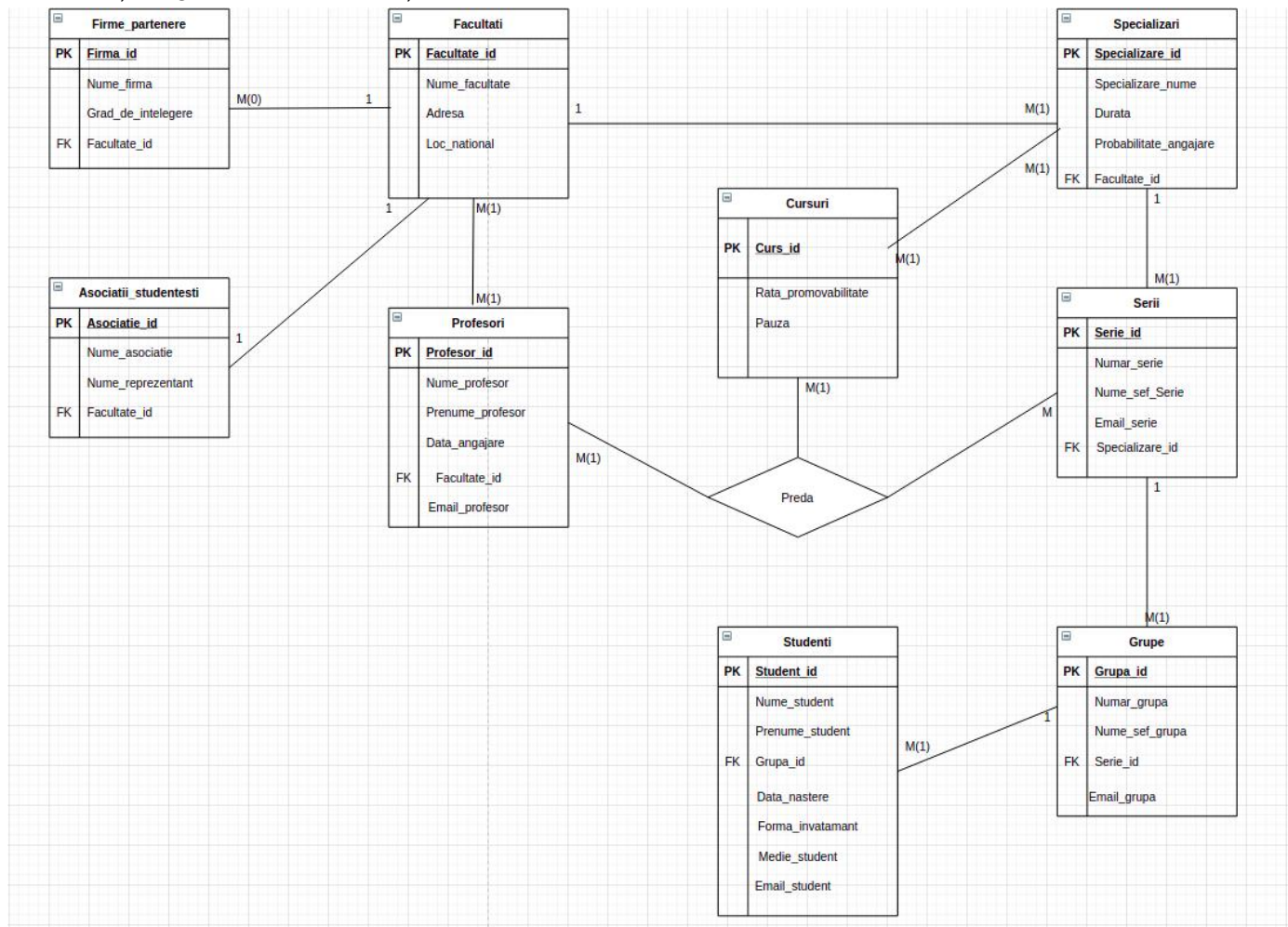


Proiect

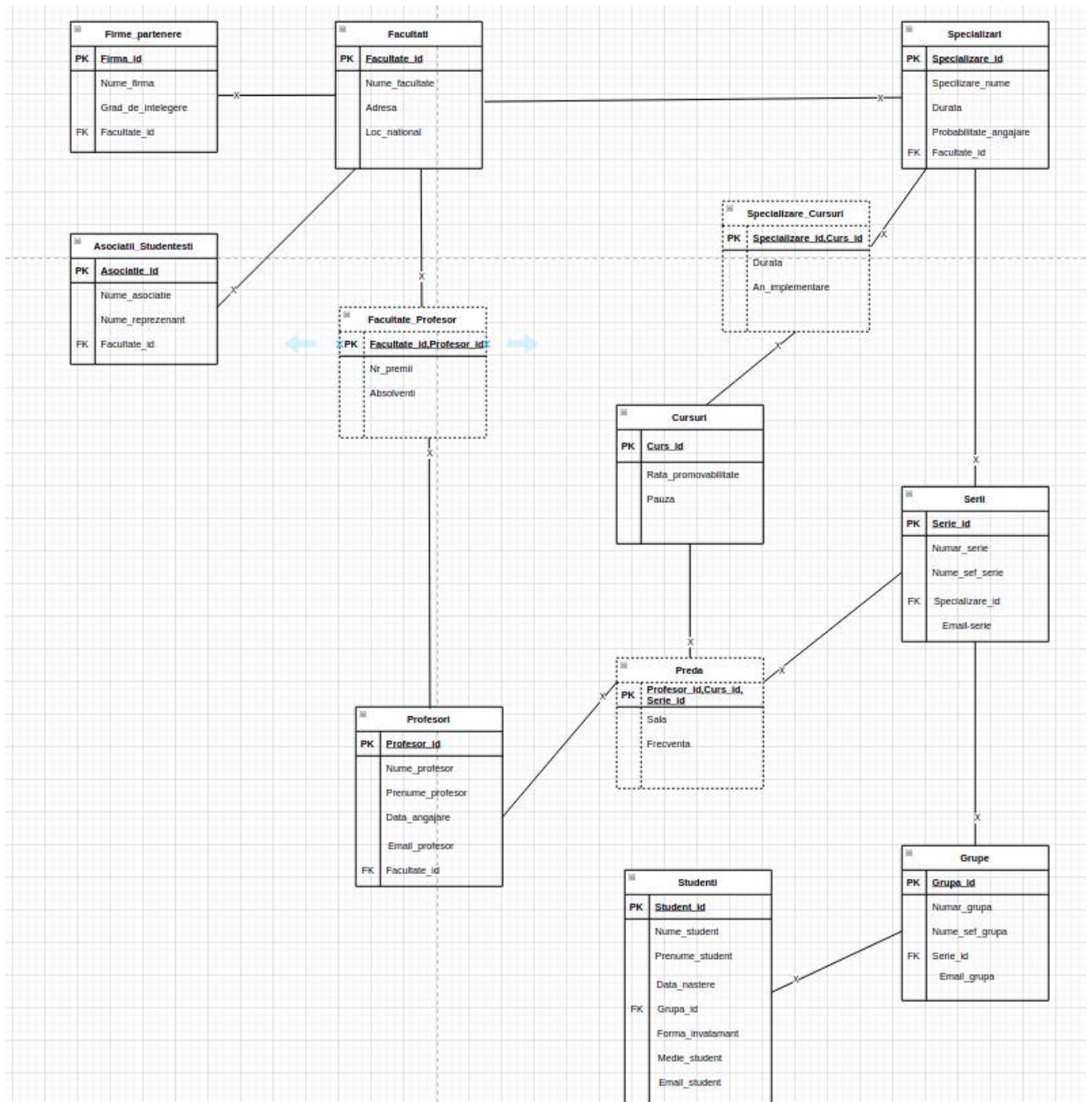
1. Prezențați pe scurt baza de date (utilitatea ei).

Modelul real este Universitatea din București, care este compusă din mai multe facultăți. Fiecare facultate are profesori, firme partenere, o asociație studentescă și specializări. Fiecare specializare este compusă din serii și cursuri. Fiecare serie este compusă din grupe, iar grupele sunt compuse din studenți. Asocierea dintre profesori, serii și cursuri rezultă într-o lecție de predare. Din acest model putem deduce diverse lucruri precum: câți studenți are universitatea, cei mai premiați profesori, studenții bursieri etc.

2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```
create table Facultati(  
    Facultate_id number(4) primary key,  
    Nume_facultate varchar2(30),  
    Adresa varchar(50),  
    Loc_national number(4)  
);
```

```
create table Profesor(  
    Profesor_id number(4) primary key,  
    Facultate_id number(4),  
    Nume_profesor varchar2(10),  
    Prenume_profesor varchar2(15),  
    Data_angajare date,  
    Email_profesor varchar2(30),  
    constraint fk_facultate_profesor foreign key (Facultate_id) references  
    facultati(Facultate_id)  
);
```

```
create table Facultate_Professori(  
    Facultate_id number(4),  
    Profesor_id number(4),  
    Absolvent varchar2(3),  
    Nr_premii number(2),  
    constraint pk_fac_prof primary key(Facultate_id, Profesor_id)  
);
```

```
create table Specializari(  
    Specializare_id number(4) primary key,  
    Facultate_id number(4),  
    Specializare_nume varchar2(15),  
    Durata number(1),  
    Probabilitate_angajare number(4,2),  
    constraint fk_fac_spec foreign key (Facultate_id) references facultati(Facultate_id)  
