, 'Mate2000', 'contact@mate2000.com', '0725566778');
```



```
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES  
INSERT INTO EDITURA (id_editura, nume_editura, email, nr_telefon) VALUES
```

ID_EDITURA	NUME_EDITURA	EMAIL	NR_TELEFON
1	1 Aramis	contact@aramis.com	0721122334
2	2 Polirom	contact@polirom.ro	0722233445
3	3 Corint	contact@corint.com	0723344556
4	4 Niculescu	contact@niculescu.ro	0724455667
5	5 Mate2000	contact@mate2000.com	0725566778

## 11.3 Creare și inserare în tabelul *CARTE*

```
CREATE TABLE CARTE (  
    id_carte NUMBER(6) CONSTRAINT pk_carte PRIMARY KEY,  
    id_editura NUMBER(6) NOT NULL,  
    titlu VARCHAR2(40) NOT NULL,  
    an_publicare NUMBER(4),  
    numar_pagini NUMBER(4),  
    CONSTRAINT fk_carte_editura FOREIGN KEY (id_editura)  
    REFERENCES EDITURA(id_editura)  
);
```

```
CREATE SEQUENCE seq_carte  
START WITH 100  
INCREMENT BY 1
```

MAXVALUE 3000  
NOCYCLE;

```
INSERT INTO CARTE (id_carte, id_editura, titlu, numar_pagini)
VALUES (seq_carte.NEXTVAL, 1, 'Introducere in analiza matematica', 350);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Analiza - ecuatii differentiale', 2010, 450);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare)
VALUES (seq_carte.NEXTVAL, 1, 'Geometrie euclidiană', 2009);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Introducere in geometria differentiala', 2011,
450);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 5, 'Geometrie computatională', 2007, 400);
INSERT INTO CARTE (id_carte, id_editura, titlu, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Teoria grupurilor si aplicatii', 400);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare)
VALUES (seq_carte.NEXTVAL, 1, 'Mecanica: Fundamente si aplicatii', 2005);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Introducere in optica geometrica', 2010, 250);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 3, 'Astrofizica: Galaxii si stele', 2018, 400);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 4, 'Mecanica cuantica si statistica', 2008, 280);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 5, 'Originea si evolutia Universului', 2015, 350);
INSERT INTO CARTE (id_carte, id_editura, titlu, numar_pagini)
VALUES (seq_carte.NEXTVAL, 1, 'Baze de date relationale', 400);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Dezvoltare software: tehnici moderne', 2016,
350);
INSERT INTO CARTE (id_carte, id_editura, titlu)
VALUES (seq_carte.NEXTVAL, 3, 'Aplicatii web dinamice');
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 4, 'Algoritmi si structuri de date', 2010, 450);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 5, 'Introducere in bazele de date', 2018, 280);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 1, 'Programare orientata pe obiecte', 2015, 400);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Dezvoltare web cu HTML si CSS', 2013, 320);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 3, 'Algoritmi si structuri de date avansate',
2017, 500);
INSERT INTO CARTE (id_carte, id_editura, titlu, numar_pagini)
VALUES (seq_carte.NEXTVAL, 4, 'Testare si calitatea software', 350);
```

```

INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 1, 'Mândrie si prejudecata', 1813, 320);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'Romeo si Julieta', 1597, 256);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 3, 'Faust', 1808, 480);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 4, 'Luceafarul', 1883, 160);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 1, 'Mizerabilii', 1862, 1488);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 3, 'Crima si pedeapsa', 1866, 672);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 1, 'The Feynman Lectures on Physics', 1963, 1552);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 3, 'A Brief History of Time', 1988, 256);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 4, 'The Origin of Species', 1859, 502);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 5, 'The Structure of Scientific Revolutions',
1962, 264);
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)
VALUES (seq_carte.NEXTVAL, 2, 'The Double Helix', 1968, 226);

```

```

INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini) VALUES
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini) VALUES
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare) VALUES
INSERT INTO CARTE (id_carte, id_editura, titlu, an_publicare, numar_pagini)

```

ID_CARTE	ID_EDITURA	TITLU	AN_PUBLICARE	NUMAR_PAGINI
1	100	1Introducere in analiza matematica	(null)	350
2	101	2Analiza - ecuatii differentiale	2010	450
3	102	1Geometrie euclidiană	2009	(null)
4	103	2Introducere in geometria diferentiaala	2011	450
5	104	5Geometrie computatională	2007	400
6	105	2Teoria grupurilor si aplicatii	(null)	400
7	106	1Mecanica: Fundamente si aplicatii	2005	(null)
8	107	2Introducere in optica geometrică	2010	250
9	108	3Astrofizica: Galaxii si stele	2018	400
10	109	4Mecanica cuantica si statistica	2008	280
11	110	5Originea si evolutia Universului	2015	350
12	111	1Baze de date relationale	(null)	400
13	112	2Dezvoltare software:tehnici moderne	2016	350
14	113	3Aplicatii web dinamice	(null)	(null)
15	114	4Algoritmi si structuri de date	2010	450
16	115	5Introducere in bazele de date	2018	280
17	116	1Programare orientata pe obiecte	2015	400
18	117	2Dezvoltare web cu HTML si CSS	2013	320
19	118	3Algoritmi si structuri de date avansate	2017	500
20	119	4Testare si calitatea software	(null)	350
21	120	1Mândrie si prejudecata	1813	320
22	121	2Romeo si Julieta	1597	256
23	122	3Faust	1808	480
24	123	4Luceafarul	1883	160
25	124	1Mizerabilii	1862	1488
26	125	3Crima si pedeapsa	1866	672
27	126	1The Feynman Lectures on Physics	1963	1552
28	127	3A Brief History of Time	1988	256
29	128	4The Origin of Species	1859	502
30	129	5The Structure of Scientific Revolutions	1962	264
31	130	2The Double Helix	1968	226

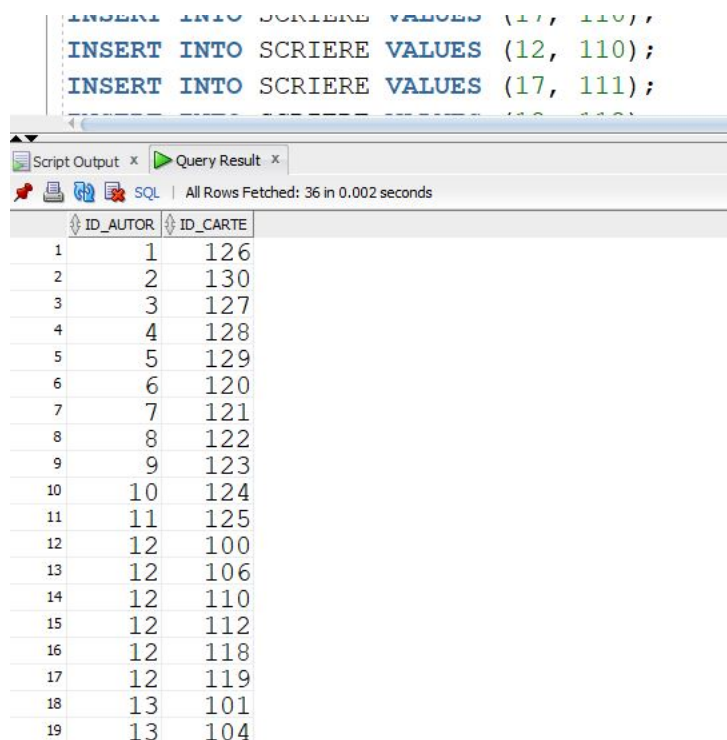
## 11.4 Creare și inserare în tabelul *SCRIERE*

```
CREATE TABLE SCRIERE (  
    id_autor NUMBER(6) NOT NULL,  
    id_carte NUMBER(6) NOT NULL,  
    CONSTRAINT fk_autor FOREIGN KEY (id_autor)  
    REFERENCES AUTOR(id_autor),  
    CONSTRAINT fk_carte FOREIGN KEY (id_carte)  
    REFERENCES CARTE(id_carte),  
    CONSTRAINT pk_autor_scrie_carte  
    PRIMARY KEY(id_autor, id_carte)  
);  
  
INSERT INTO SCRIERE VALUES (12, 100);  
INSERT INTO SCRIERE VALUES (13, 101);  
INSERT INTO SCRIERE VALUES (14, 102);  
INSERT INTO SCRIERE VALUES (15, 103);  
INSERT INTO SCRIERE VALUES (16, 104);  
INSERT INTO SCRIERE VALUES (13, 104);  
INSERT INTO SCRIERE VALUES (17, 105);  
INSERT INTO SCRIERE VALUES (12, 106);  
INSERT INTO SCRIERE VALUES (13, 107);  
INSERT INTO SCRIERE VALUES (14, 108);  
INSERT INTO SCRIERE VALUES (15, 109);  
INSERT INTO SCRIERE VALUES (16, 110);  
INSERT INTO SCRIERE VALUES (17, 110);  
INSERT INTO SCRIERE VALUES (12, 110);  
INSERT INTO SCRIERE VALUES (17, 111);  
INSERT INTO SCRIERE VALUES (12, 112);  
INSERT INTO SCRIERE VALUES (13, 113);  
INSERT INTO SCRIERE VALUES (14, 114);  
INSERT INTO SCRIERE VALUES (15, 115);  
INSERT INTO SCRIERE VALUES (17, 115);  
INSERT INTO SCRIERE VALUES (16, 116);  
INSERT INTO SCRIERE VALUES (17, 117);  
INSERT INTO SCRIERE VALUES (12, 118);  
INSERT INTO SCRIERE VALUES (13, 119);  
INSERT INTO SCRIERE VALUES (12, 119);  
INSERT INTO SCRIERE (id_autor, id_carte)  
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Austen'  
AND prenume = 'Jane'), 120);  
INSERT INTO SCRIERE (id_autor, id_carte)  
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Shakespeare'  
AND prenume = 'William'), 121);  
INSERT INTO SCRIERE (id_autor, id_carte)  
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Goethe'  
AND prenume = 'Johann Wolfgang'), 122);  
INSERT INTO SCRIERE (id_autor, id_carte)
```

```

VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Eminescu'
AND prenume = 'Mihai'), 123);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Hugo'
AND prenume = 'Victor'), 124);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Dostoevsky'
AND prenume = 'Fyodor'), 125);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Feynman'
AND prenume = 'Richard P.'), 126);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Hawking'
AND prenume = 'Stephen'), 127);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Darwin'
AND prenume = 'Charles'), 128);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Kuhn'
AND prenume = 'Thomas S.'), 129);
INSERT INTO SCRIERE (id_autor, id_carte)
VALUES ((SELECT id_autor FROM AUTOR WHERE nume = 'Watson'
AND prenume = 'James D.'), 130);

```



