**Gestionarea unui club sportiv**

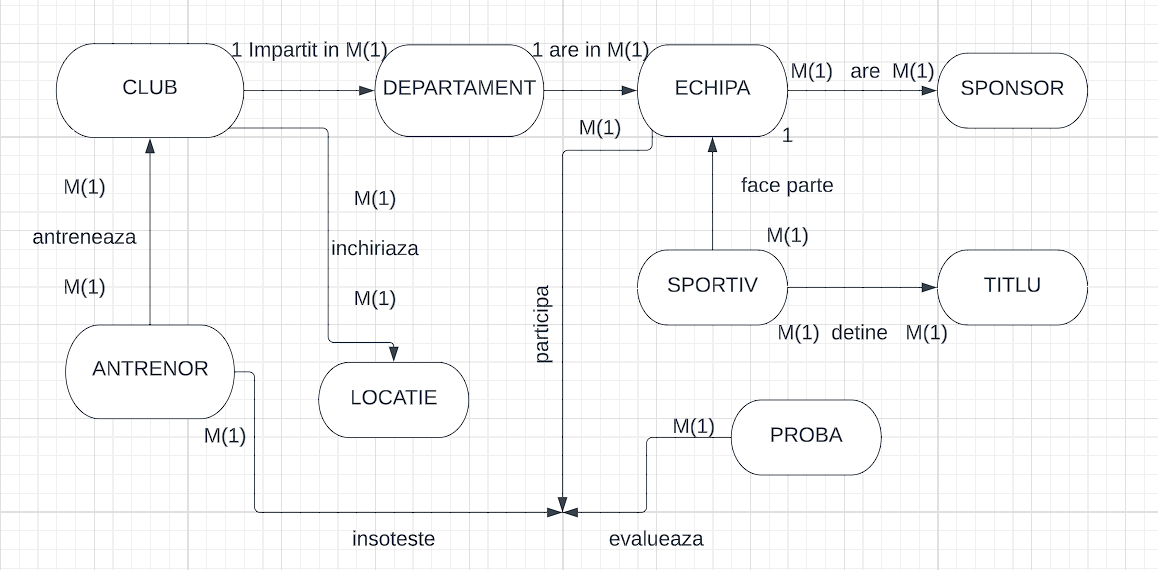
Hurloi Selena-Andreea

Grupa 242

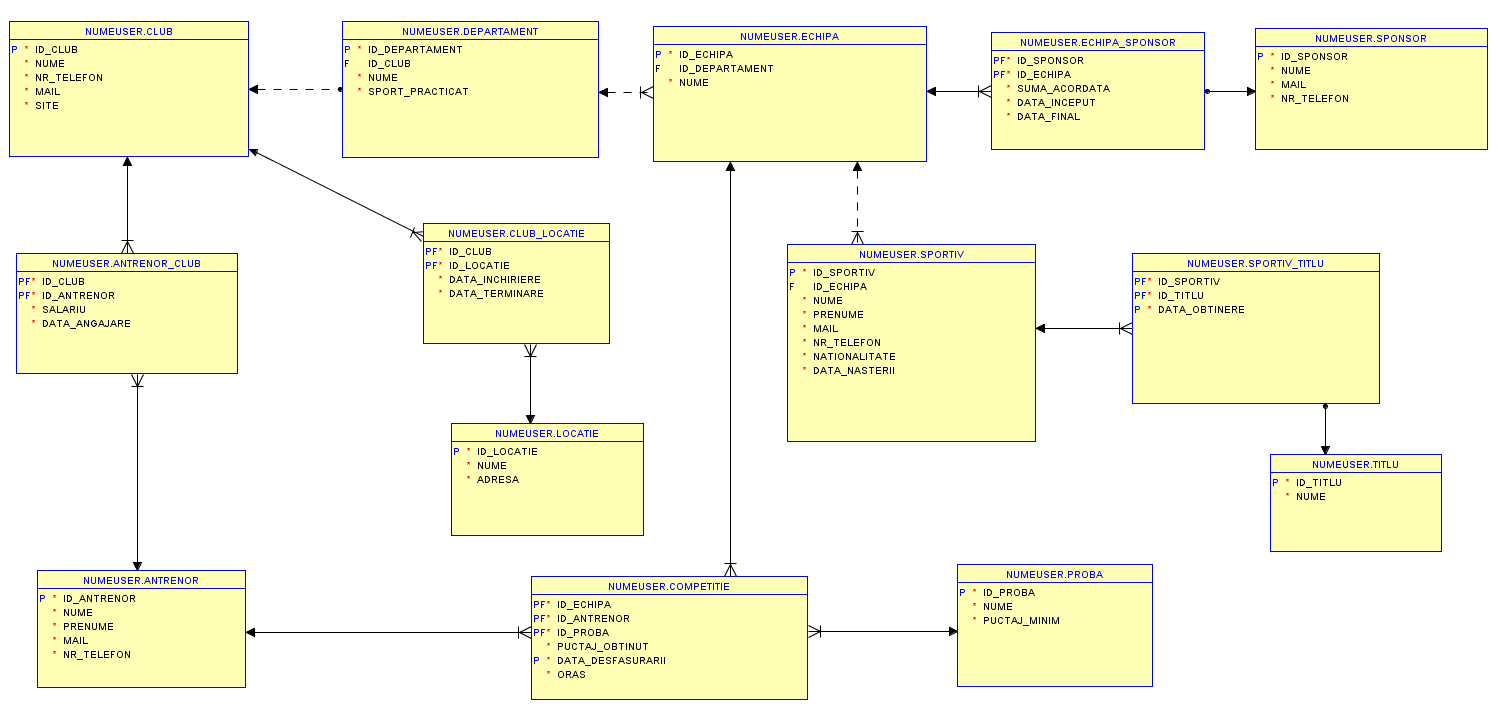
# 1. Prezentați pe scurt baza de date (utilitatea ei)

Scopul proiectului este de a tine evidenta informatiilor specifice unui club sportiv. Cluburile sportive se impart in mai multe departamente, iar in cadrul departamentelor se gasesc echipele. Aceste echipe pot beneficia de sponsori. Antrenorii lucreaza pentru cluburi, insa acestia se pot angaja la mai multe. Cluburile inchiriaza locatii pentru antrenamente unde sportivii isi desfasoara pregatirea. La competitii, echipele sunt