);
```

```
create table Cursuri(  
    Curs_id number(4) primary key,  
    Rata_promovabilitate number(4,2),  
    Pauza varchar2(3)  
);
```

```
create table Specializare_Cursuri(  
    Specializare_id number(4),  
    Curs_id number(4),
```

```
Durata number(3),
An_implementare number(4),
constraint pk_spec_curs primary key(Specializare_id,Curs_id)
);
```

```
create table Serii(
    Serie_id number(4) primary key,
    Numar_serie number(2),
    Nume_sef_serie varchar2(15),
    Email_serie varchar2(25),
    Specializare_id number(4),
    constraint fk_spec_ser foreign key (Specializare_id) references
specializari(Specializare_id)
);
```

```
create table Preda(
    Profesor_id number(4),
    Curs_id number(4),
    Serie_id number(4),
    Sala varchar2(25),
    Frecventa number(2),
    constraint pk_preda primary key(Profesor_id,Curs_id,Serie_id)
);
```

```
create table Grupe(
    Grupa_id number(4) primary key,
    Numar_grupa number(3),
    Nume_sef_grupa varchar2(15),
    Email_grupa varchar2(30),
    Serie_id number(4),
    constraint fk_ser_gr foreign key (Serie_id) references serii(Serie_id)
);
```

```
create table Studenti(
    Student_id number(4) primary key,
    Nume_student varchar2(15),
    Prenume_student varchar2(15),
    Data_nastere date,
    Grupa_id number(4),
    Forma_invatamant varchar2(15),
    Medie_student number(3,2),
    Data_admis date,
    constraint fk_gr_stud foreign key (Grupa_id) references grupe(Grupa_id)
);
```

```
create table Firme_Partenere(
    Firma_id number(4),
    Nume_firma varchar2(20),
```

```

Grad_de_intelegere number(4,2),
Facultate_id number(4),
constraint fk_fac_firma foreign key (Facultate_id) references facultati(Facultate_id)
);

```

