

# Gestiunea studenților din România

Cataramă Antonia

Grupa 242

## 1. Utilitatea bazei de date

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.

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]</sup>

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

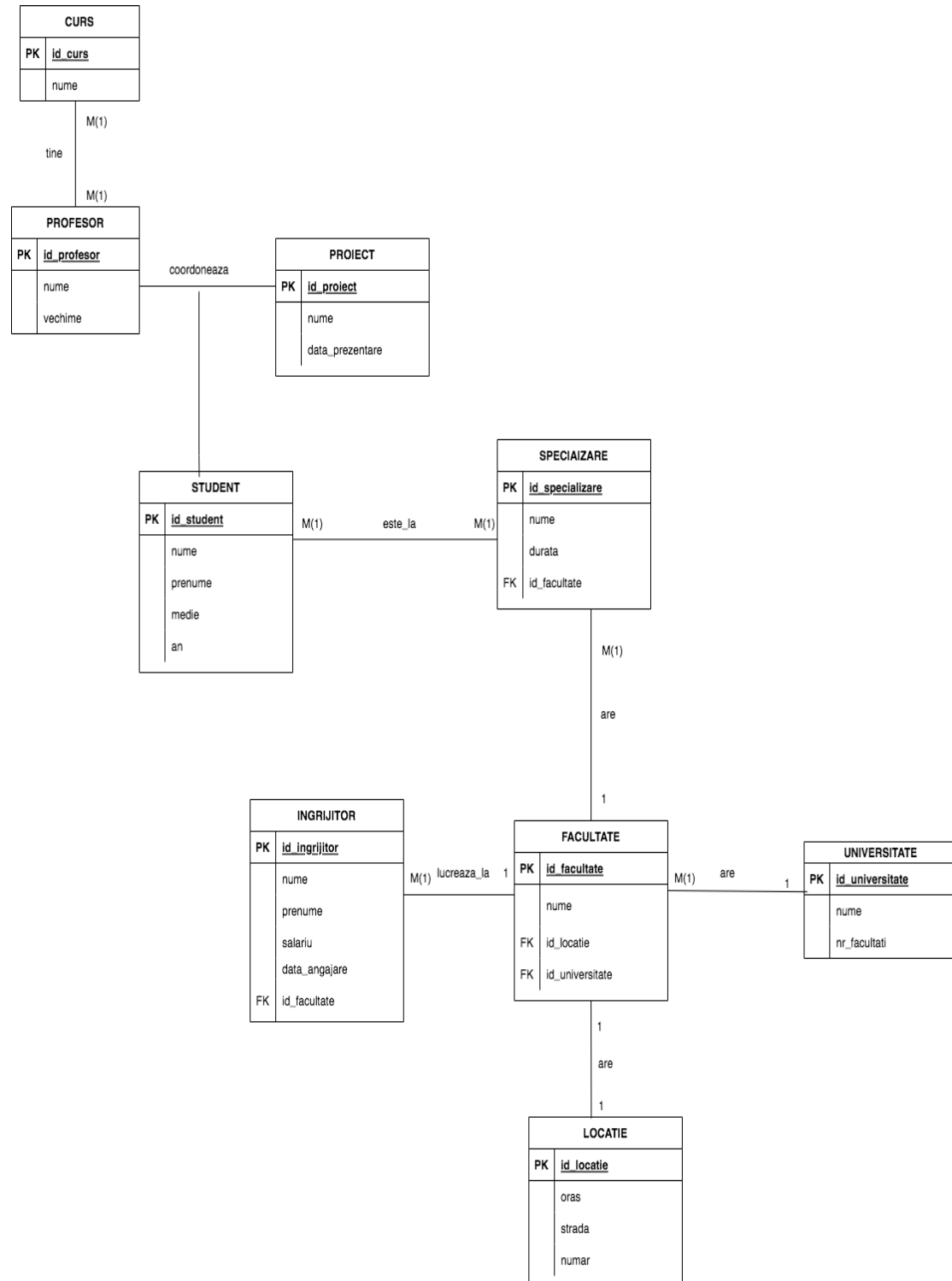
**STUDENT** = persoană fizică care urmează cursurile unui institut de învățământ superior. Cheia primară a entității este *id\_student*.

