

Proiect

Sisteme de Gestiune a Bazelor de Date

Baza de date a unui club de robotică

Buruiană Alexandra

An II, Grupa 244

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

Proiectul reprezintă o bază de date pentru un club de robotică, unde sunt incluse tabele pentru datele referitoare la **ELEVI**, **PROFESORI**, **GRUPELE** aferente, **PREZENȚA**, **ECHIPELE** pentru concursuri, lista de **CONCURSURI**, **ÎNSCRIERILE LA CONCURSURI**.

Tabelul pentru **ELEVI** va conține numele și prenumele fiecărui copil, vârsta acestuia, dacă face parte dintr-o echipă sau nu, și, dacă da - din care.

Tabelul pentru **PROFESORI** va conține numele și prenumele fiecărui profesor și de cât timp activează ca angajat al clubului.

Tabelul pentru **GRUPE** va conține indicele grupei, id-ul profesorului care va ține acea ora și ziua săptămânii în care este programată.

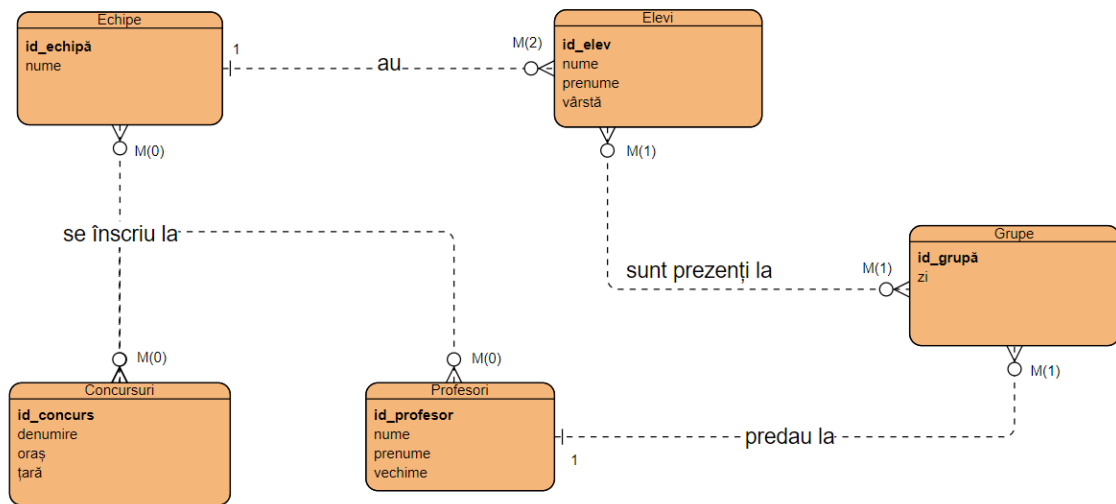
Tabelul pentru **PREZENȚĂ** va conține id-ul elevului, id-ul grupei în care s-a prezentat și data prezenței.

Tabelul pentru **ECHIPE** va conține id-ul și numele echipelor.

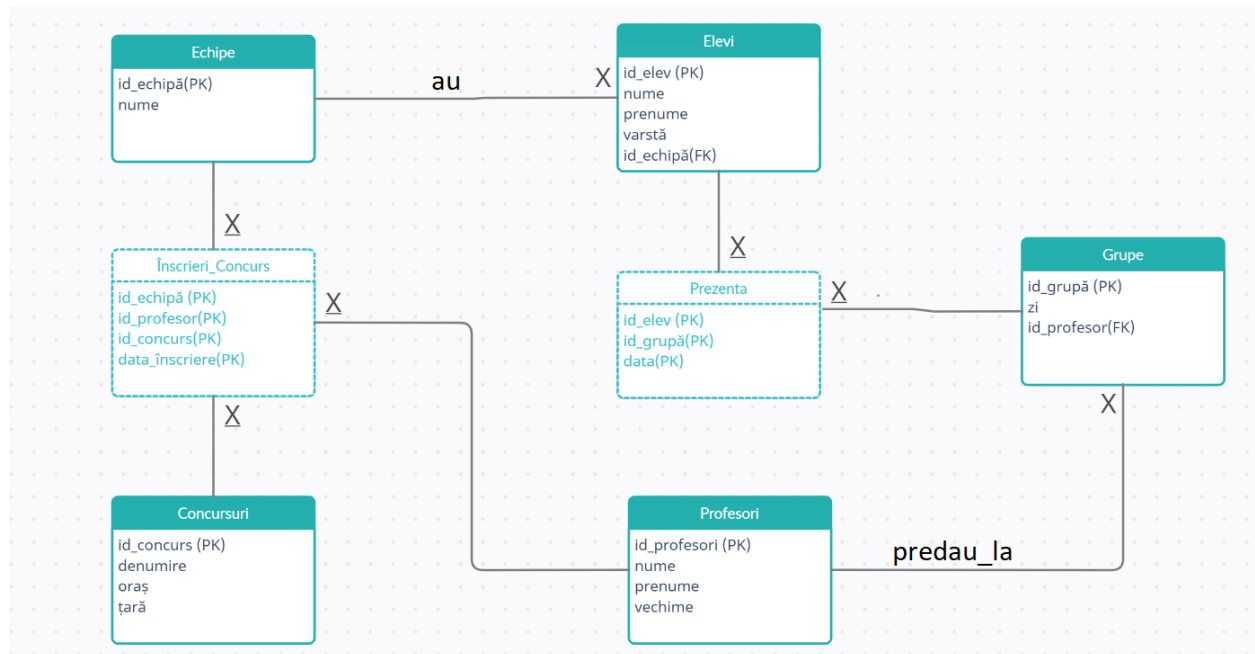
Tabelul pentru **CONCURSURI** va conține denumirea fiecărui concurs, orașul și țara în care este organizat.

Tabelul pentru **ÎNSCRIERI** va conține id-ul echipei, id-ul profesorului care face înscrierea, id-ul concursului și data înscrierii (adaugată deoarece o echipă și un profesor se pot înscrie la un concurs de mai multe ori în date diferite).

2. Realizați Diagrama Entitate-Relație.



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):

```
--creare de tabele
CREATE TABLE elevi (
    id_elev number(3),
    nume varchar2(20),
    prenume varchar2(20),
    varsta number(3),
    id echipa number(3)
);
ALTER TABLE elevi
ADD PRIMARY KEY (id_elev);

select * from elevi;

CREATE TABLE echipe (
    id echipa number(3),
    nume varchar2(20)
);
ALTER TABLE echipe
ADD PRIMARY KEY (id echipa);

select * from echipe;

CREATE TABLE profesori (
    id_profesor number(3),
    nume varchar2(20),
    prenume varchar2(20),
    vechime number(3)
);
ALTER TABLE profesori
ADD PRIMARY KEY (id_profesor);

select * from profesori;

CREATE TABLE grupe (
    id_grupa number(3),
    id_profesor number(3),
    zi varchar2(10)
);
ALTER TABLE grupe
ADD PRIMARY KEY (id_grupa);
```

```

select * from grupe;

CREATE TABLE concursuri (
    id_concurs number(3),
    denumire varchar2(20),
    oras varchar2(20),
    tara varchar2(20)
);
ALTER TABLE concursuri
ADD PRIMARY KEY (id_concurs);

--add foreign key constraints

ALTER TABLE elevi
ADD FOREIGN KEY (id echipa) REFERENCES echipe(id echipa);

ALTER TABLE grupe
ADD FOREIGN KEY (id profesor) REFERENCES profesori(id profesor);

--add tabele asociative

CREATE TABLE prezenta (
    id_elev number(3),
    id_grupa number(3),
    data date,
    CONSTRAINT PK_Prezenta PRIMARY KEY (id_elev,id_grupa,data),
    FOREIGN KEY (id_elev) REFERENCES elevi(id_elev),
    FOREIGN KEY (id_grupa) REFERENCES grupe(id_grupa)
);
CREATE TABLE inscrieri_concurs (
    id echipa number(3),
    id profesor number(3),
    id_concurs number(3),
    data_inscriere date,
    CONSTRAINT PK_Inscriere PRIMARY KEY
(id echipa,id profesor,id_concurs,data_inscriere),
    FOREIGN KEY (id echipa) REFERENCES echipe(id echipa),
    FOREIGN KEY (id profesor) REFERENCES profesori(id profesor),
    FOREIGN KEY (id_concurs) REFERENCES concursuri(id_concurs)
);

```

```

1  --creare de tabele
2  CREATE TABLE elevi (
3      id_elev number(3),
4      nume varchar2(20),
5      prenume varchar2(20),
6      varsta number(3),
7      id echipa number(3)
8  );
9  ALTER TABLE elevi
10 ADD PRIMARY KEY (id_elev);
11

```

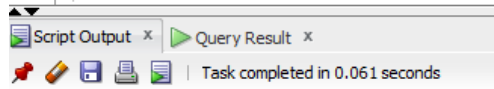


Table ELEVI created.

Table ELEVI altered.

```

34 CREATE TABLE grupe (
35     id_grupa number(3),
36     id_profesor number(3),
37     zi number(3)
38 );
39 ALTER TABLE grupe
40 ADD PRIMARY KEY (id_grupa);
41

```

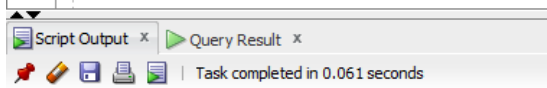


Table GRUPE created.

Table GRUPE altered.

```

14 CREATE TABLE echipa (
15     id echipa number(3),
16     nume varchar2(20)
17 );
18 ALTER TABLE echipa
19 ADD PRIMARY KEY (id echipa);
20

```

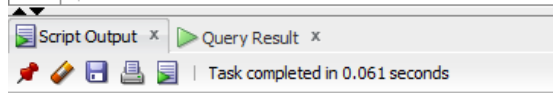


Table ECHIPE created.

