

# **Gestiunea studenților din România**

**Cataramă Antonia**

**Grupa 142**

## **1. Descrierea modelului real. Utilitatea modelului de date proiectat**

Modelul de date va gestiona informații legate de gestiunea studenților din România. Astfel, în cadrul bazei de date există mai mulți studenți care pot urma una sau mai multe specializări ce aparțin de o facultate. Mai multe facultăți aparțin de o universitate, se află la o singură locație și pot avea mai mulți îngrijitori. Studenții sunt coordonați de cel puțin un profesor în cadrul unui proiect sau mai multe, iar un profesor poate ține mai multe cursuri și un curs poate fi ținut de mai mulți profesori.

### **Reguli de funcționare**

Modelul de date respectă anumite reguli de funcționare.

- O universitate poate avea mai multe facultăți.
- O facultate aparține unei singure universități.
- O facultate are o singură locație.
- O locație găzduiește o singură facultate.
- O facultatea poate avea mai mulți îngrijitori.
- Un îngrijitor poate lucra la o singură facultate.
- O facultate poate avea mai multe specializări.
- O specializare poate aparține de o singură facultate.
- Un student poate fi la mai multe specializări.
- O specializare poate fi urmată de mai mulți studenți.
- Studentul realizează mai multe proiecte, are unul sau mai mulți profesori coordonatori, iar un profesor coordonează mai mulți studenți în cadrul mai

multor proiecte

- Un profesor poate ține mai multe cursuri.
- Un curs poate fi ținut de mai mulți profesori.

## **2. Constrângeri**

- O universitate poate avea mai multe facultăți.
- O facultate aparține unei singure universități.
- O facultate are o singură locație.
- O locație găzduiește o singură facultate.
- O facultatea poate avea mai mulți îngrijitori.
- Un îngrijitor poate lucra la o singură facultate.
- O facultate poate avea mai multe specializări.
- O specializare poate aparține de o singură facultate.
- Un student poate fi la mai multe specializări.
- O specializare poate fi urmată de mai mulți studenți.
- Un student realizează mai multe proiecte.
- Un student are unul sau mai mulți profesori coordonatori.
- Un profesor coordonează mai mulți studenți în cadrul mai multor proiecte.
- Un profesor poate ține mai multe cursuri.
- Un curs poate fi ținut de mai mulți profesori.

## **3. Entități**

Pentru modelul de date referitor la evidența studenților din România, structurile UNIVERSITATE, FACULTATE, LOCAȚIE, ÎNGRIJITOR, SPECIALIZARE, STUDENT, PROIECT, PROFESOR, CURS reprezintă entități.

Vom prezenta entitățile modelului de date, dând o descriere completă a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

**UNIVERSITATE** = instituție de învățământ superior cu mai multe facultăți și secții. Cheia primară a entității este *id\_universitate*.

**FACULTATE** = unitate didactică și administrativă în cadrul unei instituții de învățământ superior pentru pregătirea studenților într-un anumit domeniu de specialitate. Cheia primară a entității este *id\_facultate*.

**LOCAȚIE** = entitate care identifică locația unei facultăți. Cheia primară a entității este *id\_locatie*.

**ÎNGRIJITOR** = persoană fizică care se ocupă cu întreținerea curățeniei într-o facultate. Cheia primară a entității este *id\_ingrijitor*.<sup>[1]  
[S&P]</sup>

**SPECIALIZARE** = domeniu în cadrul unei facultăți. Cheia primară a entității este *id\_specializare*.

**STUDENT** = persoană fizică care urmează cursurile unui institut de învățământ superior. Cheia primară a entității este *id\_student*.

**PROIECT** = lucrare tehnică întocmită pe baza unei teme date. Cheia primară a entității este *id\_proiect*.

**PROFESOR** = persoană fizică cu o pregătire specială într-un anumit domeniu de activitate și care predă o materie de învățământ. Cheia primară a entității este *id\_profesor*.

**CURS** = predare a unei materii, făcută la o catedră de învățământ superior. Cheia primară a entității este *id\_curs*.

#### 4. Relații

Vom prezenta relațiile modelului de date, dând o descriere completă a fiecăreia. Pentru fiecare relație se va preciza cardinalitatea minimă și maximă.

*UNIVERSITATE\_are\_FACULTATE* = relație care leagă entitățile UNIVERSITATE și FACULTATE, reflectând legătura dintre acestea (ce facultate aparține de o

universitate). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

*FACULTATE\_are\_LOCAȚIE* = relație care leagă entitățile FACULTATE și LOCAȚIE, reflectând legătura dintre acestea (o facultate are o locație). Relația are cardinalitatea minimă și cardinalitatea maximă 1:1 (o facultate se află în exact o locație).

*ÎNGRIJITOR\_lucrează\_la\_FACULTATE* = relație care leagă entitățile ÎNGRIJITOR și FACULTATE, reflectând legătura dintre acestea (ce îngrijitor lucrează la o facultate). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

*FACULTATE\_are\_SPECIALIZARE* = relație care leagă entitățile FACULTATE și SPECIALIZARE, reflectând legătura dintre acestea (ce specializare aparține de o facultate). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

*STUDENT\_este\_la\_SPECIALIZARE* = relație care leagă entitățile STUDENT și SPECIALIZARE, reflectând legătura dintre acestea (un student este la o specializare). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă m:n.

*PROFESOR\_coordonează\_STUDENT\_la\_PROIECT* = relație de tip 3 ce leagă entitățile PROFESOR, STUDENT și PROIECT, reflectând ce profesor coordonează ce student la ce proiect. Denumirea acestei relații va fi *coordonează*.

*PROFESOR\_tine\_CURS* = relație care leagă entitățile PROFESOR și CURS, reflectând legătura dintre acestea (un profesor ține un curs). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă m:n.

## 5. Attribute

Entitatea **UNIVERSITATE** are ca attribute:

*id\_universitate* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei universități.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unei universități.

*nr\_facultati* = variabilă de tip întreg, de lungime maximă 5, care reprezintă numărul de facultăți al unei universități.

Entitatea **LOCAȚIE** are ca attribute:

*id\_locatie* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei locații.