**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*.

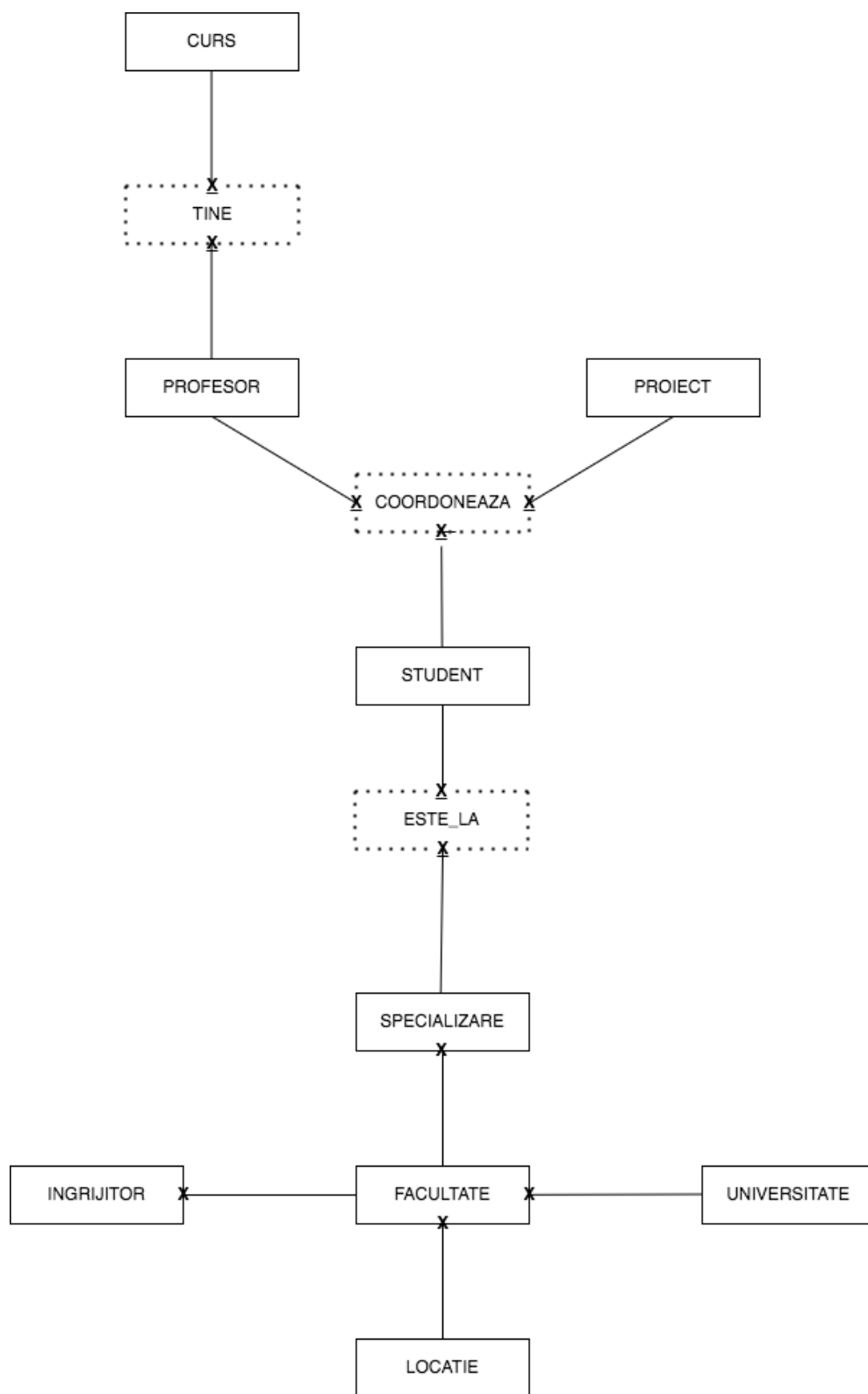
**CURS** = predare a unei materii, făcută la o catedră de învățământ superior. Cheia primară a entității este *id\_curs*.