```

create table Asociatii_studentesti(
    Asociatie_id number(4),
    Nume_asociatie varchar2(45),
    Nume_reprezentant varchar2(15),
    Facultate_id number(4),
    constraint fk_fac_asoc foreign key (Facultate_id) references facultati(Facultate_id)
);

```

5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```

insert into facultati(facultate_id, nume_facultate,adresa,loc_national)
values(1,'Facultatea de Matematica-Info','Universitate',1);
insert into facultati(facultate_id, nume_facultate,adresa,loc_national)
values(2,'Facultatea de Biologie','Gradina Botanica',3);
insert into facultati(facultate_id, nume_facultate,adresa,loc_national)
values(3,'Facultatea de Jurnalism','Universitate',10);
insert into facultati(facultate_id, nume_facultate,adresa,loc_national)
values(4,'Facultatea de Limbi Straine','Universitate',9);
insert into facultati(facultate_id, nume_facultate,adresa,loc_national)
values(5,'Facultatea de Filozofie','Grozavesti',12);

```

```

insert into
profesor(profesor_id,facultate_id,nume_profesor,prenume_profesor,data_angajare,email_profesor)
values(10,1,'Pop','Ionel',to_date('06-06-2002','dd-mm-yyyy'),'pop.ionel@unibuc.ro');
insert into
profesor(profesor_id,facultate_id,nume_profesor,prenume_profesor,data_angajare,email_profesor)
values(11,2,'Popa','Ionela',to_date('06-06-2005','dd-mm-yyyy'),'popa.ionela@unibuc.ro');
insert into
profesor(profesor_id,facultate_id,nume_profesor,prenume_profesor,data_angajare,email_profesor)
values(12,3,'Magda','Ana',to_date('06-09-2012','dd-mm-yyyy'),'ana.magda@unibuc.ro');
insert into
profesor(profesor_id,facultate_id,nume_profesor,prenume_profesor,data_angajare,email_profesor)
values(13,4,'Anton','Margel',to_date('06-06-2019','dd-mm-yyyy'),'margel.anton@unibuc.ro');
insert into
profesor(profesor_id,facultate_id,nume_profesor,prenume_profesor,data_angajare,email_profesor)

```

```
values(14,5,'Mugur','Alexandru',to_date('08-12-1998','dd-mm-yyyy'),'mugur_alexandru@unibuc.ro');
```

```
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(1,10,'da',5);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(1,14,'nu',0);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(2,11,'nu',7);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(2,13,'nu',1);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(3,12,'da',1);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(3,13,'nu',0);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(4,13,'da',19);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(5,14,'da',12);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(5,13,'nu',4);
insert into Facultate_Professori(facultate_id,profesor_id,absolvent,nr_premii)
values(2,10,'nu',2);
```

```
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(21,1,'cti',4,96.70);
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(22,1,'informatica',3,99.36);
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(23,2,'microorganisme',3,70.02);
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(24,3,'media digitala',3,82.52);
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(25,4,'engleza-rusa',3,91.22);
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(26,4,'chineza-araba',3,40.94);
insert into
specializari(specializare_id,facultate_id,specializare_nume,durata,probabilitate_angajare)
values(27,5,'filozofie',3,22.08);
```

```
insert into cursuri(curs_id,rata_promovabilitate,pauza)
values(31,82,'nu');
```

```
insert into cursuri(curs_id,rata_promovabilitate,pauza)
values(32,40,'nu');
insert into cursuri(curs_id,rata_promovabilitate,pauza)
values(33,97.99,'da');
insert into cursuri(curs_id,rata_promovabilitate,pauza)
values(34,50,'da');
insert into cursuri(curs_id,rata_promovabilitate,pauza)
values(35,91.25,'nu');
insert into cursuri(curs_id,rata_promovabilitate,pauza)
values(36,87.41,'da');
```

```
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(21,31,100,2015);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(22,31,100,2014);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(22,32,100,2018);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(23,33,100,2007);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(24,34,100,2011);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(24,33,100,2020);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(25,35,100,2018);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(26,36,100,2016);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(26,35,100,2016);
insert into specializare_cursuri(specializare_id,curs_id,durata,an_implementare)
values(27,35,100,2020);
```

```
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(800,16,'Dragut','seria16@serie.unibuc.ro',21);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(801,13,'Emanuel','seria13@serie.unibuc.ro',22);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(802,14,'Balaur','seria14@serie.unibuc.ro',22);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(803,20,'Lazar','seria20@serie.unibuc.ro',23);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(804,30,'Paiu','seria30@serie.unibuc.ro',24);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(805,40,'Pop','seria40@serie.unibuc.ro',25);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(806,41,'Ionescu','seria41@serie.unibuc.ro',25);
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(807,42,'Ionescu','seria42@serie.unibuc.ro',26);
```

```
insert into serii(serie_id,numar_serie,nume_sef_serie,email_serie,specializare_id)
values(808,50,'Dragut','seria50@serie.unibuc.ro',27);
```

```
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(10,31,800,'amfiteatrul 1',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(10,31,801,'sala google',2);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(11,33,803,'sala 40',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(10,33,803,'amfiteatrul bio',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(12,34,804,'amfiteatrul 2',2);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(13,35,805,'amfiteatrul 1',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(13,35,806,'amfiteatrul 1',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(14,35,808,'amfiteatrul 1',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(14,35,801,'amfiteatrul 1',1);
insert into preda(profesor_id,curs_id,serie_id,sala,frecventa)
values(14,35,802,'amfiteatrul 1',1);
```

```
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(901,161,'Mihalcea','grupa161@grupa.unibuc.ro,80',800);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(902,162,'Mihaila','grupa162@grupa.unibuc.ro,80',800);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(903,201,'Iepure','grupa201@grupa.unibuc.ro,80',803);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(904,301,'Stan','grupa301@grupa.unibuc.ro,80',804);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(905,401,'Stanescu','grupa401@grupa.unibuc.ro,80',805);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(906,402,'Popescu','grupa402@grupa.unibuc.ro,80',805);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(907,411,'Pop','grupa411@grupa.unibuc.ro,80',806);
insert into grupe(grupa_id,numar_grupa,nume_sef_grupa,email_grupa,serie_id)
values(908,501,'Chelaru','grupa501@grupa.unibuc.ro,80',807);
```

```
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatamant,medie_student,data_admis)
values(1000,'Escobar','Paul',to_date('08-12-2001','dd-mm-yyyy'),901,'buget',9.12,to_date('30-07-2020','dd-mm-yyyy'));
```



```

insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1001,'Rugina','Andrei',to_date('08-08-2000','dd-mm-yyyy'),901,'taxa',8.74,to_date('29-
07-2020','dd-mm-yyyy'));
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1002,'Mihai','Irina',to_date('11-12-1999','dd-mm-yyyy'),902,'buget',9.98,to_date('31-07
-2020','dd-mm-yyyy'));
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1002,'Begu','Maria',to_date('08-12-2001','dd-mm-yyyy'),903,'taxa',7.80,to_date('30-07
-2020','dd-mm-yyyy'));
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1003,'Bratu','Vlad',to_date('03-01-2002','dd-mm-yyyy'),907,'buget',9.02,to_date('30-0
7-2020','dd-mm-yyyy'));
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1004,'Andries','Alexanda',to_date('08-04-2001','dd-mm-yyyy'),906,'buget',8.68,to_dat
e('14-09-2019','dd-mm-yyyy'));
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1005,'Almas','Alexandru',to_date('08-07-2000','dd-mm-yyyy'),906,'taxa',6.12,to_date('
29-07-2019','dd-mm-yyyy'));
insert into
studenti(student_id,nume_student,prenume_student,data_nastere,grupa_id,forma_invatama
nt,medie_student,data_admis)
values(1006,'Panaite','Aurelian',to_date('09-09-1999','dd-mm-yyyy'),908,'buget',9.71,to_date(
'31-07-2018','dd-mm-yyyy'));