*oras* = variabilă de tip caracter, de lungime maximă 50, care reprezintă orașul

în care se află o facultate.

*strada* = variabilă de tip caracter, de lungime maximă 50, care reprezintă strada la care se află o facultate.

*numar* = variabilă de tip întreg, de lungime maximă 5, care reprezintă numărul la care se află o facultate.

Entitatea **FACULTATE** are ca attribute:

*id\_facultate* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei facultăți.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unei facultăți.

*id\_locatie* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul locației unde se află facultatea. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul LOCAȚIE.

Entitatea **ÎNGRIJITOR** are ca attribute:

*id\_ingrijitor* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui îngrijitor.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unui îngrijitor.

*salariu* = variabilă de tip real, de lungime maximă 5, care reprezintă salariul unui îngrijitor.

*data\_angajare* = variabilă de tip dată calendaristică, care reprezintă data angajării.

*id\_facultate* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul facultății la care lucrează îngrijitorul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FACULTATE.

Entitatea **SPECIALIZARE** are ca attribute:

*id\_specializare* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei specializări.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unei specializări.

*durata* = variabilă de tip întreg, de lungime maximă 5, care reprezintă durata unei specializări.

*medie* = variabilă de tip real, de lungime maximă 5, care reprezintă media necesară pentru a urma o specializare.

*id\_facultate* = variabilă de tip întreg, de lungime maximă 5, care reprezintă

*id\_ul* facultății de care aparține specializarea. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FACULTATE.

Entitatea **STUDENT** are ca atribute:

*id\_student* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui student.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unui student.

*an* = variabilă de tip întreg, de lungime maximă 5, care reprezintă anul de studiu al unui student.

Entitatea **PROIECT** are ca atribute:

*id\_proiect* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui proiect.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unui proiect.

*data\_prezentare* = variabilă de tip calendaristică, care reprezintă data în care studentul va prezenta proiectul.

Entitatea **PROFESOR** are ca atribute:

*id\_profesor* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui profesor.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unui profesor.

Entitatea **CURS** are ca atribute:

*id\_curs* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui curs.

*nume* = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unui curs.

Relația **PROFESOR\_tine\_CURS** are ca atribute:

*id\_profesor* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui profesor. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PROFESOR.

*id\_curs* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui curs. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CURS.

Relația **STUDENT\_este\_la\_SPECIALIZARE** are ca attribute:

*id\_student* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui student. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul STUDENT.

*id\_specializare* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei specializări. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SPECIALIZARE.

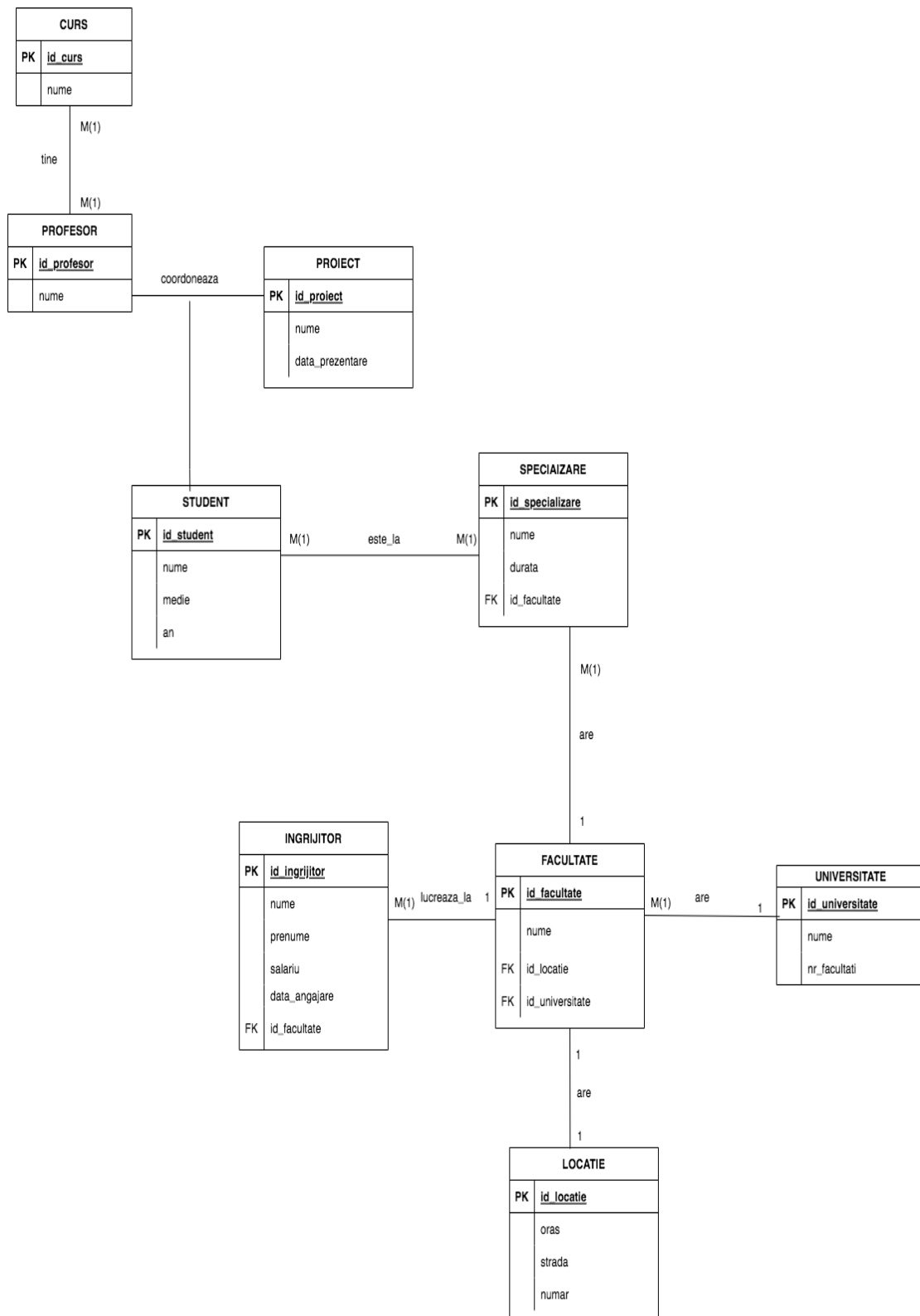
Relația **PROFESOR\_coordonează\_STUDENT\_la\_PROIECT** are ca attribute:

*id\_profesor* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui profesor. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PROFESOR.