## 2. Diagrama entitate-relație



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

### 3. Diagrama conceptuală



**Fig. 2.** Diagrama conceptuală.

#### 4-5. 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 Ploie?ti', 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  
Kog?lniceanu', 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 Ploieşti	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', 'Iuliana',  
1200,to\_date('12.03.2010','dd/mm/yyyy'), 1);

insert into ingrijitor values (seq\_ingrijitor.nextval, 'Mormocea', 'Dorina',  
2000,to\_date('01.01.1998','dd/mm/yyyy'), 1);

insert into ingrijitor values (11, 'Ursu', 'Alina',  
2500,to\_date('25.02.2001','dd/mm/yyyy'), 4);

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),

prenume 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),

vechime number(5)

);

create sequence seq\_profesor

increment by 1

start with 0

minvalue 0

maxvalue 10

nocycle;

insert into profesor values (seq\_profesor.nextval, 'Dumitrescu Dumitru', 2);

insert into profesor values (seq\_profesor.nextval, 'Petrescu Maria', 4);

insert into profesor values (seq\_profesor.nextval, 'Negrescu Alexandra', 13);

insert into profesor values (seq\_profesor.nextval, 'Vladimirescu Vlad', 12);

insert into profesor values (seq\_profesor.nextval, 'Iliescu Ilie', 5);

insert into profesor values (seq\_profesor.nextval, 'Lungu Andrei', 10);

insert into profesor values (seq\_profesor.nextval, 'Badea Ion', 9);

```

insert into profesor values (seq_profesor.nextval, 'Cernat Mihai', 12);

insert into profesor values (seq_profesor.nextval, 'Ivanov Virginia', 20);

insert into profesor values (seq_profesor.nextval, 'Nicoara Tania', 2);

commit;

```

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

**6. Afisati studentii care sunt la o specializare cu id-ul dat si sunt coordonati de profesori care au cel putin 10 ani vechime.**

create or replace procedure c6(cod\_specializare  
specializare.id\_specializare%type) is

type tablou\_imbricat is table of student.id\_student%type;

v\_student tablou\_imbricat := tablou\_imbricat();

type vector is varray(30) of profesor.id\_profesor%type;

v\_profesor vector := vector();

cursor c is

select id\_student, id\_profesor

from coordoneaza;

n number(3);

begin

select count(\*)

into n

```

from profesor;

for i in 1..n loop
    v_profesor.extend;
end loop;

select id_student
bulk collect into v_student
from student s join este_la e using (id_student)
where e.id_specializare = cod_specializare;

select id_profesor
bulk collect into v_profesor
from profesor
where vechime >= 10;

for i in c loop
    for j in 1..v_student.count loop
        if v_student(j) = i.id_student then
            for k in 1..v_profesor.count loop
                if v_profesor(k) = i.id_profesor then
                    DBMS_OUTPUT.PUT_LINE('Studentul cu id-ul ' ||
i.id_student || ' este coordonat de profesorul cu id-ul ' || i.id_profesor);
                end if;
            end loop;
        end loop;
    end loop;

```

```

        end if;

    end loop;

end loop;

end c6;

begin

    c6(2);

end;

Studentul cu id-ul 5 este coordonat de profesorul cu id-ul 9

Studentul cu id-ul 6 este coordonat de profesorul cu id-ul 3

```

**7. Pentru fiecare facultate, afisati numele si lista ingrijitorilor (nume si salariu) care se ocupa de aceasta. Tratati cazul in care nu exista ingrijitori.**

```

create or replace procedure c7 is

    type tablou_imbricat is table of facultate.id_facultate%type;

    v_id_facultate tablou_imbricat := tablou_imbricat();

    v_numa_facultate facultate.numa%type;

    v_numa_ingrijitor ingrijitor.numa%type;

    v_prename_ingrijitor ingrijitor.prename%type;

    v_salariu ingrijitor.salariu%type;

    contor number(3);

```

```

cursor c(cod_facultate facultate.id_facultate%type) is

    select i.nume, i.prenume, i.salariu

    from ingrijitor i, facultate f

    where i.id_facultate = f.id_facultate and f.id_facultate = cod_facultate;

begin

    select id_facultate

    bulk collect into v_id_facultate

    from facultate;

    for i in v_id_facultate.first..v_id_facultate.last loop

        select nume

        into v_nume_facultate

        from facultate

        where id_facultate = v_id_facultate(i);

        contor := 0;

        DBMS_OUTPUT.PUT_LINE(v_nume_facultate || ':');

        open c(v_id_facultate(i));

        loop

            fetch c into v_nume_ingrijitor, v_prenume_ingrijitor, v_salariu;

            exit when c%NOTFOUND;

            contor := contor + 1;

            DBMS_OUTPUT.PUT_LINE('      ' || v_nume_ingrijitor || ' ' ||
v_prenume_ingrijitor || ' ' || v_salariu);

