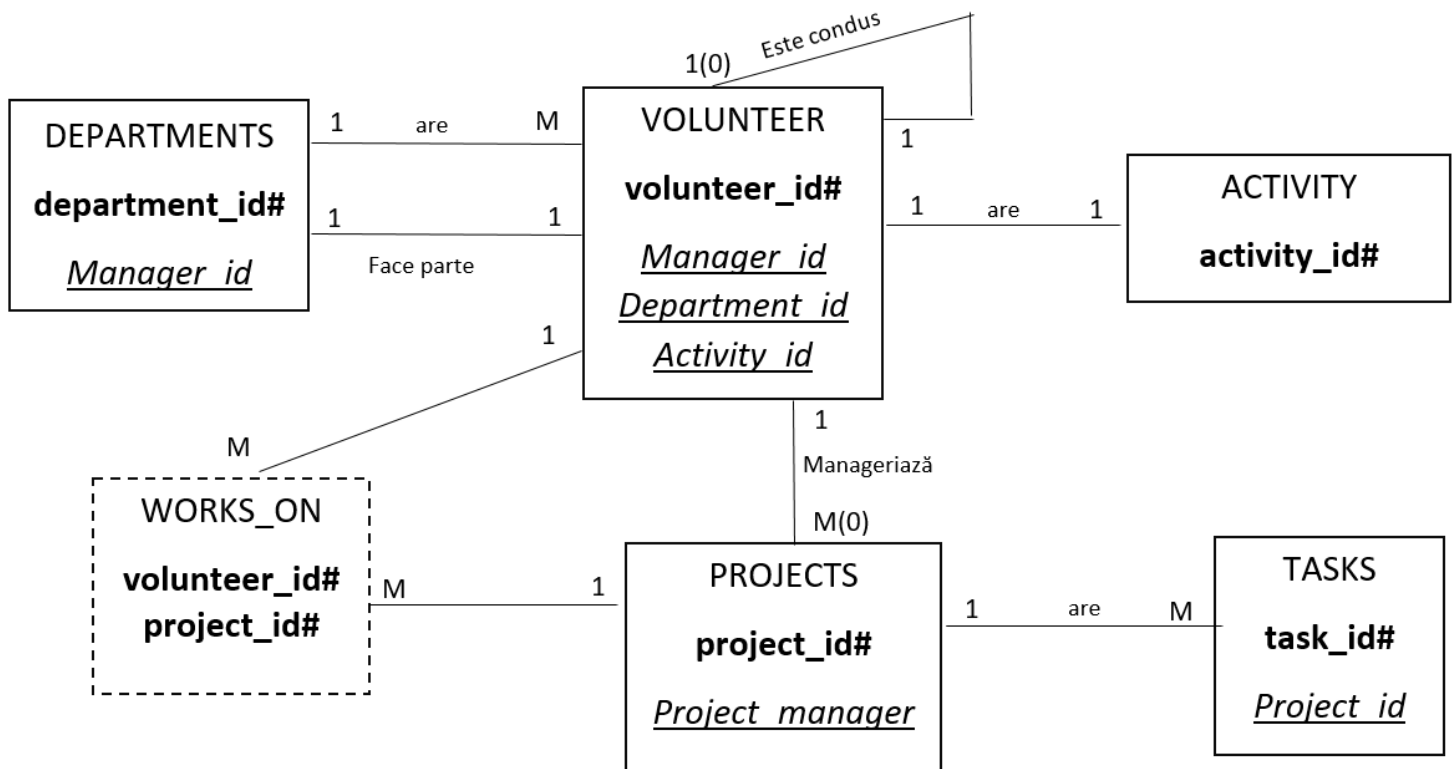


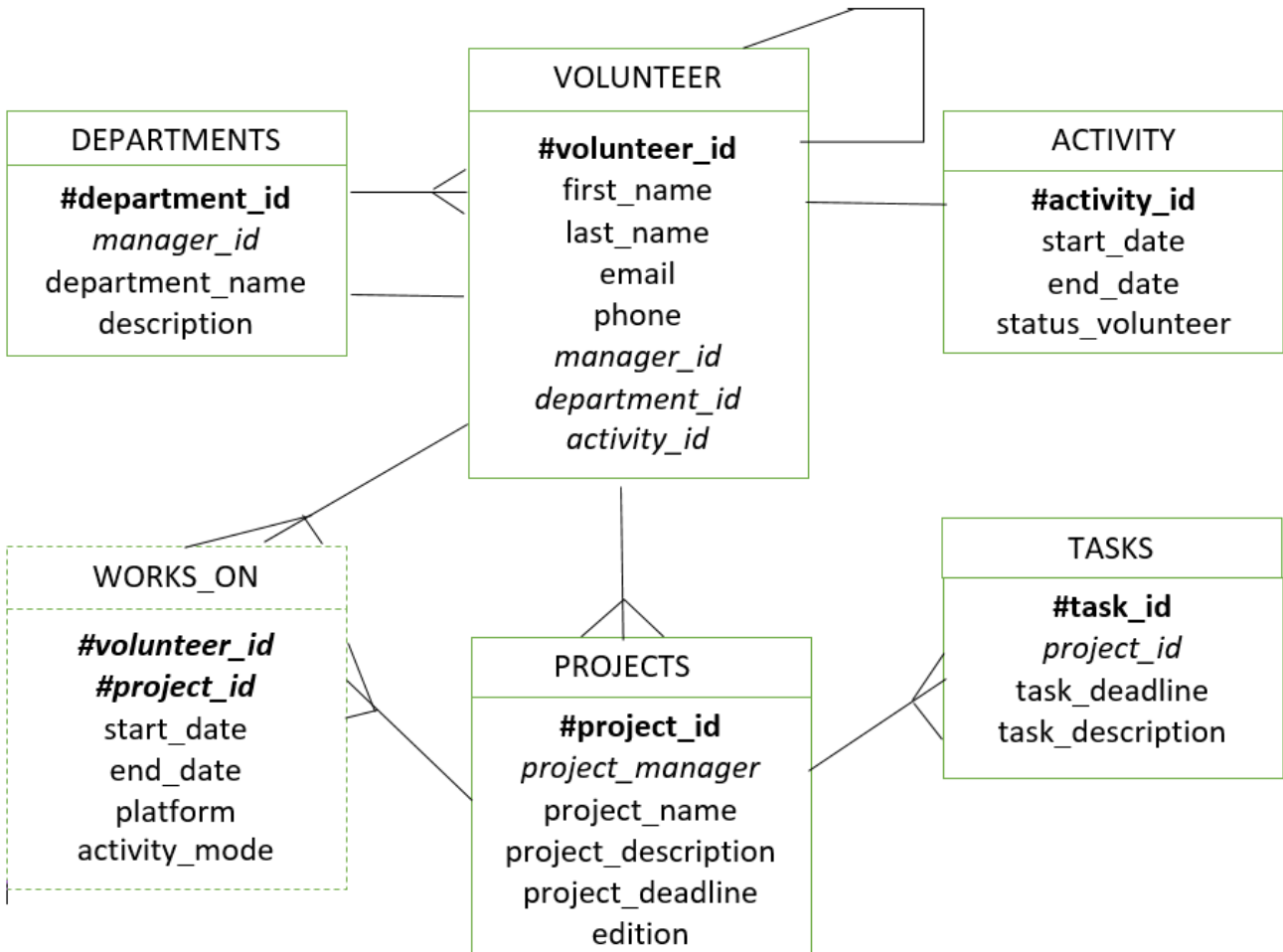
Baza de date Asociația Studenților  
Dima Oana-Teodora  
Grupa 241

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

După 11 ani de existență a Asociației Studenților la Matematică și Informatică, președintele hotărăște că trebuie să existe o îmbunătățire considerabilă în managerierea proiectelor, a voluntarilor și a task-urilor. Își dorește ca toate să se afle într-un singur loc, evidența lor să fie ținută în timp real, voluntarii să-și poată îndeplini task-urile mai ușor, iar directorii de departament să scape de rapoartele de activitate întocmite în fiecare lună. De asemenea urmărește păstrarea istoricului asociației, a proiectelor desfășurate de-a lungul timpului, împreună cu edițiile evenimentelor, dar și a membrilor onorifici. Astfel că, în decursul acestui an, apelează la Departamentul IT al FMI pentru crearea unei baze de date corespunzătoare. În urma unei ședințe de brainstorming, membrii departamentului ajung la concluzia că vor fi necesare 6 tabele dintre care unul va fi de legătură. Datele importante ale voluntarilor care se doresc a fi păstrate sunt: numele, prenumele, numărul de telefon și email-ul. Fiecare voluntar poate fi : activ, inactiv, alumn sau președinte și face parte dintr-un singur departament. Fiecare departament este condus de un singur manager, iar fiecare manager la rândul său este condus de președinte. Există o listă de task-uri comune care se aplică proiectelor. De asemenea, există și proiecte care sunt la prima ediție, dar majoritatea proiectelor au avut ediții și în anii precedenți, păstrând formula de desfășurare. Mai mulți voluntari pot lucra la mai multe proiecte, mai ales că unele dintre acestea se pot suprapune, iar fiecare proiect este manageriat de un voluntar, dar același voluntar poate manageria mai multe proiecte.

**2. Realizați diagrama entitate-relație (ERD).****Baza de date Asociația Studenților -> Diagrama Entitate-Relație**

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



Baza de date Asociația Studenților -> **Diagrama conceptuală**

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

5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```

create table volunteer
(
    volunteer_id number(4) constraint PK_VOLUNTEER_ID primary key,
    first_name varchar(20) not null,
    last_name varchar(20) not null,
    email varchar(20) not null,
    phone varchar(10),
    manager_id number(4),
    department_id number(4),
    activity_id number(4)
);
    
```

```
create table departments
( department_id number(4) constraint PK_DEPARTMENT_ID primary key,
  manager_id number(4),
  department_name varchar(20),
  description varchar(50)
);
```

```
--adaug cheile externe
--legatura cu departments
alter table volunteer
add constraint FK_DEPARTEMNT_ID
foreign key (department_id) references departments(department_id);
```

```
--self-joinul de la volunteer
alter table volunteer
add constraint FK_VOLUNTEER_ID
foreign key(manager_id) references volunteer(volunteer_id);
```

```
--departments cu volunteer prin manager_id
alter table departments
add constraint FK_MANAGER_ID
foreign key(manager_id) references volunteer(volunteer_id);
```

```
create table activity
(
  activity_id number(4) constraint PK_ACTIVITY_ID primary key,
  start_date Date,
  end_date Date,
  status_volunteer varchar(10) default 'activ'
);
```

```
--adaug legatura cu volunteer
alter table volunteer
add constraint FK_ACTIVITY_ID
foreign key (activity_id) references activity(activity_id);
```

```
create table tasks
(
  task_id number(4) constraint PK_TASK_ID primary key,
  project_id number(4),
  task_deadline Date,
  task_description varchar(50)
);
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
create table projects
(
project_id number(4) constraint PK_PROJECT_ID primary key,
project_manager number(4),
project_name varchar(20),
project_description varchar(50),
project_deadline date
);
alter table projects
add(edition number(2) not null);

--legatura projects tasks

alter table tasks
add constraint FK_PROJECT_ID
foreign key (project_id) references projects(project_id)
on delete cascade;

--legatura volunteer- projects

alter table projects
add constraint FK_PROJECT_MANAGER
foreign key (project_manager) references volunteer(volunteer_id)
on delete cascade;

create table works_on
(
volunteer_id number(4),
project_id number(4),
start_date date,
end_date date,
platform varchar(20),
activity_mode varchar(20) not null
);
--cheia primara compusa
alter table works_on
add constraint PK_WORKS_ON
primary key(volunteer_id, project_id);

--legatura cu tabela volunteer
alter table works_on
add constraint FK_WROKS_ON_VOLUNTEER
foreign key (volunteer_id) references volunteer(volunteer_id)
on delete cascade;
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
--legatura cu projects
alter table works_on
add constraint FK_WORKS_ON_PROJECTS
foreign key (project_id) references projects(project_id)
on delete cascade;
```