*id\_student* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui student. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul STUDENT.

*id\_proiect* = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui proiect.

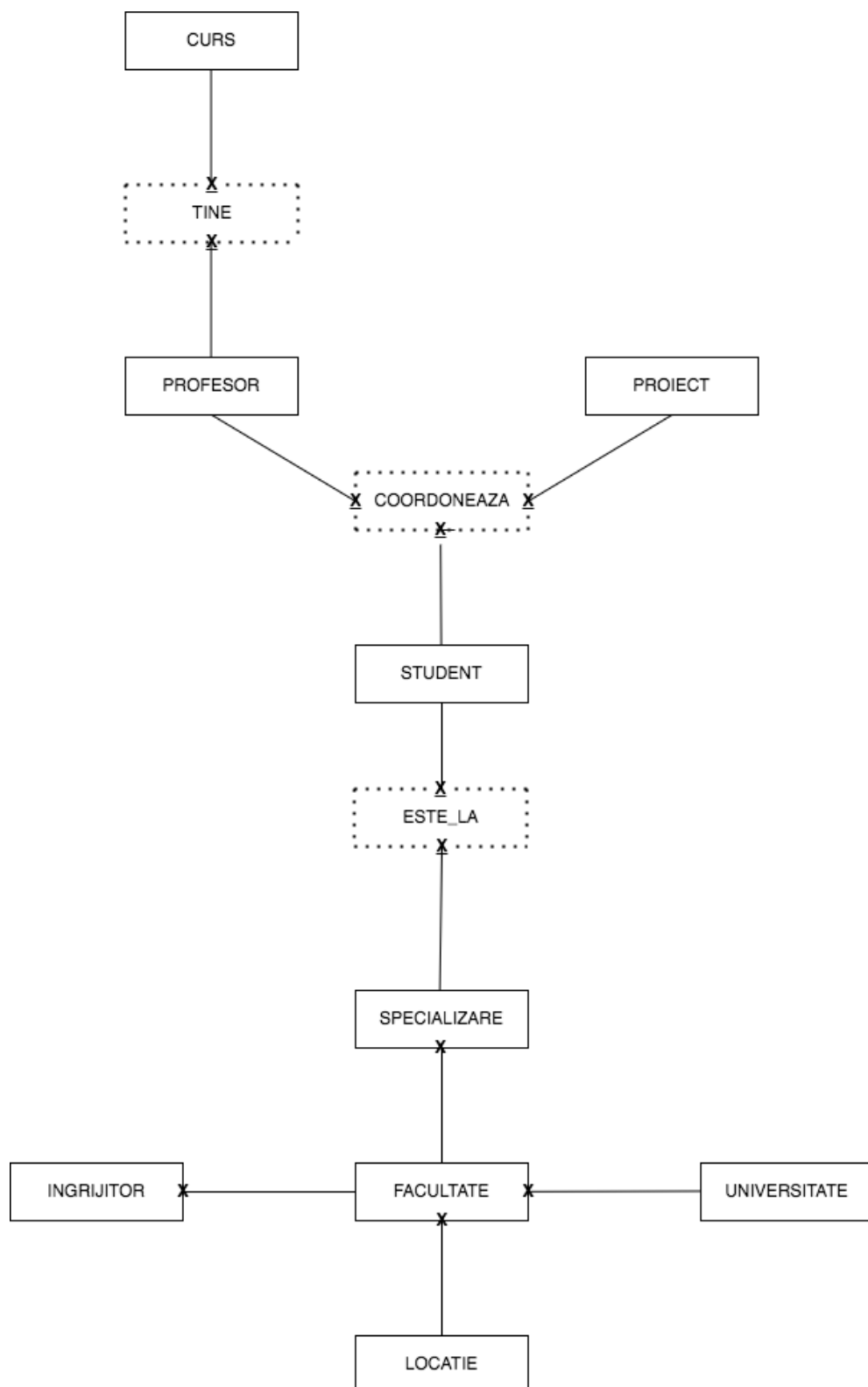
## 6. Diagrama entitate-relație



**Fig. 1.** Diagrama E/R.



## 7. Diagrama conceptuală



**Fig. 2.** Diagrama conceptuală.

## 8. Scheme relaționale

Schemele relaționale corespunzătoare diagramei conceptuale din figura 2 sunt următoarele:

UNIVERSITATE (id\_universitate#, nume, nr\_facultati)

LOCAȚIE (id\_locatie#, oras, strada, numar)

FACULTATE (id\_facultate#, id\_locatie, id\_universitate, nume)

ÎNGRIJITOR (id\_ingrijitor#, id\_facultate, nume, salariu, data\_angajare)

SPECIALIZARE (id\_specializare#, id\_facultate, nume, durata, medie)

STUDENT (id\_student#, nume, an)

PROIECT (id\_proiect#, nume, data\_prezentare)

PROFESOR (id\_profesor#, nume)

ȚINE (id\_tine#, id\_profesor, id\_curs)

ESTE\_LA (id\_este\_la#, id\_student, id\_specializare)

COORDONEAZĂ (id\_coordoneaza#, id\_profesor, id\_student, id\_proiect)

## 9. Realizarea normalizării (FN1-FN3)

### 9.1. Forma normală 1 (FN1)

O relație este în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă (atomică).

Pentru a exemplifica normalizarea, considerăm următorul exemplu:

STUDENT	PROIECT
S1	P1,P2
S2	P1,P3,P4

Exemplu non-FN1

STUDENT	PROIECT
S1	P1
S1	P2
S2	P1
S2	P3
S2	P4

Exemplu FN1

## 9.2. Forma normală 2 (FN2)

O relație  $R$  este în a doua formă normală dacă și numai dacă relația  $R$  este în FN1 și fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară.