```

```

insert into Firme_partenere(firma_id, nume_firma,grad_de_intelegere,facultate_id)
values( 1 ,'Bitdefender',80,1);
insert into Firme_partenere(firma_id, nume_firma,grad_de_intelegere,facultate_id)
values( 2 ,'Gradina Botanica',97.56,2);
insert into Firme_partenere(firma_id, nume_firma,grad_de_intelegere,facultate_id)
values( 3 ,'Monitorul oficial',54,3);
insert into Firme_partenere(firma_id, nume_firma,grad_de_intelegere,facultate_id)
values(4,'Thales',91.14,4);
insert into Firme_partenere(firma_id, nume_firma,grad_de_intelegere,facultate_id)
values(5,'Carturesti',67.18,5);

```

```

insert into
asociatii_studentesti(asociatie_id,nume_asociatie,nume_reprezentant,facultate_id)
values(1 ,'ASMI','Grigorescu',1);
insert into
asociatii_studentesti(asociatie_id,nume_asociatie,nume_reprezentant,facultate_id)
values( 2 ,'ASB','Petrache',2);
insert into
asociatii_studentesti(asociatie_id,nume_asociatie,nume_reprezentant,facultate_id)
values(3 ,'ASJ','Teodor',3);
insert into
asociatii_studentesti(asociatie_id,nume_asociatie,nume_reprezentant,facultate_id)
values(4,'ASLS','Minodora',4);
insert into
asociatii_studentesti(asociatie_id,nume_asociatie,nume_reprezentant,facultate_id)
values(5,'ASF','Paulescu',5);

```

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze două tipuri de colecție studiate. Apelați subprogramul.

Sa se afiseze pentru firmele partenere și asociațiile studențești pentru o facultate data.

```

create or replace procedure firmeAsociatii(facultate facultati.nume_facultate%type)
is

```

```

type asoc_tabel is table of varchar2(50) index by binary_integer;
type firme_vec is varray(100) of varchar2(50);

```

```

asoc asoc_tabel;
firme firme_vec;
v_id facultati.facultate_id%type;
v_fac facultati.nume_facultate%type;

```

```

begin
begin
select nume_facultate into v_fac
from facultati
where lower(nume_facultate)=lower(facultate);

```

```

exception
when no_data_found then
raise_application_error(-20017,' ' || facultate || ' nu este in baza de date');
end;

```

```

select facultate_id into v_id
from facultati
where lower(facultate)=lower(ume_facultate);

select ume_firma bulk collect into firme
from firme_partenere
where v_id=facultate_id;

select ume_asociatie bulk collect into asoc
from asociatii_studentesti
where v_id=facultate_id;

dbms_output.put_line('Firmele partenere ale ' || facultate || ' sunt: ');
for firma in firme.first..firme.last
loop
dbms_output.put_line(firme(firma));
end loop;

dbms_output.put_line('Asociatiile studentesti ale ' || facultate || ' sunt: ');

for a in asoc.first..asoc.last
loop
dbms_output.put_line(asoc(a));
end loop;
end firmeAsociatii;
/

```

```

Procedure FIRMEASOCIATII compiled
.

```

Apelarea subprogramului:

```

exec firmeAsociatii('Facultatea de Matematica-Info');

```

```

Firmele partenere ale Facultatea de Matematica-Info sunt:
Bitdefender
Asociatiile studentesti ale Facultatea de Matematica-Info sunt:
ASMI

PL/SQL procedure successfully completed.