```
--5 departamente
```

```
/*
```

```
10 HR
```

```
11 D&PR
```

```
12 FR
```

```
13 Proiecte
```

```
14 Edu
```

```
*/
```

```
--activity_id numerele de la 101 la 107
```

```
insert into activity
```

```
values (101, '1-OCT-2019', null, 'activ' );
```

```
insert into activity
```

```
values (102, '1-OCT-2019', null, 'inactiv' );
```

```
insert into activity
```

```
values (103, '1-OCT-2009', '1-OCT-2019', 'alumn' );
```

```
insert into activity
```

```
values (104, '1-OCT-2019', null, 'presedinte' );
```

```
insert into activity
```

```
values (105, '1-OCT-2018', '1-NOV-2020', 'alumn' );
```

```
insert into activity
```

```
values (106, '1-NOV-2020', null, 'activ' );
```

```
insert into activity
```

```
values (107, '1-NOV-2019', null, 'inactiv' );
```

```
--departments
```

```
insert into departments(department_id)
```

```
values(10);
```

```
insert into departments(department_id)
```

```
values(11);
```

```
insert into departments(department_id)
```

```
values(12);
```

```
insert into departments(department_id)
```

```
values(13);
```

```
insert into departments(department_id)
```

```
values(14);
```

```
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
```

```
values (1, 'Silistru', 'Delia-Stefania', 'silistru@s.unibuc.ro', '0123456789');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (3, 'Dorneanu', 'Alina-Mihaela', 'dorneanu@s.unibuc.ro', '0113156189');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (4, 'Popa', 'Iulia-Andreea', 'popa@s.unibuc.ro', '0122456722');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (5, 'Bigan', 'Marian-Antonio', 'bigan@s.unibuc.ro', '0100456789');--manager Proiecte
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (6, 'Nimara', 'Dan-Gabriel', 'nimara@s.unibuc.ro', '0183486788');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (7, 'Dirstariu', 'Daria', 'dirstar@s.unibuc.ro', '0125556789');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (8, 'Daineanu', 'Denis', 'daineanu@s.unibuc.ro', '0123456000');--manager design
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (9, 'Marcu', 'Florian', 'marcu@s.unibuc.ro', '0124444489');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (10, 'Fusneica', 'Florin', 'fusneica@s.unibuc.ro', '0120456789');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (11, 'Badescu', 'Gabriel', 'badescu@s.unibuc.ro', '0123456785');--manager edu
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (12, 'Pavalasc', 'Irina', 'pavalasc@s.unibuc.ro', '0122226789');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (13, 'Bertalan', 'Victor', 'bertalan@s.unibuc.ro', '0123456711');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (14, 'Negulescu', 'Radu', 'negules@s.unibuc.ro', '0123400089');--manager fr
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (15, 'Guta', 'Razvan', 'guta@s.unibuc.ro', '0123666789');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (16, 'Vijulan', 'Stefania-Claudia', 'vijulan@s.unibuc.ro', '0123333789');
insert into volunteer(volunteer_id, last_name, first_name, email, phone)
values (2, 'Neagu', 'Andrei-Stefan', 'neagu@s.unibuc.ro', '0123223789');

update departments
set manager_id=1 , department_name ='Human Resources', description='recrutarea noilor membrii si de team
building'
where department_id=10;
update departments
set manager_id=8 , department_name ='Design si PR', description='imaginea asociatiei'
where department_id=11;
update departments
set manager_id=14 , department_name ='Fundraising', description='se ocupa de gasirea sponsorilor'
where department_id=12;
update departments
set manager_id=5 , department_name ='Proiecte', description='se ocupa de orgnizarea petrecerilor'
where department_id=13;
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
update departments
set manager_id=11 , department_name ='Educational', description='se ocupa de relatiile cu profesorii'
where department_id=14;
```

```
update volunteer
set activity_id=104
where volunteer_id=2;--presedinte
```

```
update volunteer
set manager_id=2, department_id=10, activity_id=101
where volunteer_id=1;--manager HR
```

```
update volunteer
set manager_id=1, department_id=10, activity_id=101
where volunteer_id=3;
```

```
update volunteer
set manager_id=1, department_id=10, activity_id=101
where volunteer_id=4;
```

```
update volunteer
set manager_id=2, department_id=13, activity_id=101
where volunteer_id=5;
```

```
update volunteer
set manager_id=5, department_id=13, activity_id=106
where volunteer_id=6;
```

```
update volunteer
set manager_id=5, department_id=13, activity_id=107
where volunteer_id=7;
```

```
update volunteer
set manager_id=2, department_id=11, activity_id=106
where volunteer_id=8;
```

```
update volunteer
set manager_id=8, department_id=11, activity_id=102
where volunteer_id=9;
```

```
update volunteer
set manager_id=8, department_id=11, activity_id=102
where volunteer_id=10;
```

```
update volunteer
set manager_id=2, department_id=14, activity_id=106
```



Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
where volunteer_id=11;
```

```
update volunteer
```

```
set manager_id=11, department_id=14, activity_id=107
```

```
where volunteer_id=12;
```

```
update volunteer
```

```
set manager_id=11, department_id=14, activity_id=101
```

```
where volunteer_id=13;
```

```
update volunteer
```

```
set manager_id=2, department_id=12, activity_id=101
```

```
where volunteer_id=14;
```

```
update volunteer
```

```
set manager_id=14, department_id=12, activity_id=106
```

```
where volunteer_id=15;
```

```
update volunteer
```

```
set manager_id=14, department_id=12, activity_id=106
```

```
where volunteer_id=16;
```

```
--inserare in works_on, projects, tasks
```

```
--proiecte:
```

```
--Recrutari 21
```

```
--Arta'n dar 22
```

```
--TeamBuilding 23
```

```
--Carriere 24
```

```
--SmartHack 25
```

```
--UBSU 26
```

```
insert into projects
```

```
values(21, 3, 'Recrutari', 'alegerea noilor membrii', '1-DEC-2020', 11);
```

```
insert into projects
```

```
values(22, 9, 'Arta-n dar', 'sustinere cauza carotabila', '25-DEC-2020', 5);
```

```
insert into projects
```

```
values(23,10 , 'TeamBuilding', 'creeare comunitate', '1-OCT-2021', 2);
```

```
insert into projects
```

```
values(24,16 , 'Carriere', 'oferire internship', '10-MAY-2021', 5);
```

```
insert into projects
```

```
values(25, 13, 'SmartHack', 'Hackathon FMI', '8-NOV-2021', 20);
```

```
insert into projects
```

```
values(26,7 , 'UBSU', 'atragera viitorilor studenti', '2-MAY-2020', 3);
```

```
insert into projects
```

```
values(27,3 , 'UBSU', 'atragera viitorilor studenti', '2-MAY-2019', 2);
```

```
insert into projects
```

```
values(28,4, 'UBSU', 'atragerea viitorilor studenti', '2-MAY-2018', 1);
```

```
insert into tasks
```

```
values( 200, 21, '29-NOV-2020', 'creare formular de inscriere');
```

```
insert into tasks
```

```
values( 201, 22, '29-NOV-2020', 'impodobire sediu');
```

```
insert into tasks
```

```
values( 202, 22, '1-DEC-2020', 'printare afise');
```

```
insert into tasks
```

```
values( 203, 22, '29-NOV-2020', 'materiale stand');
```

```
insert into tasks
```

```
values( 207, 24, '10-JAN-2021', 'organizare prezentari');
```

```
insert into tasks
```

```
values( 208, 25, '3-NOV-2021', 'printat diplome');
```

```
insert into tasks
```

```
values( 209, 25, '3-NOV-2021', 'inchiriat sala');
```

```
insert into tasks
```

```
values( 210, 25, '1-NOV-2021', 'contact firma catering');
```

```
insert into tasks
```

```
values( 211, 25, '28-OCT-2021', 'trimis invitatii');
```

```
insert into tasks
```

```
values( 212, 25, '29-OCT-2021', 'contactare echipe');
```

```
insert into tasks
```

```
values( 213, 26, '29-JAN-2020', 'contact participantii');
```

```
insert into tasks
```

```
values( 214, 26, '1-MAY-2020', 'imprimare tricouri');
```

```
insert into tasks
```

```
values( 215, 27, '26-JAN-2019', 'contact ASUB');
```

```
insert into tasks
```

```
values( 216, 28, '25-FEB-2018', 'contact profesori');
```

```
insert into tasks
```

```
values( 217, 28, '1-MAY-2018', 'organizare camere');
```

```
select * from tasks;
```

```
select * from projects;
```

```
insert into works_on
```

```
values( 4, 21, '1-NOV-2020', '30-NOV-2020', 'teams', 'online');
```

```
insert into works_on
```

```
values( 3, 21, '2-NOV-2020', '30-NOV-2020', 'Discord', 'online');
```

```
insert into works_on
```

```
values( 5, 21, '3-NOV-2020', '29-NOV-2020', 'teams', 'online');
```

```
insert into works_on
```

```
values( 6, 21, '1-NOV-2020', '28-NOV-2020', 'teams', 'online');
```

```
insert into works_on
```

```
values( 1, 22, '10-NOV-2020', '24-DEC-2020', null, 'fizic');
```

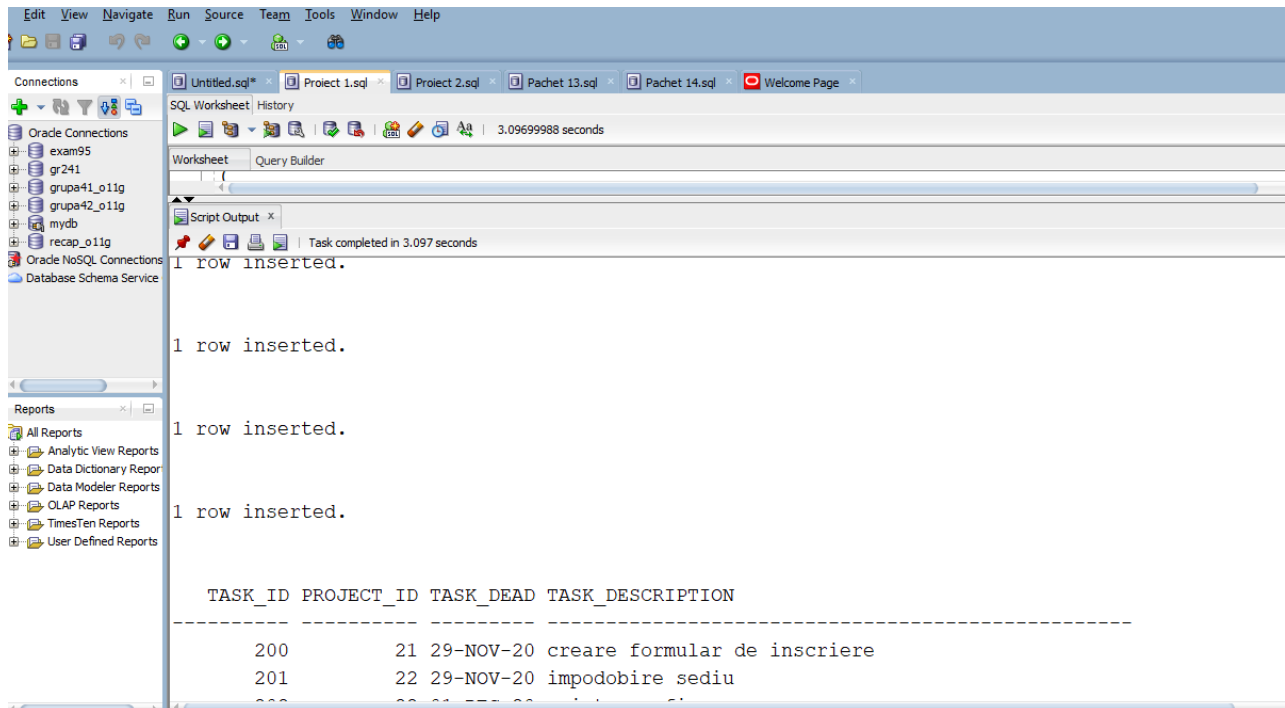
```
insert into works_on
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
values( 13, 22, '10-NOV-2020', '30-NOV-2020', null, 'fizic');
insert into works_on
values( 15, 23, '11-SEP-2021', '31-OCT-2021', null, 'fizic');
insert into works_on
values( 16, 23, '12-SEP-2021', '29-OCT-2021', 'Discord', 'online');
insert into works_on
values( 10, 25, '1-OCT-2021', '8-NOV-2021', null, 'fizic');
insert into works_on
values( 11, 25, '5-OCT-2021', '5-NOV-2021', 'teams', 'online');
insert into works_on
values( 11, 23, '11-SEP-2021', '31-OCT-2021', null, 'fizic');
insert into works_on
values( 4, 22, '15-NOV-2020', '20-DEC-2020', 'Webex', 'online');

insert into projects
values(29, null, 'ProjectASMI', 'introducere boboci in organizare proiecte', null, 1);
insert into departments
values(15, null, 'IT', null);
insert into projects values
(30, null, 'Caravana Edu', 'promovare facultate', null, 1);
commit;
```