insotite de antrenor, iar in cadrul acestora primesc punctaje. Pentru premiile obtinute in decursul carierei, sportivii sunt premiati cu titluri.

# 2. Diagrama Entitate-Relație(ERD)



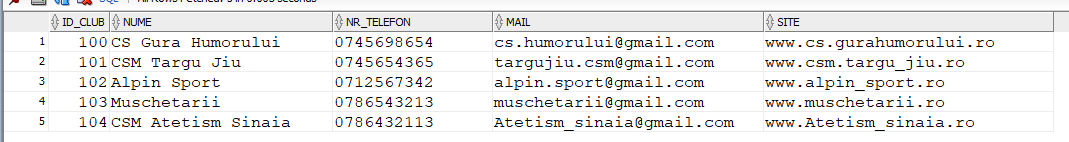
# 3. Diagrama Conceptuală



# 4. Implementare si inserare

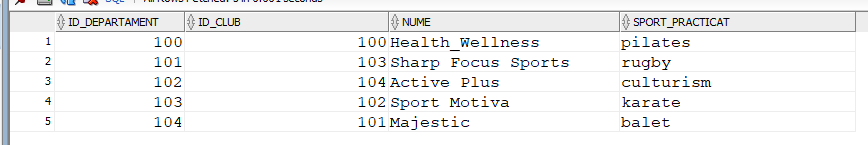
Create table CLUB  
 (id\_club int constraint id\_club\_pk PRIMARY KEY,  
 nume VARCHAR(20) CONSTRAINT name\_club not null,  
 nr\_telefon VARCHAR(20) CONSTRAINT nr\_telefon\_club not null,  
 mail VARCHAR(200) CONSTRAINT mail\_club not null,  
 site VARCHAR (200) CONSTRAINT site\_facultate not null  
 );