```

INSERT INTO SCRIERE VALUES (11, 110);
INSERT INTO SCRIERE VALUES (12, 110);
INSERT INTO SCRIERE VALUES (17, 111);

```

	ID_AUTOR	ID_CARTE
1	1	126
2	2	130
3	3	127
4	4	128
5	5	129
6	6	120
7	7	121
8	8	122
9	9	123
10	10	124
11	11	125
12	12	100
13	12	106
14	12	110
15	12	112
16	12	118
17	12	119
18	13	101
19	13	104



## 11.5 Creare și inserare în tabelul *CITITOR*

```
CREATE TABLE CITITOR (  
    id_cititor NUMBER(6) CONSTRAINT pk_cititor PRIMARY KEY,  
    nume VARCHAR2(25) NOT NULL,  
    prenume VARCHAR2(25) NOT NULL,  
    email VARCHAR2(25) NOT NULL,  
    nr_telefon VARCHAR2(10) NOT NULL,  
    data_inscriere DATE  
);  
  
CREATE SEQUENCE seq_cititor  
    START WITH 1  
    INCREMENT BY 1  
    MAXVALUE 9000  
    NOCACHE  
    NOCYCLE;  
  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Popovici', 'Elena',  
'popovici.elena@gmail.com', '0721122334', TO_DATE('2010-03-12', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Iorgulescu', 'Andrei',  
'iorgu.andrei@yahoo.com', '0732233445', TO_DATE('2011-06-25', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Popescu', 'Maria',  
'popescu.maria@gmail.com', '0743344556', TO_DATE('2012-08-10', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Georgescu', 'Alexandru',  
'georgescu.alex@yahoo.com', '0754455667', TO_DATE('2013-02-18', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Radu', 'Cristina',  
'radu.cristina@gmail.com', '0765566778', TO_DATE('2014-11-05', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Petrescu', 'Ionut',  
'petrescu.ionut@yahoo.com', '0776677889', TO_DATE('2015-09-20', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Constantin', 'Andreea',  
'andreea.const@gmail.com', '0787788990', TO_DATE('2016-07-03', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Mihai', 'Adrian',  
'mihai.adrian@yahoo.com', '0798899001', TO_DATE('2017-04-15', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Popa', 'Alexandra',  
'popa.alexandra@gmail.com', '0700011122', TO_DATE('2018-12-28', 'YYYY-MM-DD'));  
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon,  
data_inscriere) VALUES (seq_cititor.NEXTVAL, 'Balan', 'Razvan',  
'balan.razvan@yahoo.com', '0711122233', TO_DATE('2019-10-30', 'YYYY-MM-DD'));
```

```

CREATE SEQUENCE seq_cititor
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 9000
  NOCACHE
  NOCYCLE;

INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon, data_
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon, data_
INSERT INTO CITITOR (id_cititor, nume, prenume, email, nr_telefon, data_

```

ID_CITITOR	NUME	PRENUME	EMAIL	NR_TELEFON	DATA_INSCRIERE
1	1 Popovici	Elena	popovici.elena@gmail.com	0721122334	12-MAR-10
2	2 Iorquiescu	Andrei	iorqu.andrei@yahoo.com	0732233445	25-JUN-11
3	3 Popescu	Maria	popescu.maria@gmail.com	0743344556	10-AUG-12
4	4 Georgescu	Alexandru	georgescu.alex@yahoo.com	0754455667	18-FEB-13
5	5 Radu	Cristina	radu.cristina@gmail.com	0765566778	05-NOV-14
6	6 Petrescu	Ionut	petrescu.ionut@yahoo.com	0776677889	20-SEP-15
7	7 Constantin	Andreea	andreea.const@gmail.com	0787788990	03-JUL-16
8	8 Mihai	Adrian	mihai.adrian@yahoo.com	0798899001	15-APR-17
9	9 Popa	Alexandra	popa.alexandra@gmail.com	0700011122	28-DEC-18
10	10 Balan	Razvan	balan.razvan@yahoo.com	0711122233	30-OCT-19

## 11.6 Creare și inserare în tabelul *BIBLIOTECAR*

```

CREATE TABLE BIBLIOTECAR (
  id_biblioteca NUMBER(6) CONSTRAINT pk_biblioteca PRIMARY KEY,
  nume VARCHAR2(25) NOT NULL,
  prenume VARCHAR2(25) NOT NULL,
  email VARCHAR2(25) NOT NULL,
  nr_telefon VARCHAR2(10),
  salariu NUMBER NOT NULL,
  data_angajare DATE NOT NULL
);

```

```

CREATE SEQUENCE seq_biblioteca_id
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 100
  NOCYCLE;

```

```

INSERT INTO BIBLIOTECAR (id_biblioteca, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES(seq_biblioteca_id.NEXTVAL, 'Popescu', 'Ion',
'ion.popescu@yahoo.com', '0712345678', 2000, TO_DATE('01-01-1990',
'DD-MM-YYYY'));

```

```

INSERT INTO BIBLIOTECAR (id_biblioteca, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_biblioteca_id.NEXTVAL, 'Ionescu', 'Maria',
'maria.ionescu@gmail.com', '0723456789', 3500, TO_DATE('01-02-1995',
'DD-MM-YYYY'));

```

```

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Popa', 'Mihai',
'mihai.popa@gmail.com', '0734567890', 3200, TO_DATE('01-03-2000',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Georgescu',
'Ana', 'ana.georgescu@gmail.com', '0745678901', 3100, TO_DATE('01-04-2005',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Radu', 'Adrian',
'adrian.radu@yahoo.com', '0756789012', 3300, TO_DATE('01-05-2010',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Pop', 'Andreea',
'andreea.pop@yahoo.com', '0767890123', 3400, TO_DATE('01-06-2015',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Badea', 'Mircea',
'mircea.badea@gmail.com', '0778901234', 2100, TO_DATE('01-07-2020',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Diaconu',
'Elena', 'elena.diaconu@gmail.com', '0789012345', 3200, TO_DATE('01-08-2021',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Marinescu',
'George', 'george@yahoo.com', '0790123456', 3100, TO_DATE('01-09-2022',
'DD-MM-YYYY'));

INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon,
salariu, data_angajare) VALUES (seq_bibliotecar_id.NEXTVAL, 'Dumitru',
'Raluca', 'raluca.dumitru@yahoo.com', '0701234567', 1800,
TO_DATE('01-10-2023', 'DD-MM-YYYY'));

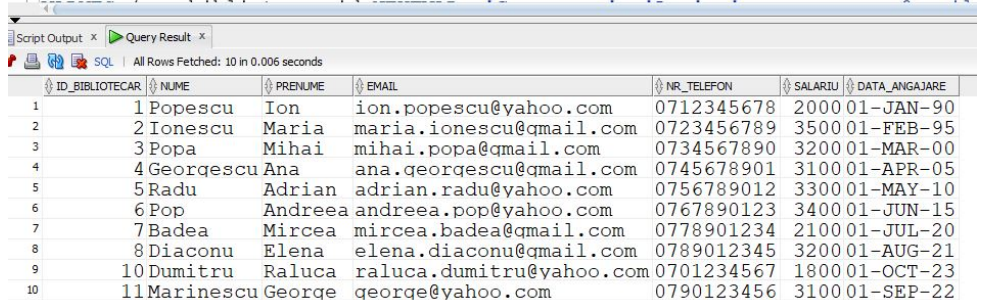
```

```

CREATE SEQUENCE seq_bibliotecar_id
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 100
  NOCYCLE;

--INSERARE IN BIBLIOTECAR
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s
VALUES (seq_bibliotecar_id.NEXTVAL, 'Popescu', 'Ion', 'ion.popescu@yahoo.com'
TO_DATE('01-01-1990', 'DD-MM-YYYY'));
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s
VALUES (seq_bibliotecar_id.NEXTVAL, 'Ionescu', 'Maria', 'maria.ionescu@gmail
TO_DATE('01-02-1995', 'DD-MM-YYYY'));
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s
VALUES (seq_bibliotecar_id.NEXTVAL, 'Popa', 'Mihai', 'mihai.popa@gmail.com',
TO_DATE('01-03-2000', 'DD-MM-YYYY'));
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s

```



ID_BIBLIOTECAR	NUME	PRENUME	EMAIL	NR_TELEFON	SALARIU	DATA_ANGAJARE
1	1 Popescu	Ion	ion.popescu@yahoo.com	0712345678	2000	01-JAN-90
2	2 Ionescu	Maria	maria.ionescu@gmail.com	0723456789	3500	01-FEB-95
3	3 Popa	Mihai	mihai.popa@gmail.com	0734567890	3200	01-MAR-00
4	4 Georgescu	Ana	ana.georgescu@gmail.com	0745678901	3100	01-APR-05
5	5 Radu	Adrian	adrian.radu@yahoo.com	0756789012	3300	01-MAY-10
6	6 Pop	Andreea	andreea.pop@yahoo.com	0767890123	3400	01-JUN-15
7	7 Badea	Mircea	mircea.badea@gmail.com	0778901234	2100	01-JUL-20
8	8 Diaconu	Elena	elena.diaconu@gmail.com	0789012345	3200	01-AUG-21
9	10 Dumitru	Raluca	raluca.dumitru@yahoo.com	0701234567	1800	01-OCT-23
10	11 Marinescu	George	george@yahoo.com	0790123456	3100	01-SEP-22

## 11.7 Creare și inserare în tabelul *SECTIUNE*

```

CREATE TABLE SECTIUNE (
  id_sectiune NUMBER(6) CONSTRAINT pk_sectiune PRIMARY KEY,
  id_bibliotecar NUMBER(6) NOT NULL,
  nume_sectiune VARCHAR2(10) NOT NULL,
  nr_raft NUMBER(3) NOT NULL,
  CONSTRAINT fk_sectiune_bibliotecar FOREIGN KEY (id_bibliotecar)
  REFERENCES BIBLIOTECAR(id_bibliotecar)
);

```

```

INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar, nume_sectiune, nr_raft)
VALUES (1, 1, 'Matematica', 1);
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar, nume_sectiune, nr_raft)
VALUES (2, 2, 'IT', 2);
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar, nume_sectiune, nr_raft)
VALUES (3, 3, 'Fizica', 3);
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar, nume_sectiune, nr_raft)
VALUES (4, 4, 'Literatura', 4);
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar, nume_sectiune, nr_raft)
VALUES (5, 5, 'Diverse', 5);

```