Table ECHIPE altered.

```

23 CREATE TABLE profesori (
24     id_profesor number(3),
25     nume varchar2(20),
26     prenume varchar2(20),
27     vechime number(3)
28 );
29 ALTER TABLE profesori
30 ADD PRIMARY KEY (id_profesor);
31

```

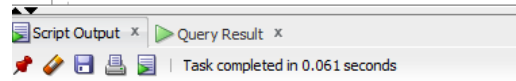


Table PROFESORI created.

Table PROFESORI altered.

```

44 CREATE TABLE concursuri (
45     id_concurs number(3),
46     denumire varchar2(20),
47     locatie varchar2(20)
48 );
49 ALTER TABLE concursuri
50 ADD PRIMARY KEY (id_concurs);
51

```

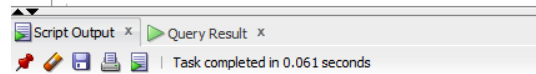


Table CONCURSURI created.

Table CONCURSURI altered.

```

64 CREATE TABLE prezenta (
65     id_elev number(3),
66     id_grupa number(3),
67     data date,
68     CONSTRAINT PK_Prezenta PRIMARY KEY (id_elev,id_grupa,data),
69     FOREIGN KEY (id_elev) REFERENCES elevi(id_elev),
70     FOREIGN KEY (id_grupa) REFERENCES grupe(id_grupa)
71
72 );

```

Script Output x Query Result x
 Task completed in 0.061 seconds

*Cause:
 *Action:

Table PREZENTA created.

```

73 CREATE TABLE inscrieri_concurs (
74     id echipa number(3),
75     id_profesor number(3),
76     id_concurs number(3),
77     data_inscriere date,
78     CONSTRAINT PK_Inscriere PRIMARY KEY (id_echipa,id_profesor,id_concurs,data_inscriere),
79     FOREIGN KEY (id_echipa) REFERENCES echipe(id_echipa),
80     FOREIGN KEY (id_profesor) REFERENCES profesori(id_profesor),
81     FOREIGN KEY (id_concurs) REFERENCES concursuri(id_concurs)
82
83 );

```

Script Output x Query Result x
 Task completed in 0.061 seconds

Table INSCRIERI_CONCURS created.

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

```
--inseram valori in tabele
select * from elevi;
select * from echipe;
select * from profesori;
select * from grupe;
/
insert into profesori values (1,'Buruiana','Alexandra',3);
insert into profesori values (2,'Buruiana','Marinela',10);
insert into profesori values (3,'Andreescu','Ana',2);
insert into profesori values (4,'Vasilache','Andrei',5);
insert into profesori values (5,'Enache','Sorin',2);
/
insert into grupe values (1,2,'LUNI');
insert into grupe values (2,2,'MIERCURI');
insert into grupe values (3,1,'MARTI');
insert into grupe values (4,2,'JOI');
insert into grupe values (5,3,'MARTI');
insert into grupe values (6,1,'VINERI');
insert into grupe values (7,3,'VINERI');
insert into grupe values (8,4,'VINERI');
insert into grupe values (9,5,'LUNI');
/
select * from elevi;
/
insert into elevi values (1,'Popescu','Elena',11,3);
insert into elevi values (2,'Panaite','Iris',12,3);
insert into elevi values (3,'Stoian','Mihai',8,4);
insert into elevi values (4,'Olea','Iarina',10,5);
insert into elevi values (5,'Stoian','Stefan',15,1);
insert into elevi values (6,'Mihai','David',15,1);
insert into elevi values (7,'Calugaru','David',15,1);
insert into elevi values (8,'Miron','Matei',12,null);
insert into elevi values (9,'Ifrim','Teodor',11,5);
insert into elevi values (10,'Butunoi','Matei',11,5);
insert into elevi values (11,'Enaceanu','Iustin',9,2);
insert into elevi values (12,'Elisei','David',10,4);
insert into elevi values (13,'Parvu','Luca',14,null);
insert into elevi values (14,'Musat','Sofia',11,null);
insert into elevi values (15,'Mantu','Ana',8,2);
insert into elevi values (16,'Tudor','Radu',8,2);
insert into elevi values (17,'Chirica','Silviu',9,null);
insert into elevi values (18,'Calin','Andrei',9,null);

insert into echipe values(1,'MICROBOTS');
```



```

insert into echipe values(2,'ROBO1');
insert into echipe values(3,'ROBOT RULER');
insert into echipe values(4,'ROBOTIX');
insert into echipe values(5,'CYBER WHISPER');

select * from concursuri;

insert into concursuri values(1,'WORLD ROBOT OLYMPIAD','San
Jose','Costa Rica');
insert into concursuri values(2,'FTC','Detroit','SUA');
insert into concursuri values(3,'STELELE
STIINTEI','Bucharest','Romania');
insert into concursuri values(4,'FLL','Denver','SUA');
insert into concursuri values(5,'WRO2020x','Montreal','Canada');

commit;

select * from prezenta;
insert into prezenta values
(1,1,to_date('05/10/2020','DD/MM/YYYY'));
insert into prezenta values
(16,9,to_date('05/10/2020','DD/MM/YYYY'));
insert into prezenta values
(4,3,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(9,3,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(10,3,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(5,5,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(6,5,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(7,5,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(15,3,to_date('06/10/2020','DD/MM/YYYY'));
insert into prezenta values
(11,2,to_date('07/10/2020','DD/MM/YYYY'));
insert into prezenta values
(17,2,to_date('07/10/2020','DD/MM/YYYY'));
insert into prezenta values
(12,4,to_date('08/10/2020','DD/MM/YYYY'));
insert into prezenta values
(3,4,to_date('08/10/2020','DD/MM/YYYY'));
insert into prezenta values
(18,4,to_date('08/10/2020','DD/MM/YYYY'));

```

```

insert into prezenta values
(2,6,to_date('09/10/2020','DD/MM/YYYY'));
insert into prezenta values
(8,6,to_date('09/10/2020','DD/MM/YYYY'));
insert into prezenta values
(13,8,to_date('09/10/2020','DD/MM/YYYY'));
insert into prezenta values
(14,7,to_date('09/10/2020','DD/MM/YYYY'));
insert into prezenta values
(1,1,to_date('12/10/2020','DD/MM/YYYY'));
insert into prezenta values
(16,9,to_date('12/10/2020','DD/MM/YYYY'));
insert into prezenta values
(4,3,to_date('13/10/2020','DD/MM/YYYY'));
insert into prezenta values
(10,3,to_date('13/10/2020','DD/MM/YYYY'));
insert into prezenta values
(5,5,to_date('13/10/2020','DD/MM/YYYY'));
insert into prezenta values
(6,5,to_date('13/10/2020','DD/MM/YYYY'));
insert into prezenta values
(15,3,to_date('13/10/2020','DD/MM/YYYY'));
insert into prezenta values
(11,2,to_date('14/10/2020','DD/MM/YYYY'));
insert into prezenta values
(17,2,to_date('14/10/2020','DD/MM/YYYY'));
insert into prezenta values
(12,4,to_date('15/10/2020','DD/MM/YYYY'));
insert into prezenta values
(3,4,to_date('15/10/2020','DD/MM/YYYY'));
insert into prezenta values
(18,4,to_date('15/10/2020','DD/MM/YYYY'));
insert into prezenta values
(2,6,to_date('16/10/2020','DD/MM/YYYY'));
insert into prezenta values
(8,6,to_date('16/10/2020','DD/MM/YYYY'));
insert into prezenta values
(14,7,to_date('16/10/2020','DD/MM/YYYY'));
insert into prezenta values
(1,1,to_date('19/10/2020','DD/MM/YYYY'));
insert into prezenta values
(4,3,to_date('20/10/2020','DD/MM/YYYY'));
insert into prezenta values
(9,3,to_date('20/10/2020','DD/MM/YYYY'));
insert into prezenta values
(10,3,to_date('20/10/2020','DD/MM/YYYY'));

```

```

insert into presenta values
(5,5,to_date('20/10/2020','DD/MM/YYYY'));
insert into presenta values
(11,2,to_date('21/10/2020','DD/MM/YYYY'));
insert into presenta values
(17,2,to_date('21/10/2020','DD/MM/YYYY'));
insert into presenta values
(12,4,to_date('22/10/2020','DD/MM/YYYY'));
insert into presenta values
(2,6,to_date('23/10/2020','DD/MM/YYYY'));
insert into presenta values
(8,6,to_date('23/10/2020','DD/MM/YYYY'));
insert into presenta values
(13,8,to_date('23/10/2020','DD/MM/YYYY'));
insert into presenta values
(14,7,to_date('23/10/2020','DD/MM/YYYY'));
insert into presenta values
(1,1,to_date('26/10/2020','DD/MM/YYYY'));
insert into presenta values
(16,9,to_date('26/10/2020','DD/MM/YYYY'));
insert into presenta values
(4,3,to_date('27/10/2020','DD/MM/YYYY'));
insert into presenta values
(7,5,to_date('27/10/2020','DD/MM/YYYY'));
insert into presenta values
(15,3,to_date('27/10/2020','DD/MM/YYYY'));
insert into presenta values
(17,2,to_date('28/10/2020','DD/MM/YYYY'));
insert into presenta values
(12,4,to_date('29/10/2020','DD/MM/YYYY'));
insert into presenta values
(3,4,to_date('29/10/2020','DD/MM/YYYY'));
insert into presenta values
(18,4,to_date('29/10/2020','DD/MM/YYYY'));
insert into presenta values
(2,6,to_date('30/10/2020','DD/MM/YYYY'));
insert into presenta values
(8,6,to_date('30/10/2020','DD/MM/YYYY'));
insert into presenta values
(13,8,to_date('30/10/2020','DD/MM/YYYY'));
insert into presenta values
(14,7,to_date('30/10/2020','DD/MM/YYYY'));

insert into presenta values
(1,1,to_date('02/11/2020','DD/MM/YYYY'));
insert into presenta values
(16,9,to_date('02/11/2020','DD/MM/YYYY'));