FN2 interzice manifestarea unor dependențe funcționale parțiale în cadrul relației  $R$ .

Pentru a exemplifica normalizarea, considerăm următorul exemplu:

id_ingrijitor#	id_facultate#	ore	salariu
1	1	8	2000
2	3	9	2500
4	1	8	2000
5	2	10	3000
3	2	10	3000

Exemplu non-FN2

Avem următoarele dependențe:

$\{id\_ingrijitor\} \Rightarrow \{ore\}$  -  $id\_ingrijitor$  determină funcțional  $ore$

$\{id\_ingrijitor\#, id\_facultate\# \} \Rightarrow \{salariu\}$ .

$id\_ingrijitor\#$	$id\_facultate\#$	salariu
1	1	2000
2	3	2500
4	1	2000
5	2	3000
3	2	3000

$id\_ingrijitor\#$	ore
1	8
2	9
4	8
5	10
3	10

Exemplu FN2

### 9.3. Forma normală 3 (FN3)

Intuitiv, o relație  $R$  este în a treia formă normală dacă și numai dacă relația  $R$  este în FN2 și fiecare atribut care nu este cheie (nu participă la o cheie) depinde direct de cheia primară.

Pentru a exemplifica normalizarea, considerăm următorul exemplu:

$id\_specializare\#$	$id\_student\#$	an
----------------------	-----------------	----

1	2	1
1	3	3
2	2	1
3	1	2

Exemplu non-FN3

Pentru a aduce relația  $R$  în FN3 se aplică regula Casey-Delobel. Descompunem relația prin eliminarea dependențelor funcționale tranzitive în proiecțiile:

$R1(id\_student\#, an)$

$R2(id\_specializare\#, id\_student\#)$

id_student	an
2	1
3	3
1	2

id_specializare#	id_student
1	2
1	3
2	2
3	1

Exemplu FN3

## 10. Creare. Inserare

```
--UNIVERSITATE
```

```
create table universitate
```

```
(id_universitate number(5) constraint pk_universitate primary key,
```

```

nume varchar2(50),
nr_facultati number(5)
);

```

```

create sequence seq_universitate
increment by 1
start with 0
minvalue 0
maxvalue 5
nocycle;

```

```

insert into universitate values (seq_universitate.nextval, 'Universitatea din Bucuresti',
6);

```

```

insert into universitate values (seq_universitate.nextval, 'Universitatea Babes-Bolyai',
7);

```

```

insert into universitate values (seq_universitate.nextval, 'Universitatea Politehnica',
10);

```

```

insert into universitate values (seq_universitate.nextval, 'Universitatea „Alexandru
Ioan Cuza”', 9);

```

```

insert into universitate values (seq_universitate.nextval, 'Academia de Studii
Economice', 10);

```

```

commit;

```

	ID_UNIVERSITATE	NUME	NR_FACULTATI
1	1	Universitatea din Bucuresti	6
2	2	Universitatea Babes-Bolyai	7
3	3	Universitatea Politehnica	10
4	4	Universitatea „Alexandru Ioan Cuza”	9
5	5	Academia de Studii Economice	10

```

--LOCATIE

```

```
create table locatie
(id_locatie number(5) constraint pk_locatie primary key,
oras varchar2(50),
strada varchar2(50),
numar number(5)
);
```

```
create sequence seq_locatie
increment by 1
start with 0
minvalue 0
maxvalue 10
nocycle;
```

```
insert into locatie values (seq_locatie.nextval, 'Bucuresti', 'Strada Academiei', 14);
insert into locatie values (seq_locatie.nextval, 'Cluj', ' Strada Ploiești', 22);
insert into locatie values (seq_locatie.nextval, 'Iasi', 'Bulevardul Carol I', 11);
insert into locatie values (seq_locatie.nextval, 'Bucuresti', 'Piata Romana', 6);
insert into locatie values (seq_locatie.nextval, 'Bucuresti', 'Splaiul Independentei',
313);
insert into locatie values (seq_locatie.nextval, 'Bucuresti', ' Strada Edgar Quinet', 5);
insert into locatie values (seq_locatie.nextval, 'Bucuresti', 'Splaiul Independentei',
314);
insert into locatie values (seq_locatie.nextval, 'Bucuresti', 'Bulevardul Dacia', 41);
insert into locatie values (seq_locatie.nextval, 'Cluj', 'Strada Mihail Kogălniceanu', 1);
insert into locatie values (seq_locatie.nextval, 'Iasi', 'Bulevardul Carol I', 20);

commit;
```

	ID_LOCATIE	ORAS	STRADA	NUMAR
1	1	Bucuresti	Strada Academiei	14
2	2	Cluj	Strada Ploiesti	22
3	3	Iasi	Bulevardul Carol I	11
4	4	Bucuresti	Piata Romana	6
5	5	Bucuresti	Splaiul Independentei	313
6	6	Bucuresti	Strada Edgar Quinet	5
7	7	Bucuresti	Splaiul Independentei	314
8	8	Bucuresti	Bulevardul Dacia	41
9	9	Cluj	Strada Mihail Kogalniceanu	1
10	10	Iasi	Bulevardul Carol I	20

--FACULTATE

create table facultate

(id\_facultate number(5) constraint pk\_facultate primary key,

nume varchar2(50),

id\_locatie number(5),

constraint fk\_facultate1 foreign key (id\_locatie) references locatie(id\_locatie),

id\_universitate number(5),

constraint fk\_facultate2 foreign key (id\_universitate) references  
universitate(id\_universitate)

);

create sequence seq\_facultate

increment by 1

start with 0

minvalue 0

maxvalue 10

nocycle;

insert into facultate values (seq\_facultate.nextval, 'Facultatea de Matematica si  
Informatica', 1, 1);



```

insert into facultate values (seq_facultate.nextval, 'Facultatea de Matematica si
Informatica', 2, 2);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Litere', 3, 4);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Drept', 4, 5);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Energetica', 5, 3);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Litere', 6, 1);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Automatica si
Calculatoare', 7, 3);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Business si Turism',
8, 5);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Fizica', 9, 2);

insert into facultate values (seq_facultate.nextval, 'Facultatea de Biologie', 10, 4);

commit;