```
--7. TABELA SECTIUNE
--INSERARE IN SECTIUNE
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar,
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar,
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar,
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar,
INSERT INTO SECTIUNE (id_sectiune, id_bibliotecar,
SELECT * FROM SECTIUNE;
```

ID_SECTIUNE	ID_BIBLIOTECAR	NUME_SECTIUNE	NR_RAFT
1	1	1Matematica	1
2	3	3Fizica	3
3	4	4Literatura	4
4	5	5Diverse	5
5	2	2IT	2

## 11.8 Creare și inserare în tabelul *CATEGORIE*

```
CREATE TABLE CATEGORIE (
    id_categorie NUMBER(6) CONSTRAINT pk_categorie PRIMARY KEY,
    id_sectiune NUMBER(6) NOT NULL,
    nume_categorie VARCHAR2(25) NOT NULL,
    CONSTRAINT fk_categorie_sectiune FOREIGN KEY (id_sectiune)
    REFERENCES SECTIUNE(id_sectiune)
);
```

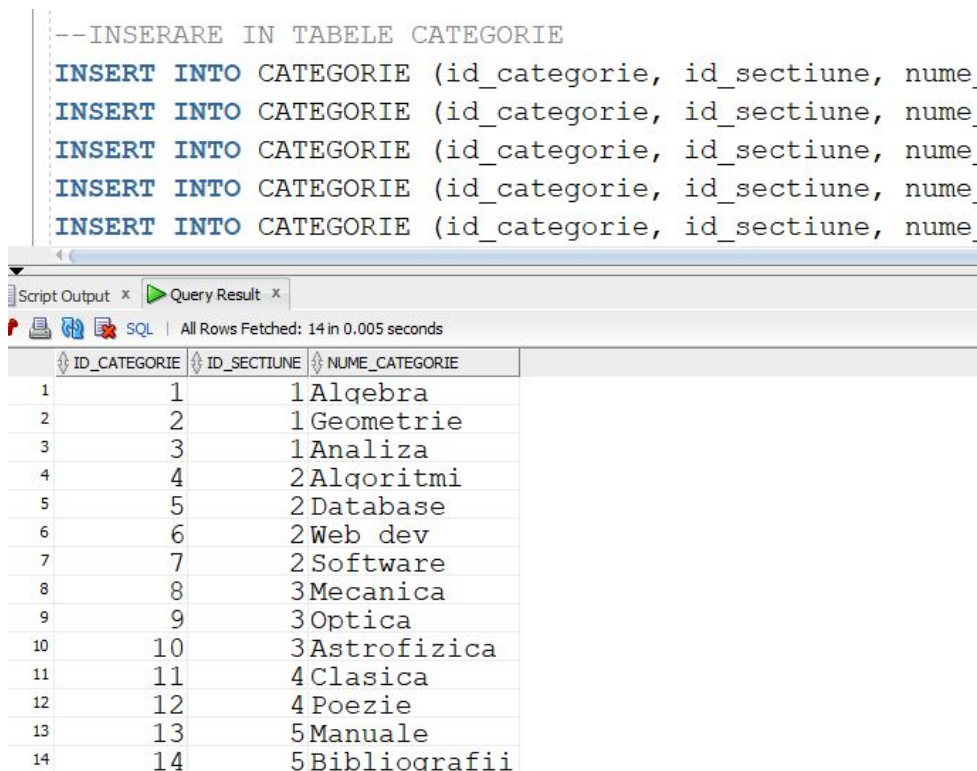
```
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (1, 1, 'Algebra');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (2, 1, 'Geometrie');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (3, 1, 'Analiza');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (4, 2, 'Algoritmi');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (5, 2, 'Database');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (6, 2, 'Web_dev');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (7, 2, 'Software');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (8, 3, 'Mecanica');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (9, 3, 'Optica');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
```

```

VALUES (10, 3, 'Astrofizica');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (11, 4, 'Clasica');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (12, 4, 'Poezie');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (13, 5, 'Manuale');
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
VALUES (14, 5, 'Bibliografii');

```

```
--INSERARE IN TABELE CATEGORIE
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
INSERT INTO CATEGORIE (id_categorie, id_sectiune, nume_categorie)
```



ID_CATEGORIE	ID_SECTIUNE	NUME_CATEGORIE
1	1	1Algebra
2	2	1Geometrie
3	3	1Analiza
4	4	2Algoritmi
5	5	2Database
6	6	2Web dev
7	7	2Software
8	8	3Mecanica
9	9	3Optica
10	10	3Astrofizica
11	11	4Clasica
12	12	4Poezie
13	13	5Manuale
14	14	5Bibliografii

## 11.9 Creare și inserare în tabelul *REZERVARE*

```

-- Creare tabela REZERVARE
CREATE TABLE REZERVARE (
    id_rezervare NUMBER(6) PRIMARY KEY,
    id_carte NUMBER(6),
    id_cititor NUMBER(6),
    CONSTRAINT fk_rezervare_carte FOREIGN KEY (id_carte)
    REFERENCES CARTE (id_carte),
    CONSTRAINT fk_rezervare_cititor FOREIGN KEY (id_cititor)
    REFERENCES CITITOR (id_cititor)
);

```



```

CREATE SEQUENCE seq_rezervare
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 10000
  NOCACHE
  NOCYCLE;

```

```

INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 100, 1);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 130, 2);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 127, 3);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 114, 4);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 115, 5);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 114, 6);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 114, 6);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 114, 4);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 123, 5);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 126, 6);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 113, 9);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 123, 9);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 115, 7);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 117, 1);
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor)
VALUES (seq_rezervare.NEXTVAL, 117, 3);

```

INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES
INSERT INTO REZERVARE (id_rezervare, id_carte, id_cititor) VALUES

ID_REZERVARE	ID_CARTE	ID_CITITOR
1	1	100
2	2	130
3	3	127
4	4	114
5	5	115
6	6	114
7	7	114
8	8	114
9	9	123
10	10	126
11	11	113
12	12	123
13	13	115
14	14	117
15	15	117

## 11.10 Creare și inserare în tabelul *TAXĂ*

```
CREATE TABLE TAXA (
    id_taxa NUMBER(6) PRIMARY KEY,
    suma NUMBER,
    status VARCHAR2(10),
    data_achitare DATE
);
```

```
CREATE SEQUENCE seq_taxa
START WITH 1
INCREMENT BY 1
MAXVALUE 10000
NOCACHE
NOCYCLE;
```

```
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
```



```

INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);

```

```

CREATE SEQUENCE seq_taxa
START WITH 1
INCREMENT BY 1
MAXVALUE 10000
NOCACHE
NOCYCLE;

--INSERARE IN TAXA
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);

```

Script Output x Query Result x

SQL | All Rows Fetched: 15 in 0.002 seconds

ID_TAXA	SUMA	STATUS	DATA_ACHITARE
1	(null)	(null)	(null)
2	(null)	(null)	(null)
3	(null)	(null)	(null)
4	(null)	(null)	(null)
5	(null)	(null)	(null)
6	(null)	(null)	(null)
7	(null)	(null)	(null)
8	(null)	(null)	(null)
9	(null)	(null)	(null)
10	(null)	(null)	(null)
11	(null)	(null)	(null)
12	(null)	(null)	(null)
13	(null)	(null)	(null)
14	(null)	(null)	(null)
15	(null)	(null)	(null)

## 11.11 Creare și inserare în tabelul *EXEMPLAR*

```

CREATE TABLE EXEMPLAR (
    id_exemplar NUMBER(6) CONSTRAINT pk_exemplar PRIMARY KEY,
    id_carte NUMBER(6) NOT NULL,
    id_categorie NUMBER(6) NOT NULL,
    status VARCHAR2(10),
    CONSTRAINT fk_exemplar_carte FOREIGN KEY (id_carte)

```

```

REFERENCES CARTE(id_carte),
CONSTRAINT fk_exemplar_categorie FOREIGN KEY (id_categorie)
REFERENCES CATEGORIE(id_categorie)
);

```

```

CREATE SEQUENCE seq_exemplar
START WITH 1000
INCREMENT BY 1
MAXVALUE 9000
NOCACHE
NOCYCLE;

```

```

INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 102, 2, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 102, 2, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 103, 2, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 104, 2, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 105, 1, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 105, 1, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 105, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 106, 8, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 107, 9, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 107, 2, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 108, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 109, 8, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 110, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 111, 5, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 112, 7, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 112, 7, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 113, 6, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 114, 5, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 115, 5, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 115, 5, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 116, 4, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 117, 6, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 118, 4, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 119, 7, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 119, 7, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 119, 7, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 119, 7, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 120, 11, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 121, 11, 'disponibil');

```

```

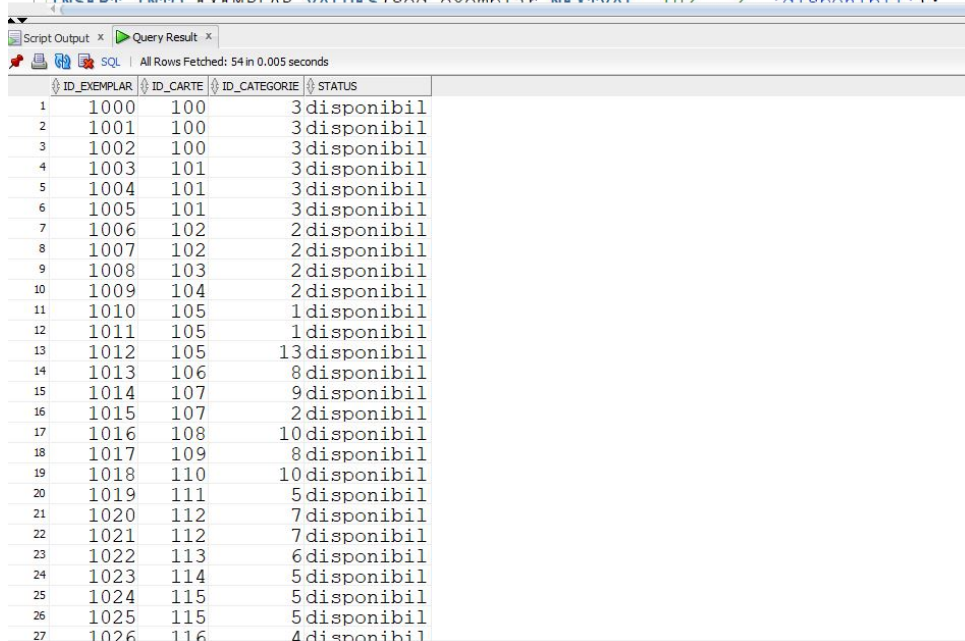
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 122, 11, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 123, 12, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 124, 11, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 125, 11, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 125, 11, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 126, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 126, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 127, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 127, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 127, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 128, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 128, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 128, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 128, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 129, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 129, 13, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 130, 8, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 130, 10, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 130, 13, 'disponibil');

```