```

```
exec firmeAsociatii('Facultatea de Geologie');
```

```
Error starting at line : 56 in command -  
BEGIN firmeAsociatii('Facultatea de Geologie'); END;  
Error report -  
ORA-20017:  Facultatea de Geologie nu este in baza de date  
ORA-06512: at "TEST.FIRMEASOCIATII", line 21  
ORA-06512: at line 1
```

7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

Sa se afiseze numele,prenumele si media studentilor care au media mai mare sau egala cu 9, necontand in cadrul carei facultati sunt inmatriculati. Datele vor fi sortate descrescator in functie de medie.

```
create or replace procedure studenti9  
is
```

```
lista_studenti studenti%rowtype;
```

```
cursor students return studenti%rowtype  
is  
select *  
from studenti  
order by medie_student desc;
```

```
begin  
open students;  
dbms_output.put_line('Studentii care au media peste 9 sunt: ');  
loop  
fetch students into lista_studenti;  
exit when students%notfound;
```

```
if lista_studenti.medie_student>=9 then  
    dbms_output.put_line(lista_studenti.nume_student || ' ' || lista_studenti.prenume_student ||  
' ' || lista_studenti.medie_student);  
end if;  
end loop;  
end studenti9;  
/
```

```
Procedure STUDENTI9 compiled
```

Apelarea subprogramului:

exec studenti9;

```
Studentii care au media peste 9 sunt:  
Decebal Daniel 9.99  
Mihai Irina 9.98  
Panaite Aurelian 9.71  
Ghenea Ana-Maria 9.35  
Dima Ion 9.07  
Bratu Vlad 9.02  
  
PL/SQL procedure successfully completed.
```

8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Sa se afiseze numele specializarii care nu are niciun student. Daca nu exista o astfel de specializare se va afisa un mesaj corespunzator, iar daca sunt mai multe astfel de specializari se va afisa de asemenea un mesaj corespunzator.

```
create or replace function specializareFaraStudent return  
specializari.specializare_nume%type  
is  
v_spec specializari.specializare_nume%type;  
begin  
    select specializare_nume into v_spec  
    from specializari  
    where specializare_nume not in  
    (select spec.specializare_nume  
    from studenti s join grupe g on(s.grupa_id=g.grupa_id)  
    join serii ser on(ser.serie_id=g.serie_id)  
    join specializari spec on(spec.specializare_id=ser.specializare_id));  
    return v_spec;  
exception  
when no_data_found then  
    raise_application_error(-20005,'Toate specializarile au studenti');
```

```

when too_many_rows then
    raise_Application_error(-20006,'Cel putin 2 specializari nu au studenti');
when others then
    raise_application_error(-20061,'Alta eroare a fost intampinata');
end specializareFaraStudent;
/

```

```
Function SPECIALIZAREFARASTUDENT compiled
```

Apelare:

```

declare
v_ans specializari.specializare_nume%type;
begin
v_ans:=specializareFaraStudent;
dbms_output.put_line(v_ans);
end;
/

```

```

Error starting at line : 55 in command -
declare
v_ans specializari.specializare_nume%type;
begin
v_ans:=specializareFaraStudent;
dbms_output.put_line(v_ans);
end;
Error report -
ORA-20006: Cel putin 2 specializari nu au studenti
ORA-06512: at "TEST.SPECIALIZAREFARASTUDENT", line 17
ORA-06512: at line 4

```

9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Pentru o facultate data afisati id-ul studentului eliminat(studentul care are media egala cu 5). Daca nu este niciun student sau sunt mai multi studenti in aceasta situatie niciun un student nu va fi eliminat si se vor trata cazurile in mod corespunzator.

```
create or replace procedure studentEliminat(facultate facultati.nume_facultate%type)
is
```

```
v_nume studenti.nume_student%type;
v_prenume studenti.nume_student%type;
v_fac facultati.nume_facultate%type;
v_id studenti.student_id%type;
```

```
begin
```

```
begin
```

```
    select nume_facultate into v_fac
    from facultati
    where lower(nume_facultate)=lower(facultate);
exception
```

```
    when no_data_found then
```

```
        raise_application_error(-20002,'Facultatea nu exista');
```

```
    when others then
```

```
        raise_application_error(-20071,'O eroare necunoscuta a aparut.');
```

```
end;
```

```
    select s.student_id s_id into v_id
    from grupe g join studenti s on (s.grupa_id=g.grupa_id)
    join serii ser on (ser.serie_id=g.serie_id)
    join specializari spec on (spec.specializare_id=ser.specializare_id)
    join facultati f on (spec.facultate_id=f.facultate_id)
    where lower(f.nume_facultate)=lower(facultate)
    group by s.student_id,s.medie_student
    having s.medie_student=5;
```

```
dbms_output.put_line('Studentul exmatriculat va fi: ' || v_id);
```

```
exception
```

```
    when no_data_found then
```

```
        raise_application_error(-20003,'Niciun student nu este eligibil pentru exmatriculare');
```

```
    when too_many_rows then
```

```
        raise_application_error(-20004,'Prea multi studenti eligibili epntru exmatriculare. Deci
nu mai exmatriculam pe nimeni');
```

```
end studentEliminat;
```

```
/
```

```
Procedure STUDENTELIMINAT compiled
```

Apelare:

```
exec studentEliminat('Facultatea de Limbi Straine');
```

```
Studentul exmatriculat va fi: 1005
```

```
PL/SQL procedure successfully completed.
```

```
exec studentEliminat('Facultatea de Biologie');
```

```
Error starting at line : 106 in command -  
BEGIN studentEliminat('Facultatea de Biologie'); END;  
Error report -  
ORA-20003: Niciun student nu este eligibil pentru exmatriculare  
ORA-06512: at "TEST.STUDENTELIMINAT", line 35  
ORA-06512: at line 1
```

```
exec studentEliminat('Facultatea de Matematica-Info');
```

```
Error starting at line : 106 in command -  
BEGIN studentEliminat('Facultatea de Matematica-Info'); END;  
Error report -  
ORA-20004: Prea multi studenti eligibili epntru exmatriculare. Deci nu mai exmatriculam pe nimeni  
ORA-06512: at "TEST.STUDENTELIMINAT", line 37  
ORA-06512: at line 1
```

```
exec studentEliminat('Facultatea de geologie');
```

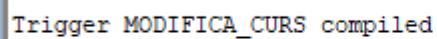
```
Error starting at line : 106 in command -  
BEGIN studentEliminat('Facultatea de geologie'); END;  
Error report -  
ORA-20002: Facultatea nu exista  
ORA-06512: at "TEST.STUDENTELIMINAT", line 17  
ORA-06512: at line 1
```