```

	ID_FACULTATE	NUME	ID_LOCATIE	ID_UNIVERSITATE
1	1	Facultatea de Matematica si Informatica	1	1
2	2	Facultatea de Matematica si Informatica	2	2
3	3	Facultatea de Litere	3	4
4	4	Facultatea de Drept	4	5
5	5	Facultatea de Energetica	5	3
6	6	Facultatea de Litere	6	1
7	7	Facultatea de Automatica si Calculatoare	7	3
8	8	Facultatea de Business si Turism	8	5
9	9	Facultatea de Fizica	9	2
10	10	Facultatea de Biologie	10	4

--INGRIJITOR

```
create table ingrijitor
```

```
(id_ingrijitor number(5) constraint pk_ingrijitor primary key,
```

```
nume varchar2(50),
```

```
prenume varchar2(50),
```

```
salariu number(5),
```

```
data_angajarii date,
```

```
id_facultate number(5),  
constraint fk_ingrijitor foreign key (id_facultate) references facultate(id_facultate)  
);
```

```
create sequence seq_ingrijitor  
increment by 1  
start with 0  
minvalue 0  
maxvalue 10  
nocycle;
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Argint', 'Cornel',  
2000,to_date('24.09.2001','dd/mm/yyyy'), 4);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Coca', 'Eugen',  
1500,to_date('24.09.1998','dd/mm/yyyy'), 1);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Mariut', 'Elena',  
1500,to_date('04.04.2001','dd/mm/yyyy'), 3);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Muresan', 'Teofil',  
2600,to_date('19.11.2022','dd/mm/yyyy'), 2);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Neamt', 'Liviu',  
1400,to_date('01.09.2022','dd/mm/yyyy'), 5);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Ciobanu', 'Ioana',  
1300,to_date('01.08.2021','dd/mm/yyyy'), 5);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Ursu', 'Camelia',  
2000,to_date('01.08.2003','dd/mm/yyyy'), 7);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Paduraru', 'Ion',  
2300,to_date('08.08.2003','dd/mm/yyyy'), 1);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Mitronici', 'Tuliana',  
1200,to_date('12.03.2010','dd/mm/yyyy'), 1);
```

```
insert into ingrijitor values (seq_ingrijitor.nextval, 'Mormoceea', 'Dorina',  
2000,to_date('01.01.1998','dd/mm/yyyy'), 1);
```

commit;

	ID_INGRIJITOR	NUME	PRENUME	SALARIU	DATA_ANGAJARII	ID_FACULTATE
1	1	Argint	Cornel	2000	24-SEP-01	4
2	2	Coca	Eugen	1500	24-SEP-98	1
3	3	Mariut	Elena	1500	04-APR-01	3
4	4	Muresan	Teofil	2600	19-NOV-22	2
5	5	Neamt	Liviu	1400	01-SEP-22	5
6	6	Ciobanu	Ioana	1300	01-AUG-21	5
7	7	Ursu	Camelia	2000	01-AUG-03	7
8	8	Paduraru	Ion	2300	08-AUG-03	1
9	9	Mitronici	Iuliana	1200	12-MAR-10	1
10	10	Mormocea	Dorina	2000	01-JAN-98	1

--SPECIALIZARE

create table specializare

(id\_specializare number(5) constraint pk\_specializare primary key,

nume varchar2(50),

durata number(5),

id\_facultate number(5),

constraint fk\_specializare foreign key (id\_facultate) references facultate(id\_facultate)

);

create sequence seq\_specializare

increment by 1

start with 0

minvalue 0

maxvalue 10

nocycle;

insert into specializare values (seq\_specializare.nextval, 'Matematica', 3, 1);

insert into specializare values (seq\_specializare.nextval, 'Informatica', 3, 1);

```
insert into specializare values (seq_specializare.nextval, 'Limba si literatura romana',
3, 3);
```

```
insert into specializare values (seq_specializare.nextval, 'Drept', 4, 4);
```

```
insert into specializare values (seq_specializare.nextval, 'Biochimie', 3, 10);
```

```
insert into specializare values (seq_specializare.nextval, 'Matematica', 3, 2);
```

```
insert into specializare values (seq_specializare.nextval, 'Limba si literatura romana',
3, 6);
```

```
insert into specializare values (seq_specializare.nextval, 'Ingineria sistemelor', 4, 7);
```

```
insert into specializare values (seq_specializare.nextval, 'Fizica', 3, 9);
```

```
insert into specializare values (seq_specializare.nextval, 'Ingineria sistemelor
electroenergetice', 4, 5);
```

```
commit;
```

	ID_SPECIALIZARE	NUME	DURATA	ID_FACULTATE
1	1	Matematica	3	1
2	2	Informatica	3	1
3	3	Limba si literatura romana	3	3
4	4	Drept	4	4
5	5	Biochimie	3	10
6	6	Matematica	3	2
7	7	Limba si literatura romana	3	6
8	8	Ingineria sistemelor	4	7
9	9	Fizica	3	9
10	10	Ingineria sistemelor electroenergetice	4	5

```
--STUDENT
```

```
create table student
```

```
(id_student number(5) constraint pk_student primary key,
```

```
nume varchar2(50),
```

```
medie number(3, 2),
```

```
an number(5)
```

```
);
```

```
create sequence seq_student
```

```
increment by 1
```

```
start with 0
```

```
minvalue 0
```

```
maxvalue 10
```

```
nocycle;
```

```
insert into student values (seq_student.nextval, 'Teodoro Emil', 9.2, 2);
```

```
insert into student values (seq_student.nextval, 'Stefanescu Florian', 9.5, 1);
```

```
insert into student values (seq_student.nextval, 'Stanescu Constantin', 8.6, 4);
```

```
insert into student values (seq_student.nextval, 'Popa Valentin', 9.9, 3);
```

```
insert into student values (seq_student.nextval, 'Ofrim Dragos', 8.3, 3);
```

```
insert into student values (seq_student.nextval, 'Nicolae George', 9, 3);
```

```
insert into student values (seq_student.nextval, 'Munteanu Toader', 7.9, 2);
```

```
insert into student values (seq_student.nextval, 'Munteanu Calin', 9.9, 1);
```

```
insert into student values (seq_student.nextval, 'Cobzaru Alina', 8.9, 1);
```

```
insert into student values (seq_student.nextval, 'Voicu Aida', 7.8, 4);
```

```
commit;
```