```

```

insert into prezenta values
(10,3,to_date('03/11/2020','DD/MM/YYYY'));
insert into prezenta values
(5,5,to_date('03/11/2020','DD/MM/YYYY'));
insert into prezenta values
(6,5,to_date('03/11/2020','DD/MM/YYYY'));
insert into prezenta values
(7,5,to_date('03/11/2020','DD/MM/YYYY'));
insert into prezenta values
(15,3,to_date('03/11/2020','DD/MM/YYYY'));
insert into prezenta values
(11,2,to_date('04/11/2020','DD/MM/YYYY'));
insert into prezenta values
(17,2,to_date('04/11/2020','DD/MM/YYYY'));
insert into prezenta values
(12,4,to_date('05/11/2020','DD/MM/YYYY'));
insert into prezenta values
(3,4,to_date('05/11/2020','DD/MM/YYYY'));
insert into prezenta values
(18,4,to_date('05/11/2020','DD/MM/YYYY'));
insert into prezenta values
(2,6,to_date('06/11/2020','DD/MM/YYYY'));
insert into prezenta values
(8,6,to_date('06/11/2020','DD/MM/YYYY'));
insert into prezenta values
(13,8,to_date('06/11/2020','DD/MM/YYYY'));
insert into prezenta values
(14,7,to_date('06/11/2020','DD/MM/YYYY'));
insert into prezenta values
(16,9,to_date('09/11/2020','DD/MM/YYYY'));
insert into prezenta values
(4,3,to_date('10/11/2020','DD/MM/YYYY'));
insert into prezenta values
(9,3,to_date('10/11/2020','DD/MM/YYYY'));
insert into prezenta values
(5,5,to_date('10/11/2020','DD/MM/YYYY'));
insert into prezenta values
(7,5,to_date('10/11/2020','DD/MM/YYYY'));
insert into prezenta values
(15,3,to_date('10/11/2020','DD/MM/YYYY'));
insert into prezenta values
(11,2,to_date('11/11/2020','DD/MM/YYYY'));
insert into prezenta values
(17,2,to_date('11/11/2020','DD/MM/YYYY'));
insert into prezenta values
(12,4,to_date('12/11/2020','DD/MM/YYYY'));

```

```

insert into prezenta values
(8,6,to_date('13/11/2020','DD/MM/YYYY'));
insert into prezenta values
(13,8,to_date('13/11/2020','DD/MM/YYYY'));
insert into prezenta values
(14,7,to_date('13/11/2020','DD/MM/YYYY'));
insert into prezenta values
(1,1,to_date('16/11/2020','DD/MM/YYYY'));
insert into prezenta values
(16,9,to_date('16/11/2020','DD/MM/YYYY'));
insert into prezenta values
(4,3,to_date('17/11/2020','DD/MM/YYYY'));
insert into prezenta values
(6,5,to_date('17/11/2020','DD/MM/YYYY'));
insert into prezenta values
(15,3,to_date('17/11/2020','DD/MM/YYYY'));
insert into prezenta values
(11,2,to_date('18/11/2020','DD/MM/YYYY'));
insert into prezenta values
(17,2,to_date('18/11/2020','DD/MM/YYYY'));
insert into prezenta values
(3,4,to_date('19/11/2020','DD/MM/YYYY'));
insert into prezenta values
(18,4,to_date('19/11/2020','DD/MM/YYYY'));
insert into prezenta values
(2,6,to_date('20/11/2020','DD/MM/YYYY'));
insert into prezenta values
(8,6,to_date('20/11/2020','DD/MM/YYYY'));
insert into prezenta values
(13,8,to_date('20/11/2020','DD/MM/YYYY'));
insert into prezenta values
(14,7,to_date('20/11/2020','DD/MM/YYYY'));
insert into prezenta values
(1,1,to_date('23/11/2020','DD/MM/YYYY'));
insert into prezenta values
(16,9,to_date('23/11/2020','DD/MM/YYYY'));
insert into prezenta values
(4,3,to_date('24/11/2020','DD/MM/YYYY'));
insert into prezenta values
(9,3,to_date('24/11/2020','DD/MM/YYYY'));
insert into prezenta values
(10,3,to_date('24/11/2020','DD/MM/YYYY'));
insert into prezenta values
(5,5,to_date('24/11/2020','DD/MM/YYYY'));
insert into prezenta values
(6,5,to_date('24/11/2020','DD/MM/YYYY'));

```

```

insert into prezenta values
(7,5,to_date('24/11/2020','DD/MM/YYYY'));
insert into prezenta values
(15,3,to_date('24/11/2020','DD/MM/YYYY'));
insert into prezenta values
(11,2,to_date('25/11/2020','DD/MM/YYYY'));
insert into prezenta values
(17,2,to_date('25/11/2020','DD/MM/YYYY'));
insert into prezenta values
(12,4,to_date('26/11/2020','DD/MM/YYYY'));
insert into prezenta values
(3,4,to_date('26/11/2020','DD/MM/YYYY'));
insert into prezenta values
(18,4,to_date('26/11/2020','DD/MM/YYYY'));
insert into prezenta values
(2,6,to_date('27/11/2020','DD/MM/YYYY'));
insert into prezenta values
(8,6,to_date('27/11/2020','DD/MM/YYYY'));
insert into prezenta values
(13,8,to_date('27/11/2020','DD/MM/YYYY'));
insert into prezenta values
(14,7,to_date('27/11/2020','DD/MM/YYYY'));
commit;
select * from inscrieri_concurs;
select * from concursuri;
select * from echipe;
select * from profesori;

insert into inscrieri_concurs values
(1,2,2,to_date('01/09/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(1,3,1,to_date('12/11/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(1,1,5,to_date('12/10/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(1,2,2,to_date('20/11/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(5,2,5,to_date('22/10/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(5,2,4,to_date('30/10/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(5,1,5,to_date('12/11/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(4,2,4,to_date('14/11/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(4,2,3,to_date('11/10/2020','DD/MM/YYYY'));

```

```

insert into inscrieri_concurs values
(2,1,4,to_date('03/11/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(2,1,3,to_date('05/10/2020','DD/MM/YYYY'));
insert into inscrieri_concurs values
(3,5,1,to_date('12/11/2020','DD/MM/YYYY'));

```

```

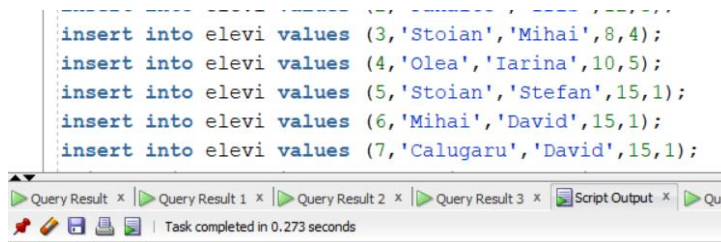
select * from elevi; --entitate independenta
select * from grupe; --entitate independenta
select * from profesori; --entitate independenta
select * from prezenta; --tabela asociativa
select * from echipe; --entitate independenta
select * from concursuri; --entitate independenta
select * from inscrieri_concurs; --tabela asociativa

```

```

insert into elevi values (3,'Stoian','Mihai',8,4);
insert into elevi values (4,'Olea','Iarina',10,5);
insert into elevi values (5,'Stoian','Stefan',15,1);
insert into elevi values (6,'Mihai','David',15,1);
insert into elevi values (7,'Calugaru','David',15,1);

```



```

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

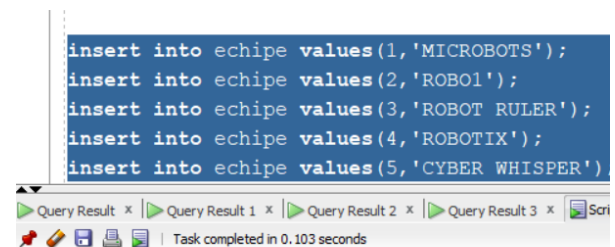
1 row inserted.

```

```

insert into echipe values(1,'MICROBOTS');
insert into echipe values(2,'ROBO1');
insert into echipe values(3,'ROBOT RULER');
insert into echipe values(4,'ROBOTIX');
insert into echipe values(5,'CYBER WHISPER')

```



```

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

```

```

insert into profesori values (1,'Buruiana','Alexandra',3);
insert into profesori values (2,'Buruiana','Marinela',10);
insert into profesori values (3,'Andreescu','Ana',2);
insert into profesori values (4,'Vasilache','Andrei',5);
insert into profesori values (5,'Enache','Sorin',2);
/

```

Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Script Output x
Task completed in 0.162 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

```

insert into grupe values (1,2,'SOL');
insert into grupe values (2,2,'MIERCURI');
insert into grupe values (3,1,'MARTI');
insert into grupe values (4,2,'JOI');
insert into grupe values (5,3,'MARTI');
insert into grupe values (6,1,'VINERI');
insert into grupe values (7,3,'VINERI');
insert into grupe values (8,4,'VINERI');
insert into grupe values (9,5,'LUNI');
/

```

Query Result x Query Result 1 x Query Result 2 x Query Result 3 x
Task completed in 0.207 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

```

insert into concursuri values(1,'WORLD ROBOT OLYMPIAD','San Jose','Costa Rica');
insert into concursuri values(2,'FTC','Detroit','SUA');
insert into concursuri values(3,'STELELE STIINTEI','Bucharest','Romania');
insert into concursuri values(4,'FLL','Denver','SUA');
insert into concursuri values(5,'WRO2020x','Montreal','Canada');
/

```

Script Output x Query Result x
Task completed in 0.124 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.


```
insert into prezenta values (12,4,to_date('29/10/2020','DD/MM/YYYY'));
insert into prezenta values (3,4,to_date('29/10/2020','DD/MM/YYYY'));
insert into prezenta values (18,4,to_date('29/10/2020','DD/MM/YYYY'));
insert into prezenta values (2,6,to_date('30/10/2020','DD/MM/YYYY'));
insert into prezenta values (8,6,to_date('30/10/2020','DD/MM/YYYY'));
insert into prezenta values (13,8,to_date('30/10/2020','DD/MM/YYYY'));
insert into prezenta values (14,7,to_date('30/10/2020','DD/MM/YYYY'));
```

Script Output x Query Result x

Task completed in 0.97 seconds

1 row inserted.