10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

```
create or replace trigger modifica_curs
before
insert or update or delete on cursuri
begin

    if to_char(sysdate,'mm')>'09' or to_char(sysdate,'mm')<'07' then
        raise_application_error(-20050,'Modificarea unui curs este disponibila doar in vacanta de
vara');
    end if;

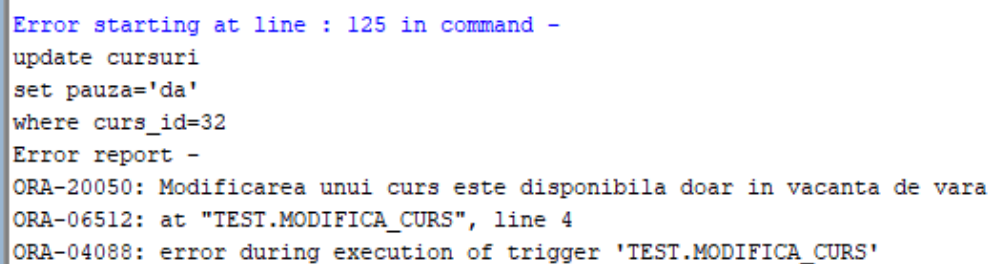
end;
/
```



Trigger MODIFICA_CURS compiled

Declansarea triggerului:

```
update cursuri
set pauza='da'
where curs_id=32;
```



```
Error starting at line : 125 in command -
update cursuri
set pauza='da'
where curs_id=32
Error report -
ORA-20050: Modificarea unui curs este disponibila doar in vacanta de vara
ORA-06512: at "TEST.MODIFICA_CURS", line 4
ORA-04088: error during execution of trigger 'TEST.MODIFICA_CURS'
```

11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

```
create or replace trigger durata_cursuri
after update or insert on specializare_cursuri
for each row
begin
```

```
if(:NEW.durata>150) then
  raise_application_error(-20019,'Cursul nu poate tine mai mult de 3 ore');
end if;

end;
/
```

```
Trigger DURATA_CURSURI compiled
```

Declansarea triggerului:

```
update specializare_cursuri
set durata=160;
```

```
Error starting at line : 142 in command -
update specializare_cursuri
set durata=160
Error report -
ORA-20019: Cursul nu poate tine mai mult de 3 ore
ORA-06512: at "TEST.DURATA_CURSURI", line 4
ORA-04088: error during execution of trigger 'TEST.DURATA_CURSURI'
```

```
update specializare_cursuri
set durata=100;
```

```
10 rows updated.
```

```
insert into specializare_cursuri
values(28,36,50,2021);
```

```
1 row inserted.
```

```
insert into specializare_cursuri  
values(28,36,151,2021);
```

```
Error starting at line : 3 in command -  
insert into specializare_cursuri  
values(28,36,151,2021)  
Error report -  
ORA-20019: Cursul nu poate tine mai mult de 3 ore  
ORA-06512: at "TEST.DURATA_CURSURI", line 4  
ORA-04088: error during execution of trigger 'TEST.DURATA_CURSURI'
```

12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

```
create table modificari(  
    nume_baza_de_date varchar2(50),  
    nume_user varchar(30),  
    modificare varchar2(15),  
    tip_obiect_referit varchar2(50),  
    nume_obiect_referit varchar2(75),  
    data timestamp);
```

```
create or replace trigger protectie  
after create or alter or drop on schema  
begin
```

```
insert into modificari  
values(sys.database_name,sys.login_user,ora_sysevent,ora_dict_obj_type,ora_dict_obj_name,sys.timestamp);
```

```
end;  
/
```

```
Trigger PROTECTIE compiled
```

Declansarea triggerului:

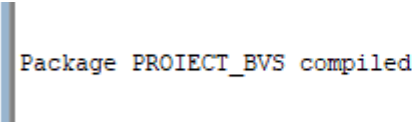
```
alter table profesor
add grad142 varchar(50);
```

```
select * from modificari;
```

128 XE	TEST	CREATE	TRIGGER	DURATA_CURSURI	07-JAN-22 06.10.26.004000000 PM
129 XE	TEST	ALTER	TABLE	PROFESOR	07-JAN-22 06.27.42.922000000 PM
130 XE	TEST	ALTER	TABLE	PROFESOR	04-JAN-22 11.12.27.821000000 PM
131 XE	TEST	CREATE	PACKAGE	EX_14	06-JAN-22 01.00.25.310000000 AM
132 XE	TEST	CREATE	PACKAGE BODY	EX_14	06-JAN-22 01.19.07.264000000 AM

13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

```
create or replace package proiect_bvs
as
procedure firmeAsociatii(facultate facultati.ume_facultate%type);
cursor students return studenti%rowtype;
procedure studenti9;
function specializareFaraStudent return specializari.specializare_ume%type;
procedure studentEliminat(facultate facultati.ume_facultate%type);
end proiect_bvs;
/
```



Package PROIECT_BVS compiled

```
create or replace package body proiect_bvs
as
```

```
cursor students return studenti%rowtype
is
select *
from studenti
order by medie_student;
```

```
procedure firmeAsociatii(facultate facultati.ume_facultate%type)
is
```

```
type asoc_tabel is table of varchar2(50) index by binary_integer;
type firme_vec is varray(100) of varchar2(50);
```

```
asoc asoc_tabel;
```

```

firme firme_vec;
v_id facultati.facultate_id%type;
v_fac facultati.num_e_facultate%type;

begin
begin
select num_e_facultate into v_fac
from facultati
where lower(num_e_facultate)=lower(facultate);

exception
when no_data_found then
raise_application_error(-20017,' ' || facultate || ' nu este in baza de date');
end;

select facultate_id into v_id
from facultati
where lower(facultate)=lower(num_e_facultate);

select num_e_firma bulk collect into firme
from firme_partenere
where v_id=facultate_id;

select num_e_asociatie bulk collect into asoc
from asociatii_studentesti
where v_id=facultate_id;

dbms_output.put_line('Firmele partenere ale ' || facultate || ' sunt: ');
for firma in firme.first..firme.last
loop
dbms_output.put_line(firme(firma));
end loop;

dbms_output.put_line('Asociatiile studentesti ale ' || facultate || ' sunt: ');

for a in asoc.first..asoc.last
loop
dbms_output.put_line(asoc(a));
end loop;

end;

procedure studenti9
is

lista_studenti studenti%rowtype;