CREATE SEQUENCE ID  
INCREMENT by 1  
START WITH 100  
MAXVALUE 10000  
NOCYCLE;  
   
Insert into Club  
values (ID.nextval,'CS Gura Humorului','0745698654','cs.humorului@gmail.com','www.cs.gurahumorului.ro');  
Insert into Club  
values (ID.nextval,'CSM Targu Jiu','0745654365','targujiu.csm@gmail.com','www.csm.targu\_jiu.ro');  
Insert into Club  
values (ID.nextval,'Alpin Sport','0712567342','alpin.sport@gmail.com','www.alpin\_sport.ro');  
Insert into Club  
values (ID.nextval,'Muschetarii','0786543213','muschetarii@gmail.com','www.muschetarii.ro');  
Insert into Club  
values (ID.nextval,'CSM Atetism Sinaia','0786432113','atetism\_sinaia@gmail.com','www.Atetism\_sinaia.ro');  
   
select \* from club

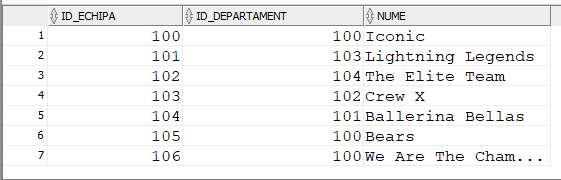


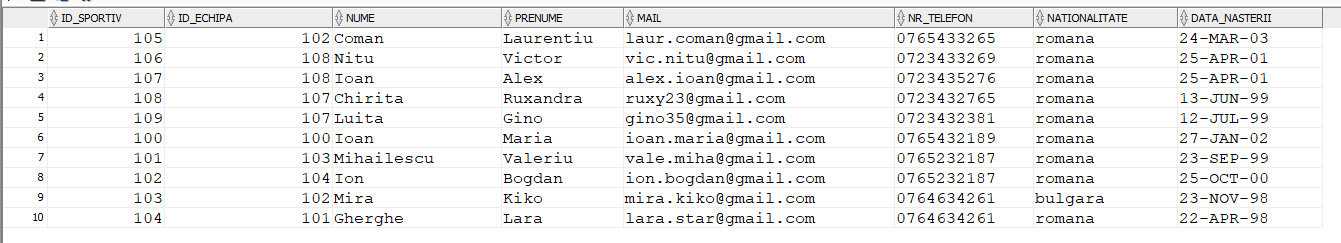
create table DEPARTAMENT  
 (id\_departament int constraint id\_departament\_pk PRIMARY KEY,  
 id\_club int, constraint id\_club\_fk FOREIGN KEY (id\_club) references CLUB(id\_club),  
 nume varchar (20) not null,  
 sport\_practicat varchar (20) not null  
 );

CREATE SEQUENCE ID\_DEP  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;   
   
INSERT INTO Departament  
VALUES (ID\_DEP.NEXTVAL,100,'Health\_Wellness','pilates');  
INSERT INTO Departament  
VALUES (ID\_DEP.NEXTVAL,103,'Sharp Focus Sports','rugby');  
INSERT INTO Departament  
VALUES (ID\_DEP.NEXTVAL,104,'Active Plus','culturism');  
INSERT INTO Departament  
VALUES (ID\_DEP.NEXTVAL,102,'Sport Motiva','karate');  
INSERT INTO Departament  
VALUES (ID\_DEP.NEXTVAL,101,'Majestic','balet');  
   
 SELECT \* FROM departament

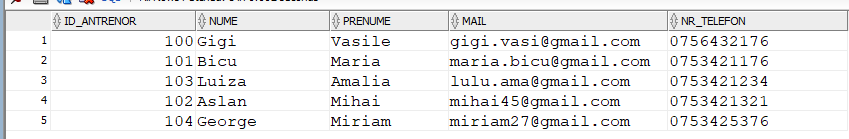


CREATE TABLE ECHIPA (id\_echipa int constraint id\_echila\_pk PRIMARY KEY,  
 id\_departament int, constraint id\_departament\_fk FOREIGN KEY (id\_departament) references DEPARTAMENT(id\_departament),  
 nume VARCHAR(20) NOT NULL  
 );   
CREATE SEQUENCE ID\_ECH  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;   
   