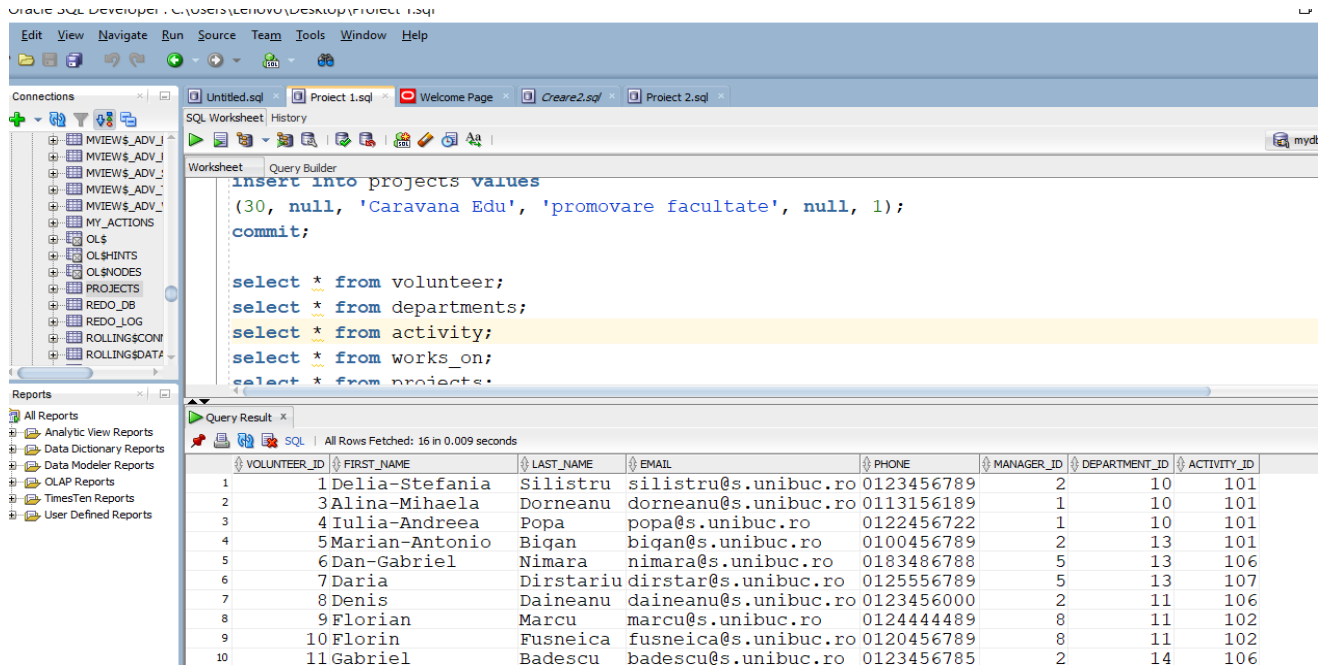


```
select * from volunteer;
select * from departments;
select * from activity;
select * from works_on;
select * from projects;
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

select \* from tasks;



6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

Vreau să șterg și să afișez (împreună cu numele proiectelor de care aparțin) task-urile a căror deadline a expirat, apoi vreau să adaug aceleași task-uri cu deadline-ul modificat (aceeași zi, lună, dar anul următor).

- Colecții utilizate: **tablou indexat, înregistrare**.
- Subprogram de tip **procedură** stocat.

```
CREATE OR REPLACE PROCEDURE punctul_6
IS
```

```
    Type rec IS RECORD
    ( cod tasks.task_id%type,
      prj tasks.project_id%type,
      deadline tasks.task_deadline%type,
      descriere tasks.task_description%type);
```

```
    TYPE tabel IS TABLE OF rec INDEX BY PLS_INTEGER;
    t tabel :=tabel();
```

```
    nume projects.project_name%type;
```

```
BEGIN
```

```
--stergera cu retinerea informatiilor
```

```
DELETE FROM tasks
```

```
WHERE task_deadline< sysdate
```

```
RETURNING task_id, project_id, task_deadline, task_description BULK COLLECT INTO t ;
```

```
DBMS_OUTPUT.PUT_LINE('Lista de task-uri expirate: ');
```

```

FOR i IN t.first..t.last LOOP
    --iau numele proiectului de care apartine task-ul
    SELECT project_name
    INTO nume
    FROM projects
    WHERE project_id=t(i).prj;
    DBMS_OUTPUT.PUT_LINE('Taskul: '|| t(i).descriere ||' apartine de proiectul: '||nume );
END LOOP;

FOR i IN t.first..t.last LOOP
    --reactualizarea tabelii de task-uri
    INSERT INTO tasks
    VALUES(t(i).cod, t(i).prj, t(i).deadline+365*(2021-to_number(to_char(t(i).deadline, 'yyyy'))),
    t(i).descriere);
END LOOP;

EXCEPTION
    --in cazul in care nu avem task-uri in care
    --deadlineul a expirat
    WHEN value_error THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt date de actualizat');
END punctul_6;
/
EXECUTE punctul_6;
SELECT * FROM tasks;
ROLLBACK;
BEFORE:

```

**AFTER:**

Query Result x

SQL | All Rows Fetched: 15 in 0.014 seconds

	TASK_ID	PROJECT_ID	TASK_DEADLINE	TASK_DESCRIPTION
1	200	21	29-NOV-20	creare formulă
2	201	22	29-NOV-20	impodobire sec
3	202	22	01-DEC-20	printare afise
4	203	22	29-NOV-20	materiale star
5	207	24	10-JAN-21	organizare pre
6	208	25	03-NOV-21	printat diplor
7	209	25	03-NOV-21	inchiriat sala
8	210	25	01-NOV-21	contact firma
9	211	25	28-OCT-21	trimis invitati
10	212	25	29-OCT-21	contactare ech
11	213	26	29-JAN-20	contact partic
12	214	26	01-MAY-20	imprimare tric
13	215	27	26-JAN-19	contact ASUB
14	216	28	25-FEB-18	contact profes
15	217	28	01-MAY-18	organizare car

Query Result x

SQL | All Rows Fetched: 15 in 0.005 seconds

	TASK_ID	PROJECT_ID	TASK_DEADLINE	TASK_DESCRIPTION
1	207	24	10-JAN-21	organizare prezentari
2	208	25	03-NOV-21	printat diplome
3	209	25	03-NOV-21	inchiriat sala
4	210	25	01-NOV-21	contact firma catering
5	211	25	28-OCT-21	trimis invitatii
6	212	25	29-OCT-21	contactare echipe
7	200	21	29-NOV-21	creare formular de inscriere
8	201	22	29-NOV-21	impodobire sediu
9	202	22	01-DEC-21	printare afise
10	203	22	29-NOV-21	materiale stand
11	213	26	28-JAN-21	contact participant
12	214	26	01-MAY-21	imprimare tricouri
13	215	27	25-JAN-21	contact ASUB
14	216	28	24-FEB-21	contact profesori
15	217	28	30-APR-21	organizare camere

```
Dbms Output
+ | Buffer Size: 20000
mydb x
Lista de task-uri expirate:
Taskul: creare formular de inscriere apartine de proiectul: Recrutari
Taskul: impodobire sediu apartine de proiectul: Arta-n dar
Taskul: printare afise apartine de proiectul: Arta-n dar
Taskul: materiale stand apartine de proiectul: Arta-n dar
Taskul: contact participanti apartine de proiectul: UBSU
Taskul: imprimare tricouri apartine de proiectul: UBSU
Taskul: contact ASUB apartine de proiectul: UBSU
Taskul: contact profesori apartine de proiectul: UBSU
Taskul: organizare camere apartine de proiectul: UBSU
```

**7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.**

În funcție de o opțiune introdusă de la tastatură (una dintre șirurile activ, inactiv, alumn), deschid un cursor astfel încât să se regăsească:

- lista voluntarilor activi împreună cu numele departamentelor din care fac parte(pentru opțiunea activ);
  - lista voluntarilor inactivi împreună cu data în care au intrat în asociație(pentru opțiunea inactiv);
  - lista alumnilor împreună cu anul în care au ieșit din asociație(pentru opțiunea alumn).
- **Cursor dinamic**
  - Subprogram de tip **procedură** stocat

```
CREATE OR REPLACE PROCEDURE punctul_7(optiune VARCHAR2)
IS
```

```
TYPE rec1 IS RECORD
( nume volunteer.last_name%type,
prenume volunteer.first_name%type,
nume_dep departments.department_name%type);
TYPE rec23 IS RECORD
( nume volunteer.last_name%type,
prenume volunteer.first_name%type,
activ activity.start_date%type);
```

```
TYPE tip1 IS REF CURSOR RETURN rec1;
TYPE tip23 IS REF CURSOR RETURN rec23;
cer1 tip1;
cer23 tip23;
```

```
v1 rec1;
v23 rec23;
ok_found BOOLEAN:=false;
```

BEGIN

IF optiune ='activ' THEN

OPEN cer1 FOR

SELECT last\_name, first\_name, department\_name

FROM volunteer v JOIN departments d ON (v.department\_id=d.department\_id)

JOIN activity a On (v.activity\_id=a.activity\_id)

WHERE LOWER(status\_volunteer)='activ';

--afisare

LOOP

FETCH cer1 INTO v1;

EXIT WHEN cer1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v1.numc || ' ' || v1.prenume || ' ' || v1.numc\_dep);

ok\_found:=true;

END LOOP;

CLOSE cer1;

ELSIF optiune = 'inactiv' THEN

OPEN cer23 FOR

SELECT last\_name, first\_name, start\_date

FROM volunteer v JOIN activity a ON (v.activity\_id=a.activity\_id)

WHERE LOWER(status\_volunteer)='inactiv';

--afisare

LOOP

FETCH cer23 into v23;

EXIT WHEN cer23%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v23.numc||' ' || v23.prenume||' ' || v23.activ);

ok\_found:=true;

END LOOP;

CLOSE cer23;

ELSIF optiune = 'alumn' THEN

OPEN cer23 FOR

SELECT last\_name, first\_name, end\_date

FROM volunteer v JOIN activity a ON (v.activity\_id=a.activity\_id)

WHERE LOWER(status\_volunteer)='alumn';

--afisare

LOOP

FETCH cer23 into v23;

EXIT WHEN cer23%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v23.numc||' ' || v23.prenume ||' ' || v23.activ);

ok\_found:=true;

END LOOP;

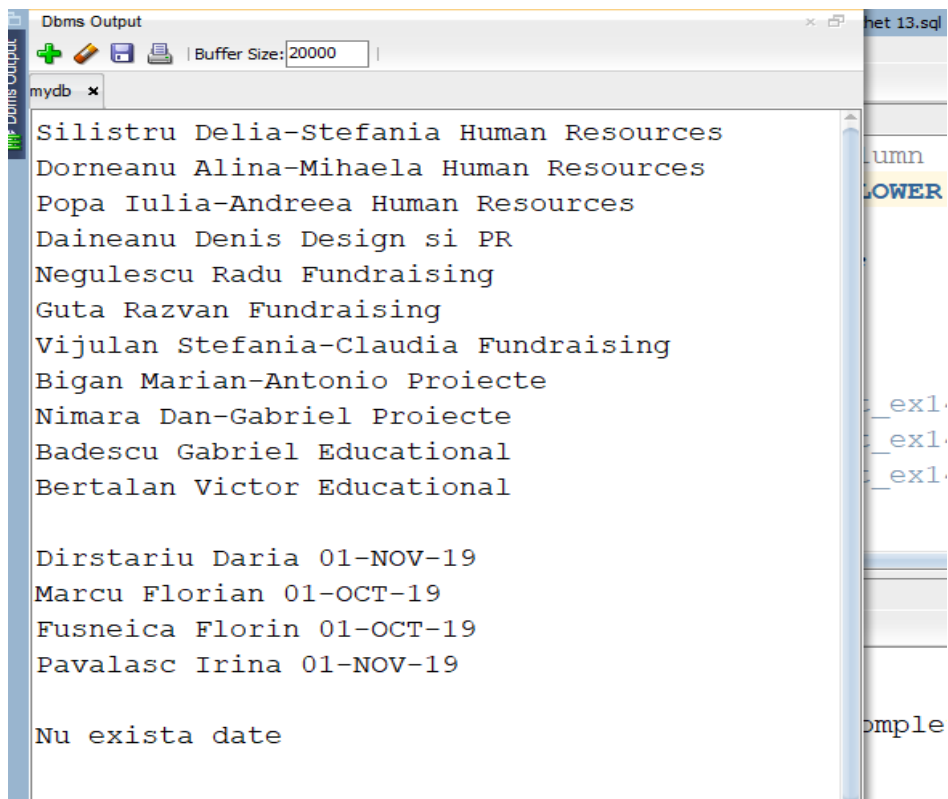
CLOSE cer23;

ELSE

```
DBMS_OUTPUT.PUT_LINE('Optiune incorecta');
END IF;

IF ok_found=false THEN
  DBMS_OUTPUT.PUT_LINE('Nu exista date');
END IF;

END punctul_7;
/
--Apel:
DECLARE
  --activ , inactiv sau alumn
  optiune VARCHAR2(8) := LOWER('&p_optiune');
BEGIN
  punctul_7(optiune);
END;
/
```



```
Dbms Output
+-----+
+ Buffer Size: 20000
+-----+
mydb x
Silistru Delia-Stefania Human Resources
Dorneanu Alina-Mihaela Human Resources
Popa Iulia-Andreea Human Resources
Daineanu Denis Design si PR
Negulescu Radu Fundraising
Guta Razvan Fundraising
Vijulan Stefania-Claudia Fundraising
Bigan Marian-Antonio Proiecte
Nimara Dan-Gabriel Proiecte
Badescu Gabriel Educational
Bertalan Victor Educational

Dirstariu Daria 01-NOV-19
Marcu Florian 01-OCT-19
Fusneica Florin 01-OCT-19
Pavalasc Irina 01-NOV-19