1 row inserted.

1 row inserted.

```
274 insert into inscrieri_concurs values (5,1,5,to_date('12/11/2020','DD/MM/YYYY'));
275 insert into inscrieri_concurs values (4,2,4,to_date('14/11/2020','DD/MM/YYYY'));
276 insert into inscrieri_concurs values (4,2,3,to_date('11/10/2020','DD/MM/YYYY'));
277 insert into inscrieri_concurs values (2,1,4,to_date('03/11/2020','DD/MM/YYYY'));
278 insert into inscrieri_concurs values (2,1,3,to_date('05/10/2020','DD/MM/YYYY'));
279 insert into inscrieri_concurs values (3,5,1,to_date('12/11/2020','DD/MM/YYYY'));
280
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x

Task completed in 0.28 seconds

1 row inserted.

1 row inserted.

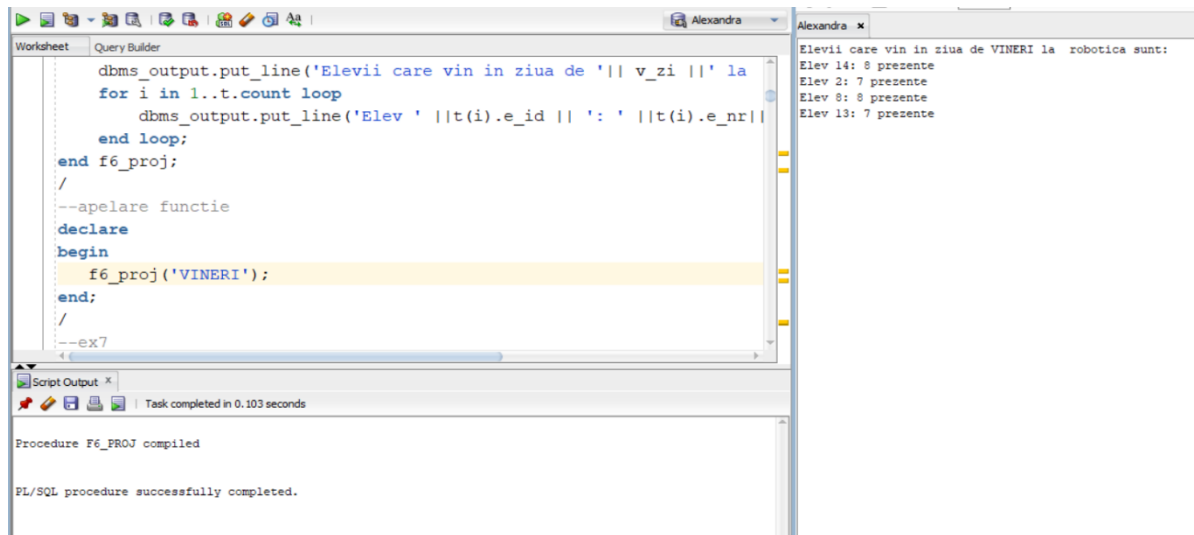
1 row inserted.

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

```
--afisati cate prezente au copiii
--care se prezinta intr-o anumita zi la robotica

create or replace procedure f6_proj
    ( v_zi varchar2 )
is
    TYPE prez_record IS RECORD --am creeat un record pentru a
selecta datele necesare intr-un singur tabel
        (e_id number,
         e_nr number);
    TYPE v_prez IS TABLE OF prez_record;
    t v_prez;
begin
    select id_elev,count(data)
    bulk collect into t
    from prezenta p join grupe g
    on p.id_grupa=g.id_grupa
    where g.zi=v_zi
    group by id_elev;
    if t.count=0 then
        dbms_output.put_line('Nu sunt elevi care vin in aceasta
zi la robotica');
        return;
    end if;
    dbms_output.put_line('Elevii care vin in ziua de '|| v_zi
||' la robotica sunt:');
    for i in 1..t.count loop
        dbms_output.put_line('Elev ' ||t(i).e_id || ': '
||t(i).e_nr|| ' prezente');
    end loop;
end f6_proj;
/

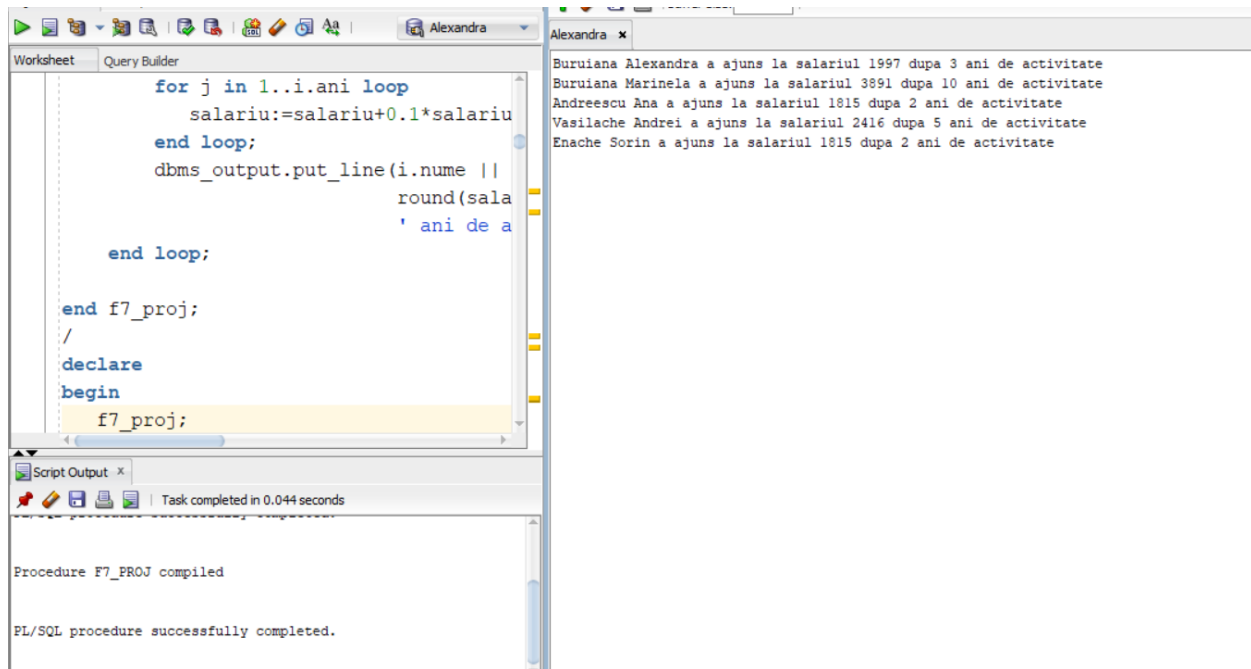
--apelare functie
declare
begin
    f6_proj('VINERI');
end;
/
```



```
-- stiind ca salariul unui profesor porneste de la 1500
-- si pentru fiecare an de vechime se adauga un comision de 2%
-- afisati profesorii si salariul lor

create or replace procedure f7_proj
is
    CURSOR c IS
        SELECT nume || ' ' || prenume nume, vechime ani
        FROM profesori;
    salariu number;
begin
    for i in c loop
        salariu:=1500;
        for j in 1..i.ani loop
            salariu:=salariu+0.1*salariu;
        end loop;
        dbms_output.put_line(i.nume || ' a ajuns la salariul '
||
                                round(salariu) || ' dupa ' || i.ani
||
                                ' ani de activitate');
    end loop;
end f7_proj;
/
```

```
--apelare functie
declare
begin
    f7_proj;
end;
/
```



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.