INSERT INTO Echipa  
VALUES (ID\_ECH.NEXTVAL,100,'Iconic');  
INSERT INTO Echipa  
VALUES (ID\_ECH.NEXTVAL,103,'Lightning Legends');  
INSERT INTO Echipa  
VALUES (ID\_ECH.NEXTVAL,104,'The Elite Team');  
INSERT INTO Echipa  
VALUES (ID\_ECH.NEXTVAL,102,'Crew X');  
INSERT INTO Echipa  
VALUES (ID\_ECH.NEXTVAL,101,'Ballerina Bellas');  
INSERT INTO Echipa  
VALUES (105,100,'Bears');  
INSERT INTO Echipa  
VALUES (106,100,'We Are The Champions');  
   
SELECT \* FROM echipa



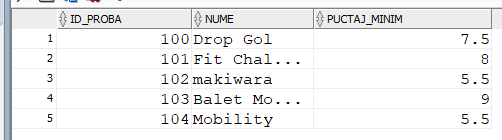
CREATE TABLE SPORTIV ( id\_sportiv int constraint id\_sportiv\_pk PRIMARY KEY,  
 id\_echipa int, constraint id\_echipa\_fk FOREIGN KEY (id\_echipa) references ECHIPA(id\_echipa),  
 nume VARCHAR(20) CONSTRAINT name\_sportiv NOT NULL,  
 prenume VARCHAR(20) CONSTRAINT prenume\_sportiv NOT NULL,  
 mail VARCHAR(20) CONSTRAINT mail\_sportiv NOT NULL,  
 nr\_telefon VARCHAR(20) NOT NULL,  
 nationalitate VARCHAR(20) NOT NULL,  
 data\_nasterii DATE NOT NULL  
 );  
CREATE SEQUENCE ID\_sportiv2  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;   
   
INSERT INTO Sportiv  
VALUES(ID\_sportiv2.NEXTVAL, 100, 'Ioan', 'Maria', 'ioan.maria@gmail.com','0765432189','romana',  
TO\_DATE('27-01-2002','dd-mm-yyyy'));  
INSERT INTO SPORTIV  
VALUES(ID\_sportiv2.NEXTVAL, 103, 'Mihailescu', 'Valeriu', 'vale.miha@gmail.com','0765232187','romana',  
TO\_DATE('23-09-1999','dd-mm-yyyy'));  
INSERT INTO SPORTIV  
VALUES(ID\_SPORTIV2.NEXTVAL, 104, 'Ion', 'Bogdan', 'ion.bogdan@gmail.com','0765232187','romana',  
TO\_DATE('25-10-2000','dd-mm-yyyy'));  
INSERT INTO SPORTIV  
VALUES(ID\_SPORTIV2.NEXTVAL, 102, 'Mira', 'Kiko', 'mira.kiko@gmail.com','0764634261','bulgara',  
TO\_DATE('23-11-1998','dd-mm-yyyy'));  
INSERT INTO SPORTIV  
VALUES(ID\_SPORTIV2.NEXTVAL, 101, 'Gherghe', 'Lara', 'lara.star@gmail.com','0764634261','romana',  
TO\_DATE('22-04-1998','dd-mm-yyyy'));  
INSERT INTO Sportiv  
VALUES(105, 102, 'Coman', 'Laurentiu', 'laur.coman@gmail.com','0765433265','romana',  
TO\_DATE('24-03-2003','dd-mm-yyyy'));  
INSERT INTO Sportiv  
VALUES(106, 108, 'Nitu', 'Victor', 'vic.nitu@gmail.com','0723433269','romana',  
TO\_DATE('25-04-2001','dd-mm-yyyy'));   
INSERT INTO Sportiv  
VALUES(107, 108, 'Ioan', 'Alex', 'alex.ioan@gmail.com','0723435276','romana',  
TO\_DATE('25-04-2001','dd-mm-yyyy'));  
INSERT INTO Sportiv  
VALUES(108, 107, 'Chirita', 'Ruxandra', 'ruxy23@gmail.com','0723432765','romana',  
TO\_DATE('13-06-1999','dd-mm-yyyy'));  
INSERT INTO Sportiv  
VALUES(109, 107, 'Luita', 'Gino', 'gino35@gmail.com','0723432381','romana',  
TO\_DATE('12-07-1999','dd-mm-yyyy'));  
   