```



```

end loop;

close c;

if contor = 0 then

    DBMS_OUTPUT.PUT_LINE('    Nu exista ingrijitori.');
```

end if;

```

    DBMS_OUTPUT.PUT_LINE('-----
-----');

end loop;

end c7;

begin

    c7;

end;

Facultatea de Matematica si Informatica:
    Coca Eugen,1500
    Paduraru Ion,2300
    Mitronici Iuliana,1200
    Mormocea Dorina,2000
-----
Facultatea de Matematica si Informatica:
    Muresan Teofil,2600
-----
Facultatea de Litere:
    Mariut Elena,1500
-----
Facultatea de Drept:
    Argint Cornel,2000
-----
Facultatea de Energetica:
    Neamt Liviu,1400
    Ciobanu Ioana,1300
-----
Facultatea de Litere:
    Nu exista ingrijitori.
-----
Facultatea de Automatica si Calculatoare:
    Ursu Camelia,2000
-----
Facultatea de Business si Turism:
    Nu exista ingrijitori.
-----
Facultatea de Fizica:
    Nu exista ingrijitori.
-----
Facultatea de Biologie:
    Nu exista ingrijitori.
-----

```

**8. Afisati universitatea de care apartine facultatea unde lucreaza un ingrijitor cu numele dat ca parametru.**

```
create or replace function c8(ume_ingrijitor ingrijitor.ume%type)

    return universitate.ume%type

    is v_universitate universitate.ume%type;

begin

    select u.ume

    into v_universitate

    from universitate u join facultate f using (id_universitate)

        join ingrijitor i using (id_facultate)

    where i.ume = ume_ingrijitor;

    return v_universitate;

exception

    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('Nu exista ingrijitori cu numele dat.');
```

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista ingrijitori cu numele dat.');

```
    when too_many_rows then

        DBMS_OUTPUT.PUT_LINE('Exista prea multi ingrijitori cu numele dat.');
```

RAISE\_APPLICATION\_ERROR(-20001, 'Exista prea multi ingrijitori cu numele dat.');

```
    when others then

        DBMS_OUTPUT.PUT_LINE('Alta eroare.');
```

RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare.');

```
end c8;
```

```
begin
```

```
    DBMS_OUTPUT.PUT_LINE(c8('Mormoceea')); -- fara erori
```

```
end;
```

```
begin
```

```
    DBMS_OUTPUT.PUT_LINE(c8('Gheorghe')); --no data found
```

```
end;
```

```
begin
```

```
    DBMS_OUTPUT.PUT_LINE(c8('Ursu')); -- too many rows
```

```
end;
```

```
Universitatea din Bucuresti  
  
Nu exista ingrijitori cu numele dat.  
  
Exista prea multi ingrijitori cu numele dat.
```

## **9. Afisati orasul facultatii unde merge studentul cu nume dat ca parametru.**

```
create or replace procedure c9 (v_student student.nume%type)is
```

```
    v_locatie locatie.oras%type;
```

```
begin
```

```
    select l.oras
```

```
    into v_locatie
```

```
    from locatie l join facultate f using (id_locatie)
```

```

        join specializare s using (id_facultate)

        join este_la_e using (id_specializare)

        join student st using (id_student)

where st.nume = v_student;

DBMS_OUTPUT.PUT_LINE(v_locatie);

exception

when no_data_found then

    DBMS_OUTPUT.PUT_LINE('Nu exista studenti cu numele dat.');
```

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista studenti cu numele dat.');

```

when too_many_rows then

    DBMS_OUTPUT.PUT_LINE('Exista prea multi studenti cu numele
dat.');
```

RAISE\_APPLICATION\_ERROR(-20001, 'Exista prea multi studenti cu numele dat.');

```

when others then

    DBMS_OUTPUT.PUT_LINE('Alta eroare.');
```

RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare.');

```

end c9;

begin

    c9('Popa'); -- fara erori

end;

begin

    c9('Gheorghe'); --no data found

end;
```

```
begin

    c9('Munteanu'); -- too many rows

end;
```

```
Iasi
```

```
Nu exista studenti cu numele dat.
```

```
Exista prea multi studenti cu numele dat.
```