```
--sa se inscrie echipa x si profesorul y
--la fiecare concurs din lista cu data curenta
--afisati numele concursului, data inscrierii (si la cele vechi
si la cele noi)
--apelarea se face cu numele de familie al profesorului
--concurs + inscrieri + profesori
```

```
create or replace procedure f8_proj
    ( v_prof varchar2, v_ech number)
is
    TYPE v_concurs is table of number;
    cons v_concurs;
```

```

TYPE inscrieri_record IS RECORD
    (id echipa number,
    prof profesori.prenume%TYPE,
    concurs concursuri.denumire%TYPE,
    data_inscriere date);
TYPE v_inscrieri IS TABLE OF inscrieri_record;
t v_inscrieri;
aux_p number;
begin
    select id_concurs
    bulk collect into cons
    from concursuri;
    select id_profesor into aux_p
    from profesori
    where upper( nume)=upper(v_prof);

    for i in 1..cons.count loop
        insert into inscrieri_concurs values(v_ech, aux_p,
cons(i),sysdate);
    end loop;
    select id echipa, p.nume || ' ' || p.prenume, c.denumire ,
data_inscriere
    bulk collect into t
    from profesori p join inscrieri_concurs ic on
p.id_profesor=ic.id_profesor
    join concursuri c on c.id_concurs=ic.id_concurs
    where p.nume=v_prof;
    dbms_output.put_line('Profesorul ' ||v_prof || ' a inscris
echipele: ');
    for i in 1..t.count loop
        dbms_output.put_line('Echipa ' ||t(i).id echipa || ' la
concursul '
|| t(i).concurs || ' pe data de '
|| t(i).data_inscriere);
    end loop;
exception
WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista acest
profesor in baza de date');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi
profesori cu numele dat');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

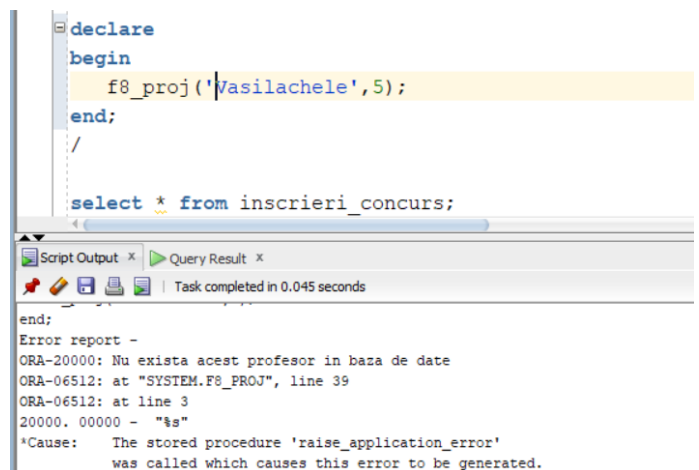
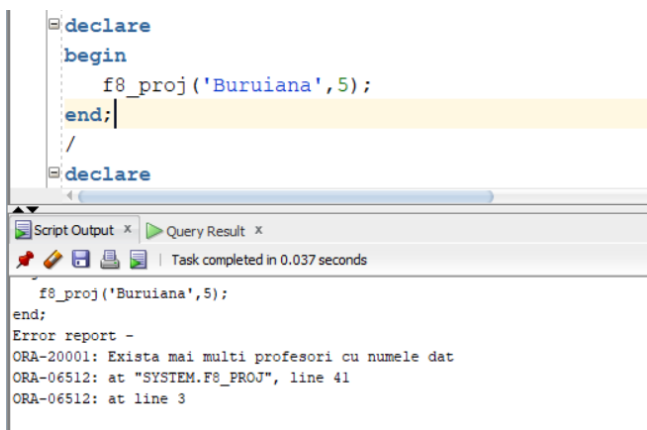
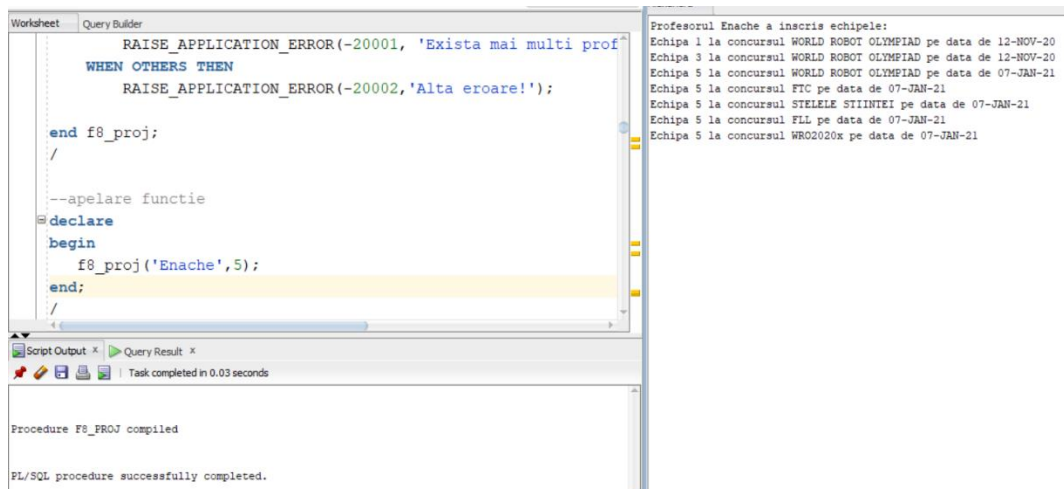
end f8_proj;
/

```

```

--apelare functie
declare
begin
    f8_proj('Enache',5);
end;
/
declare
begin
    f8_proj('Buruiana',5);
end;
/
declare
begin
    f8_proj('Vasilachele',5);
end;
/

```



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.

```
create or replace procedure f9_proj
    ( v_prof varchar2 )
is
    p profesori.nume%TYPE;
    aux_c concursuri.denumire%TYPE;
    aux_e echipe.nume%TYPE;
    aux_n profesori.prenume%TYPE;
    TYPE profesori_record IS RECORD
        (nume profesori.nume%TYPE,
         prenume profesori.prenume%TYPE,
         echipa echipe.nume%TYPE,
         concurs concursuri.denumire%TYPE);
    TYPE v_profesori IS TABLE OF profesori_record;
    c v_profesori;
    TYPE elevi_record IS RECORD
        (e_nume elevi.nume%TYPE,
         e_varsta elevi.varsta%TYPE,
         e_nr number,
         e_gr number,
         e_echipa echipe.nume%TYPE);
    TYPE v_elevi IS TABLE OF elevi_record;
    t v_elevi;
    aux number;

begin

    select  count(distinct p.prenume)
    into aux
    from    echipe ech join inscrieri_concurs ic on
ech.id_echipa=ic.id_echipa
    join concursuri c on c.id_concurs=ic.id_concurs
    join profesori p on ic.id_profesor=p.id_profesor
    where p.nume=v_prof;

    --dbms_output.put_line('aux e ' || aux);

    if aux = 0 then
        RAISE NO_DATA_FOUND;
    elsif aux > 1 then
        RAISE TOO_MANY_ROWS;
    end if;
```

```

        select p.num nume, p.prenume prenume, ech.num echipa,
c.denumire concurs
        bulk collect into c
        from echipe ech join inscrieri_concurs ic on
ech.id_echipa=ic.id_echipa
        join concursuri c on c.id_concurs=ic.id_concurs
        join profesori p on ic.id_profesor=p.id_profesor
        where p.num=v_prof;

        select e.num || ' ' || e.prenume elev,
e.varsta varsta,
(select count(data) from prezenta where id_elev=e.id_elev
group by id_elev) nr_prezente,
(select distinct id_grupa from prezenta where
id_elev=e.id_elev) id_grupa,
ech.num
        bulk collect into t
        from elevi e join echipe ech on e.id_echipa=ech.id_echipa;
aux_c:=' ';
aux_e:=' ';
aux_n:=c(1).prenume;

FOR i in 1..c.count LOOP
    if aux_n<>c(i).prenume then
        RAISE TOO_MANY_ROWS;
    end if;
    if aux_e<>c(i).echipa then
        aux_e:=c(i).echipa;
        aux_c:=' ';
        dbms_output.put_line('-----
-----');
        dbms_output.put_line(c(i).num || ' ' ||c(i).prenume
|| ' a inscrie echipa ' || c(i).echipa);
        dbms_output.new_line();
        dbms_output.put_line('Echipa este formata din: ');

        for j in 1..t.count loop
            if c(i).echipa = t(j).e_echipa then
                dbms_output.put_line(t(j).e_num || ' ' ||
t(j).e_varsta || ' ani '
|| t(j).e_nr || ' prezente
G' ||t(j).e_gr);
            end if;
        end loop;

        dbms_output.new_line();
        dbms_output.put_line('Lista concursuri: ');

```



```

        end if;

        if aux_c<>c(i).concurs then
            dbms_output.put_line(c(i).concurs);
            aux_c:=c(i).concurs;
        end if;

    end loop;

exception
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista echipe
inscrise de acest profesor');
    WHEN TOO_MANY_ROWS THEN
        --dbms_output.put_line(SQL%ROWCOUNT);
        RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi
profesori cu numele dat');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

end f9_proj;
/
--apelare functie
declare
begin
    f9_proj('Enache');
end;
/
declare
begin
    f9_proj('Buruiana');
end;
/
declare
begin
    f9_proj('Vasilache');
end;
/

```

```

        RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi prof
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

    end f9_proj;
    /
    --apelare functie
    declare
    begin
        f9_proj('Enache');
    end;
    /
    declare

```

Enache Sorin a inscris echipa MICROBOTS

Echipa este formata din:
 Stoian Stefan 15 ani 6 prezente G5
 Mihai David 15 ani 5 prezente G5
 Calugaru David 15 ani 5 prezente G5

Lista concursuri:
 WORLD ROBOT OLYMPIAD

Enache Sorin a inscris echipa ROBOT RULER

Echipa este formata din:
 Popescu Elena 11 ani 7 prezente G1
 Panaite Iris 12 ani 7 prezente G6

Lista concursuri:
 WORLD ROBOT OLYMPIAD

Script Output x Query Result x

Task completed in 0.033 seconds

the application administrator or DBA for more information.

Procedure F9_PROJ compiled

PL/SQL procedure successfully completed.