SELECT \* FROM sportiv

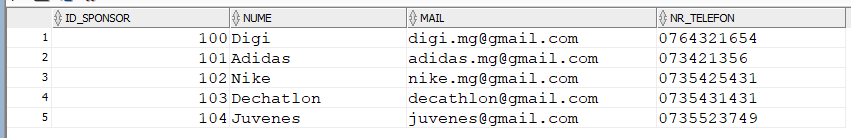
CREATE TABLE ANTRENOR (id\_antrenor int constraint id\_antrenor\_pk PRIMARY KEY,  
 nume VARCHAR(20) CONSTRAINT name\_antrenor NOT NULL,  
 prenume VARCHAR(20) CONSTRAINT prenume\_antrenor NOT NULL,  
 mail VARCHAR(20) CONSTRAINT mail\_antrenor NOT NULL,  
 nr\_telefon VARCHAR(20) NOT NULL);  
CREATE SEQUENCE ID\_antrenor  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;   
   
INSERT INTO ANTRENOR  
VALUES (ID\_antrenor.NEXTVAL,'Gigi','Vasile','gigi.vasi@gmail.com','0756432176');  
INSERT INTO ANTRENOR  
VALUES (ID\_antrenor.NEXTVAL,'Bicu','Maria','maria.bicu@gmail.com','0753421176');  
INSERT INTO ANTRENOR  
VALUES (ID\_antrenor.NEXTVAL,'Luiza','Amalia','lulu.ama@gmail.com','0753421234');  
INSERT INTO ANTRENOR  
VALUES (102,'Aslan','Mihai','mihai45@gmail.com','0753421321');  
INSERT INTO ANTRENOR  
VALUES (ID\_antrenor.NEXTVAL,'George','Miriam','miriam27@gmail.com','0753425376');  
   
SELECT \* FROM antrenor



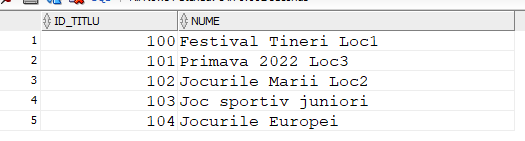
CREATE TABLE PROBA (id\_proba int constraint id\_proba\_pk PRIMARY KEY,  
 nume VARCHAR(20) CONSTRAINT name\_proba NOT NULL,  
 puctaj\_minim NUMBER(7,2) NOT NULL);

CREATE SEQUENCE id\_proba  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;   
   
INSERT INTO proba  
VALUES(id\_proba.NEXTVAL,'Drop Gol',7.5)  
INSERT INTO proba  
VALUES(id\_proba.NEXTVAL,'Fit Challenge',8)  
INSERT INTO proba  
VALUES(id\_proba.NEXTVAL,'makiwara',5.5)  
INSERT INTO proba  
VALUES(id\_proba.NEXTVAL,'Balet Modobradi',9)  
INSERT INTO proba  
VALUES(id\_proba.NEXTVAL,'Mobility',5.5)  
   