```

INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 102, 2, 'disponibil');

```



	ID_EXEMPLAR	ID_CARTE	ID_CATEGORIE	STATUS
1	1000	100	3	disponibil
2	1001	100	3	disponibil
3	1002	100	3	disponibil
4	1003	101	3	disponibil
5	1004	101	3	disponibil
6	1005	101	3	disponibil
7	1006	102	2	disponibil
8	1007	102	2	disponibil
9	1008	103	2	disponibil
10	1009	104	2	disponibil
11	1010	105	1	disponibil
12	1011	105	1	disponibil
13	1012	105	13	disponibil
14	1013	106	8	disponibil
15	1014	107	9	disponibil
16	1015	107	2	disponibil
17	1016	108	10	disponibil
18	1017	109	8	disponibil
19	1018	110	10	disponibil
20	1019	111	5	disponibil
21	1020	112	7	disponibil
22	1021	112	7	disponibil
23	1022	113	6	disponibil
24	1023	114	5	disponibil
25	1024	115	5	disponibil
26	1025	115	5	disponibil
27	1026	116	4	disponibil

## 11.12 Creare și inserare în tabelul ÎMPRUMUT

```
CREATE TABLE IMPRUMUT (  
    id_imprumut NUMBER(6) CONSTRAINT pk_imprumut PRIMARY KEY,  
    id_exemplar NUMBER(6) NOT NULL,  
    id_cititor NUMBER(6) NOT NULL,  
    id_taxa NUMBER(6) NOT NULL,  
    data_imprumut DATE,  
    data_returnare DATE,  
    CONSTRAINT fk_exemplar FOREIGN KEY (id_exemplar)  
    REFERENCES EXEMPLAR(id_exemplar),  
    CONSTRAINT fk_cititor FOREIGN KEY (id_cititor)  
    REFERENCES CITITOR(id_cititor),  
    CONSTRAINT fk_taxa FOREIGN KEY (id_taxa)  
    REFERENCES TAXA(id_taxa)  
);  
  
CREATE SEQUENCE seq_imprumut  
    START WITH 1  
    INCREMENT BY 1  
    MAXVALUE 10000  
    NOCACHE  
    NOCYCLE;  
  
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,  
    data_imprumut, data_returnare)  
VALUES (seq_imprumut.NEXTVAL, 1008, 1, 1, TO_DATE('2018-05-15', 'YYYY-MM-DD'),  
    TO_DATE('2018-06-10', 'YYYY-MM-DD'));  
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,  
    data_imprumut, data_returnare)  
VALUES (seq_imprumut.NEXTVAL, 1004, 2, 2, TO_DATE('2019-02-20', 'YYYY-MM-DD'),  
    TO_DATE('2019-03-25', 'YYYY-MM-DD'));  
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,  
    data_imprumut, data_returnare)  
VALUES (seq_imprumut.NEXTVAL, 1023, 2, 3, TO_DATE('2020-09-10', 'YYYY-MM-DD'),  
    TO_DATE('2020-10-20', 'YYYY-MM-DD'));  
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,  
    data_imprumut, data_returnare)  
VALUES (seq_imprumut.NEXTVAL, 1023, 4, 4, TO_DATE('2021-07-05', 'YYYY-MM-DD'),  
    TO_DATE('2021-08-05', 'YYYY-MM-DD'));  
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,  
    data_imprumut, data_returnare)  
VALUES (seq_imprumut.NEXTVAL, 1025, 5, 5, TO_DATE('2022-03-15', 'YYYY-MM-DD'),  
    TO_DATE('2022-05-15', 'YYYY-MM-DD'));  
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,  
    data_imprumut, data_returnare)  
VALUES (seq_imprumut.NEXTVAL, 1034, 6, 6, TO_DATE('2022-08-10', 'YYYY-MM-DD'),  
    TO_DATE('2022-09-10', 'YYYY-MM-DD'));
```

```

INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut, data_returnare)
VALUES (seq_imprumut.NEXTVAL, 1033, 6, 7, TO_DATE('2022-12-20', 'YYYY-MM-DD'),
TO_DATE('2023-01-20', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1030, 7, 8, TO_DATE('2023-04-05', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1027, 9, 9, TO_DATE('2023-03-01', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1006, 10, 10, TO_DATE('2022-12-30', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1033, 10, 11, TO_DATE('2023-04-05', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1007, 8, 12, TO_DATE('2023-04-15', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1043, 3, 13, TO_DATE('2023-02-21', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1003, 2, 14, TO_DATE('2023-03-05', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1011, 1, 15, TO_DATE('2023-05-10', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1028, 1, 16, TO_DATE('2023-05-20', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1041, 4, 17, TO_DATE('2023-04-28', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1044, 4, 18, TO_DATE('2023-04-01', 'YYYY-MM-DD'));
INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
data_imprumut)
VALUES (seq_imprumut.NEXTVAL, 1048, 4, 19, TO_DATE('2023-04-23', 'YYYY-MM-DD'));

```

```

VALUES (seq_imprumut.NEXTVAL, 1025, 5, 5, TO_DATE('2022-03-15', 'YYY
1 INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
2 VALUES (seq_imprumut.NEXTVAL, 1034, 6, 6, TO_DATE('2022-08-10', 'YYY
3 INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
4 VALUES (seq_imprumut.NEXTVAL, 1033, 6, 7, TO_DATE('2022-12-20', 'YYY
5 INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,
6 VALUES (seq_imprumut.NEXTVAL, 1030, 7, 8, TO_DATE('2023-04-05', 'YYY
7 INSERT INTO IMPRUMUT (id_imprumut, id_exemplar, id_cititor, id_taxa,

```

Script Output x Query Result x

SQL | All Rows Fetched: 19 in 0.002 seconds

ID_IMPRUMUT	ID_EXEMPLAR	ID_CITITOR	ID_TAXA	DATA_IMPRUMUT	DATA_RETUNARE
1	2	1004	2	220-FEB-19	25-MAR-19
2	3	1023	2	310-SEP-20	20-OCT-20
3	4	1023	4	405-JUL-21	05-AUG-21
4	5	1025	5	515-MAR-22	15-MAY-22
5	6	1034	6	610-AUG-22	10-SEP-22
6	7	1033	6	720-DEC-22	20-JAN-23
7	8	1030	7	805-APR-23	(null)
8	9	1027	9	901-MAR-23	(null)
9	10	1006	10	1030-DEC-22	(null)
10	11	1033	10	1105-APR-23	(null)
11	12	1007	8	1215-APR-23	(null)
12	13	1043	3	1321-FEB-23	(null)
13	14	1003	2	1405-MAR-23	(null)
14	15	1011	1	1510-MAY-23	(null)
15	16	1028	1	1620-MAY-23	(null)
16	17	1041	4	1728-APR-23	(null)
17	18	1044	4	1801-APR-23	(null)
18	19	1048	4	1923-APR-23	(null)
19	1	1008	1	115-MAY-18	10-JUN-18

## 12 Cereri SQL

### 12.1 Cererea I

Să se afișeze pentru fiecare taxă ID-ul, suma inițială (dacă nu există atunci se va afișa 0), suma mărită cu 15% în cazul în care exemplarul împrumutat aparține secțiunii Matematică” și cu 20% în cazul în care exemplarul împrumutat aparține secțiunii Literatură”. În plus se vor afișa și codul exemplarului împrumutat, titlul cărții, precum și numele secțiunii din care face parte. Datele vor fi ordonate în ordinea descrescătoare a sumelor mărite.

*În cadrul acestei cereri am utilizat funcțiile NVL și Decode, precum și clauza ORDER BY.*

```

SELECT T.ID_TAXA, NVL(T.SUMA, 0) AS SUMA_INITIALA,
       NVL(T.SUMA *
           DECODE(UPPER(S.NUME_SECTIUNE), 'MATEMATICA', 1.15, 'LITERATURA', 1.20, 1),
           0) AS SUMA_MARITA, E.ID_EXEMPLAR, C.TITLU, S.NUME_SECTIUNE
FROM TAXA T
LEFT JOIN IMPRUMUT I ON T.ID_TAXA = I.ID_TAXA
LEFT JOIN EXEMPLAR E ON I.ID_EXEMPLAR = E.ID_EXEMPLAR
LEFT JOIN CATEGORIE CG ON E.ID_CATEGORIE = CG.ID_CATEGORIE
LEFT JOIN SECTIUNE S ON CG.ID_SECTIUNE = S.ID_SECTIUNE
LEFT JOIN CARTE C ON E.ID_CARTE = C.ID_CARTE
ORDER BY SUMA_MARITA DESC;
```



```

--CEREREA I
SELECT t.id_taxa, NVL(t.suma, 0) AS suma_inicială,
       NVL(t.suma *
           DECODE(s.ume_sectiune, 'Matematica', 1.15, 'Literatura', 1.20, 1), 0) AS suma_mărită,
       e.id_exemplar, c.titlu, s.ume_sectiune
FROM TAXA t
LEFT JOIN IMPRUMUT i ON t.id_taxa = i.id_taxa
LEFT JOIN EXEMPLAR e ON i.id_exemplar = e.id_exemplar
LEFT JOIN CATEGORIE cg ON e.id_categorie = cg.id_categorie
LEFT JOIN SECTIUNE s ON cg.id_sectiune = s.id_sectiune
LEFT JOIN CARTE c ON e.id_carte = c.id_carte
ORDER BY suma_mărită DESC;

```

ID_TAXA	SUMA_INICIALĂ	SUMA_MĂRITĂ	ID_EXEMPLAR	TITLU	NUME_SECTIUNE
5	31	31	1025	Introducere in bazele de date	IT
3	10	10	1023	Algoritmi si structuri de date	IT
2	3	3.45	1004	Analiza - ecuatii diferentiale	Matematica
6	1	1.2	1034	Romeo si Julieta	Literatura
7	1	1.2	1033	Mândrie si prejudecata	Literatura
4	1	1	1023	Algoritmi si structuri de date	IT
16	0	0	1028	Algoritmi si structuri de date avansate	IT
8	0	0	1030	Testare si calitatea software	IT
11	0	0	1033	Mândrie si prejudecata	Literatura
17	0	0	1041	The Feynman Lectures on Physics	Diverse
13	0	0	1043A	Brief History of Time	Fizica
18	0	0	1044A	Brief History of Time	Diverse
19	0	0	1048	The Origin of Species	Fizica
20	0	0	(null)	(null)	(null)
21	0	0	(null)	(null)	(null)
9	0	0	1027	Dezvoltare web cu HTML si CSS	IT
15	0	0	1011	Teoria grupurilor si aplicatii	Matematica

## 12.2 Cererea II

Să se afișeze pentru fiecare carte care are numărul de pagini mai mare decât media numărului de pagini a tuturor cărților titlul concatenat cu numărul de pagini, numărul de exemplare ale cărții, numărul de autori care au scris cartea și numărul de rezervări înregistrate pentru carte. Se vor lua în calcul doar acele cărți ale căror exemplare se găsesc în secțiunea IT”.

În cadrul acestei cereri am utilizat un bloc de cerere cu clauza *WITH*, subcereri sincronizate cu min 3 tabele, precum și funcții pe șiruri de caractere (*UPPER*, *CONCAT*).