```

    declare
    begin
        f9_proj('Buruiana');
    end;
    /
    declare
    begin

```

Script Output x Query Result x

Task completed in 0.037 seconds

f9_proj('Buruiana');

end;

Error report -

ORA-20001: Exista mai multi profesori cu numele dat

ORA-06512: at "SYSTEM.F9_PROJ", line 94

ORA-06512: at line 3

```

    declare
    begin
        f9_proj('Vasilache');
    end;
    /

```

Script Output x Query Result x

Task completed in 0.044 seconds

end;

Error report -

ORA-20000: Nu exista echipe inscrise de acest profesor

ORA-06512: at "SYSTEM.F9_PROJ", line 91

ORA-06512: at line 3

20000. 00000 - "%s"

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

```
CREATE OR REPLACE TRIGGER trig10_prezenta
BEFORE insert or update or delete on prezenta
begin
    if upper(to_char(sysdate,'d')) =7 or
    upper(to_char(sysdate,'d')) =1 then
        RAISE_APPLICATION_ERROR(-20003,
                                'Data curenta nu este una valida
pentru prezenta');
    end if;
end;
/
--declasare
declare
begin
    for i in 1..9 loop
        insert into prezenta values(i,5,sysdate);
    end loop;
end;
/

DROP TRIGGER trig10_prezenta;
```

The image displays two side-by-side screenshots of the Oracle SQL Developer interface, showing the execution of a trigger.

Left Screenshot: The script editor shows the trigger definition and a loop to insert data. The status bar indicates "Task completed in 0.036 seconds". The "Script Output" pane shows an error report:

```
end;
Error report -
ORA-20003: Data curenta nu este una valida pentru prezenta
ORA-06512: at "SYSTEM.TRIG10_PREZENTA", line 3
ORA-04088: error during execution of trigger 'SYSTEM.TRIG10_PREZENTA'
ORA-06512: at line 4
```

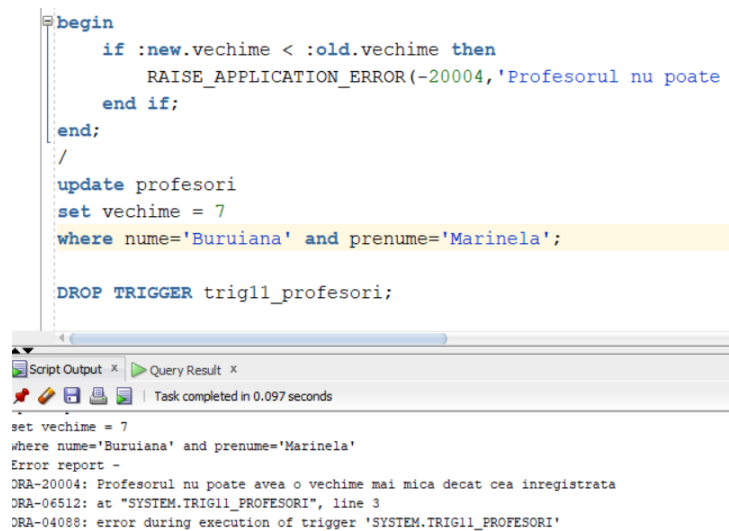
Right Screenshot: The script editor shows the same trigger definition. The status bar indicates "Task completed in 0.033 seconds". The "Script Output" pane shows the successful compilation of the trigger:

```
Trigger TRIG10_PREZENTA compiled
PL/SQL procedure successfully completed.
```

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

```
CREATE OR REPLACE TRIGGER trig11_profesori
  BEFORE update of vechime on profesori
  for each row
begin
  if :new.vechime < :old.vechime then
    RAISE_APPLICATION_ERROR(-20004,'Profesorul nu poate avea
o vechime mai mica decat cea inregistrata');
  end if;
end;
/
update profesori
set vechime = 7
where nume='Buruiana' and prenume='Marinela';

DROP TRIGGER trig11_profesori;
```



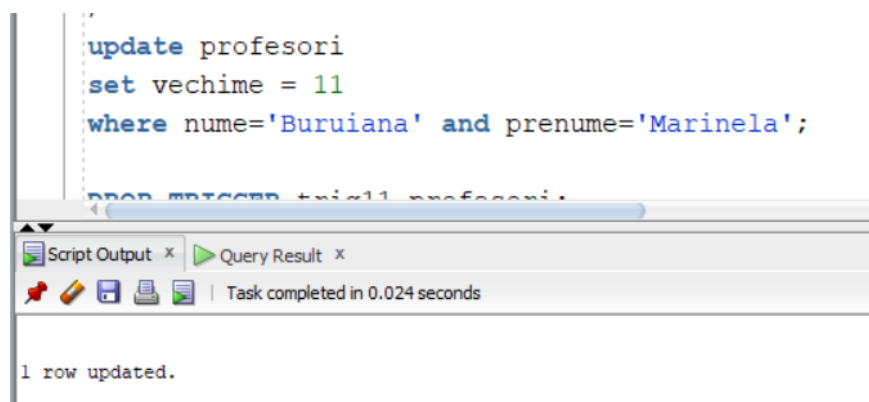
The screenshot shows a SQL script editor with the following code:

```
begin
  if :new.vechime < :old.vechime then
    RAISE_APPLICATION_ERROR(-20004,'Profesorul nu poate avea
o vechime mai mica decat cea inregistrata');
  end if;
end;
/
update profesori
set vechime = 7
where nume='Buruiana' and prenume='Marinela';

DROP TRIGGER trig11_profesori;
```

Below the script, the 'Script Output' pane shows the execution results:

```
set vechime = 7
where nume='Buruiana' and prenume='Marinela'
Error report -
ORA-20004: Profesorul nu poate avea o vechime mai mica decat cea inregistrata
ORA-06512: at "SYSTEM.TRIG11_PROFESORI", line 3
ORA-04088: error during execution of trigger 'SYSTEM.TRIG11_PROFESORI'
```



The screenshot shows a SQL script editor with the following code:

```
update profesori
set vechime = 11
where nume='Buruiana' and prenume='Marinela';

DROP TRIGGER trig11_profesori;
```

Below the script, the 'Script Output' pane shows the execution results:

```
1 row updated.
```

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

```
create table inscrieri_copy as select * from inscrieri_concurs;  
select * from inscrieri_copy;
```

```
CREATE TABLE audit_robotica  
(utilizator varchar2(30),  
nume_bd varchar2(50),  
eveniment varchar2(20),  
nume_obiect varchar2(30),  
data date);
```

```
CREATE OR REPLACE TRIGGER trig12_audit  
after create or drop or alter on schema  
begin  
insert into audit_robotica  
values (SYS.LOGIN_USER,  
SYS.DATABASE_NAME,  
SYS.SYSEVENT,  
SYS.DICTIONARY_OBJ_NAME,  
SYSDATE);
```

```
end;
```

```
/
```

```
CREATE INDEX ind ON inscrieri_copy(data_inscriere);  
DROP INDEX ind;  
CREATE INDEX ind ON inscrieri_copy(id_concurs);  
DROP INDEX ind;  
SELECT * FROM audit_robotica;  
DROP TRIGGER trig12_audit;  
drop table audit_robotica;
```

```
insert into audit_robotica  
values (SYS.LOGIN_USER,  
SYS.DATABASE_NAME,  
SYS.SYSEVENT,  
SYS.DICTIONARY_OBJ_NAME,  
SYSDATE);  
  
end;  
  
/  
CREATE INDEX ind ON inscrieri_copy(data_inscriere);  
DROP INDEX ind;
```

```
SELECT * FROM audit_robotica;  
DROP TRIGGER trig12_audit;  
  
drop table audit_robotica;  
drop table inscrieri_copy;
```

Script Output x Query Result x
Task completed in 0.283 seconds
Rollback complete.

Table AUDIT_ROBOTICA created.

Trigger TRIG12_AUDIT compiled

Index IND created.

Script Output x Query Result x
All Rows Fetched: 4 in 0.009 seconds

	UTILIZATOR	NUME_BD	EVENIMENT	NUME_OBIECT	DATA
1	SYSTEM	ORCL	CREATE	IND	07-JAN-21
2	SYSTEM	ORCL	DROP	IND	07-JAN-21
3	SYSTEM	ORCL	CREATE	IND	07-JAN-21
4	SYSTEM	ORCL	DROP	IND	07-JAN-21

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

```
CREATE OR REPLACE PACKAGE pachet_functii_1
AS
    procedure f6_proj( v_zi varchar2 );
    -----
    procedure f7_proj;
    -----
    procedure f8_proj( v_prof varchar2,
                      v_ech number);
    -----
    procedure f9_proj( v_prof varchar2 );
    -----
END pachet_functii_1;
/

CREATE OR REPLACE PACKAGE BODY pachet_functii_1
AS
    procedure f6_proj( v_zi varchar2 )
    is
        TYPE prez_record IS RECORD --am creeat un record
        pentru a selecta datele necesare intr-un singur tabel
            (e_id number,
             e_nr number);
        TYPE v_prez IS TABLE OF prez_record;
        t v_prez;
    begin
        select id_elev,count(data)
        bulk collect into t
        from prezenta p join grupe g
        on p.id_grupa=g.id_grupa
        where g.zi=v_zi
        group by id_elev;
        if t.count=0 then
            dbms_output.put_line('Nu sunt elevi care vin in
aceasta zi la robotica');
            return;
        end if;

        dbms_output.put_line('Elevii care vin in ziua de '||
v_zi ||' la  robotica sunt:');
        for i in 1..t.count loop
            dbms_output.put_line('Elev ' ||t(i).e_id || ': '
||t(i).e_nr|| ' prezente');
        end loop;
    end;
```

```

-----
-----
procedure f7_proj
is
    CURSOR c IS
        SELECT nume || ' ' || prenume nume, vechime
ani
        FROM profesori;
        salariu number;
begin
    for i in c loop
        salariu:=1500;
        for j in 1..i.ani loop
            salariu:=salariu+0.1*salariu;
        end loop;
        dbms_output.put_line(i.nume || ' a ajuns la
salariul ' ||
                                round(salariu) || '
dupa ' || i.ani ||
                                ' ani de activitate');
        end loop;

    end;

```

```

-----
-----
procedure f8_proj( v_prof varchar2,
                  v_ech number)
is
    TYPE v_concurs is table of number;
    cons v_concurs;
    TYPE inscrieri_record IS RECORD
        (id_echipa number,
         prof profesori.prenume%TYPE,
         concurs concursuri.denumire%TYPE,
         data_inscriere date);
    TYPE v_inscrieri IS TABLE OF inscrieri_record;
    t v_inscrieri;
    aux_p number;
begin
    select id_concurs
    bulk collect into cons
    from concursuri;
    select id_profesor into aux_p

```