SELECT \* FROM proba

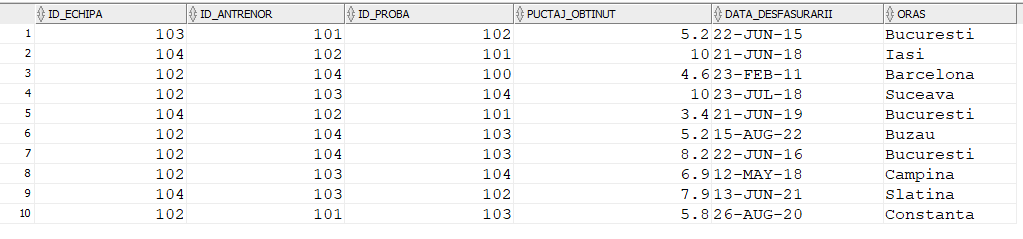


CREATE TABLE SPONSOR (id\_sponsor int constraint id\_sponsor\_pk PRIMARY KEY,  
 nume VARCHAR(20) CONSTRAINT name\_sponsor NOT NULL,  
 mail VARCHAR(20) CONSTRAINT mail\_sponsor NOT NULL,  
 nr\_telefon VARCHAR(20) NOT NULL);  
CREATE SEQUENCE id\_sponsor  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;  
   
INSERT INTO sponsor  
VALUES(id\_sponsor.NEXTVAL,'Digi','digi.mg@gmail.com','0764321654')  
INSERT INTO sponsor  
VALUES(id\_sponsor.NEXTVAL,'Adidas','adidas.mg@gmail.com','073421356')  
INSERT INTO sponsor  
VALUES(id\_sponsor.NEXTVAL,'Nike','nike.mg@gmail.com','0735425431')  
INSERT INTO sponsor  
VALUES(id\_sponsor.NEXTVAL,'Dechatlon','decathlon@gmail.com','0735431431')  
INSERT INTO sponsor  
VALUES(id\_sponsor.NEXTVAL,'Juvenes','juvenes@gmail.com','0735523749')  
   
SELECT \* FROM sponsor

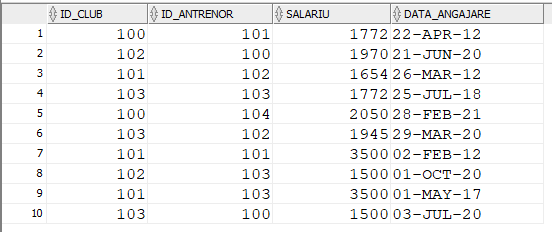
CREATE TABLE TITLU (id\_titlu int constraint id\_titlu\_pk PRIMARY KEY,  
 nume VARCHAR(20) CONSTRAINT name\_titlu NOT NULL);  
CREATE SEQUENCE id\_titlu  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;  
   
INSERT INTO titlu  
VALUES(id\_titlu.NEXTVAL,'Festival Tineri Loc1')  
INSERT INTO titlu  
VALUES(id\_titlu.NEXTVAL,'Primava 2022 Loc3')  
INSERT INTO titlu  
VALUES(id\_titlu.NEXTVAL,'Jocurile Marii Loc2')  
INSERT INTO titlu  
VALUES(id\_titlu.NEXTVAL,'Joc sportiv juniori')  
INSERT INTO titlu  
VALUES(id\_titlu.NEXTVAL,'Jocurile Europei')  
   