```

```

cursor students return studenti%rowtype
is
select *
from studenti
order by medie_student desc;

```

```

begin
open students;
dbms_output.put_line('Studentii care au media peste 9 sunt: ');
loop
fetch students into lista_studenti;
exit when students%notfound;

```

```

if lista_studenti.medie_student >= 9 then
    dbms_output.put_line(lista_studenti.num_e_student || ' ' || lista_studenti.prenume_student ||
' ' || lista_studenti.medie_student);
end if;
end loop;

end;

```

```

function specializareFaraStudent return specializari.specializare_num_e%type
is
v_spec specializari.specializare_num_e%type;
begin
    select specializare_num_e into v_spec
    from specializari
    where specializare_num_e not in
(select spec.specializare_num_e
from studenti s join grupe g on(s.grupa_id=g.grupa_id)
    join serii ser on(ser.serie_id=g.serie_id)
    join specializari spec on(spec.specializare_id=ser.specializare_id));
    return v_spec;
exception
when no_data_found then
    raise_application_error(-20005,'Toate specializarile au studenti');
when too_many_rows then
    raise_Application_error(-20006,'Cel putin 2 specializari nu au studenti');
when others then
    raise_application_error(-20061,'Alta eroare a fost intampinata');
end;

```

```

procedure studentEliminat(facultate facultati.num_e_facultate%type)
is

```

```

v_nume studenti.nume_student%type;
v_prenume studenti.nume_student%type;
v_fac facultati.nume_facultate%type;
v_id studenti.student_id%type;

begin
begin
    select nume_facultate into v_fac
    from facultati
    where lower(nume_facultate)=lower(facultate);
exception
    when no_data_found then
        raise_application_error(-20002,'Facultatea nu exista');
    when others then
        raise_application_error(-20071,'O eroare necunoscuta a aparut.');
```

end;

```

        select s.student_id s_id into v_id
        from grupe g join studenti s on (s.grupa_id=g.grupa_id)
        join serii ser on (ser.serie_id=g.serie_id)
        join specializari spec on (spec.specializare_id=ser.specializare_id)
        join facultati f on (spec.facultate_id=f.facultate_id)
        where lower(f.nume_facultate)=lower(facultate)
        group by s.student_id,s.medie_student
        having s.medie_student=5;

dbms_output.put_line('Studentul exmatriculat va fi: ' || v_id);
exception
    when no_data_found then
        raise_application_error(-20003,'Niciun student nu este eligibil pentru exmatriculare');
    when too_many_rows then
        raise_application_error(-20004,'Prea multi studenti eligibili epntru exmatriculare. Deci
nu mai exmatriculam pe nimeni');
```

end;

```

end proiect_bvs;
/
```

Package Body PROIECT_BVS compiled

Apelari:

```
begin
proiect_bvs.firmeAsociatii('Facultatea de Matematica-Info');
end;
/
```

```
Firmele partenere ale Facultatea de Matematica-Info sunt:
Bitdefender
Asociatiile studentesti ale Facultatea de Matematica-Info sunt:
ASMI

PL/SQL procedure successfully completed.
```

```
begin
  proiect_bvs.studenti9;
end;
/
```

```
Studentii care au media peste 9 sunt:
Decebal Daniel 9.99
Mihai Irina 9.98
Panaite Aurelian 9.71
Ghenea Ana-Maria 9.35
Dima Ion 9.07
Bratu Vlad 9.02

PL/SQL procedure successfully completed.
```

```
declare
var specializari.specializare_nume%type;
begin
  var:= proiect_bvs.specializareFaraStudent;
  dbms_output.put_line(var);
end;
/
```



```

Error starting at line : 173 in command -
declare
var  specializari.specializare_nume$type;
begin
    var:= proiect_bvs.specializareFaraStudent;
    dbms_output.put_line(var);
end;
Error report -
ORA-20006: Cel puțin 2 specializari nu au studenti
ORA-06512: at "TEST.PROIECT_BVS", line 105
ORA-06512: at line 4

```

```

begin
    proiect_bvs.studentEliminat('Facultatea de Biologie');
end;
/

```

```

Error starting at line : 169 in command -
begin
    proiect_bvs.studentEliminat('Facultatea de Biologie');
end;
Error report -
ORA-20003: Niciun student nu este eligibil pentru exmatriculare
ORA-06512: at "TEST.PROIECT_BVS", line 144
ORA-06512: at line 2

```

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

```

create or replace package ex_14
as
-- returneaza numarul ce reprezinta 20% din numarul studentilor inmatriculati la o facultate
data
function nrStudenti(facultate facultati.nume_facultate$type) return integer;

--afiseaza id-ul studentilor bursieri(primii 20% din nr total de studenti). Daca sunt mai putin
de 5 studenti afiseaza 1.
procedure studentiBursieri(facultate facultati.nume_facultate$type);

--afiseaza id-ul primilor 10% profesori clasati in functie de numarul de premii obtinute. Daca
sunt mai putin de 10 profesori se afiseaza primul.
procedure topProfesori;