	ID_STUDENT	NUME	MEDIE	AN
1	1	Teodoro Emil	9.2	2
2	2	Stefanescu Florian	9.5	1
3	3	Stanescu Constantin	8.6	4
4	4	Popa Valentin	9.9	3
5	5	Ofrim Dragos	8.3	3
6	6	Nicolae George	9	3
7	7	Munteanu Toader	7.9	2
8	8	Munteanu Calin	9.9	1
9	9	Cobzaru Alina	8.9	1
10	10	Voicu Aida	7.8	4

--PROIECT

create table proiect

(id\_proiect number(5) constraint pk\_proiect primary key,

nume varchar2(50),

data\_prezentare date

);

create sequence seq\_proiect

increment by 1

start with 0

minvalue 0

maxvalue 5

nocycle;

insert into proiect values (seq\_proiect.nextval, 'Proiect  
1',to\_date('29.01.2022','dd/mm/yyyy'));

insert into proiect values (seq\_proiect.nextval, 'Proiect  
2',to\_date('03.03.2022','dd/mm/yyyy'));

insert into proiect values (seq\_proiect.nextval, 'Proiect  
3',to\_date('24.01.2022','dd/mm/yyyy'));

insert into proiect values (seq\_proiect.nextval, 'Proiect  
4',to\_date('24.01.2022','dd/mm/yyyy'));

insert into proiect values (seq\_proiect.nextval, 'Proiect  
5',to\_date('24.01.2022','dd/mm/yyyy'));

commit;

	ID_PROIECT	NUME	DATA_PREZENTARE
1	1	Proiect 1	29-JAN-22
2	2	Proiect 2	03-MAR-22
3	3	Proiect 3	24-JAN-22
4	4	Proiect 4	24-JAN-22
5	5	Proiect 5	24-JAN-22

--PROFESOR

create table profesor

(id\_profesor number(5) constraint pk\_profesor primary key,

nume varchar2(50)

);

create sequence seq\_profesor

increment by 1

start with 0

minvalue 0

maxvalue 10

nocycle;

insert into profesor values (seq\_profesor.nextval, 'Dumitrescu Dumitru');

insert into profesor values (seq\_profesor.nextval, 'Petrescu Maria');

insert into profesor values (seq\_profesor.nextval, 'Negrescu Alexandra');

insert into profesor values (seq\_profesor.nextval, 'Vladimirescu Vlad');

insert into profesor values (seq\_profesor.nextval, 'Iliescu Ilie');

insert into profesor values (seq\_profesor.nextval, 'Lungu Andrei');

insert into profesor values (seq\_profesor.nextval, 'Badea Ion');

insert into profesor values (seq\_profesor.nextval, 'Cernat Mihai');

insert into profesor values (seq\_profesor.nextval, 'Ivanov Virginia');

```
insert into profesor values (seq_profesor.nextval, 'Nicoara Tania');
```

```
commit;
```

ID_PROFESOR	NUME
1	Dumitrescu Dumitru
2	Petrescu Maria
3	Negrescu Alexandra
4	Vladimirescu Vlad
5	Iliescu Ilie
6	Lungu Andrei
7	Badea Ion
8	Cernat Mihai
9	Ivanov Virginia
10	Nicoara Tania

```
--CURS
```

```
create table curs
```

```
(id_curs number(5) constraint pk_curs primary key,
```

```
nume varchar2(50)
```

```
);
```

```
create sequence seq_curs
```

```
increment by 1
```

```
start with 0
```

```
minvalue 0
```

```
maxvalue 10
```

```
nocycle;
```

```
insert into curs values (seq_curs.nextval, 'Programare Orientata pe Obiecte');
```

```
insert into curs values (seq_curs.nextval, 'Algebra');
```

```
insert into curs values (seq_curs.nextval, 'Literatura universala');
```



```

insert into curs values (seq_curs.nextval, 'Biologie celulara');
insert into curs values (seq_curs.nextval, 'Electricitate');
insert into curs values (seq_curs.nextval, 'Sisteme de operare');
insert into curs values (seq_curs.nextval, 'Limba si literatura romana');
insert into curs values (seq_curs.nextval, 'Microeconomie');
insert into curs values (seq_curs.nextval, 'Drept civil');
insert into curs values (seq_curs.nextval, 'Analiza');

```

```
commit;
```

ID_CURS	NUME
1	1 Programare Orientata pe Obiecte
2	2 Algebra
3	3 Literatura universala
4	4 Biologie celulara
5	5 Electricitate
6	6 Sisteme de operare
7	7 Limba si literatura romana
8	8 Microeconomie
9	9 Drept civil
10	10 Analiza

```
--TINE
```

```
create table tine
```

```
(id_tine number(5) constraint pk_tine primary key,
```

```
id_curs number(5),
```

```
id_profesor number(5),
```

```
constraint fk_tine_curs foreign key (id_curs) references curs(id_curs),
```

```
constraint fk_tine_profesor foreign key (id_profesor) references profesor(id_profesor)
```

```
);
```

```
create sequence seq_tine
```

increment by 1

start with 0

minvalue 0

maxvalue 10

nocycle;

insert into tine values (seq\_tine.nextval, 1, 3);

insert into tine values (seq\_tine.nextval, 2, 10);

insert into tine values (seq\_tine.nextval, 3, 6);

insert into tine values (seq\_tine.nextval, 4, 4);

insert into tine values (seq\_tine.nextval, 5, 7);

insert into tine values (seq\_tine.nextval, 6, 9);

insert into tine values (seq\_tine.nextval, 7, 2);

insert into tine values (seq\_tine.nextval, 8, 8);

insert into tine values (seq\_tine.nextval, 9, 5);

insert into tine values (seq\_tine.nextval, 10, 1);

commit;