Nu exista date
```

8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Vreau să afișez pentru un departament dat, raportul dintre numărul total de membrii activi și numărul total de membrii din departament și pentru departamentele care au doar membrii activi, să afișez lista membrilor.



dep de Proiecte nu are doar membrii activi

dep 10 Human Resources, 12 Fundraising - au doar membrii activi

pt dep IT am impartire la 0 (0 membrii in departament)=>zero\_divide

+lista vida de membrii => collection\_is\_null

- Tabele: **departments, volunteer, activity**
- Subprogram de tip **funcție** stocat
- Excepții tratate: **no\_data\_found, zero\_divide, collection\_is\_null**

```
CREATE OR REPLACE FUNCTION punctul_8 (dep departments.department_name%type)
RETURN VARCHAR2
```

```
Is
```

```
nr_vol NUMBER(3);
```

```
nr_activ NUMBER(3);
```

```
TYPE vect IS VARRAY(100) OF volunteer.first_name%type;
```

```
v vect;
```

```
nr NUMBER(3);
```

```
id_dep departments.department_id%type;
```

```
raport FLOAT:=0;
```

```
BEGIN
```

```
SELECT COUNT(*)
```

```
INTO nr_vol
```

```
FROM volunteer v JOIN activity a On (v.activity_id=a.activity_id)
```

```
WHERE LOWER(status_volunteer)='activ';
```

```
SELECT d.department_id,COUNT(volunteer_id)
```

```
INTO id_dep, nr
```

```
FROM departments d LEFT JOIN volunteer v ON (d.department_id=v.department_id)
```

```
WHERE LOWER(department_name)=lower(dep)
```

```
GROUP BY d.department_id;
```

```
SELECT COUNT(volunteer_id)
```

```
INTO nr_activ
```

```
FROM volunteer v JOIN activity a ON (v.activity_id=a.activity_id)
```

```
WHERE department_id=id_dep AND LOWER(status_volunteer)='activ';
```

```
raport := round(nr_vol/nr,2);
```

```
IF nr=nr_activ THEN
```

```
v:= vect();
```

```
SELECT first_name BULK COLLECT INTO v
```

```
FROM volunteer v JOIN activity a ON (v.activity_id=a.activity_id)
```

```
WHERE department_id=id_dep AND LOWER(status_volunteer)='activ';
```

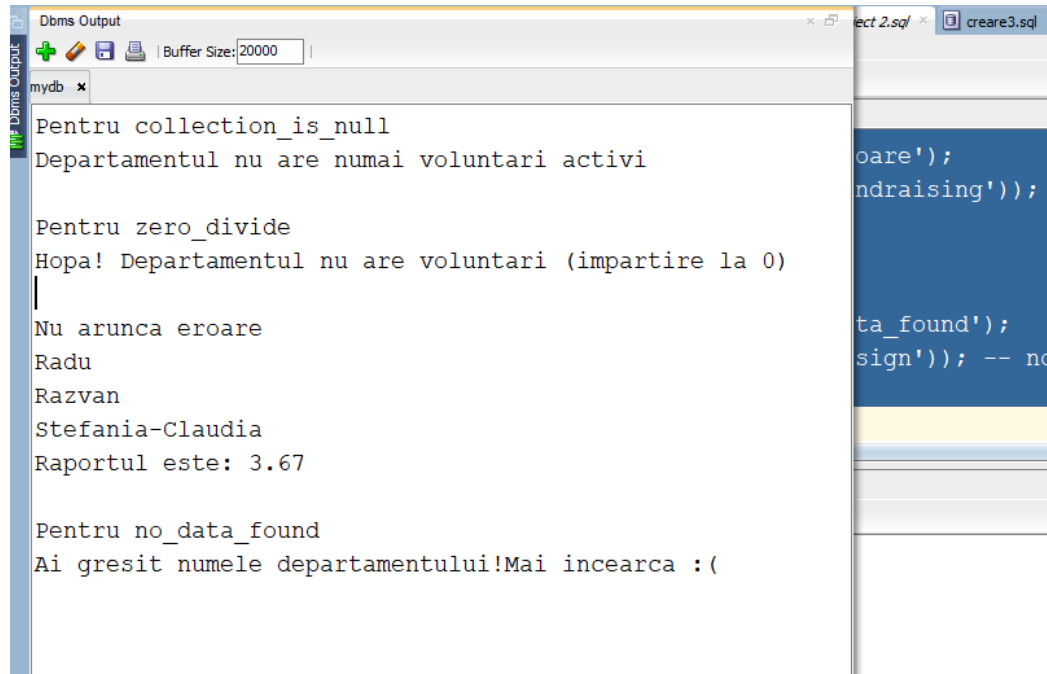
```
END IF;
```

```
FOR i IN v.first..v.last LOOP
    DBMS_OUTPUT.PUT_LINE(v(i));
END LOOP;

RETURN ('Raportul este: ' ||to_char(raport));

EXCEPTION
    WHEN no_data_found THEN
        RETURN 'Ai gresit numele departamentului! Mai incearca :(';
    WHEN zero_divide THEN
        RETURN 'Hopa! Departamentul nu are voluntari (impartire la 0)';
    WHEN collection_is_null THEN
        RETURN 'Departamentul nu are numai voluntari activi';

END punctul_8;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Pentru collection_is_null');
    DBMS_OUTPUT.PUT_LINE(punctul_8('Proiecte')); --collection_is_null
END;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Pentru zero_divide');
    DBMS_OUTPUT.PUT_LINE(punctul_8('IT')); --zero_divide
END;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Nu arunca eroare');
    DBMS_OUTPUT.PUT_LINE(punctul_8('Fundraising')); --ok
END;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Pentru no_data_found');
    DBMS_OUTPUT.PUT_LINE(punctul_8('Design')); -- no_data_found
END;/
```



```
Dbms Output
mydb x
Pentru collection_is_null
Departamentul nu are numai voluntari activi

Pentru zero_divide
Hopa! Departamentul nu are voluntari (impartire la 0)

Nu arunca eroare
Radu
Razvan
Stefania-Claudia
Raportul este: 3.67

Pentru no_data_found
Ai gresit numele departamentului!Mai incearca :(
```

9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Pentru un nume de voluntar introdus de la tastatură afișez lista de task-uri, numele și ediția proiectului pe care îl conduce, departamentul din care face parte și dacă este sau nu alumn (se iau în considerare doar voluntarii care conduc primul lor proiect).

Teambuilding 2 nu are task-uri alocate si este condus de Fusneica=> value\_error

voluntarul Dorneanu a project manageriat 2 proiecte=> too\_many\_rows

voluntarul Badescu nu a project manageriat => no data found

- Tabele: **projects, tasks, volunteer, departments, activity**
- Subprogram tip **procedură** stocat

```
CREATE OR REPLACE PROCEDURE punctul_9 (voluntar volunteer.last_name%type)
IS
```

```
    proiect projects.project_name%type;
    editie projects.edition%type;
```

```
    TYPE tabel IS TABLE OF tasks%rowtype INDEX BY PLS_INTEGER;
```

```
    taskuri tabel;
```

```
    department departments.department_name%type;
```

```
    status activity.status_volunteer%type;
```

```
    cod projects.project_id%type;
```

```
BEGIN
```

```
    SELECT department_name, status_volunteer
```

```
    INTO departament, status
```

```
    FROM departments d JOIN volunteer v ON (d.department_id=v.department_id)
```

```
         JOIN activity a ON (v.activity_id=a.activity_id)
```

```
WHERE UPPER(last_name)=UPPER(voluntar);

SELECT project_name, edition, project_id
INTO proiect, editie, cod
FROM projects
WHERE project_manager =(SELECT volunteer_id
                        FROM volunteer
                        WHERE UPPER(last_name)=UPPER(voluntar));

DBMS_OUTPUT.PUT('Voluntarul: ' ||voluntar|| ' face parte din departamentul: ' ||departament);
IF LOWER(status)='alumn' THEN
    DBMS_OUTPUT.PUT_LINE(' si este alumn');
ELSE
    DBMS_OUTPUT.PUT_LINE(' si nu este alumn');
End If;

SELECT * BULK COLLECT INTO taskuri
FROM tasks
WHERE project_id = cod;

DBMS_OUTPUT.PUT_LINE('Proiectul: ' || proiect ||' editia: ' || editie );

FOR i IN taskuri.first..taskuri.last LOOP
    DBMS_OUTPUT.PUT_LINE(taskuri(i).task_description);
END LOOP;

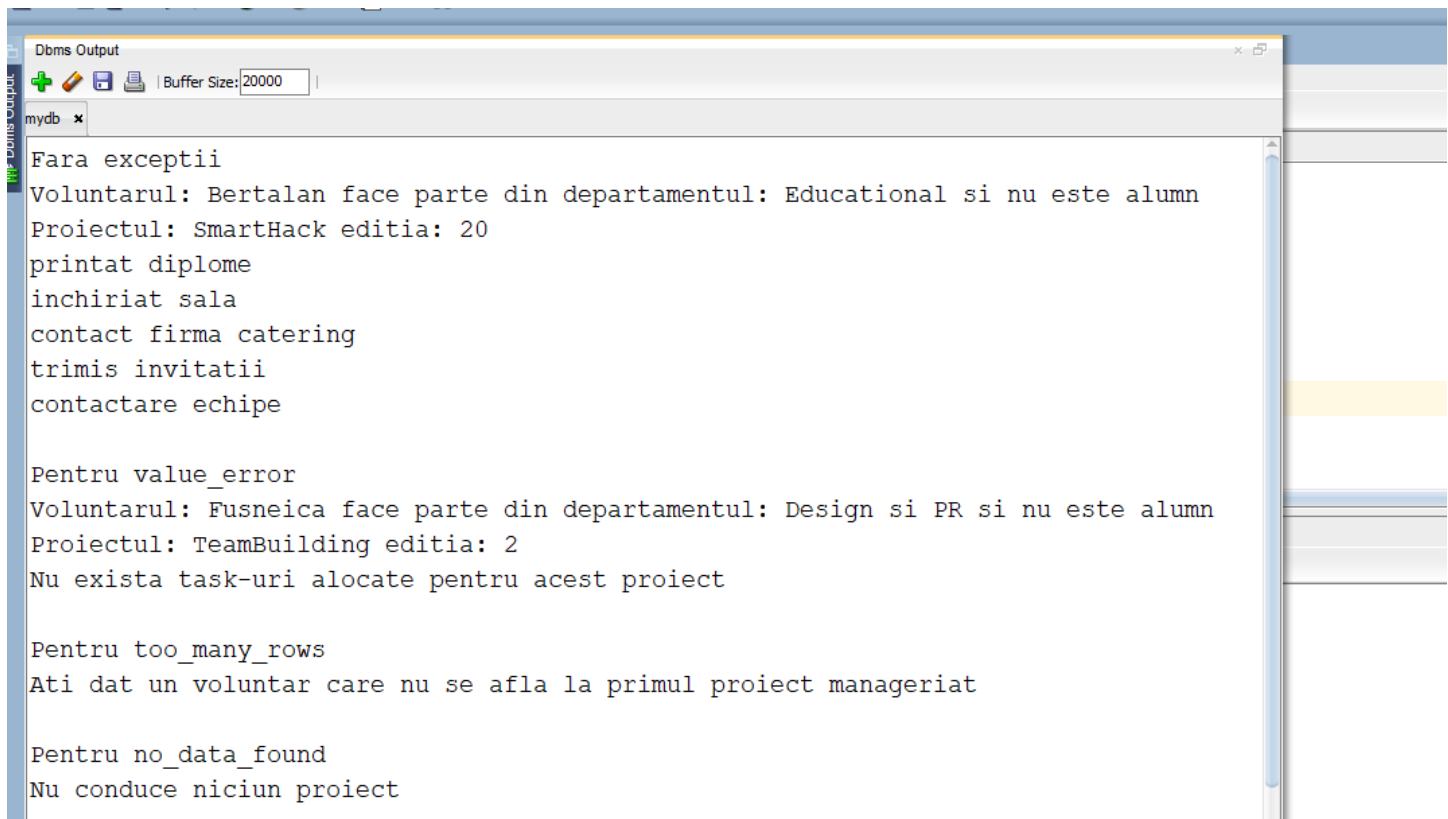
EXCEPTION
    WHEN value_error THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista task-uri alocate pentru acest proiect');
    WHEN no_data_found THEN
        DBMS_OUTPUT.PUT_LINE('Nu conduce niciun proiect');
    WHEN too_many_rows THEN
        DBMS_OUTPUT.PUT_LINE('Ati dat un voluntar care nu se afla la primul proiect manageriat');
END punctul_9;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Fara exceptii');
    punctul_9('Bertalan');--fara eroare
END;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Pentru value_error');
    punctul_9('Fusneica');-- value_error
END;
/
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Pentru too_many_rows');
  punctul_9('Dorneanu');--too_many_rows
END;
/