```

        from profesori
        where upper(ume)=upper(v_prof);

        for i in 1..cons.count loop
            insert into inscrieri_concurs values(v_ech,
aux_p, cons(i),sysdate);
        end loop;
        select id_echipa, p.ume || ' ' || p.prenume,
c.denumire , data_inscriere
        bulk collect into t
        from profesori p join inscrieri_concurs ic on
p.id_profesor=ic.id_profesor
        join concursuri c on c.id_concurs=ic.id_concurs
        where p.ume=v_prof;
        dbms_output.put_line('Profesorul ' ||v_prof || '
a inscris echipele: ');
        for i in 1..t.count loop
            dbms_output.put_line('Echipa '
||t(i).id_echipa || ' la concursul '
|| t(i).concurs || ' pe
data de ' || t(i).data_inscriere);
        end loop;

    exception
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Nu exista
acest profesor in baza de date');
        WHEN TOO_MANY_ROWS THEN
            RAISE_APPLICATION_ERROR(-20001, 'Exista mai
multi profesori cu numele dat');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20002,'Alta
eroare!');

    end;
-----
-----

    procedure f9_proj
        ( v_prof varchar2 ) --functia primeste parametrul
manager_id
    is
        p profesori.ume%TYPE;
        aux_c concursuri.denumire%TYPE;
        aux_e echipe.ume%TYPE;
        aux_n profesori.prenume%TYPE;
        TYPE profesori_record IS RECORD
            (ume profesori.ume%TYPE,

```



```

        prenume profesori.prenume%TYPE,
        echipa echipe.numa%TYPE,
        concurs concursuri.denumire%TYPE);
TYPE v_profesori IS TABLE OF profesori_record;
c v_profesori;
TYPE elevi_record IS RECORD
    (e_nume elevi.numa%TYPE,
     e_varsta elevi.varsta%TYPE,
     e_nr number,
     e_gr number,
     e_echipa echipe.numa%TYPE);
TYPE v_elevi IS TABLE OF elevi_record;
t v_elevi;
aux number;

begin

    select  count(distinct p.prenume)
    into aux
    from    echipe ech join inscrieri_concurs ic on
ech.id_echipa=ic.id_echipa
    join concursuri c on c.id_concurs=ic.id_concurs
    join profesori p on ic.id_profesor=p.id_profesor
    where p.numa=v_prof;

    --dbms_output.put_line('aux e ' || aux);

    if aux = 0 then
        RAISE NO_DATA_FOUND;
    elsif aux > 1 then
        RAISE TOO_MANY_ROWS;
    end if;
    select p.numa numa, p.prenume prenume, ech.numa
echipa, c.denumire concurs
    bulk collect into c
    from    echipe ech join inscrieri_concurs ic on
ech.id_echipa=ic.id_echipa
    join concursuri c on c.id_concurs=ic.id_concurs
    join profesori p on ic.id_profesor=p.id_profesor
    where p.numa=v_prof;

    select e.numa ||' '|| e.prenume elev,
    e.varsta varsta,
    (select count(data) from prezenta where
id_elev=e.id_elev group by id_elev) nr_prezente,
    (select distinct id_grupa from prezenta where
id_elev=e.id_elev) id_grupa,

```

```

        ech.numa
        bulk collect into t
        from elevi e join echipe ech on
e.id_echipa=ech.id_echipa;
        aux_c:=' ';
        aux_e:=' ';
        aux_n:=c(1).prenume;

        FOR i in 1..c.count LOOP
            if aux_n<>c(i).prenume then
                RAISE TOO_MANY_ROWS;
            end if;
            if aux_e<>c(i).echipa then
                aux_e:=c(i).echipa;
                aux_c:=' ';
                dbms_output.put_line('-----
-----');
                dbms_output.put_line(c(i).numa || ' '
||c(i).prenume ||' a inscrist echipa '|| c(i).echipa);
                dbms_output.new_line();
                dbms_output.put_line('Echipa este formata
din: ');

                for j in 1..t.count loop
                    if c(i).echipa = t(j).e_echipa then
                        dbms_output.put_line(t(j).e_numa || ' '
|| t(j).e_varsta ||' ani '
|| t(j).e_nr || '
prezenta G' ||t(j).e_gr);
                    end if;
                end loop;

                dbms_output.new_line();
                dbms_output.put_line('Lista concursuri: ');
                end if;

                if aux_c<>c(i).concurs then
                    dbms_output.put_line(c(i).concurs);
                    aux_c:=c(i).concurs;
                end if;

            end loop;

        exception
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu exista
echipe inscriste de acest profesor');

```

```

        WHEN TOO_MANY_ROWS THEN
            --dbms_output.put_line(SQL%ROWCOUNT);
            RAISE_APPLICATION_ERROR(-20001, 'Exista mai
multi profesori cu numele dat');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

    end;

END pachet_functii_1;
/
--apelare functii
declare
begin
    pachet_functii_1.f6_proj('VINERI');

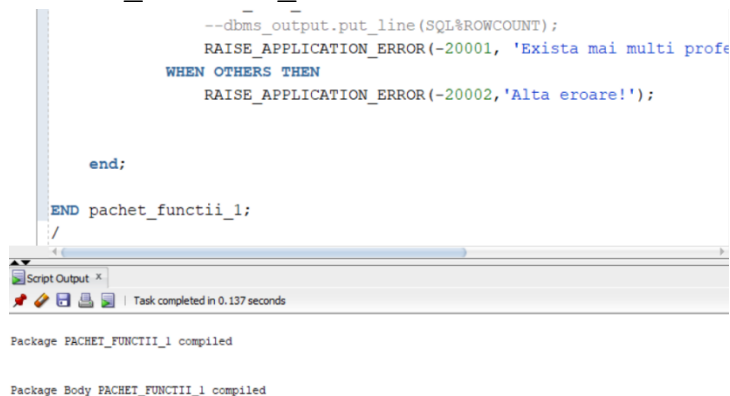
    pachet_functii_1.f7_proj;

    pachet_functii_1.f8_proj('Andreescu',5);

--    pachet_functii_1.f8_proj('Buruiana',5);
--
--    pachet_functii_1.f8_proj('Vasilachele',5);

    pachet_functii_1.f9_proj('Enache');
--
--    pachet_functii_1.f9_proj('Buruiana');
--
--    pachet_functii_1.f9_proj('Vasilache');
end;
/
drop package pachet_functii_1;

```



```

--dbms_output.put_line(SQL%ROWCOUNT);
RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi profe
WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

end;

END pachet_functii_1;
/

```

Script Output x

Task completed in 0.137 seconds

Package PACHET_FUNCITII_1 compiled

Package Body PACHET_FUNCITII_1 compiled

SQL Worksheet | History

prezentata 5 of 5

Worksheet Query Builder

```

END pachet_functii_1;
/
--apelare functii
declare
begin
    pachet_functii_1.f6_proj('VINERI');

    pachet_functii_1.f7_proj;

    pachet_functii_1.f8_proj('Andrescu',5);

    pachet_functii_1.f9_proj('Enache');

end;
/
DROP PACKAGE pachet_functii_1;

```

Script Output x Query Result x

Task completed in 0.051 seconds

PL/SQL procedure successfully completed.

Alexandra x

Elevii care vin in ziua de VINERI la robotica sunt:
 Elev 14: 8 prezente
 Elev 2: 7 prezente
 Elev 8: 8 prezente
 Elev 13: 7 prezente
 Buruiana Alexandra a ajuns la salariul 1997 dupa 3 ani de activitate
 Buruiana Marinela a ajuns la salariul 4280 dupa 11 ani de activitate
 Andrescu Ana a ajuns la salariul 1815 dupa 2 ani de activitate
 Vasilache Andrei a ajuns la salariul 2416 dupa 5 ani de activitate
 Enache Sorin a ajuns la salariul 1815 dupa 2 ani de activitate
 Profesorul Andrescu a inscris echipele:
 Echipa 5 la concursul WORLD ROBOT OLYMPIAD pe data de 07-JAN-21
 Echipa 5 la concursul FTC pe data de 07-JAN-21
 Echipa 5 la concursul STELELE STIINTEI pe data de 07-JAN-21
 Echipa 5 la concursul FLL pe data de 07-JAN-21
 Echipa 5 la concursul WRO2020x pe data de 07-JAN-21

 Enache Sorin a inscris echipa MICROBOTS

 Echipa este formata din:
 Stoian Stefan 15 ani 6 prezente G5
 Mihai David 15 ani 5 prezente G5
 Calugaru David 15 ani 5 prezente G5

 Lista concursuri:
 WORLD ROBOT OLYMPIAD

 Enache Sorin a inscris echipa ROBOT RULER

 Echipa este formata din:
 Popescu Elena 11 ani 7 prezente G1
 Panaite Iris 12 ani 7 prezente G6

 Lista concursuri:
 WORLD ROBOT OLYMPIAD

Activate Windows
Go to Settings to activate Windows

Worksheet Query Builder