	ID_TINE	ID_CURS	ID_PROFESOR
1	1	1	3
2	2	2	10
3	3	3	6
4	4	4	4
5	5	5	7
6	6	6	9
7	7	7	2
8	8	8	8
9	9	9	5
10	10	10	1

--COORDONEAZA

```
create table coordoneaza  
  
(id_coordoneaza number(5) constraint pk_coordoneaza primary key,  
  
id_profesor number(5),  
  
id_student number(5),  
  
id_proiect number(5),  
  
constraint fk_coordoneaza_profesor foreign key (id_profesor) references  
profesor(id_profesor),  
  
constraint fk_coordoneaza_student foreign key (id_student) references  
student(id_student),  
  
constraint fk_coordoneaza_proiect foreign key (id_proiect) references  
proiect(id_proiect)  
  
);
```

```
create sequence seq_coordoneaza  
  
increment by 1  
  
start with 0  
  
minvalue 0  
  
maxvalue 10  
  
nocycle;
```

```
insert into coordoneaza values (seq_coordoneaza.nextval, 1, 3, 5);  
insert into coordoneaza values (seq_coordoneaza.nextval, 2, 10, 1);  
insert into coordoneaza values (seq_coordoneaza.nextval, 3, 6, 2);  
insert into coordoneaza values (seq_coordoneaza.nextval, 4, 4, 5);  
insert into coordoneaza values (seq_coordoneaza.nextval, 5, 7, 5);  
insert into coordoneaza values (seq_coordoneaza.nextval, 6, 9, 3);  
insert into coordoneaza values (seq_coordoneaza.nextval, 7, 2, 2);  
insert into coordoneaza values (seq_coordoneaza.nextval, 8, 8, 1);
```

insert into coordoneaza values (seq\_coordoneaza.nextval, 9, 5, 1);

insert into coordoneaza values (seq\_coordoneaza.nextval, 10, 1, 1);

commit;

	ID_COORDONEAZA	ID_PROFESOR	ID_STUDENT	ID_PROIECT
1	1	1	3	5
2	2	2	10	1
3	3	3	6	2
4	4	4	4	5
5	5	5	7	5
6	6	6	9	3
7	7	7	2	2
8	8	8	8	1
9	9	9	5	1
10	10	10	1	1

--ESTE\_LA

create table este\_la

(id\_este\_la number(5) constraint pk\_este\_la primary key,

id\_student number(5),

id\_specializare number(5),

constraint fk\_este\_la\_student foreign key (id\_student) references student(id\_student),

constraint fk\_este\_la\_specializare foreign key (id\_specializare) references  
specializare(id\_specializare)

);

create sequence seq\_este\_la

increment by 1

start with 0

minvalue 0

maxvalue 10

nocycle;

```

insert into este_la values (seq_este_la.nextval, 3, 1);
insert into este_la values (seq_este_la.nextval, 10, 7);
insert into este_la values (seq_este_la.nextval, 6, 2);
insert into este_la values (seq_este_la.nextval, 4, 5);
insert into este_la values (seq_este_la.nextval, 7, 4);
insert into este_la values (seq_este_la.nextval, 9, 3);
insert into este_la values (seq_este_la.nextval, 2, 10);
insert into este_la values (seq_este_la.nextval, 8, 4);
insert into este_la values (seq_este_la.nextval, 5, 8);
insert into este_la values (seq_este_la.nextval, 1, 1);

```

commit;

	ID_ESTE_LA	ID_STUDENT	ID_SPECIALIZARE
1	1	3	1
2	2	10	7
3	3	6	2
4	4	4	5
5	5	7	4
6	6	9	3
7	7	2	10
8	8	8	4
9	9	5	8
10	10	1	1

## 11. Cereri SQL

--1) Sa se afiseze numele si prenumele, data angajarii, facultatea, locatia facultatii si universitatea de care apartine ingrijitorilor care s-au angajat intre 2-09-2000 si 2-09-2005.

--join pe 4 tabele, o functie pe siruri de caractere, filtrare la nivel de linii (where), 2 functii pe date calendaristice, ordonare

```
select concat(i.ume,' ') || i.prenume "Ingrijitor", i.data_angajarii, f.ume "Facultatea",
concat(l.oras,' ') || concat(l.strada,' ') || l.numar "Locatie", u.ume "Universitate"
```

```
from ingrijitor i join facultate f on (i.id_facultate = f.id_facultate)
```

```
join locatie l on (f.id_locatie = l.id_locatie)
```

```
join universitate u on (f.id_universitate = u.id_universitate)
```

```
where (i.data_angajarii > to_date('02-09-2000','dd-mm-yyyy') and i.data_angajarii <
add_months('02-SEP-2000',60))
```

```
order by i.data_angajarii desc;
```

Ingrijitor	DATA_ANGAJARII	Facultatea	Locatie	Universitate
1 Paduraru Ion	08-AUG-03	Facultatea de Matematica si Informatica	Bucuresti, Strada Academiei, 14	Universitatea din Bucuresti
2 Ursu Camelia	01-AUG-03	Facultatea de Automatica si Calculatoare	Bucuresti, Splaiul Independentei, 314	Universitatea Politehnica
3 Argint Cornel	24-SEP-01	Facultatea de Drept	Bucuresti, Piata Romana, 6	Academia de Studii Economice
4 Mariut Elena	04-APR-01	Facultatea de Litere	Iasi, Bulevardul Carol I, 11	Universitatea „Alexandru Ioan Cuza”

--2) Sa se afiseze id-ul si numele specializarii care se afla in cadrul facultatii care este la locatia cu id-ul maxim.

--cerere nesincronizata pe 3 tabele

```
select id_specializare, nume
```

```
from specializare s
```

```
where s.id_facultate in
```

```
(select id_facultate
```

```
from facultate f
```

```
where f.id_locatie =
```

```
(select max(id_locatie)
```

```
from locatie l
```

```
)
```

```
);
```

ID_SPECIALIZARE	NUME
1	5 Biochimie

--3) Sa se afiseze numele, prenumele, data angajarii, salariul actual, salariul marit cu 20% pentru ingrijitorii angajati in 2021 si 2022 si salariul miscorat cu 5% pentru ingrijitorii angajati in 2001 si 1998.