BEGIN
  DBMS_OUTPUT.PUT_LINE('Pentru no_data_found');
  punctul_9('Badescu');--no_data_found
END;
/
```



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

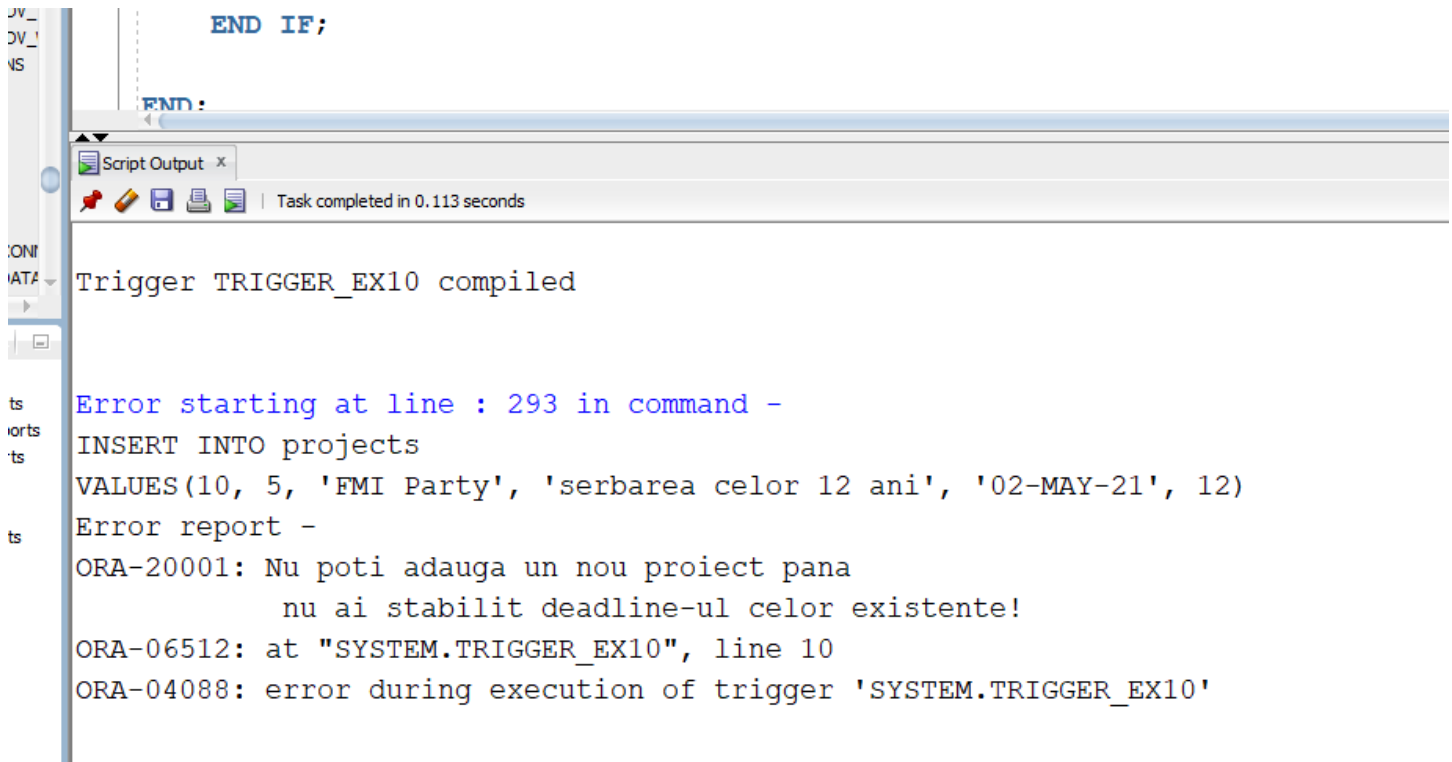
Vreau să setez un trigger care nu lasă să fie inserat un nou proiect în tabelul de proiecte, dacă există cel puțin un proiect care încă nu are setat un deadline.

```
CREATE OR REPLACE TRIGGER trigger_ex10
  BEFORE INSERT ON projects
DECLARE
  nr NUMBER:=0;
BEGIN
  SELECT COUNT(project_id)
  INTO nr
```

```
FROM projects
WHERE project_deadline IS NULL;

IF (nr>0) AND INSERTING THEN
    RAISE_APPLICATION_ERROR(-20001,'Nu poti adauga un nou proiect pana
    nu ai stabilit deadline-ul celor existente!');
END IF;

END;
/
--declansare trigger
INSERT INTO projects
VALUES(10, 5, 'FMI Party', 'serbarea celor 12 ani', '02-MAY-21', 12);
--stergere trigger
DROP TRIGGER trigger_ex10;
```



```
END IF;
END;
Script Output x
Task completed in 0.113 seconds
Trigger TRIGGER_EX10 compiled
Error starting at line : 293 in command -
INSERT INTO projects
VALUES(10, 5, 'FMI Party', 'serbarea celor 12 ani', '02-MAY-21', 12)
Error report -
ORA-20001: Nu poti adauga un nou proiect pana
          nu ai stabilit deadline-ul celor existente!
ORA-06512: at "SYSTEM.TRIGGER_EX10", line 10
ORA-04088: error during execution of trigger 'SYSTEM.TRIGGER_EX10'
```

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

Vreau să setez un trigger care șterge automat voluntarii inactivi în momentul în care data de final a activității (end\_date) a unui voluntar cu status inactiv, este updatată din null în data curentă(înseamnă că voluntarul este dat afară din asociație). Afișez și informații despre aceștia.

- Mă folosesc de **on delete cascade** de la cheile externe.
- Fac o **procedură** prin care șterg voluntarii.

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

```
CREATE OR REPLACE PROCEDURE sterge_voluntar(activ_id activity.activity_id%type)
IS
```

```
    TYPE rec IS RECORD
    (nume volunteer.last_name%type,
    prenume volunteer.first_name%type,
    telefon volunteer.phone%type);
```

```
    TYPE tabel IS TABLE OF rec;
    t tabel:=tabel();
```

```
BEGIN
```

```
    DELETE FROM volunteer
    WHERE activity_id=activ_id
    RETURNING last_name, first_name, phone BULK COLLECT INTO t;
```

```
    FOR i IN t.first..t.last Loop
        DBMS_OUTPUT.PUT_LINE(t(i).nume||' '|| t(i).prenume||' '|| t(i).telefon);
    END LOOP;
```

```
END;
```

```
/
```

```
CREATE OR REPLACE TRIGGER trigger_ex11
```

```
    BEFORE UPDATE of end_date ON activity
```

```
    FOR EACH ROW
```

```
    WHEN (NEW.end_date=sysdate)
```

```
BEGIN
```

```
    sterge_voluntar(:old.activity_id);
```

```
END;
```

```
/
```

```
UPDATE activity
```

```
SET end_date=sysdate
```

```
WHERE LOWER(status_volunteer)='inactiv';
```

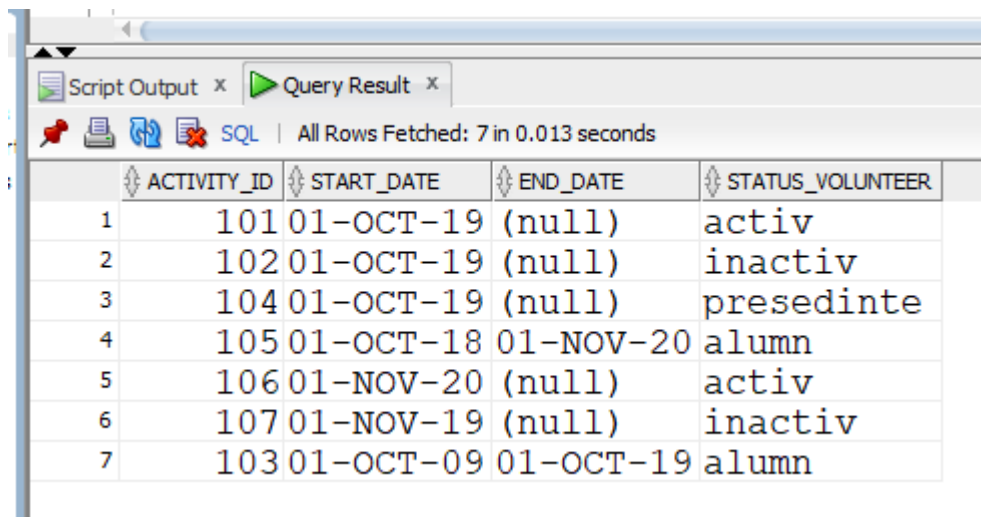
```
SELECT * FROM activity;
```

```
SELECT * FROM volunteer;
```

```
ROLLBACK;
```

```
DROP TRIGGER trigger_ex11;
```

**BEFORE:**



The screenshot shows a SQL query result window with a table containing 7 rows of data. The table has four columns: ACTIVITY\_ID, START\_DATE, END\_DATE, and STATUS\_VOLUNTEER. The data is as follows:

	ACTIVITY_ID	START_DATE	END_DATE	STATUS_VOLUNTEER
1	101	01-OCT-19	(null)	activ
2	102	01-OCT-19	(null)	inactiv
3	104	01-OCT-19	(null)	presedinte
4	105	01-OCT-18	01-NOV-20	alumn
5	106	01-NOV-20	(null)	activ
6	107	01-NOV-19	(null)	inactiv
7	103	01-OCT-09	01-OCT-19	alumn

Script Output x Query Result x

SQL | All Rows Fetched: 16 in 0.003 seconds

VOLUNTEER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	MANAGER_ID	DEPARTMENT_ID	ACTIVITY_ID
1	Delia-Stefania	Silistru	silistru@s.unibuc.ro	0123456789	2	10	101
2	Alina-Mihaela	Dorneanu	dorneanu@s.unibuc.ro	0113156189	1	10	101
3	Iulia-Andreea	Popa	popa@s.unibuc.ro	0122456722	1	10	101
4	Marian-Antonio	Bişan	bişan@s.unibuc.ro	0100456789	2	13	101
5	Dan-Gabriel	Nimara	nimara@s.unibuc.ro	0183486788	5	13	106
6	Daria	Dirstariu	dirstar@s.unibuc.ro	0125556789	5	13	107
7	Denis	Daineanu	daineanu@s.unibuc.ro	0123456000	2	11	106
8	Florian	Marcu	marcu@s.unibuc.ro	0124444489	8	11	102
9	10 Florin	Fusneica	fusneica@s.unibuc.ro	0120456789	8	11	102
10	11 Gabriel	Badescu	badescu@s.unibuc.ro	0123456785	2	14	106
11	12 Irina	Pavalasc	pavalasc@s.unibuc.ro	0122226789	11	14	107
12	13 Victor	Bertalan	bertalan@s.unibuc.ro	0123456711	11	14	101
13	14 Radu	Negulescu	negules@s.unibuc.ro	0123400089	2	12	101
14	15 Razvan	Guta	guta@s.unibuc.ro	0123666789	14	12	106
15	16 Stefania-Claudia	Vijulan	vijulan@s.unibuc.ro	0123333789	14	12	106
16	2 Andrei-Stefan	Neagu	neagu@s.unibuc.ro	0123223789	(null)	(null)	104

**AFTER**

UPDATE activity

Script Output x Query Result x

SQL | All Rows Fetched: 12 in 0.004 seconds

VOLUNTEER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	MANAGER_ID	DEPARTMENT_ID	ACTIVITY_ID
1	Delia-Stefania	Silistru	silistru@s.unibuc.ro	0123456789	2	10	101
2	Alina-Mihaela	Dorneanu	dorneanu@s.unibuc.ro	0113156189	1	10	101
3	Iulia-Andreea	Popa	popa@s.unibuc.ro	0122456722	1	10	101
4	Marian-Antonio	Bişan	bişan@s.unibuc.ro	0100456789	2	13	101
5	Dan-Gabriel	Nimara	nimara@s.unibuc.ro	0183486788	5	13	106
6	8 Denis	Daineanu	daineanu@s.unibuc.ro	0123456000	2	11	106
7	11 Gabriel	Badescu	badescu@s.unibuc.ro	0123456785	2	14	106
8	13 Victor	Bertalan	bertalan@s.unibuc.ro	0123456711	11	14	101
9	14 Radu	Negulescu	negules@s.unibuc.ro	0123400089	2	12	101
10	15 Razvan	Guta	guta@s.unibuc.ro	0123666789	14	12	106
11	16 Stefania-Claudia	Vijulan	vijulan@s.unibuc.ro	0123333789	14	12	106
12	2 Andrei-Stefan	Neagu	neagu@s.unibuc.ro	0123223789	(null)	(null)	104

Script Output x Query Result x

SQL | All Rows Fetched: 7 in 0.008 seconds

ACTIVITY_ID	START_DATE	END_DATE	STATUS_VOLUNTEER
1	101 01-OCT-19	(null)	activ
2	102 01-OCT-19	09-JAN-21	inactiv
3	104 01-OCT-19	(null)	presedinte
4	105 01-OCT-18	01-NOV-20	alumn
5	106 01-NOV-20	(null)	activ
6	107 01-NOV-19	09-JAN-21	inactiv
7	103 01-OCT-09	01-OCT-19	alumn



Marcu Florian	0124444489
Fusneica Florin	0120456789
Dirstariu Daria	0125556789
Pavalasc Irina	0122226789

Bertala
Negules
Guta
Vișulan
Neagu

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

Vreau să setez un trigger care, în tabelul my\_actions pune doar comenzile care au avut loc fără erori, celelalte fiind doar avertizate printr-un mesaj.