## **10. Definiti un trigger care sa permita modificarea tabelului ingrijitor doar intre 8 si 12 de luni pana vineri.**

```
create or replace trigger c10
```

```
    before insert or update or delete on ingrijitor
```

```
begin
```

```
    if to_char(sysdate,'d') = 1 or to_char(sysdate,'d') = 7 or to_char(sysdate,
'hh24') not between 8 and 12 then
```

```
        RAISE_APPLICATION_ERROR(-20001, 'Tabelul nu poate fi
actualizat.');
```

```
    end if;
```

```
end;
```

```
/
```

```
drop trigger c10;
```

```
--insert into ingrijitor values (12, 'Ion', 'Alexandra',
3000,to_date('25.02.2001','dd/mm/yyyy'), 4);
```

---

```
Error starting at line : 549 in command -
insert into ingrijitor values (12, 'Ion', 'Alexandra', 3000,to_date('25.02.2001','dd/mm/yyyy'), 4)
Error report -
ORA-20001: Tabelul nu poate fi actualizat.
ORA-06512: at "ANTONIA.C10", line 3
ORA-04088: error during execution of trigger 'ANTONIA.C10'
```

## **11. Definiti un trigger prin care sa nu se poata mari cu mai mult de 20% salariile ingrijitorilor.**

create or replace trigger c11

before update of salariu on ingrijitor

for each row

begin

if :new.salariu > 1.2 \* :old.salariu then

RAISE\_APPLICATION\_ERROR(-20002, 'Salariul nu poate fi marit.');

end if;

end;

/

drop trigger c11;

--update ingrijitor set salariu = 1.5 \* salariu

```
Error starting at line : 563 in command -
update ingrijitor set salariu = 1.5 * salariu
Error report -
ORA-20002: Salariul nu poate fi marit.
ORA-06512: at "ANTONIA.C11", line 3
ORA-04088: error during execution of trigger 'ANTONIA.C11'
```

## **12. Definiti un trigger care sa introduca date in tabelul audit\_12 dupa ce utilizatorul a folosit o comanda LDD.**

create table audit\_12

(utilizator varchar2(50),

```

nume_bd varchar2(50),

eveniment varchar2(50),

nume_obiect varchar2(50),

data date);

```

```

create or replace trigger c12

```

```

after create or drop or alter on schema

```

```

begin

```

```

insert into audit_12

```

```

values (sys.login_user, sys.database_name, sys.sysevent,
sys.dictionary_obj_name, sysdate);

```

```

end;

```

```

drop trigger c12;

```

```

create table test

```

```

(id_test number(5),

```

```

nume varchar2(50)

```

```

);

```

```

select * from audit_12;

```

	UTILIZATOR	NUME_BD	EVENIMENT	NUME_OBIECT	DATA
1	ANTONIA	011G	CREATE	TEST	24-AUG-22

- 13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.** L  
SEP

create or replace package pachet as

    procedure c6(cod\_specializare specializare.id\_specializare%type);

    procedure c7;

    function c8(ume\_ingrijitor ingrijitor.ume%type)

        return universitate.ume%type;

    procedure c9 (v\_student student.ume%type);

end;

create or replace package body pachet as

--6)

    procedure c6(cod\_specializare specializare.id\_specializare%type) is

        type tablou\_imbricat is table of student.id\_student%type;

        v\_student tablou\_imbricat := tablou\_imbricat();

        type vector is varray(30) of profesor.id\_profesor%type;

        v\_profesor vector := vector();

        cursor c is

            select id\_student, id\_profesor

            from coordoneaza;

        n number(3);

    begin



```

select count(*)

into n

from profesor;

for i in 1..n loop
    v_profesor.extend;
end loop;

select id_student

bulk collect into v_student

from student s join este_la e using (id_student)

where e.id_specializare = cod_specializare;

select id_profesor

bulk collect into v_profesor

from profesor

where vechime >= 10;

for i in c loop

    for j in 1..v_student.count loop

        if v_student(j) = i.id_student then

            for k in 1..v_profesor.count loop

                if v_profesor(k) = i.id_profesor then

                    DBMS_OUTPUT.PUT_LINE('Studentul cu id-ul ' ||
i.id_student || ' este coordonat de profesorul cu id-ul ' || i.id_profesor);

                end if;
            end loop;
        end if;
    end loop;
end loop;

```

```

        end loop;

    end if;

    end loop;

    end loop;

end c6;