```

WITH DATE_CERUTE AS (SELECT AVG(NUMAR_PAGINI) AS MEDIE_PAGINI FROM CARTE)
SELECT CONCAT(C.TITLU, C.NUMAR_PAGINI) AS "TITLU SI NR.PAGINI",
       (SELECT COUNT(*)
        FROM SCRIERE S
        WHERE S.ID_CARTE=C.ID_CARTE) "NR_AUTORI",
       (SELECT COUNT(*)
        FROM EXEMPLAR E
        WHERE E.ID_CARTE=C.ID_CARTE) "NR_EXEMPLARE",
       (SELECT COUNT(*)
        FROM REZERVARE R
        WHERE R.ID_CARTE=C.ID_CARTE) "NR_REZERVARI"
FROM CARTE C, DATE_CERUTE D
WHERE C.NUMAR_PAGINI>D.MEDIE_PAGINI AND C.ID_CARTE IN
(SELECT E.ID_CARTE
 FROM EXEMPLAR E
 JOIN CATEGORIE CAT ON CAT.ID_CATEGORIE=E.ID_CATEGORIE
 JOIN SECTIUNE SEC ON SEC.ID_SECTIUNE=CAT.ID_SECTIUNE
 WHERE E.ID_CARTE=C.ID_CARTE AND UPPER(SEC.NUME_SECTIUNE)='IT');

```

```

WITH DATE_CERUTE AS (SELECT AVG(NUMAR_PAGINI) AS MEDIE_PAGINI FROM CARTE)
SELECT CONCAT(C.TITLU, C.NUMAR_PAGINI) AS "TITLU SI NR.PAGINI",
       (SELECT COUNT(*)
        FROM SCRIERE S
        WHERE S.ID_CARTE=C.ID_CARTE) "NR_AUTORI",
       (SELECT COUNT(*)
        FROM EXEMPLAR E
        WHERE E.ID_CARTE=C.ID_CARTE) "NR_EXEMPLARE",
       (SELECT COUNT(*)
        FROM REZERVARE R
        WHERE R.ID_CARTE=C.ID_CARTE) "NR_REZERVARI"
FROM CARTE C, DATE_CERUTE D
WHERE C.NUMAR_PAGINI>D.MEDIE_PAGINI AND C.ID_CARTE IN
      (SELECT E.ID_CARTE
       FROM EXEMPLAR E
       JOIN CATEGORIE CAT ON CAT.ID_CATEGORIE=E.ID_CATEGORIE
       JOIN SECTIUNE SEC ON SEC.ID_SECTIUNE=CAT.ID_SECTIUNE
       WHERE E.ID_CARTE=C.ID_CARTE AND UPPER(SEC.NUME_SECTIUNE)='IT');

```

Query Result

All Rows Fetched: 2 in 0.009 seconds

TITLU SI NR.PAGINI	NR_AUTORI	NR_EXEMPLARE	NR_REZERVARI
Algoritmi si structuri de date450	1	1	4
Algoritmi si structuri de date avansate500	1	1	0

## 12.3 Cererea III

Să se afișeze date despre edituri, precum și numărul total de împrumuturi pe care le-au avut exemplarele cărților publicate de aceste edituri. Datele vor fi afișate în ordinea descrescătoare numărului de împrumuturi, de asemenea se vor omite editurile care au avut mai puțin de 2 împrumuturi.

*În cadrul acestei cereri am utilizat funcții de grup și filtrare la nivel de grupuri, precum și subcereri sincronizate în care intervin minim 3 tabele.*

```

SELECT E.id_editura, E.numa_editura,
SUM(TOTAL_IMPRUMUTURI) AS numar_total_imprumuturi
FROM EDITURA E
JOIN (
    SELECT C.id_editura, COUNT(I.id_exemplar) AS TOTAL_IMPRUMUTURI
    FROM CARTE C
    LEFT JOIN EXEMPLAR EX ON C.id_carte = EX.id_carte
    LEFT JOIN IMPRUMUT I ON EX.id_exemplar = I.id_exemplar
    GROUP BY C.id_editura
) T ON E.id_editura = T.id_editura
GROUP BY E.id_editura, E.numa_editura
HAVING SUM(TOTAL_IMPRUMUTURI) > 2
ORDER BY numar_total_imprumuturi DESC;

```



```

5  --CEREREA III
6  SELECT E.id_editura, E.ume_editura,
7         SUM(TOTAL_IMPRUMUTURI) AS numar_total_imprumuturi
8  FROM EDITURA E
9  JOIN (
10     SELECT C.id_editura, COUNT(I.id_exemplar) AS TOTAL_IMPRUMUTURI
11     FROM CARTE C
12     LEFT JOIN EXEMPLAR EX ON C.id_carte = EX.id_carte
13     LEFT JOIN IMPRUMUT I ON EX.id_exemplar = I.id_exemplar
14     GROUP BY C.id_editura
15 ) T ON E.id_editura = T.id_editura
16 GROUP BY E.id_editura, E.ume_editura
17 HAVING SUM(TOTAL_IMPRUMUTURI) > 2

```

ID_EDITURA	NUME_EDITURA	NUMAR_TOTAL_IMPRUMUTURI
2	Polirom	6
1	Aramis	5
4	Niculescu	4
3	Corint	3

## 12.4 Cererea IV

Să se afișeze id-ul cărților, titlurile cărților precum și numărul de exemplare existente în bibliotecă din fiecare carte, urmat de frecvența cărții (carte rară” dacă numărul de exemplare e mai mic decât 3, carte comună” dacă numărul de exemplare e cuprins între 3 și 5, respectiv carte populară” dacă numărul de exemplare este mai mare de 5).

*În cadrul acestei cereri am utilizat o subcere nesincronizată în clauza FROM, precum și o expresie CASE.*

```

SELECT C.ID_CARTE, C.TITLU, EXP.NR "NR EXEMPLARE",
       CASE
         WHEN EXP.NR < 3 THEN 'CARTE RARĂ'
         WHEN EXP.NR >= 3 AND EXP.NR < 5 THEN 'CARTE COMUNĂ'
         WHEN EXP.NR >= 5 THEN 'CARTE POPULARĂ'
       END AS "FRECVENTA CARTE"
FROM CARTE C,
     (SELECT COUNT(*) NR, ID_CARTE
      FROM EXEMPLAR
      GROUP BY ID_CARTE
      HAVING COUNT(*) > 1 ) EXP
WHERE C.ID_CARTE = EXP.ID_CARTE;

```

```

42 |
43 | --CEREREA IV
44 | SELECT C.ID_CARTE, C.TITLU, EXP.NR "NR EXEMPLARE",
45 |     CASE
46 |         WHEN EXP.NR < 3 THEN 'CARTE RARĂ'
47 |         WHEN EXP.NR >= 3 AND EXP.NR < 5 THEN 'CARTE COMUNĂ'
48 |         WHEN EXP.NR >= 5 THEN 'CARTE POPULARĂ'
49 |     END AS "FRECVENTA CARTE"
50 |
51 | FROM CARTE C,
52 |     (SELECT COUNT(*) NR, ID_CARTE
53 |     FROM EXEMPLAR
54 |     GROUP BY ID_CARTE
55 |     HAVING COUNT(*) > 1 ) EXP

```

ID_CARTE	TITLU	NR EXEMPLARE	FRECVENTA CARTE
1	100 Introducere in analiza matematica	6	CARTE POPULARĂ
2	101 Analiza - ecuatii differentiale	6	CARTE POPULARĂ
3	102 Geometrie euclidiană	4	CARTE COMUNĂ
4	103 Introducere in geometria diferentia	2	CARTE RARĂ
5	104 Geometrie computatională	2	CARTE RARĂ
6	105 Teoria grupurilor si aplicatii	6	CARTE POPULARĂ
7	106 Mecanica: Fundamente si aplicatii	2	CARTE RARĂ
8	107 Introducere in optica geometrica	4	CARTE COMUNĂ
9	108 Astrofizica: Galaxii si stele	2	CARTE RARĂ
10	109 Mecanica cuantica si statistica	2	CARTE RARĂ
11	110 Originea si evolutia Universului	2	CARTE RARĂ
12	111 Baze de date relationale	2	CARTE RARĂ
13	112 Dezvoltare software: tehnici moderne	4	CARTE COMUNĂ
14	113 Aplicatii web dinamice	2	CARTE RARĂ

## 12.5 Cererea V

Să se afișeze id-ul, numele, prenumele și anul în care s-au înscris la bibliotecă al cititorilor care au împrumutat cărți cu un număr de pagini mai mare decât media numărului de pagini al tuturor cărților disponibile. În plus se va afișa pentru fiecare cititor și numărul de astfel de împrumuturi.

*În cadrul acestei cereri am utilizat subcereri nesincronizate cu minim 3 tabele precum și o funcție pe date calendaristice (extract)*

```

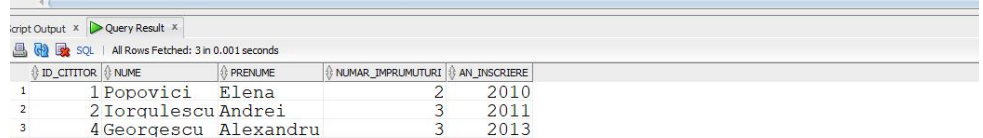
SELECT C.id_cititor, C.nume, C.prenume, COUNT(*) AS numar_imprumuturi,
EXTRACT(YEAR FROM C.DATA_INSCRIERE) AS AN_INSCRIERE
FROM CITITOR C
INNER JOIN (
    SELECT I.id_cititor, E.id_exemplar
    FROM IMPRUMUT I
    INNER JOIN EXEMPLAR E ON I.id_exemplar = E.id_exemplar
    INNER JOIN CARTE CR ON E.id_carte = CR.id_carte
    WHERE CR.numar_pagini > (
        SELECT AVG(numar_pagini)
        FROM CARTE
    )
) IM ON C.id_cititor = IM.id_cititor
GROUP BY C.id_cititor, C.nume, C.prenume, EXTRACT(YEAR FROM C.DATA_INSCRIERE);

```

```

2  --CEREREA V
3  SELECT C.id_cititor, C.num, C.prenume, COUNT(*) AS numar_imprumuturi, EXTRACT(YEAR F
4  FROM CITITOR C
5  INNER JOIN (
6  SELECT I.id_cititor, E.id_exemplar
7  FROM IMPRUMUT I
8  INNER JOIN EXEMPLAR E ON I.id_exemplar = E.id_exemplar
9  INNER JOIN CARTE CR ON E.id_carte = CR.id_carte
10 WHERE CR.numar_pagini > (
11     SELECT AVG(numar_pagini)
12     FROM CARTE
13 )
14 ) IM ON C.id_cititor = IM.id_cititor
15 GROUP BY C.id_cititor, C.num, C.prenume, EXTRACT(YEAR FROM C.DATA_INSCRIERE);

```



ID_CITITOR	NUME	PRENUME	NUMAR_IMPRUMUTURI	AN_INSCRIERE
1	Popovici	Elena	2	2010
2	Iorqulescu	Andrei	3	2011
3	Georgescu	Alexandru	3	2013

## 13 Operații de actualizare și suprimare a datelor

### 13.1 Operația 1

Să se actualizeze statusul cărților din tabela EXEMPLAR pentru acele exemplare care figurează ca împrumutate și încă nereturnate în tabela ÎMPRUMUT.