```
CREATE TABLE MY_ACTIONS
```

```
(utilizator VARCHAR2(30),  
eveniment VARCHAR2(20),  
data TIMESTAMP(3));
```

```
SELECT * FROM my_actions;
```

```
CREATE OR REPLACE TRIGGER trigger_ex12
```

```
AFTER CREATE OR DROP OR ALTER OR SERVERERROR ON DATABASE
```

```
BEGIN
```

```
IF(DBMS_UTILITY.FORMAT_ERROR_STACK IS NULL) THEN
```

```
INSERT INTO my_actions
```

```
VALUES (SYS.LOGIN_USER, SYS.SYSEVENT, SYSTIMESTAMP);
```

```
ELSE
```

```
RAISE_APPLICATION_ERROR(-20000, 'Au aparut erori in aplicatie :(');
```

```
END IF;
```

```
END;
```

```
/
```

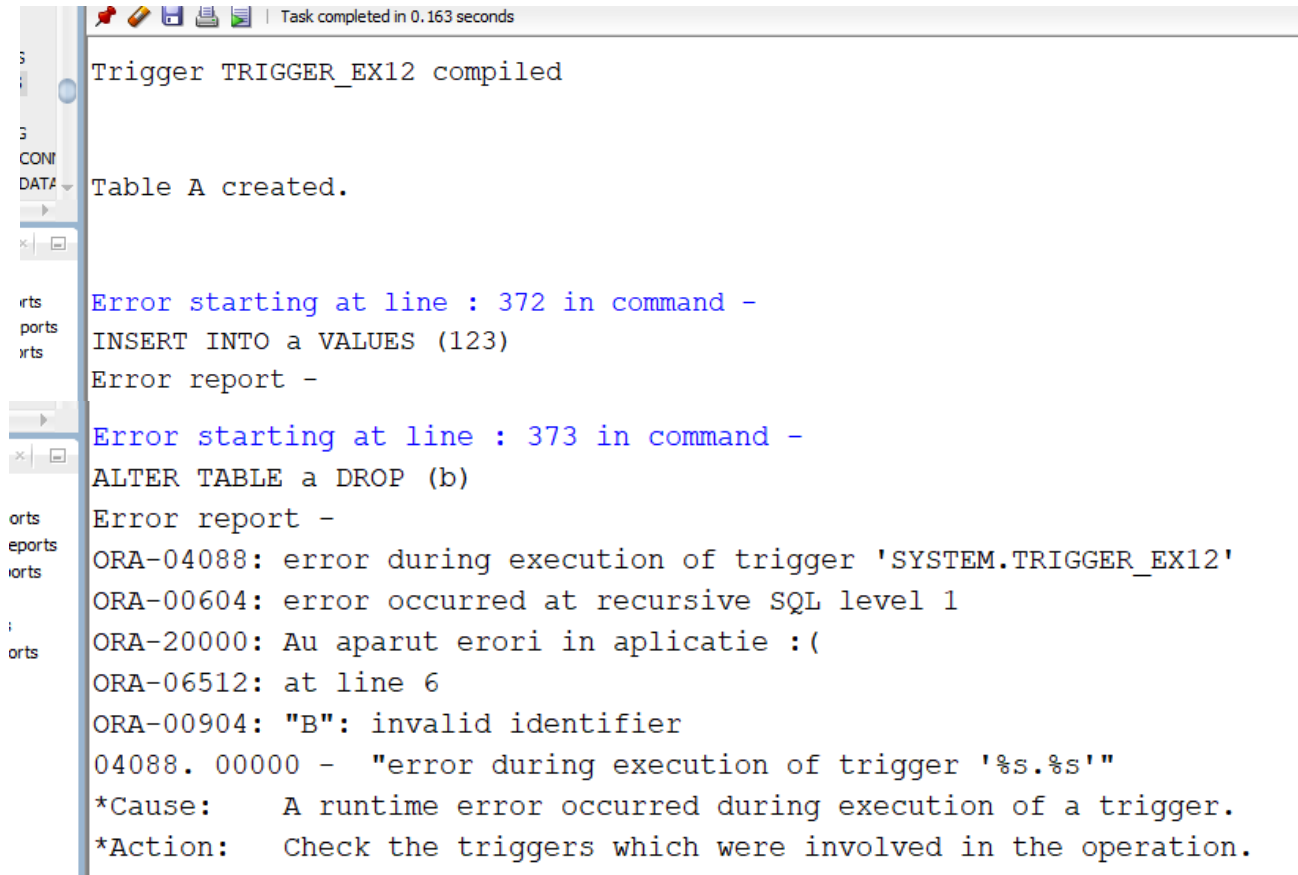
```
DROP TRIGGER trigger_ex12;
```

```
/
```

```
CREATE TABLE a (id NUMBER(2));
```

```
INSERT INTO a VALUES (123);
```

```
ALTER TABLE a DROP (b);
```



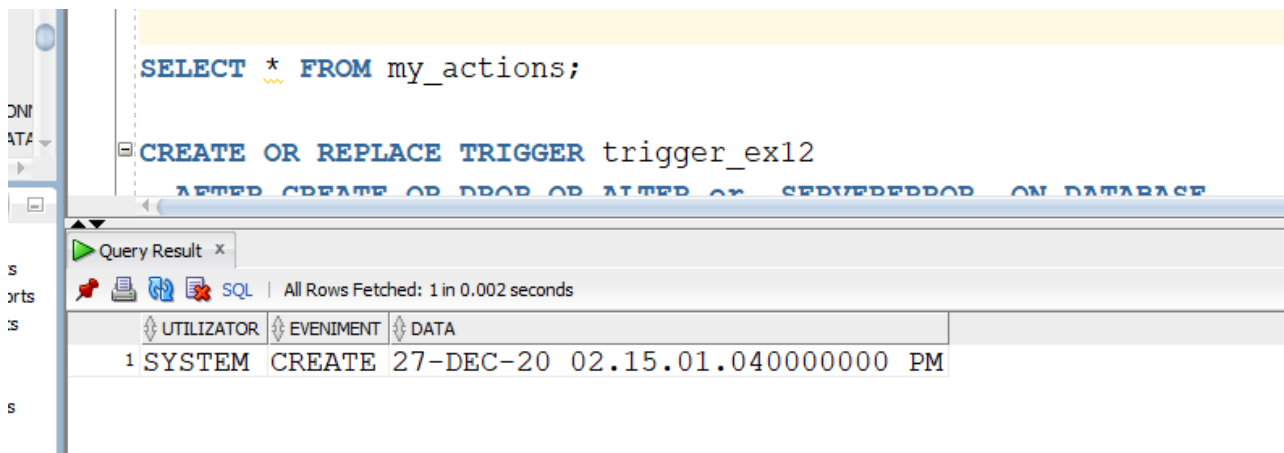
Task completed in 0.163 seconds

```
Trigger TRIGGER_EX12 compiled

Table A created.

Error starting at line : 372 in command -
INSERT INTO a VALUES (123)
Error report -

Error starting at line : 373 in command -
ALTER TABLE a DROP (b)
Error report -
ORA-04088: error during execution of trigger 'SYSTEM.TRIGGER_EX12'
ORA-00604: error occurred at recursive SQL level 1
ORA-20000: Au aparut erori in aplicatie :(
ORA-06512: at line 6
ORA-00904: "B": invalid identifier
04088. 00000 - "error during execution of trigger '%s.%s'"
*Cause:      A runtime error occurred during execution of a trigger.
*Action:     Check the triggers which were involved in the operation.
```



```
SELECT * FROM my_actions;

CREATE OR REPLACE TRIGGER trigger_ex12
AFTER CREATE OR DROP OR ALTER ON DATABASE
```

Query Result x

SQL | All Rows Fetched: 1 in 0.002 seconds

	UTILIZATOR	EVENTIMENT	DATA
1	SYSTEM	CREATE	27-DEC-20 02.15.01.040000000 PM

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

```
CREATE OR REPLACE PACKAGE pachet_ex13 AS
  PROCEDURE punctul_6;
  PROCEDURE punctul_7(optiune VARCHAR2);
  FUNCTION punctul_8(dep departments.department_name%type) RETURN VARCHAR2;
  PROCEDURE punctul_9 (voluntar volunteer.last_name%type);
  PROCEDURE sterge_voluntar(activ_id activity.activity_id%type);
```

Proiect realizat de: Dima Oana-Teodora 241

Baza de date: **Asociația Studenților**

END pachet\_ex13;

/

CREATE OR REPLACE PACKAGE BODY pachet\_ex13 AS

    PROCEDURE punctul\_6

IS

    TYPE rec IS RECORD

    ( cod tasks.task\_id%type,

      prj tasks.project\_id%type,

      deadline tasks.task\_deadline%type,

      descriere tasks.task\_description%type);

    TYPE tabel IS TABLE OF rec INDEX BY PLS\_INTEGER;

    t tabel :=tabel();

    nume projects.project\_name%type;

BEGIN

    --stergereea cu retinerea informatiilor

    DELETE FROM tasks

    WHERE task\_deadline< sysdate

    RETURNING task\_id, project\_id, task\_deadline, task\_description BULK COLLECT INTO t ;

    DBMS\_OUTPUT.PUT\_LINE('Lista de task-uri expirate: ');

    FOR i IN t.first..t.last LOOP

        --iau numele proiectului de care apartine task-ul

        SELECT project\_name

        INTO nume

        FROM projects

        WHERE project\_id=t(i).prj;

        DBMS\_OUTPUT.PUT\_LINE('Taskul: '|| t(i).descriere ||' apartine de proiectul: '||nume );

    END LOOP;

    FOR i IN t.first..t.last LOOP

        --reactualizarea tabelii de task-uri

        INSERT INTO tasks

        VALUES(t(i).cod, t(i).prj, t(i).deadline+365\*(2021-to\_number(to\_char(t(i).deadline, 'yyyy'))),

t(i).descriere);

    END LOOP;

EXCEPTION

    --in cazul in care nu avem task-uri in care

    --deadlineul a expirat

    WHEN value\_error THEN

        DBMS\_OUTPUT.PUT\_LINE('Nu sunt date de actualizat');

```
END punctul_6;
PROCEDURE punctul_7(optiune VARCHAR2)
IS
    TYPE rec1 IS RECORD
    ( nume volunteer.last_name%type,
      prenume volunteer.first_name%type,
      nume_dep departments.department_name%type);
    TYPE rec23 IS RECORD
    ( nume volunteer.last_name%type,
      prenume volunteer.first_name%type,
      activ activity.start_date%type);

    TYPE tip1 IS REF CURSOR RETURN rec1;
    TYPE tip23 IS REF CURSOR RETURN rec23;
    cer1 tip1;
    cer23 tip23;

    v1 rec1;
    v23 rec23;
    ok_found BOOLEAN:=false;

BEGIN
    IF optiune ='activ' THEN
        OPEN cer1 FOR
        SELECT last_name, first_name, department_name
        FROM volunteer v JOIN departments d ON (v.department_id=d.department_id)
        JOIN activity a On (v.activity_id=a.activity_id)
        WHERE LOWER(status_volunteer)='activ';
        --afisare
        LOOP
            FETCH cer1 INTO v1;
            EXIT WHEN cer1%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE(v1.nume || ' ' || v1.prenume || ' ' || v1.nume_dep);
            ok_found:=true;
        END LOOP;
        CLOSE cer1;

    ELSIF optiune = 'inactiv' THEN
        OPEN cer23 FOR
        SELECT last_name, first_name, start_date
        FROM volunteer v JOIN activity a ON (v.activity_id=a.activity_id)
        WHERE LOWER(status_volunteer)='inactiv';
        --afisare
        LOOP
            FETCH cer23 into v23;
```

```
EXIT WHEN cer23%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v23.num||' ' || v23.prenume||' ' || v23.activ);
ok_found:=true;
END LOOP;
CLOSE cer23;

ELSIF optiune = 'alumn' THEN
    OPEN cer23 FOR
    SELECT last_name, first_name, end_date
    FROM volunteer v JOIN activity a ON (v.activity_id=a.activity_id)
    WHERE LOWER(status_volunteer)='alumn';
    --afisare
    LOOP
        FETCH cer23 into v23;
    EXIT WHEN cer23%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v23.num||' ' || v23.prenume ||' ' || v23.activ);
    ok_found:=true;
    END LOOP;
    CLOSE cer23;
ELSE
    DBMS_OUTPUT.PUT_LINE('Optiune incorecta');
END IF;

IF ok_found=false THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista date');
END IF;

END punctul_7;
FUNCTION punctul_8 (dep departments.department_name%type)
RETURN VARCHAR2
Is
    nr_vol NUMBER(3);
    nr_activ NUMBER(3);
    TYPE vect IS VARRAY(100) OF volunteer.first_name%type;
    v vect;

    nr NUMBER(3);
    id_dep departments.department_id%type;
    raport FLOAT:=0;

BEGIN
    SELECT COUNT(*)
    INTO nr_vol
    FROM volunteer v JOIN activity a On (v.activity_id=a.activity_id)
    WHERE LOWER(status_volunteer)='activ';
```