SELECT \* FROM titlu



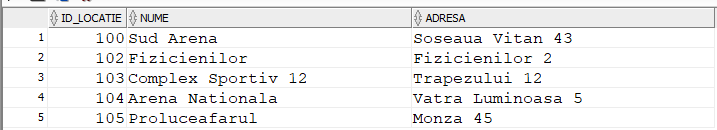
CREATE TABLE COMPETITIE (id\_echipa int, constraint id\_echipa\_fk\_comp FOREIGN KEY (id\_echipa) references ECHIPA(id\_echipa),  
 id\_antrenor int, constraint id\_antrenor\_fk\_comp FOREIGN KEY (id\_antrenor) references ANTRENOR(id\_antrenor),  
 id\_proba int, constraint id\_proba\_fk\_comp FOREIGN KEY (id\_proba) references PROBA(id\_proba),  
 puctaj\_obtinut NUMBER(7,2) NOT NULL,  
 data\_desfasurarii DATE NOT NULL,  
 oras VARCHAR(20) NOT NULL,  
 CONSTRAINT pk\_competitie primary key(id\_echipa,id\_antrenor,id\_proba)  
 );  
INSERT INTO competitie  
VALUES(103,101,102,5.2,TO\_DATE('22-06-2015','dd-mm-yyyy'),'Bucuresti')  
INSERT INTO competitie  
VALUES(104,102,101,10,TO\_DATE('21-06-2018','dd-mm-yyyy'),'Iasi')  
INSERT INTO competitie  
VALUES(102,104,100,4.6,TO\_DATE('23-02-2011','dd-mm-yyyy'),'Barcelona')  
INSERT INTO competitie  
VALUES(102,103,104,10,TO\_DATE('23-07-2018','dd-mm-yyyy'),'Suceava')  
INSERT INTO competitie  
VALUES(104,102,101,3.4,TO\_DATE('21-06-2019','dd-mm-yyyy'),'Bucuresti')  
INSERT INTO competitie  
VALUES(102,104,103,5.2,TO\_DATE('15-08-2022','dd-mm-yyyy'),'Buzau')  
INSERT INTO competitie  
VALUES(102,104,103,8.2,TO\_DATE('22-06-2016','dd-mm-yyyy'),'Bucuresti')  
INSERT INTO competitie  
VALUES(102,103,104,6.9,TO\_DATE('12-05-2018','dd-mm-yyyy'),'Campina')  
INSERT INTO competitie  
VALUES(104,103,102,7.9,TO\_DATE('13-06-2021','dd-mm-yyyy'),'Slatina')  
INSERT INTO competitie  
VALUES(102,101,103,5.8,TO\_DATE('26-08-2020','dd-mm-yyyy'),'Constanta')  
INSERT INTO competitie  
VALUES(108,101,102,10,TO\_DATE('22-06-2015','dd-mm-yyyy'),'Bucuresti')  
INSERT INTO competitie  
VALUES(107,102,104,3,TO\_DATE('22-06-2015','dd-mm-yyyy'),'Galati')

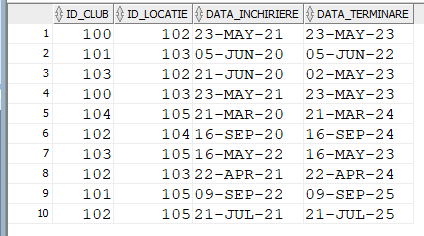


CREATE TABLE ANTRENOR\_CLUB (id\_club int, constraint id\_club\_fk\_comp FOREIGN KEY (id\_club) references CLUB(id\_club),  
 id\_antrenor int, constraint id\_antrenor\_fk\_compus FOREIGN KEY (id\_antrenor) references ANTRENOR(id\_antrenor),  
 salariu int NOT NULL,  
 data\_angajare DATE NOT NULL,  
 CONSTRAINT pk\_antrenor\_club primary key(id\_club,id\_antrenor)  
 );  
INSERT INTO Antrenor\_Club  
VALUES(100,101,1772,TO\_DATE('22-04-2012','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(102,100,1970,TO\_DATE('21-06-2020','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(101,102,1654,TO\_DATE('26-03-2012','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(103,103,1772,TO\_DATE('25-07-2018','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(100,104,2050,TO\_DATE('28-02-2021','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(103,102,1945,TO\_DATE('29-03-2020','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(101,101,3500,TO\_DATE('02-02-2012','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(102,103,1500,TO\_DATE('01-10-2020','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(101,103,3500,TO\_DATE('01-05-2017','dd-mm-yyyy'))  
INSERT INTO Antrenor\_Club  
VALUES(103,100,1500,TO\_DATE('03-07-2020','dd-mm-yyyy'))  
   