```

UPDATE EXEMPLAR
SET status = (
    SELECT
        CASE
            WHEN data_returnare IS NOT NULL THEN 'disponibil'
            ELSE 'imprumutat'
        END
    FROM IMPRUMUT
    WHERE id_exemplar = EXEMPLAR.id_exemplar
    ORDER BY data_imprumut DESC
    FETCH FIRST 1 ROW ONLY
)
WHERE id_exemplar IN (
    SELECT id_exemplar
    FROM IMPRUMUT
);

```

Tabelul EXEMPLAR înainte de a fi actualizate informațiile:

```

INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 100, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 101, 3, 'disponibil');
INSERT INTO EXEMPLAR VALUES(seq_exemplar.NEXTVAL, 102, 2, 'disponibil');

```

ID_EXEMPLAR	ID_CARTE	ID_CATEGORIE	STATUS
1	1000	100	3 disponibil
2	1001	100	3 disponibil
3	1002	100	3 disponibil
4	1003	101	3 disponibil
5	1004	101	3 disponibil
6	1005	101	3 disponibil
7	1006	102	2 disponibil
8	1007	102	2 disponibil
9	1008	103	2 disponibil
10	1009	104	2 disponibil
11	1010	105	1 disponibil
12	1011	105	1 disponibil
13	1012	105	13 disponibil
14	1013	106	8 disponibil
15	1014	107	9 disponibil
16	1015	107	2 disponibil
17	1016	108	10 disponibil
18	1017	109	8 disponibil
19	1018	110	10 disponibil
20	1019	111	5 disponibil
21	1020	112	7 disponibil
22	1021	112	7 disponibil
23	1022	113	6 disponibil
24	1023	114	5 disponibil
25	1024	115	5 disponibil
26	1025	115	5 disponibil
27	1026	116	4 disponibil

Tabelul EXEMPLAR după actualizarea informațiilor:

```

511 UPDATE EXEMPLAR
512 SET status = (
513     SELECT
514         CASE
515             WHEN data_returnare IS NOT NULL THEN 'disponibil'
516             ELSE 'imprumutat'
517         END

```

ID_EXEMPLAR	ID_CARTE	ID_CATEGORIE	STATUS
7	1006	102	2 imprumutat
8	1007	102	2 imprumutat
9	1008	103	2 disponibil
10	1009	104	2 disponibil
11	1010	105	1 disponibil
12	1011	105	1 imprumutat
13	1012	105	13 disponibil
14	1013	106	8 disponibil
15	1014	107	9 disponibil
16	1015	107	2 disponibil
17	1016	108	10 disponibil
18	1017	109	8 disponibil
19	1018	110	10 disponibil
20	1019	111	5 disponibil
21	1020	112	7 disponibil
22	1021	112	7 disponibil
23	1022	113	6 disponibil
24	1023	114	5 disponibil
25	1024	115	5 disponibil
26	1025	115	5 disponibil
27	1026	116	4 disponibil
28	1027	117	6 imprumutat
29	1028	118	4 imprumutat

## 13.2 Operația 2

Modificați tabelul TAXA astfel încât pentru exemplarele împrumutate să se calculeze suma pe care cititorii trebuie să o plătescă. De asemenea schimbați informațiile despre statusul și data achitării taxei. Data achitării trebuie să coincidă cu data returnării exemplarului din tabela ÎMPRUMUT, iar statusul va fi setat la "achitat".

```
UPDATE TAXA
SET suma = CASE
    WHEN EXISTS (
        SELECT 1
        FROM IMPRUMUT
        WHERE IMPRUMUT.id_taxa = TAXA.id_taxa
            AND data_imprumut IS NOT NULL
            AND data_returnare IS NOT NULL
    ) THEN
        CASE
            WHEN (
                SELECT data_returnare - data_imprumut
                FROM IMPRUMUT
                WHERE IMPRUMUT.id_taxa = TAXA.id_taxa
            ) > 30 THEN (
                SELECT data_returnare - data_imprumut - 30
                FROM IMPRUMUT
                WHERE IMPRUMUT.id_taxa = TAXA.id_taxa
            )
            ELSE suma
        END
    ELSE suma
END,
data_achitare = (
    SELECT data_returnare
    FROM IMPRUMUT
    WHERE IMPRUMUT.id_taxa = TAXA.id_taxa
        AND data_imprumut IS NOT NULL
        AND data_returnare IS NOT NULL
),
status = CASE
    WHEN EXISTS (
        SELECT 1
        FROM IMPRUMUT
        WHERE IMPRUMUT.id_taxa = TAXA.id_taxa
            AND data_returnare IS NOT NULL
    ) THEN 'achitat'
    ELSE status
END
```

```

WHERE id_taxa IN (
    SELECT id_taxa
    FROM IMPRUMUT
    WHERE data_imprumut IS NOT NULL
    AND data_returnare IS NOT NULL
);

```

Tabelul TAXA înaintea actualizării informațiilor:

```

CREATE SEQUENCE seq_taxa
START WITH 1
INCREMENT BY 1
MAXVALUE 10000
NOCACHE
NOCYCLE;

--INSERARE IN TAXA
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);
INSERT INTO TAXA (id_taxa) VALUES (seq_taxa.NEXTVAL);

```

ID_TAXA	SUMA	STATUS	DATA_ACHITARE
1	(null)	(null)	(null)
2	(null)	(null)	(null)
3	(null)	(null)	(null)
4	(null)	(null)	(null)
5	(null)	(null)	(null)
6	(null)	(null)	(null)
7	(null)	(null)	(null)
8	(null)	(null)	(null)
9	(null)	(null)	(null)
10	(null)	(null)	(null)
11	(null)	(null)	(null)
12	(null)	(null)	(null)
13	(null)	(null)	(null)
14	(null)	(null)	(null)
15	(null)	(null)	(null)

Tabelul TAXA după actualizarea informațiilor:

```

UPDATE TAXA
SET suma = CASE
    WHEN EXISTS (
        SELECT 1
        FROM IMPRUMUT
        WHERE IMPRUMUT.id_taxa = TAXA.id_taxa
        AND data_imprumut IS NOT NULL
    )

```

ID_TAXA	SUMA	STATUS	DATA_ACHITARE
1	(null)	achitat	10-JUN-18
2	3	achitat	25-MAR-19
3	10	achitat	20-OCT-20
4	1	achitat	05-AUG-21
5	31	achitat	15-MAY-22
6	1	achitat	10-SEP-22
7	1	achitat	20-JAN-23
8	(null)	(null)	(null)
9	(null)	(null)	(null)
10	(null)	(null)	(null)
11	(null)	(null)	(null)
12	(null)	(null)	(null)
13	(null)	(null)	(null)
14	(null)	(null)	(null)
15	(null)	(null)	(null)



### 13.3 Operația 3

Creați o comandă SQL prin care să se șteargă datele tuturor bibliotecarilor care nu se ocupă de nicio secțiune a bibliotecii.

```
DELETE FROM BIBLIOTECAR
WHERE id_bibliotecar NOT IN (
SELECT id_bibliotecar
FROM SECTIUNE
);
```

Tabelul BIBLIOTECAR înainte actualizării informațiilor:

```
CREATE SEQUENCE seq_bibliotecar_id
START WITH 1
INCREMENT BY 1
MAXVALUE 100
NOCYCLE;

--INSERARE IN BIBLIOTECAR
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s.
VALUES (seq_bibliotecar_id.NEXTVAL, 'Popescu', 'Ion', 'ion.popescu@yahoo.com'
TO_DATE('01-01-1990', 'DD-MM-YYYY'));
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s.
VALUES (seq_bibliotecar_id.NEXTVAL, 'Ionescu', 'Maria', 'maria.ionescu@gmail
TO_DATE('01-02-1995', 'DD-MM-YYYY'));
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s.
VALUES (seq_bibliotecar_id.NEXTVAL, 'Popa', 'Mihai', 'mihai.popa@gmail.com',
TO_DATE('01-03-2000', 'DD-MM-YYYY'));
INSERT INTO BIBLIOTECAR (id_bibliotecar, nume, prenume, email, nr_telefon, s.
```

ID_BIBLIOTECAR	NUME	PRENUME	EMAIL	NR_TELEFON	SALARIU	DATA_ANGAJARE
1	Popescu	Ion	ion.popescu@yahoo.com	0712345678	2000	01-JAN-90
2	Ionescu	Maria	maria.ionescu@gmail.com	0723456789	3500	01-FEB-95
3	Popa	Mihai	mihai.popa@gmail.com	0734567890	3200	01-MAR-00
4	Georgescu	Ana	ana.georgescu@gmail.com	0745678901	3100	01-APR-05
5	Radu	Adrian	adrian.radu@yahoo.com	0756789012	3300	01-MAY-10
6	Pop	Andreea	andreea.pop@yahoo.com	0767890123	3400	01-JUN-15
7	Badea	Mircea	mircea.badea@gmail.com	0778901234	2100	01-JUL-20
8	Diaconu	Elena	elena.diaconu@gmail.com	0789012345	3200	01-AUG-21
9	Dumitru	Raluca	raluca.dumitru@yahoo.com	0701234567	1800	01-OCT-23
10	Marinescu	George	george@yahoo.com	0790123456	3100	01-SEP-22

Tabelul BIBLIOTECAR după actualizarea informațiilor:

```
DELETE FROM BIBLIOTECAR
WHERE id_bibliotecar NOT IN (
SELECT id_bibliotecar
FROM SECTIUNE
);
```

ID_BIBLIOTECAR	NUME	PRENUME	EMAIL	NR_TELEFON	SALARIU	DATA_ANGAJARE
1	Popescu	Ion	ion.popescu@yahoo.com	0712345678	2000	01-JAN-90
2	Ionescu	Maria	maria.ionescu@gmail.com	0723456789	3500	01-FEB-95
3	Popa	Mihai	mihai.popa@gmail.com	0734567890	3200	01-MAR-00
4	Georgescu	Ana	ana.georgescu@gmail.com	0745678901	3100	01-APR-05
5	Radu	Adrian	adrian.radu@yahoo.com	0756789012	3300	01-MAY-10

## 14 Crearea unei vizualizări complexe

```
CREATE VIEW DETALII_CARTI AS
(
    SELECT C.ID_CARTE, C.TITLU, A.NUME || ' ' || A.PRENUME AS "NUME_AUTOR",
    COUNT(E.ID_EXEMPLAR) AS "NUMAR_EXEMPLARE"
    FROM CARTE C
    JOIN SCRIERE S ON C.ID_CARTE=S.ID_CARTE
    JOIN AUTOR A ON S.ID_AUTOR=A.ID_AUTOR
    JOIN EXEMPLAR E ON C.ID_CARTE=E.ID_CARTE
    GROUP BY C.ID_CARTE, C.TITLU, A.NUME || ' ' || A.PRENUME
);
```

### Operație LMD permisă

```
CREATE VIEW DETALII_CARTI AS
(
    SELECT C.ID_CARTE, C.TITLU, A.NUME || ' ' || A.PRENUME AS "NUME_AUTOR", COUNT(E.
    FROM CARTE C
    JOIN SCRIERE S ON C.ID_CARTE=S.ID_CARTE
    JOIN AUTOR A ON S.ID_AUTOR=A.ID_AUTOR
    JOIN EXEMPLAR E ON C.ID_CARTE=E.ID_CARTE
    GROUP BY C.ID_CARTE, C.TITLU, A.NUME || ' ' || A.PRENUME
);
```

```
SELECT * FROM DETALII_CARTI ORDER BY "NUME_AUTOR";
```

ID_CARTE	TITLU	NUME_AUTOR	NUMAR_EXEMPLARE
120	Mândrie și prejudecată	Austen Jane	1
111	Baze de date relationale	Constantin Elena	1
117	Dezvoltare web cu HTML și CSS	Constantin Elena	1
115	Introducere în bazele de date	Constantin Elena	2
110	Originea și evoluția Universului	Constantin Elena	1
105	Teoria grupurilor și aplicații	Constantin Elena	3
128	The Origin of Species	Darwin Charles	4
125	Crimă și pedeapsă	Dostoevsky Fyodor	2
115	Introducere în bazele de date	Dumitrescu Ana	2
103	Introducere în geometria diferențială	Dumitrescu Ana	1
109	Mecanica cuantică și statistică	Dumitrescu Ana	1
123	Luceafărul	Eminescu Mihai	1
126	The Feynman Lectures on Physics	Feynman Richard P.	2
104	Geometrie computațională	Georgescu Mihai	1

### Operație LMD nepermisă

```
28 |
29 | DELETE FROM DETALII_CARTI
30 | WHERE ID_CARTE=120;
```

Script Output x Query Result x  
Task completed in 0.062 seconds

```
Error starting at line : 29 in command -
DELETE FROM DETALII_CARTI
WHERE ID_CARTE=120
Error at Command Line : 29 Column : 13
Error report -
SQL Error: ORA-01732: data manipulation operation not legal on this view
1732. 00000 - "data manipulation operation not legal on this view"
Cause:
```



## 15 Cereri SQL

### 15.1 Operația outer-join

Creați o condică pentru starea împrumuturilor bibliotecii. În condică vor fi afișați cititorii bibliotecii care au împrumutat cărți, precum și titlurile cărților împrumutate, urmați de cititorii care nu au împrumutat nicio carte. În condică se vor adăuga la final cărțile ale căror exemplare nu au fost împrumutate de niciun cititor.

```
SELECT DISTINCT C.ID_CITITOR, CONCAT(CONCAT(C.NUME, ' '), C.PRENUME)
AS "CITITOR", C1.TITLU AS "TITLU CARTE", ED.NUME_EDITURA "TITLU_EDITURA"
FROM CITITOR C
FULL OUTER JOIN IMPRUMUT I ON C.ID_CITITOR=I.ID_CITITOR
LEFT OUTER JOIN EXEMPLAR E ON E.ID_EXEMPLAR=I.ID_EXEMPLAR
FULL OUTER JOIN CARTE C1 ON E.ID_CARTE=C1.ID_CARTE
FULL OUTER JOIN EDITURA ED ON C1.ID_EDITURA=ED.ID_EDITURA
ORDER BY C.ID_CITITOR;
```

1	Popovici Elena	Algoritmi si structuri de date avansate	Corint
1	Popovici Elena	Introducere in geometria diferentia	Polirom
1	Popovici Elena	Teoria grupurilor si aplicatii	Polirom
2	Iorqulescu Andrei	Algoritmi si structuri de date	Niculescu
2	Iorqulescu Andrei	Analiza - ecuatii diferentiale	Polirom
3	Popescu Maria	A Brief History of Time	Corint
4	Georgescu Alexandru	A Brief History of Time	Corint
4	Georgescu Alexandru	Algoritmi si structuri de date	Niculescu
4	Georgescu Alexandru	The Feynman Lectures on Physics	Aramis
4	Georgescu Alexandru	The Origin of Species	Niculescu
5	Radu Cristina	Introducere in bazele de date	Mate2000
6	Petrescu Ionut	Mândrie si prejudecata	Aramis
6	Petrescu Ionut	Romeo si Julieta	Polirom
7	Constantin Andreea	Testare si calitatea software	Niculescu
8	Mihai Adrian	Geometrie euclidiana	Aramis
9	Popa Alexandra	Dezvoltare web cu HTML si CSS	Polirom
10	Balan Razvan	Geometrie euclidiana	Aramis
10	Balan Razvan	Mândrie si prejudecata	Aramis
11	Ionascu Ion	(null)	(null)
12	Popovici Elena	(null)	(null)
13	Iorqulescu Andrei	(null)	(null)
14	Popescu Maria	(null)	(null)
15	Georgescu Alexandru	(null)	(null)
16	Radu Cristina	(null)	(null)
17	Petrescu Ionut	(null)	(null)
18	Constantin Andreea	(null)	(null)
19	Mihai Adrian	(null)	(null)
20	Popa Alexandra	(null)	(null)
21	Balan Razvan	(null)	(null)
22	Ionascu Ion	(null)	(null)
(null)		Aplicatii web dinamice	Corint
(null)		Astrofizica: Galaxii si stele	Corint
(null)		Baze de date relationale	Aramis
(null)		Crima si pedeapsa	Corint

## 15.2 Operația division

Afișați informații despre cărțile ale căror exemplare se găsesc în toate secțiunile de care se ocupă bibliotecarul cu ID-ul 5.

```
SELECT C.ID_CARTE, C.TITLU
FROM CARTE C
WHERE NOT EXISTS (
    SELECT S.ID_SECTIUNE
    FROM SECTIUNE S
    WHERE S.ID_BIBLIOTECAR = 5
    AND NOT EXISTS (
        SELECT E.ID_EXEMPLAR
        FROM EXEMPLAR E
        JOIN CATEGORIE CAT ON E.ID_CATEGORIE=CAT.ID_CATEGORIE
        WHERE E.ID_CARTE = C.ID_CARTE
        AND CAT.ID_SECTIUNE = S.ID_SECTIUNE
    )
);
```

```
SELECT C.ID_CARTE, C.TITLU
FROM CARTE C
WHERE NOT EXISTS (
    SELECT S.ID_SECTIUNE
    FROM SECTIUNE S
    WHERE S.ID_BIBLIOTECAR = 5
    AND NOT EXISTS (
        SELECT E.ID_EXEMPLAR
        FROM EXEMPLAR E
        JOIN CATEGORIE CAT ON E.ID_CATEGORIE=CAT.ID_CATEGORIE
        WHERE E.ID_CARTE = C.ID_CARTE
        AND CAT.ID_SECTIUNE = S.ID_SECTIUNE
    )
);
```

ID_CARTE	TITLU
105	Teoria grupurilor si aplicatii
126	The Feynman Lectures on Physics
127	A Brief History of Time
128	The Origin of Species
129	The Structure of Scientific Revolutions
130	The Double Helix

## 15.3 Analiza top-n

Să se afișeze primele 3 cărți cu cele mai multe exemplare din bibliotecă

```
SELECT T.TITLU, T.NUMAR_EXEMPLARE
FROM (
    SELECT C.TITLU, COUNT(E.ID_EXEMPLAR) AS NUMAR_EXEMPLARE
    FROM CARTE C
    JOIN EXEMPLAR E ON C.ID_CARTE = E.ID_CARTE
    GROUP BY C.TITLU
    ORDER BY COUNT(E.ID_EXEMPLAR) DESC
) T
WHERE ROWNUM <= 3;
```

```
76 --ANALIZA TOP-N
77 --Sa se afiseze primele 3 carti cu cele mai multe exemplare
78 SELECT T.TITLU, T.NUMAR_EXEMPLARE
79 FROM (
80     SELECT C.TITLU, COUNT(E.ID_EXEMPLAR) AS NUMAR_EXEMPLARE
81     FROM CARTE C
82     JOIN EXEMPLAR E ON C.ID_CARTE = E.ID_CARTE
83     GROUP BY C.TITLU
84     ORDER BY COUNT(E.ID_EXEMPLAR) DESC
85 ) T
86 WHERE ROWNUM <= 3;
```

Script Output x Query Result x	
All Rows Fetched: 3 in 0.002 seconds	
TITLU	NUMAR_EXEMPLARE
1 Testare si calitatea software	4
2 The Origin of Species	4
3 Introducere in analiza matematica	3

## 16 Optimizarea unei cereri

### 16.1 Cererea SQL

Să se afișeze titlul și numărul de pagini al cărților scrise de autorul cu numele "Popescu" și care au fost publicate la editura cu ID-ul 1.