```
SELECT d.department_id,COUNT(volunteer_id)
INTO id_dep, nr
FROM departments d LEFT JOIN volunteer v ON (d.department_id=v.department_id)
WHERE LOWER(department_name)=lower(dep)
GROUP BY d.department_id;
```

```
SELECT COUNT(volunteer_id)
INTO nr_activ
FROM volunteer v JOIN activity a ON (v.activity_id=a.activity_id)
WHERE department_id=id_dep AND LOWER(status_volunteer)='activ';
```

```
raport := round(nr_vol/nr,2);
```

```
IF nr=nr_activ THEN
    v:= vect();
    SELECT first_name BULK COLLECT INTO v
    FROM volunteer v JOIN activity a ON (v.activity_id=a.activity_id)
    WHERE department_id=id_dep AND LOWER(status_volunteer)='activ';
END IF;
```

```
FOR i IN v.first..v.last LOOP
    DBMS_OUTPUT.PUT_LINE(v(i));
END LOOP;
```

```
RETURN ('Raportul este: ' ||to_char(raport));
```

EXCEPTION

```
WHEN no_data_found THEN
    RETURN 'Ai gresit numele departamentului! Mai incearca :(';
WHEN zero_divide THEN
    RETURN 'Hopa! Departamentul nu are voluntari (impartire la 0)';
WHEN collection_is_null THEN
    RETURN 'Departamentul nu are numai voluntari activi';
```

END punctul\_8;

PROCEDURE punctul\_9 (voluntar volunteer.last\_name%type)

Is

```
proiect projects.project_name%type;
editie projects.edition%type;
```

```
TYPE tabel IS TABLE OF tasks%rowtype INDEX BY PLS_INTEGER;
taskuri tabel;
departament departments.department_name%type;
status activity.status_volunteer%type;
```

```
cod projects.project_id%type;
BEGIN
    SELECT department_name, status_volunteer
    INTO departament, status
    FROM departments d JOIN volunteer v ON (d.department_id=v.department_id)
        JOIN activity a ON (v.activity_id=a.activity_id)
    WHERE UPPER(last_name)=UPPER(voluntar);

    SELECT project_name, edition, project_id
    INTO proiect, editie, cod
    FROM projects
    WHERE project_manager =(SELECT volunteer_id
        FROM volunteer
        WHERE UPPER(last_name)=UPPER(voluntar));

    DBMS_OUTPUT.PUT('Voluntarul: '||voluntar|| ' face parte din departamentul: ' ||departament);
    IF LOWER(status)='alumn' THEN
        DBMS_OUTPUT.PUT_LINE(' si este alumn');
    ELSE
        DBMS_OUTPUT.PUT_LINE(' si nu este alumn');
    End If;

    SELECT * BULK COLLECT INTO taskuri
    FROM tasks
    WHERE project_id = cod;

    DBMS_OUTPUT.PUT_LINE('Proiectul: ' || proiect ||' editia: ' || editie );

    FOR i IN taskuri.first..taskuri.last LOOP
        DBMS_OUTPUT.PUT_LINE(taskuri(i).task_description);
    END LOOP;

EXCEPTION
    WHEN value_error THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista task-uri alocate pentru acest proiect');
    WHEN no_data_found THEN
        DBMS_OUTPUT.PUT_LINE('Nu conduce niciun proiect');
    WHEN too_many_rows THEN
        DBMS_OUTPUT.PUT_LINE('Ati dat un voluntar care nu se afla la primul proiect managerial');
END punctul_9;
PROCEDURE sterge_voluntar(activ_id activity.activity_id%type)
IS
    TYPE rec IS RECORD
    (nume volunteer.last_name%type,
    prenume volunteer.first_name%type,
```

```
telefon volunteer.phone%type);

TYPE tabel IS TABLE OF rec;
t tabel:=tabel();
BEGIN

DELETE FROM volunteer
WHERE activity_id=activ_id
RETURNING last_name, first_name, phone BULK COLLECT INTO t;

FOR i IN t.first..t.last Loop
    DBMS_OUTPUT.PUT_LINE(t(i).nume||' '|| t(i).prenume||' '|| t(i).telefon);
END LOOP;
END sterge_voluntar;

END pachet_ex13;
/
--apel pt ex 6
EXECUTE pachet_ex13.punctul_6;
SELECT * FROM tasks;
ROLLBACK;
--apel pt ex 7
DECLARE
    --activ , inactiv sau alumn
    optiune VARCHAR2(8) := LOWER('&p_optiune');
BEGIN
    pachet_ex13.punctul_7(optiune);
END;
/
--apel pt ex 8
BEGIN
    DBMS_OUTPUT.PUT_LINE('Pentru collection_is_null');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.punctul_8('Proiecte')); --collection_is_null
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('Pentru zero_divide');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.punctul_8('IT')); --zero_divide
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('Nu arunca eroare');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.punctul_8('Fundraising')); --ok
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('Pentru no_data_found');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.punctul_8('Design')); -- no_data_found
END;
/
--apel pt ex 9
```



BEGIN

```
DBMS_OUTPUT.PUT_LINE('Fara exceptii');
pachet_ex13.punctul_9('Bertalan');--fara eroare
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('Pentru value_error');
pachet_ex13.punctul_9('Fusneica');-- value_error
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('Pentru too_many_rows');
pachet_ex13.punctul_9('Dorneanu');--too_many_rows
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('Pentru no_data_found');
pachet_ex13.punctul_9('Badescu');--no_data_found
```

END;

/

--triggerul de la ex 11 se foloseste de

--procedura sterge voluntari

CREATE OR REPLACE TRIGGER trigger\_ex11

BEFORE UPDATE of end\_date ON activity

FOR EACH ROW

WHEN (NEW.end\_date=sysdate)

BEGIN

pachet\_ex13.sterge\_voluntar(:old.activity\_id);

END;

/

UPDATE activity

SET end\_date=sysdate

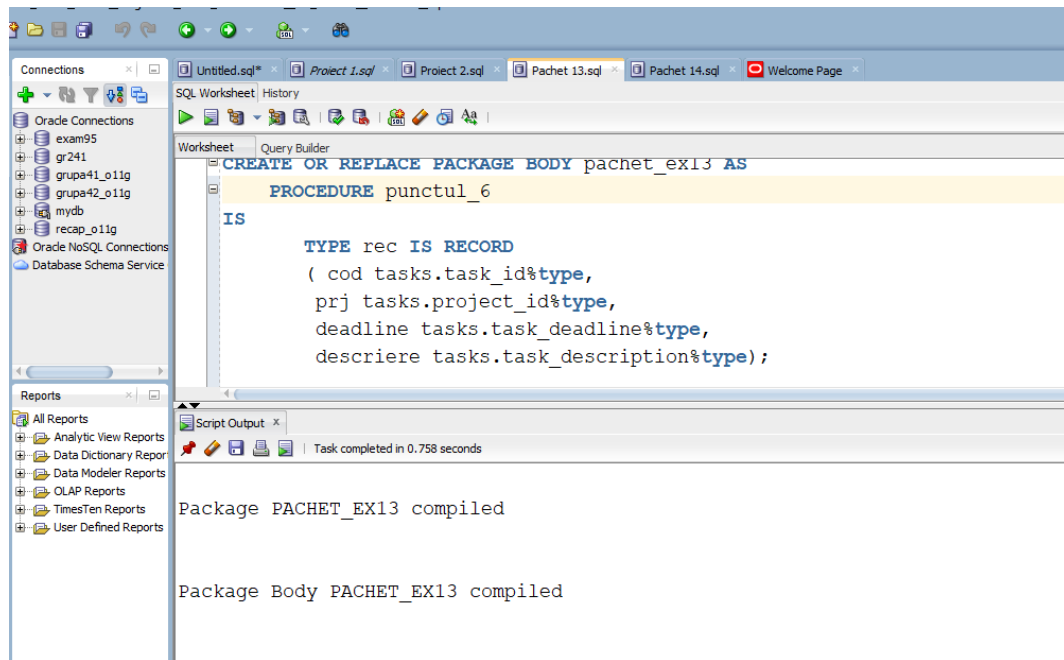
WHERE LOWER(status\_volunteer)='inactiv';

SELECT \* FROM activity;

SELECT \* FROM volunteer;

ROLLBACK;

DROP TRIGGER trigger\_ex11;



#### 14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare pentru acțiuni integrate.

Pentru fiecare voluntar vreau să rețin și să afișez lista proiectelor la care a participat cu anumite mențiuni. Dacă este voluntar activ și are numărul proiectelor la care a participat  $\geq 2$ , atunci îi afișez alături de nume, titlul de membru onorific, altfel în cazul în care este voluntar inactiv și a intrat de cel puțin 2 ani în asociație îl șterg din baza de date doar dacă nr proiectelor la care a participat este  $\leq 1$  și îi afișez un mesaj corespunzător.

- Tip de date complex: **vectori de vectori**

```
CREATE OR REPLACE PACKAGE pachet_info_voluntari AS
    PROCEDURE creare;
    FUNCTION verificare_activ(i NUMBER) RETURN activity.status_volunteer%type;
    PROCEDURE voluntar_activ(i NUMBER);
    FUNCTION ani_vechime(i NUMBER) RETURN BOOLEAN;
    PROCEDURE voluntar_inactiv(i NUMBER);
    PROCEDURE afisare;
END pachet_info_voluntari;
/

CREATE OR REPLACE PACKAGE BODY pachet_info_voluntari AS
--DECLARATII
    TYPE vector IS VARRAY (100) OF VARCHAR2(100);
    TYPE matrice IS VARRAY (100) OF vector;
    --matrice in care retin lista proiectelor pt fiecare voluntar
    mproiecte matrice:=matrice();
    n NUMBER:=0;
    --matrice cu 2 randuri
    --in care retin numele voluntarilor si nr de proiecte
```

```
mnume matrice:=matrice();
```

```
--FUNCTII/PROCEDURI
```

```
--populare matrice cu date
```

```
PROCEDURE creare
```

```
IS
```

```
BEGIN
```

```
--calculez nr de voluntari din tabel
```

```
SELECT COUNT(volunteer_id)
```

```
INTO n
```

```
FROM volunteer;
```

```
mnume.extend(n);
```

```
mproiecte.extend(n);
```

```
--stiu ca id-urile voluntarilor sunt de la 1 la n (16)
```

```
FOR i IN 1..n LOOP
```

```
    mproiecte(i):=vector();
```

```
    mproiecte(i).extend();
```

```
    mnume(i):=vector();
```

```
    mnume(i).extend(2);
```

```
    --iau numele voluntarilor
```

```
    SELECT last_name
```

```
    INTO mnume(i)(1)
```

```
    FROM volunteer
```

```
    WHERE volunteer_id=i;
```

```
    --calculez nr de proiecte ca sa evit exceptia value_error si oprirea programului
```

```
    SELECT to_char(COUNT(project_name))
```

```
    INTO mnume(i)(2)
```

```
    FROM works_on w JOIN projects p ON (w.project_id=p.project_id)
```

```
    WHERE volunteer_id=i;
```

```
    IF mnume(i)(2)!='0' THEN
```

```
        SELECT project_name BULK COLLECT INTO mproiecte(i)
```

```
        FROM works_on w JOIN projects p On (w.project_id=p.project_id)
```

```
        WHERE volunteer_id=i;
```

```
    ELSE
```

```
        mproiecte(i)(1):='Nu are proiecte';
```

```
    END IF;
```

```
END LOOP;
```

```
END creare;
```

```
--returnez statusul voluntarului cu id-ul i (activ/inactiv/alumn)
```

```
FUNCTION verificare_activ(i NUMBER) RETURN activity.status_volunteer%type
```

```
IS
```

```
status activity.status_volunteer%type;
```

```
BEGIN
    SELECT status_volunteer
    INTO status
    FROM activity a JOIN volunteer v ON (a.activity_id=v.activity_id)
    WHERE volunteer_id=i;

    RETURN LOWER(status);
END verificare_activ;
--adaug titlul de membru onorific in nume
PROCEDURE voluntar_activ(i NUMBER)
IS
BEGIN
    IF to_number(mnume(i)(2))>=2 THEN
        mnume(i)(1):=mnume(i)(1) || ' MEMBRU ONORIFIC ';
    END IF;
END voluntar_activ;
--functie care returneaza TRUE daca un voluntar
--are mai mult de 2 ani vechime
FUNCTION ani_vechime(i NUMBER) RETURN BOOLEAN
IS
    ani NUMBER;
BEGIN

    SELECT to_number(to_char(sysdate, 'yyyy'))-to_number(to_char(start_date, 'yyyy'))
    INTO ani
    FROM activity a JOIN volunteer v ON (a.activity_id=v.activity_id)
    WHERE volunteer_id=i;
    IF ani>=2 THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END ani_vechime;
--stergere voluntar i daca nr proiectelor la care a participat este <= 1
PROCEDURE voluntar_inactiv(i NUMBER)
IS
BEGIN
    IF to_number(mnume(i)(2))<=1 THEN
        DELETE FROM volunteer
        WHERE volunteer_id=i;
        mnume(i)(1):= mnume(i)(1) || ' a fost sters ';
    END IF;
END voluntar_inactiv;
--procedura de afisare si apelare functii
PROCEDURE afisare
```