```

--7)

procedure c7 is

```

    type tablou_imbricat is table of facultate.id_facultate%type;

    v_id_facultate tablou_imbricat := tablou_imbricat();

    v_num_e_facultate facultate.num_e%type;

    v_num_e_ingrijitor ingrijitor.num_e%type;

    v_prename_ingrijitor ingrijitor.prename%type;

    v_salariu ingrijitor.salariu%type;

    contor number(3);

```

cursor c(cod\_facultate facultate.id\_facultate%type) is

```

    select i.num_e, i.prename, i.salariu

    from ingrijitor i, facultate f

    where i.id_facultate = f.id_facultate and f.id_facultate = cod_facultate;

```

begin

```

    select id_facultate

    bulk collect into v_id_facultate

    from facultate;

```

```

for i in v_id_facultate.first..v_id_facultate.last loop

    select nume

    into v_num_facultate

    from facultate

    where id_facultate = v_id_facultate(i);

    contor := 0;

    DBMS_OUTPUT.PUT_LINE(v_num_facultate || ':');

    open c(v_id_facultate(i));

    loop

        fetch c into v_num_ingrijitor, v_prename_ingrijitor, v_salariu;

        exit when c%NOTFOUND;

        contor := contor + 1;

        DBMS_OUTPUT.PUT_LINE('      ' || v_num_ingrijitor || ' ' ||
v_prename_ingrijitor || ',' || v_salariu);

    end loop;

    close c;

    if contor = 0 then

        DBMS_OUTPUT.PUT_LINE('      Nu exista ingrijitori.');
```

end if;

```

    DBMS_OUTPUT.PUT_LINE('-----
-----');
```

```

        end loop;

end c7;

--8)

function c8(ume_ingrijitor ingrijitor.ume%type)

    return universitate.ume%type

    is v_universitate universitate.ume%type;

begin

    select u.ume

    into v_universitate

    from universitate u join facultate f using (id_universitate)

        join ingrijitor i using (id_facultate)

    where i.ume = ume_ingrijitor;

    return v_universitate;

exception

    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('Nu exista ingrijitori cu numele dat.');
```

```

        RAISE_APPLICATION_ERROR(-20000, 'Nu exista ingrijitori cu
numele dat.');
```

```

    when too_many_rows then

        DBMS_OUTPUT.PUT_LINE('Exista prea multi ingrijitori cu numele
dat.');
```

```

        RAISE_APPLICATION_ERROR(-20001, 'Exista prea multi ingrijitori
cu numele dat.');
```

```

    when others then

        DBMS_OUTPUT.PUT_LINE('Alta eroare.');
```

```

        RAISE_APPLICATION_ERROR(-20002, 'Alta eroare.');
```

end c8;

--9)

```

procedure c9 (v_student student.nume%type)is
    v_locatie locatie.oras%type;
begin
    select l.oras
    into v_locatie
    from locatie l join facultate f using (id_locatie)
        join specializare s using (id_facultate)
        join este_la_e using (id_specializare)
        join student st using (id_student)
    where st.nume = v_student;

    DBMS_OUTPUT.PUT_LINE(v_locatie);

exception
    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('Nu exista studenti cu numele dat.');
```

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista studenti cu numele dat.');

```

    when too_many_rows then

        DBMS_OUTPUT.PUT_LINE('Exista prea multi studenti cu numele
dat.');
```

RAISE\_APPLICATION\_ERROR(-20001, 'Exista prea multi studenti cu numele dat.');

```

    when others then

        DBMS_OUTPUT.PUT_LINE('Alta eroare.');
```

```
        RAISE_APPLICATION_ERROR(-20002, 'Alta eroare.');
```

end c9;

end;

```
-----
```

begin

pachet.c6(2);

end;

begin

pachet.c7;

end;

begin

DBMS\_OUTPUT.PUT\_LINE(pachet.c8('Mormocea'));

end;

begin

pachet.c9('Popa');

end;

```
Studentul cu id-ul 6 este coordonat de profesorul cu id-ul 3
```

```
Facultatea de Matematica si Informatica:  
    Coca Eugen,1500  
    Paduraru Ion,2300  
    Mitronici Iuliana,1200  
    Mormocea Dorina,2000
```

```
-----  
Facultatea de Matematica si Informatica:  
    Muresan Teofil,2600
```

```
-----  
Facultatea de Litere:  
    Mariut Elena,1500
```

```
-----  
Facultatea de Drept:  
    Argint Cornel,2000
```