```
SELECT C.TITLU, NVL(TO_CHAR(C.NUMAR_PAGINI), 'NESPECIFICAT')
FROM CARTE C
JOIN SCRIERE S ON C.ID_CARTE=S.ID_CARTE
JOIN AUTOR A ON A.ID_AUTOR=S.ID_AUTOR
WHERE UPPER(A.NUME) = 'POPESCU' AND C.ID_EDITURA=1;
```

### 16.2 Optimizare

Cererea este optim proiectată întrucât respectă regulile de optimizare:

- Elimină selecțiile redundante: Identifică și elimină selecțiile care pot fi deduse din selecții sau proiecții anterioare. Aceasta reduce numărul de înregistrări evaluate și accelerează cererea.
- Elimină utilizarea produselor carteziane.
- Elimină proiecțiile redundante: Elimină proiecțiile care nu afectează rezultatul final al cererii, reducând astfel timpul de execuție și numărul de înregistrări procesate.

### 16.3 Expresii algebrice

R1 = SELECT(AUTOR, NUME='POPESCU') - eliminăm attributele nefolositoare

R2 = PROJECT(R1, ID\_AUTOR) - eliminăm attributele nefolositoare

R3 = PROJECT(SCRIERE, ID\_AUTOR, ID\_CARTE)

R4 = JOIN(R2, R3, ID\_AUTOR)

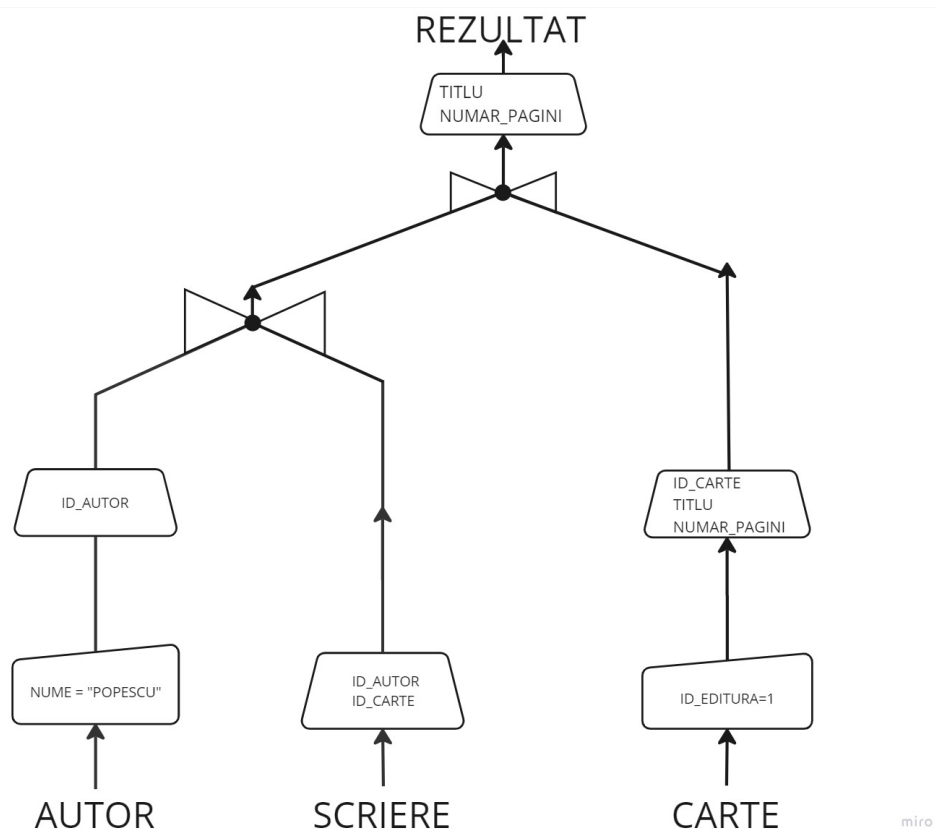
R5 = SELECT(CARTE, ID\_EDITURA=1)

R6 = PROJECT(R5, ID\_CARTE, NUME, PRET)

R7 = JOIN(R4,R6, ID\_CARTE)

R8 = PROJECT(R7, NUME, NUMAR\_PAGINI) = REZULTAT FINAL

## 16.4 Arbore algebric



## 17 Normalizarea BCNF, FN4, FN5

### 17.1 Forma Normală Boyce-Codd

O entitate se găsește în BCNF dacă și numai dacă:

- \* se găsește în a treia formă normală
- \* toate dependențele funcționale non-triviale sunt determinate de cheia primară

Să luăm în considerare tabela **EXEMPLAR** pentru exemplificare. Schema acestei tabele este: **EXEMPLAR** (**id\_exemplar#**, **id\_carte**, **id\_categorie**, **status**). Dependențele funcționale în această tabelă sunt:

- **id\_exemplar#**: **id\_carte**
- **id\_exemplar#**: **id\_categorie**

Observăm că există o dependență funcțională nedecompozabilă, respectiv  $\text{id\_exemplar\#} \rightarrow \text{id\_carte}$ . Acest lucru indică faptul că tabela EXEMPLAR nu se află în forma normală Boyce-Codd (BCNF). Pentru a normaliza această relație, putem crea o nouă tabelă pentru a separa dependența funcțională nedecompozabilă. Schema actualizată va fi:

- EXEMPLAR (id\_exemplar#, id\_categorie, status)
- CARTE\_EXEMPLAR (id\_exemplar#, id\_carte)

## 17.2 FN4

O entitate se găsește în FN4 dacă și numai dacă:

- \* se găsește în BCNF
- \* nu conține relații M:N independente.

Pentru a oferi un exemplu de normalizare în FN4, putem lua în considerare tabela REZERVARE. Schema acestei tabele este: REZERVARE (id\_rezervare#, id\_carte, id\_cititor). Dependențele funcționale în această tabelă sunt:

- id\_rezervare#: id\_carte
- id\_rezervare#: id\_cititor

Observăm că există o dependență M:N între id\_rezervare# și id\_cititor. Acest lucru indică faptul că tabela REZERVARE nu se află în FN4. Pentru a normaliza această relație conform FN4, putem crea două tabele separate pentru a gestiona dependența. Schema actualizată va fi:

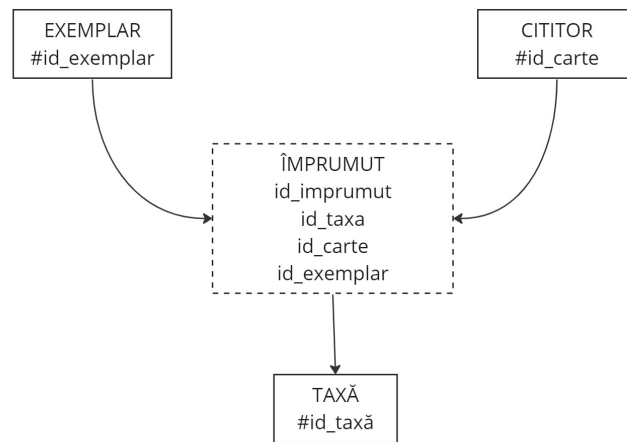
- REZERVARE (id\_rezervare#, id\_carte)
- REZERVARE\_CITITOR (id\_rezervare#, id\_cititor)

## 17.3 FN5

O entitate se găsește în FN5 dacă și numai dacă:

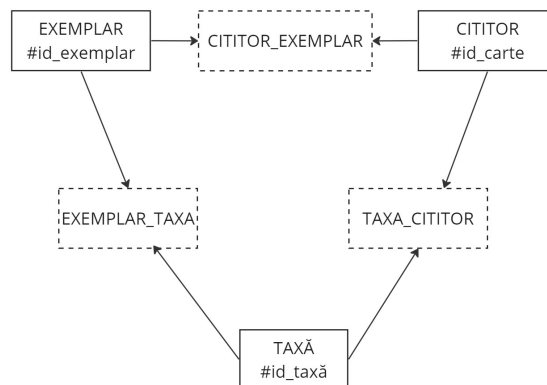
- \* se găsește în FN4
- \* nu conține dependențe ciclice.

Pentru exemplificarea normalizării vom considera următorul exemplu: un cititor poate plăti mai multe taxe pentru împrumutul mai multor exemplare.



miro

Relația de tip 3 poate fi echivalentă cu următoarea:



miro

Se observă că cele 3 relații de tip 2 compun o diagramă care conține dependente ciclice, deci relația de mai sus nu se află în FN5. Pe de altă parte, relația de tip 3 este în FN5.

## 17.4 Aplicarea denormalizării

Scopul denormalizării este de a reduce numărul de join-uri care trebuie realizate în cadrul unei interogări, prin realizarea unora dintre acestea în avans ca parte din proiectarea internă a bazei de date.

Pentru a exemplifica denormalizarea vom lua în calcul următorul exemplu. Să presupunem că avem următoarea tabelă: CARTE(id\_carte#, id\_editura, titlu, an\_publicare,

numar\_pagini). În mod normalizat, informațiile despre editură sunt stocate într-o entitate separată (tabelul EDITURĂ) și legate de tabela CARTE printr-o cheie externă (id\_editura). Cu toate acestea, dacă dorim să denormalizăm schema, putem include coloana nume\_editura direct în tabela CARTE, eliminând astfel nevoia de a accesa tabela EDITURĂ pentru a obține numele editurii. Tabela CARTE denormalizată ar putea arăta astfel: CARTE(id\_carte#, titlu, an\_publicare, numar\_pagini, nume\_editura). În această formă denormalizată, găsim informațiile despre editură (nume\_editură) direct în tabela CARTE. Aceasta poate duce la o mai bună performanță în anumite scenarii, deoarece nu mai este nevoie de o operație de join pentru a obține informațiile despre editură.

## 18 Optimizarea cererilor prin indexare

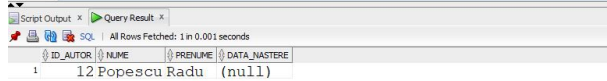
### 18.1 Cererea I

Să se afișeze eficient autorul cu numele Popescu Radu.

```

84 --19
85 --OPTIMIZAREA UNOR CERERI FOLOSIND INDEXARE
86
87 --CEREREA I
88 --Sa se afiseze eficient autorul cu numele 'Popescu Radu'
89 CREATE INDEX IDX_NUME_AUTOR
90 ON AUTOR (NUME, PRENUME);
91 SELECT * FROM AUTOR
92 WHERE NUME='Popescu' AND PRENUME='Radu';

```



ID_AUTOR	NUME	PRENUME	DATA_NASTERE
12	Popescu Radu	(null)	

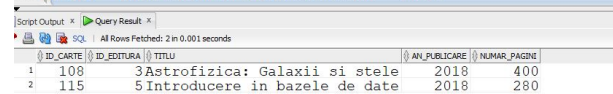
### 18.2 Cererea II

Să se afișeze cărțile publicate într-un anumit an.

```

93
94 --CEREREA II
95 --Sa se afiseze eficient cartile publicate intr-un anumit an;
96 CREATE INDEX IDX_AN_PUBLICARE ON CARTE (AN_PUBLICARE);
97 SELECT * FROM CARTE WHERE AN_PUBLICARE = 2018;

```



ID_CARTE	ID_EDITURA	TITLU	AN_PUBLICARE	NUMAR_PAGINI
108	3	Astrofizica: Galaxii si stele	2018	400
115	5	Introducere in bazele de date	2018	280