```
IS
BEGIN
creare();
FOR i IN 1..n LOOP
    --un voluntar poate fi activ/inactiv/alumn/presedinte
    --modificarile au loc doar pt cei activi/inactivi
    mnume(i)(1):='Voluntarul '|| verificare_activ(i)|| ' '|| mnume(i)(1);
    IF verificare_activ(i)='activ' THEN
        --voluntar activ
        --daca au participat la cel putin 2 proiecte
        --le adaug la afisare titlul de membru onorific
        voluntar_activ(i);

    ELSIF verificare_activ(i)='inactiv' AND ani_vechime(i)=TRUE THEN
        --voluntar inactiv
        --daca are cel putin 2 ani vechime
        --si maxim 1 proiect la care a participat
        --va fi sters din baza de date
        --si afisat un mesaj corespunzator

        voluntar_inactiv(i);

    END IF;

END LOOP;

FOR i IN 1..n LOOP
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE(mnume(i)(1));
    DBMS_OUTPUT.PUT_LINE('Proiectele la care a participat: ');
    FOR j IN mproiecte(i).first..mproiecte(i).last LOOP
        Dbms_Output.Put_Line(mproiecte(i)(j));
    END LOOP;
END LOOP;
END afisare;

END pachet_info_voluntari;
/
BEGIN
pachet_info_voluntari.afisare();
END;
/
ROLLBACK;
/
```

**BEFORE:**

159 | **ROLLBACK;**

Script Output x Query Result x

All Rows Fetched: 16 in 0.002 seconds

	VOLUNTEER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	MANAGER_ID	DEPARTMENT_ID	ACTIVITY_ID
1	1	Delia-Stefania	Silistru	silistru@s.unibuc.ro	0123456789	2	10	101
2	3	Alina-Mihaela	Dorneanu	dorneanu@s.unibuc.ro	0113156189	1	10	101
3	4	Iulia-Andreea	Popa	popa@s.unibuc.ro	0122456722	1	10	101
4	5	Marian-Antonio	Bișan	bișan@s.unibuc.ro	0100456789	2	13	101
5	6	Dan-Gabriel	Nimara	nimara@s.unibuc.ro	0183486788	5	13	106
6	7	Daria	Dirstariu	dirstar@s.unibuc.ro	0125556789	5	13	107
7	8	Denis	Daineanu	daineanu@s.unibuc.ro	0123456000	2	11	106
8	9	Florian	Marcu	marcu@s.unibuc.ro	0124444489	8	11	102
9	10	Florin	Fusneica	fusneica@s.unibuc.ro	0120456789	8	11	102
10	11	Gabriel	Badescu	badescu@s.unibuc.ro	0123456785	2	14	106
11	12	Irina	Pavalasc	pavalasc@s.unibuc.ro	0122226789	11	14	107
12	13	Victor	Bertalan	bertalan@s.unibuc.ro	0123456711	11	14	101
13	14	Radu	Negulescu	negules@s.unibuc.ro	0123400089	2	12	101
14	15	Razvan	Guta	guta@s.unibuc.ro	0123666789	14	12	106
15	16	Stefania-Claudia	Vijulan	vijulan@s.unibuc.ro	0123333789	14	12	106
16	2	Andrei-Stefan	Neagu	neagu@s.unibuc.ro	0123223789	(null)	(null)	104

**AFTER:**

154 | /

Script Output x Query Result x

All Rows Fetched: 12 in 0.002 seconds

	VOLUNTEER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	MANAGER_ID	DEPARTMENT_ID	ACTIVITY_ID
1	1	Delia-Stefania	Silistru	silistru@s.unibuc.ro	0123456789	2	10	101
2	3	Alina-Mihaela	Dorneanu	dorneanu@s.unibuc.ro	0113156189	1	10	101
3	4	Iulia-Andreea	Popa	popa@s.unibuc.ro	0122456722	1	10	101
4	5	Marian-Antonio	Bișan	bișan@s.unibuc.ro	0100456789	2	13	101
5	6	Dan-Gabriel	Nimara	nimara@s.unibuc.ro	0183486788	5	13	106
6	8	Denis	Daineanu	daineanu@s.unibuc.ro	0123456000	2	11	106
7	11	Gabriel	Badescu	badescu@s.unibuc.ro	0123456785	2	14	106
8	13	Victor	Bertalan	bertalan@s.unibuc.ro	0123456711	11	14	101
9	14	Radu	Negulescu	negules@s.unibuc.ro	0123400089	2	12	101
10	15	Razvan	Guta	guta@s.unibuc.ro	0123666789	14	12	106
11	16	Stefania-Claudia	Vijulan	vijulan@s.unibuc.ro	0123333789	14	12	106
12	2	Andrei-Stefan	Neagu	neagu@s.unibuc.ro	0123223789	(null)	(null)	104

Voluntarul activ Silistru

Proiectele la care a participat:

Arta-n dar

Voluntarul presedinte Neagu

Proiectele la care a participat:

Nu are proiecte

Voluntarul activ Dorneanu

Proiectele la care a participat:

Recrutari

Voluntarul activ Popa MEMBRU ONORIFIC

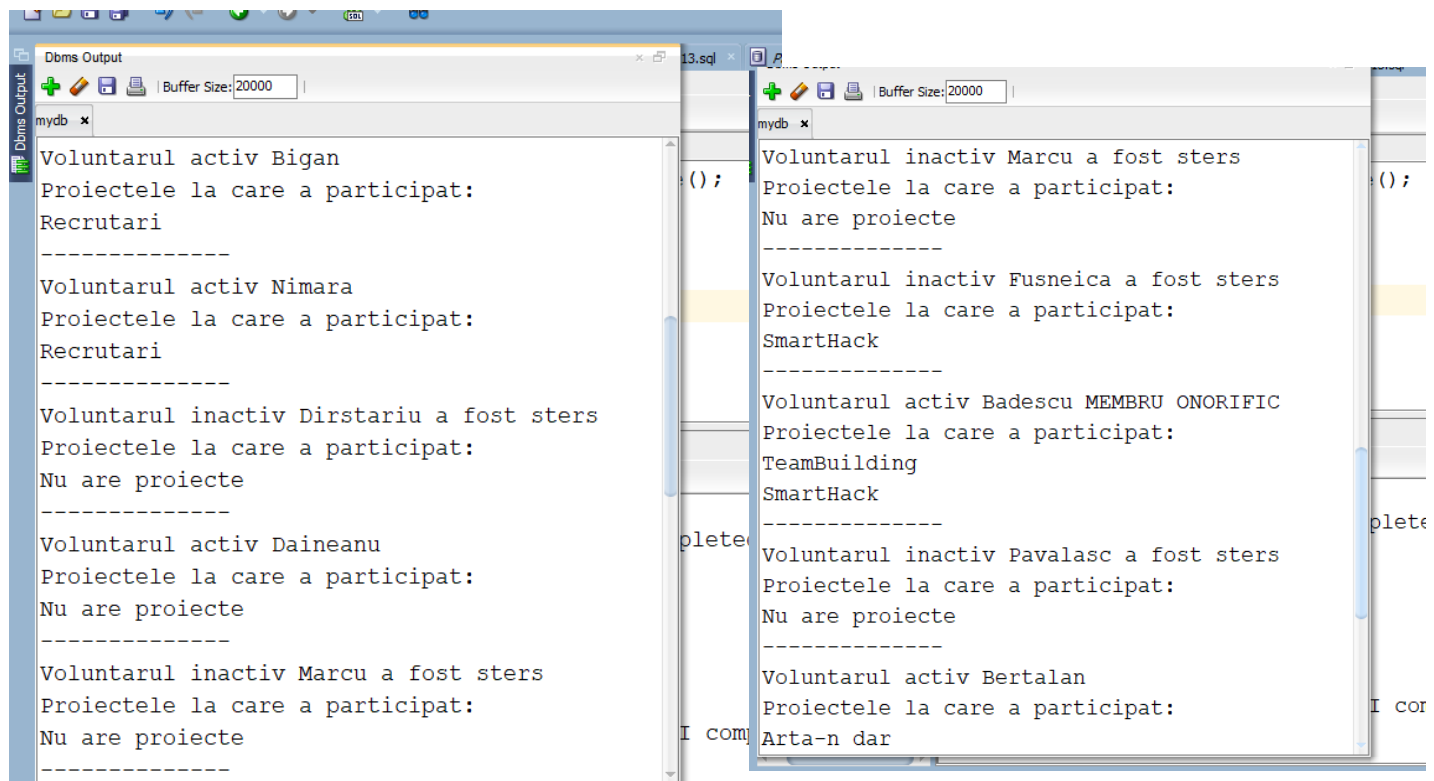
Proiectele la care a participat:

Recrutari

Arta-n dar

Voluntarul activ Bișan

Proiectele la care a participat:



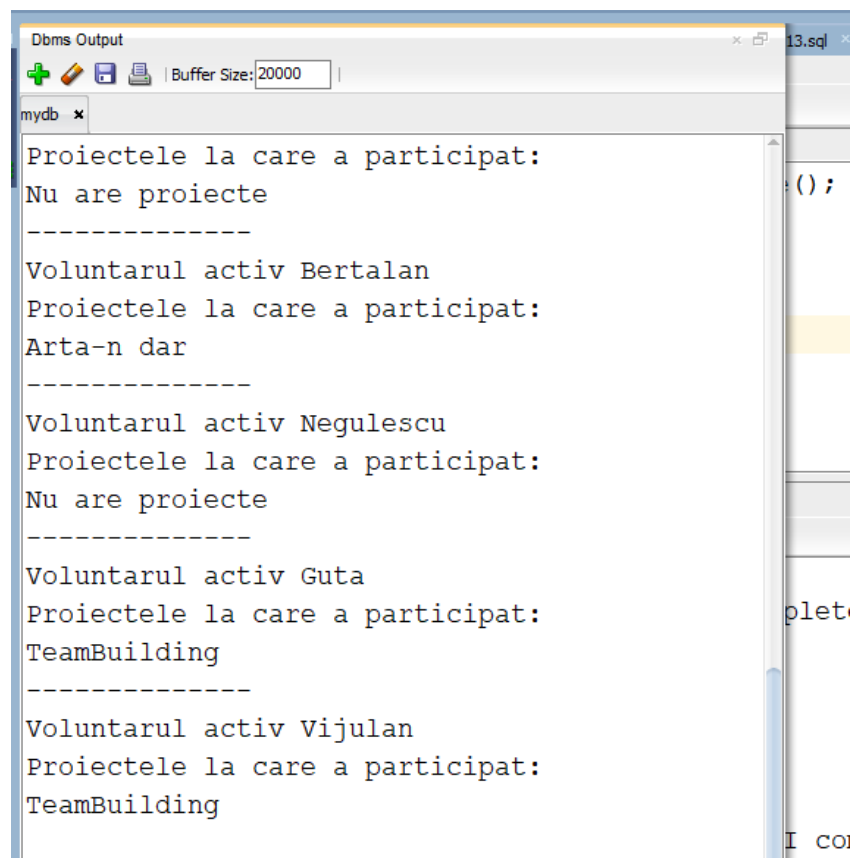
The image shows two side-by-side screenshots of a Dbms Output window. The left window displays results for active volunteers, and the right window displays results for inactive volunteers. Each result block includes the volunteer's name, their status, and a list of projects they have participated in, separated by a dashed line.

**Left Window (Active Volunteers):**

- Voluntarul activ Bigan  
Proiectele la care a participat:  
Recrutari
- Voluntarul activ Nimara  
Proiectele la care a participat:  
Recrutari
- Voluntarul inactiv Dirstariu a fost sters  
Proiectele la care a participat:  
Nu are proiecte
- Voluntarul activ Daineanu  
Proiectele la care a participat:  
Nu are proiecte
- Voluntarul inactiv Marcu a fost sters  
Proiectele la care a participat:  
Nu are proiecte

**Right Window (Inactive Volunteers):**

- Voluntarul inactiv Marcu a fost sters  
Proiectele la care a participat:  
Nu are proiecte
- Voluntarul inactiv Fusneica a fost sters  
Proiectele la care a participat:  
SmartHack
- Voluntarul activ Badescu MEMBRU ONORIFIC  
Proiectele la care a participat:  
TeamBuilding  
SmartHack
- Voluntarul inactiv Pavalasc a fost sters  
Proiectele la care a participat:  
Nu are proiecte
- Voluntarul activ Bertalan  
Proiectele la care a participat:  
Arta-n dar



The image shows a screenshot of a Dbms Output window displaying results for active volunteers. Each result block includes the volunteer's name, their status, and a list of projects they have participated in, separated by a dashed line.

**Active Volunteers:**

- Proiectele la care a participat:  
Nu are proiecte
- Voluntarul activ Bertalan  
Proiectele la care a participat:  
Arta-n dar
- Voluntarul activ Negulescu  
Proiectele la care a participat:  
Nu are proiecte
- Voluntarul activ Guta  
Proiectele la care a participat:  
TeamBuilding
- Voluntarul activ Vijulan  
Proiectele la care a participat:  
TeamBuilding