```

declare
begin
    pachet_functii_1.f6_proj('VINERI');
    pachet_functii_1.f7_proj;
    pachet_functii_1.f8_proj('Enache',5);
    rollback;
    pachet_functii_1.f8_proj('Buruiana',5);
    pachet_functii_1.f8_proj('Vasilachele',5);
    pachet_functii_1.f9_proj('Enache');
    pachet_functii_1.f9_proj('Buruiana');
    pachet_functii_1.f9_proj('Vasilache');
end;
/
DROP PACKAGE pachet_functii_1;

```

Script Output x

Task completed in 0.049 seconds

pachet_functii_1.f9_proj('Buruiana');

pachet_functii_1.f9_proj('Vasilache');

end;

Error report -
 ORA-20001: Exista mai multi profesori cu numele dat
 ORA-06512: at "SYSTEM.PACHET_FUNCTII_1", line 90
 ORA-06512: at line 11

Elevii care vin in ziua de VINERI la robotica sunt:
 Elev 14: 8 prezente
 Elev 2: 7 prezente
 Elev 8: 8 prezente
 Elev 13: 7 prezente
 Buruiana Alexandra a ajuns la salariul 1997 dupa 3 ani de activitate
 Buruiana Marinela a ajuns la salariul 3891 dupa 10 ani de activitate
 Andrescu Ana a ajuns la salariul 1815 dupa 2 ani de activitate
 Vasilache Andrei a ajuns la salariul 2416 dupa 5 ani de activitate
 Enache Sorin a ajuns la salariul 1815 dupa 2 ani de activitate
 Profesorul Enache a inscris echipele:
 Echipa 1 la concursul WORLD ROBOT OLYMPIAD pe data de 12-NOV-20
 Echipa 3 la concursul WORLD ROBOT OLYMPIAD pe data de 12-NOV-20
 Echipa 5 la concursul WORLD ROBOT OLYMPIAD pe data de 07-JAN-21
 Echipa 5 la concursul FTC pe data de 07-JAN-21
 Echipa 5 la concursul STELELE STIINTEI pe data de 07-JAN-21
 Echipa 5 la concursul FLL pe data de 07-JAN-21
 Echipa 5 la concursul WRO2020x pe data de 07-JAN-21

Activate Windows
Go to Settings to activate Windows

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

```
--primaria orasului a decis premiarea copiilor si a
profesorilor
--pentru concursurile internationale si nationale
--profesorii sunt premiati cu 200 de lei pentru fiecare
inscriere de echipa
--fiecare echipa este premiata cu 1500 de lei, suma fiind
impartita apoi intre membri
--participarea la cel putin un concurs romanesc
--este premiata cu un bonus de 100 de lei pe langa standard
--in plus, pentru concursurile din SUA, fiecare copil
primeste decontarea cheltuielilor pentru viza
--decontarea cheltuielilor pentru viza in valoare de 300 de
lei
--sa se afiseze exact cat va oferi primaria fiecarui
profesor
--si fiecarui elev participant la concursuri

CREATE OR REPLACE PACKAGE pachet_functii_2
AS
    type ech_medie is record (id echipa number,
                             nume echipe.nume%TYPE,
                             nr number,
                             suma number);
    cursor c_ech_medie return ech_medie;
    -----
    type prof is record (id_prof number,
                         nume profesori.nume%TYPE,
                         prenume
profesori.prenume%TYPE);
    cursor c_prof return prof;
    -----
    function viza_sua ( v_elev number)
        return number;
    -----

    function prima_prof ( ln elevi.nume%TYPE,
                          fn elevi.prenume%TYPE)
        return number;
    function echipa_romania ( v echipa number)
        return number;
```

```

END pachet_functii_2;
/

CREATE OR REPLACE PACKAGE BODY pachet_functii_2
AS
    --sa se declare un cursor care sa retina pentru fiecare
    echipa numele
    --si suma de bani totala pentru echipa (fara bonusul
    pentru Romania)
    cursor c_ech_medie return ech_medie
                                is select id_echipa, nume, (select
count(*)
                                from elevi where
id_echipa=ech.id_echipa) nr,
                                (select count(*) from
inscrieri_concurs where id_echipa=ech.id_echipa)*1500 suma
                                from echipe ech
                                order by id_echipa;
    cursor c_prof return prof
                                is select id_profesor id_prof, nume,
prenume
                                from profesori;
    -- cati elevi vor avea nevoie de viza pentru SUA (vor
    pleca la concurs in aceasta tara)
    function viza_sua( v_elev number)
    return number
    is
        TYPE v_sua IS TABLE OF number;
        t v_sua;
    begin
        select id_elev
        bulk collect into t
        from elevi e
        join inscrieri_concurs ic on e.id_echipa=
ic.id_echipa
        join concursuri c on c.id_concurs=ic.id_concurs
        where tara='SUA'
        group by id_elev;

        if t.count=0 then
            dbms_output.put_line('Nu este niciun elev
care sa aiba nevoie de viza pentru SUA');
            return 0;
        end if;
        for i in 1..t.count loop
            if t(i)=v_elev then
                return 1;
            end if;
        end loop;
    end function;

```

```

        end if;
    end loop;
    return 0;

end;

-----

-----

--profesorii sunt premiati cu 200 de lei pentru
fiecare inscriere de echipa
function prima_prof ( ln elevi.nume%TYPE,
                      fn elevi.prenume%TYPE)
return number
is
cursor c is select count(id echipa) nr,
                  id_profesor from inscrieri_concurs
                  group by id_profesor;
aux_id number;
begin
    select id_profesor
    into aux_id
    from profesori
    where nume=ln and prenume=fn;
    for i in c loop
        if aux_id = i.id_profesor then
            return i.nr*200;
        end if;
    end loop;
    return 0;
exception
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista
acest profesor');
    WHEN TOO_MANY_ROWS THEN
        --dbms_output.put_line(SQL%ROWCOUNT);
        RAISE_APPLICATION_ERROR(-20001, 'Exista mai
multi profesori cu numele dat');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'Alta
eroare!');
end;

function echipa_romania ( v echipa number)

```

```

return number
is
nr_concursuri_romania number;
aux_ech number:=-1;
begin

select id echipa
into aux_ech
from echipe
where id echipa=v echipa;

if aux_ech=-1 then
    raise NO_DATA_FOUND;
end if;

select count(*)
into nr_concursuri_romania
from inscrieri_concurs ic join concursuri c on
ic.id_concurs=c.id_concurs
where id echipa=v echipa and upper(c.tara)='ROMANIA';

return nr_concursuri_romania;
exception
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista
aceasta echipa');
end;

END pachet_functii_2;
/

declare
    type nume is record (id echipa number,
                        id_elev number,
                        nume elevi.nume%type,
                        prenume elevi.prenume%type);
    type v_nume is table of nume;
    t v_nume;
begin
    --aflam sumele pentru profesori
    dbms_output.put_line('Premierea profesorilor');
    dbms_output.put_line('-----');
    for i in pachet_functii_2.c_prof loop
        dbms_output.put_line('Profesorul ' || i.nume || ' '
|| i.prenume || ' va primi suma de '
||
pachet_functii_2.prima_prof(i.nume, i.prenume) || ' lei');

```



```

end loop;
dbms_output.put_line('-----');
dbms_output.new_line;
--facem lista de elevi
select id echipa, id elev, nume, prenume
bulk collect into t
from elevi
order by id echipa;
--calculam premiarea copiilor
dbms_output.put_line('Premierea copiilor');
dbms_output.put_line('-----');
for i in 1..t.count loop
    for j in pachet_functii_2.c_ech_medie loop
        if t(i).id echipa = j.id echipa then
            dbms_output.put_line(t(i).nume || ' '
||t(i).prenume || ' va primi ' ||

(round((pachet_functii_2.echipa_romania(j.id echipa)*100+j.
suma)/j.nr)+

pachet_functii_2.viza_sua(t(i).id elev)*300) || ' lei');
            if
pachet_functii_2.echipa_romania(j.id echipa) <> 0 then
                dbms_output.put_line('Acest elev va
pleca in America');
            end if;
            if pachet_functii_2.viza_sua(t(i).id elev)
<> 0 then
                dbms_output.put_line('Acest elev a avut
cel putin un concurs in Romania');
            end if;
            dbms_output.new_line;
        end if;
    end loop;
end loop;
dbms_output.put_line('-----');

end;
/

drop package pachet_functii_2;

```

Worksheet

Query Builder

```

151 bulk collect into t
152 from elevi
153 order by id echipa;
154 --calculam premiarea copiilor
155 dbms_output.put_line('Premierea copiilor');
156 dbms_output.put_line('-----');
157 for i in 1..t.count loop
158     for j in pachet_functii_2.c_ech_medie loop
159         if t(i).id echipa = j.id echipa then
160             dbms_output.put_line(t(i).nume || ' ' || t(i).prenume || ' va primi ' ||
161                 (round((pachet_functii_2.echipa_romania(j.id echipa)*100+j.suma)/j.
162                     pachet_functii_2.viza_sua(t(i).id elev)*300) || ' lei');
163             if pachet_functii_2.echipa_romania(j.id echipa) <> 0 then
164                 dbms_output.put_line('Acest elev va pleca in America');

```

Query Result

Query Result 1

Query Result 2

Script Output

Task completed in 0.049 seconds

Package PACHET_FUNCTII_2 compiled

Package Body PACHET_FUNCTII_2 compiled

PL/SQL procedure successfully completed.

Premierea profesorilor

Profesorul Buruiana Alexandra va primi suma de 800 lei
Profesorul Buruiana Marinela va primi suma de 1200 lei
Profesorul Andreescu Ana va primi suma de 0 lei
Profesorul Vasilache Andrei va primi suma de 0 lei
Profesorul Enache Sorin va primi suma de 400 lei

Premierea copiilor

Mihai David va primi 2300 lei
Acest elev a avut cel putin un concurs in Romania

Stoian Stefan va primi 2300 lei
Acest elev a avut cel putin un concurs in Romania

Calugaru David va primi 2300 lei
Acest elev a avut cel putin un concurs in Romania

Tudor Radu va primi 1333 lei
Acest elev va pleca in America
Acest elev a avut cel putin un concurs in Romania

Enacheanu Iustin va primi 1333 lei
Acest elev va pleca in America
Acest elev a avut cel putin un concurs in Romania

Mantu Ana va primi 1333 lei
Acest elev va pleca in America
Acest elev a avut cel putin un concurs in Romania

Popescu Elena va primi 750 lei