```

```
--afiseaza id-ul celui mai recent implementat curs de la o specializare data.
function cursRecent(numa specializari.specializare_nume%type) return
specializare_cursuri.curs_id%type;
end ex_14;
/
```

```
Package EX_14 compiled
```

```
create or replace package body ex_14
as
```

```
function nrStudenti(facultate facultati.nume_facultate%type) return integer
is
    v_fac facultati.nume_facultate%type;
    v_nr number;
begin
    begin
        select numa_facultate into v_fac
        from facultati
        where lower(nume_facultate)=lower(facultate);
    exception
        when no_data_found then
            raise_application_error(-20002,'Facultatea nu exista');
end;

    select sum(count(s.student_id)) into v_nr
    from grupe g join studenti s on (s.grupa_id=g.grupa_id)
    join serii ser on (ser.serie_id=g.serie_id)
    join specializari spec on (spec.specializare_id=ser.specializare_id)
    join facultati f on (spec.facultate_id=f.facultate_id)
    where lower(f.nume_facultate)=lower(facultate)
    group by s.student_id;

    if v_nr<5 then
        return 1;
    end if;

    return floor(0.2*v_nr);
exception
    when no_data_found then
        raise_application_error(-20050,'Facultatea nu are studenti');
end;
```

```
procedure studentiBursieri(facultate facultati.nume_facultate%type)
is
```

```
type tabel_studenti is table of varchar2(50) index by binary_integer;
```

```
bursieri tabel_studenti;
procent integer;
```

```
begin
procent:=nrStudenti(facultate);
if procent=0 then
dbms_output.put_line('ceva');
end if;
    select s.student_id bulk collect into bursieri
        from grupe g join studenti s on (s.grupa_id=g.grupa_id)
        join serii ser on (ser.serie_id=g.serie_id)
        join specializari spec on (spec.specializare_id=ser.specializare_id)
        join facultati f on (spec.facultate_id=f.facultate_id)
        where lower(f.nume_facultate)=lower(facultate)
        group by s.student_id,s.medie_student
        order by s.medie_student desc;

    dbms_output.put_line('Studentii bursieri ai facultatii ' || facultate || ' sunt: ');
    for stud in bursieri.first..procent
    loop
        dbms_output.put_line(bursieri(stud));
    end loop;
end;
```

```
procedure topProfesori
is
```

```
type prof_vec is varray(1000) of number;
profesori prof_vec;
v_nr number;
```

```
begin
```

```
select count(profesor_id) into v_nr
from facultate_profesori;
```

```
if v_nr>10 then
    v_nr:=floor(0.1*v_nr);
else
    v_nr:=1;
end if;
```

```
select profesor_id bulk collect into profesori
from facultate_profesori
order by nr_premii desc;
```

```
dbms_output.put_line('Cei mai premiati profesori sunt: ');
for prof in profesori.first..v_nr
loop
dbms_output.put_line(profesori(prof));
end loop;
end;
```

```
function cursRecent(ume specializari.specializare_ume%type) return
specializare_cursuri.curs_id%type
is
```

```
v_spec specializari.specializare_ume%type;
type cursuri_vec is varray(1000) of number;
curs cursuri_vec;
v_rez specializare_cursuri.curs_id%type;
```

```
begin
begin
select specializare_ume into v_spec
from specializari
where lower(ume)=lower(specializare_ume);
```

```
exception
when no_data_found then
raise_application_error(-20051,'Specializarea nu exista');
end;
```

```
select sc.curs_id bulk collect into curs
from specializare_cursuri sc join specializari s on(sc.specializare_id=s.specializare_id)
where lower(ume)=lower(s.specializare_ume)
order by sc.an_implementare desc;
```

```
v_rez:=curs.first;
return v_rez;
```

```
end;
end ex_14;
/
```

```
Package Body EX_14 compiled
```

Apelari:

```
declare
v_nr number;
begin
v_nr:=ex_14.nrStudenti('Facultatea de Matematica_Inf');
dbms_output.put_line(v_nr);
end;
/
```

```
Error starting at line : 156 in command -
declare
v_nr number;
begin
v_nr:=ex_14.nrStudenti('Facultatea de Matematica_Inf');
dbms_output.put_line(v_nr);
end;
Error report -
ORA-20002: Facultatea nu exista
ORA-06512: at "TEST.EX_14", line 15
ORA-06512: at line 4
```

```
declare
v_nr number;
begin
v_nr:=ex_14.nrStudenti('Facultatea de Biologie');
dbms_output.put_line(v_nr);
end;
/
```

```
1
```

```
PL/SQL procedure successfully completed.
```

```

begin
ex_14.studentiBursieri('Facultatea de Biologie');
end;
/

```

```

Studentii bursieri ai facultatii Facultatea de Biologie sunt:
1010

```

```

PL/SQL procedure successfully completed.

```

```

begin
ex_14.studentiBursieri('Facultatea de Geologie');
end;
/

```

```

Error starting at line : 164 in command -
begin
ex_14.studentiBursieri('Facultatea de Geologie');
end;
Error report -
ORA-20002: Facultatea nu exista
ORA-06512: at "TEST.EX_14", line 15
ORA-06512: at "TEST.EX_14", line 47
ORA-06512: at line 2

```

```

begin
ex_14.topProfesori;
end;
/

```

```

Cei mai premiati profesori sunt:
13

```

```

PL/SQL procedure successfully completed.

```

```

declare
v_ans specializare_cursuri.curs_id%type;
begin
v_ans:=ex_14.cursRecent('cti');
dbms_output.put_line(v_ans);
end;
/

```

1

PL/SQL procedure successfully completed.

```
declare
v_ans specializare_cursuri.curs_id%type;
begin
v_ans:=ex_14.cursRecent('crti');
dbms_output.put_line(v_ans);
end;
/
```

```
Error starting at line : 173 in command -
declare
v_ans specializare_cursuri.curs_id%type;
begin
v_ans:=ex_14.cursRecent('crti');
dbms_output.put_line(v_ans);
end;
Error report -
ORA-20051: Specializarea nu exista
ORA-06512: at "TEST.EX_14", line 113
ORA-06512: at line 4
```