--nvl, decode, case, o functie pe siruri de caractere

```
select nume, prenume, data_angajarii, salariu,
nvl((decode(to_char(data_angajarii, 'yyyy'), '2021', salariu * 1.2, '2022', salariu* 1.2)),
salariu) "Salariul marit",
case to_char(data_angajarii, 'yyyy')
when '2001' then salariu * 0.95
when '1998' then salariu * 0.95
else salariu
end "Salariul micsorat"
from ingrijitor;
```

	NUME	PRENUME	DATA_ANGAJARII	SALARIU	Salariul marit	Salariul micsorat
1	Argint	Cornel	24-SEP-01	2000	2000	1900
2	Coca	Eugen	24-SEP-98	1500	1500	1425
3	Mariut	Elena	04-APR-01	1500	1500	1425
4	Muresan	Teofil	19-NOV-22	2600	3120	2600
5	Neamt	Liviu	01-SEP-22	1400	1680	1400
6	Ciobanu	Ioana	01-AUG-21	1300	1560	1300
7	Ursu	Camelia	01-AUG-03	2000	2000	2000
8	Padurararu	Ion	08-AUG-03	2300	2300	2300
9	Mitronici	Iuliana	12-MAR-10	1200	1200	1200
10	Mormoceea	Dorina	01-JAN-98	2000	2000	1900

--4) Sa se afiseze numele si media duratei specializarilor fiecarei facultati.

--clauza with, grupari de date, functii grup, filtrare la nivel de grupuri

```
with durata_medie as (select f.nume, avg(s.durata) "Durata Medie"
from specializare s join facultate f on(s.id_facultate = f.id_facultate)
group by f.nume)
```

having avg(s.durata) >= 3)

select\*

from durata\_medie;

	NUME	Durata Medie
1	Facultatea de Matematica si Informatica	3
2	Facultatea de Biologie	3
3	Facultatea de Drept	4
4	Facultatea de Fizica	3
5	Facultatea de Litere	3
6	Facultatea de Energetica	4
7	Facultatea de Automatica si Calculatoare	4

--5) Sa se afiseze facultatile care contin sirul "MAT", salariul ingrijitorului si orasul in care se afla.

-- cerere sincronizata pe 3 tabele, functie pe siruri de caractere

select id\_facultate, f.numa,

(select round(avg(salariu),2)

from ingrijitor i

where f.id\_facultate = i.id\_facultate) "Salariu Mediu",

(select oras

from locatie l

where l.id\_locatie = f.id\_locatie) "Localitate"

from facultate f

where upper(f.numa) like '%MAT%';

	ID_FACULTATE	NUME	Salariu Mediu	Localitate
1	1	Facultatea de Matematica si Informatica	1750	Bucuresti
2	2	Facultatea de Matematica si Informatica	2600	Cluj
3	7	Facultatea de Automatica si Calculatoare	2000	Bucuresti

## 12. Operații de actualizare și suprimare a datelor folosind subcereri



--1) Actualizare salarii ingrijitori cu salariul mediu daca salariul lor este mai mare decat salariul mediu.

update ingrijitor

set (salariu) = (select avg(salariu)

from ingrijitor)

where salariu > (select avg(salariu)

from ingrijitor);

rollback;

select\* from ingrijitor;

	ID_INGRIJITOR	NUME	PRENUME	SALARIU	DATA_ANGAJARII	ID_FACULTATE
1	1	Argint	Cornel	1780	24-SEP-01	4
2	2	Coca	Eugen	1500	24-SEP-98	1
3	3	Mariut	Elena	1500	04-APR-01	3
4	4	Muresan	Teofil	1780	19-NOV-22	2
5	5	Neamt	Liviu	1400	01-SEP-22	5
6	6	Ciobanu	Ioana	1300	01-AUG-21	5
7	7	Ursu	Camelia	1780	01-AUG-03	7
8	8	Paduraru	Ion	1780	08-AUG-03	1
9	9	Mitronici	Iuliana	1200	12-MAR-10	1
10	10	Mormoceea	Dorina	1780	01-JAN-98	1

--2) Sa se stearga universitatile care au numarul de facultati mai mare decat media.

--tabela facultate depinde de tabela universitate, asadar nu puteam sterge nimic

--am sters constrangerea care facea ca facultate sa depinda de universitate

alter table facultate

drop constraint fk\_facultate2;

--si am adaugat constrangerea asta, astfel incat atunci cand sterg o universitate, id\_universitate din facultate va fi null pentru universitatea stearta

alter table facultate

```

add constraint fk_facultate2
foreign key (id_universitate)
references universitate(id_universitate)
on delete set null;

```

```

delete from universitate
where nr_facultati >
(select avg(nr_facultati)
from universitate );

```

```
rollback;
```

```
select* from universitate;
```

	ID_UNIVERSITATE	NUME	NR_FACULTATI
1	1	Universitatea din Bucuresti	6
2	2	Universitatea Babes-Bolyai	7

--3) Actualizare durata specializare din 3 ani in 4.

```
update specializare
```

```
set (durata) = 4
```

```
where durata = 3;
```

```
rollback;
```

```
select* from specializare;
```

	ID_SPECIALIZARE	NUME	DURATA	ID_FACULTATE
1	1	Matematica	4	1
2	2	Informatica	4	1
3	3	Limba si literatura romana	4	3
4	4	Drept	4	4
5	5	Biochimie	4	10
6	6	Matematica	4	2
7	7	Limba si literatura romana	4	6
8	8	Ingineria sistemelor	4	7
9	9	Fizica	4	9
10	10	Ingineria sistemelor electroenergetice	4	5

### 13.Crearea unei secvențe

```
create sequence seq_universitate
increment by 1
start with 0
minvalue 0
maxvalue 5
nocycle;
```

### 14.Crearea unei vizualizări compuse

```
create view angajati as
select i.num, i.prenume, i.salariu
from ingrijitor i join facultate f on (i.id_facultate = f.id_facultate)
where i.salariu > 2000 and f.num = 'Facultatea de Matematica si Informatica';

--operatie lmd permisa
select * from angajati;
```

	NUME	PRENUME	SALARIU
1	Muresan	Teofil	2600
2	Paduraru	Ion	2300

--operatie lmd nepermisa

insert into angajati values('nume', 'prenume', 1000);