SELECT \* FROM antrenor\_club

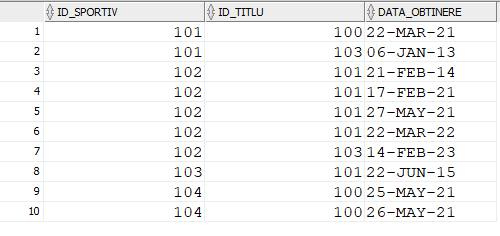


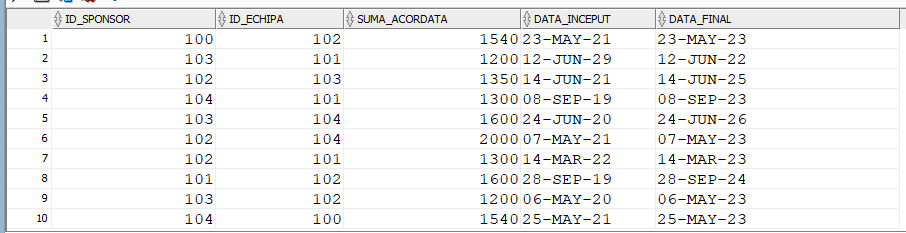
CREATE TABLE LOCATIE (id\_locatie int constraint id\_locatie\_pk PRIMARY KEY,  
 nume VARCHAR(20) CONSTRAINT name\_locatie NOT NULL,  
 adresa VARCHAR(20) CONSTRAINT adresa\_locatie NOT NULL  
 );  
CREATE SEQUENCE id\_locatie  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCYCLE;  
   
INSERT INTO locatie   
VALUES(id\_locatie.NEXTVAL,'Sud Arena','Soseaua Vitan 43')  
INSERT INTO locatie   
VALUES(id\_locatie.NEXTVAL,'Fizicienilor','Fizicienilor 2')  
INSERT INTO locatie   
VALUES(id\_locatie.NEXTVAL,'Complex Sportiv 12','Trapezului 12')  
INSERT INTO locatie   
VALUES(id\_locatie.NEXTVAL,'Arena Nationala','Vatra Luminoasa 5')  
INSERT INTO locatie   
VALUES(id\_locatie.NEXTVAL,'Proluceafarul','Monza 45')  
   
SELECT \* FROM locatie

CREATE TABLE CLUB\_LOCATIE (id\_club int, constraint id\_club\_fk\_comp\_loc FOREIGN KEY (id\_club) references CLUB(id\_club),   
 id\_locatie int, constraint id\_loc\_fk\_comp FOREIGN KEY (id\_locatie) references LOCATIE(id\_locatie),  
 data\_inchiriere DATE NOT NULL,  
 data\_terminare DATE NOT NULL,  
 CONSTRAINT pk\_club\_locatie primary key(id\_club,id\_locatie)  
 );  
INSERT INTO Club\_Locatie  
VALUES(100,102,TO\_DATE('23-05-2021','dd-mm-yyyy'),TO\_DATE('23-05-2023','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(101,103,TO\_DATE('05-06-2020','dd-mm-yyyy'),TO\_DATE('05-06-2022','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(103,102,TO\_DATE('21-06-2020','dd-mm-yyyy'),TO\_DATE('2-05-2023','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(100,103,TO\_DATE('23-05-2021','dd-mm-yyyy'),TO\_DATE('23-05-2023','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(104,105,TO\_DATE('21-03-2020','dd-mm-yyyy'),TO\_DATE('21-03-2024','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(102,104,TO\_DATE('16-09-2020','dd-mm-yyyy'),TO\_DATE('16-09-2024','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(100,102,TO\_DATE('23-05-2021','dd-mm-yyyy'),TO\_DATE('23-05-2023','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(103,105,TO\_DATE('16-05-2022','dd-mm-yyyy'),TO\_DATE('16-05-2023','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(102,103,TO\_DATE('22-04-2021','dd-mm-yyyy'),TO\_DATE('22-04-2024','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(101,105,TO\_DATE('09-09-2022','dd-mm-yyyy'),TO\_DATE('09-09-2025','dd-mm-yyyy'))  
INSERT INTO Club\_Locatie  
VALUES(102,105,TO\_DATE('21-07-2021','dd-mm-yyyy'),TO\_DATE('21-07-2025','dd-mm-yyyy'))  
   
SELECT \* FROM club\_locatie

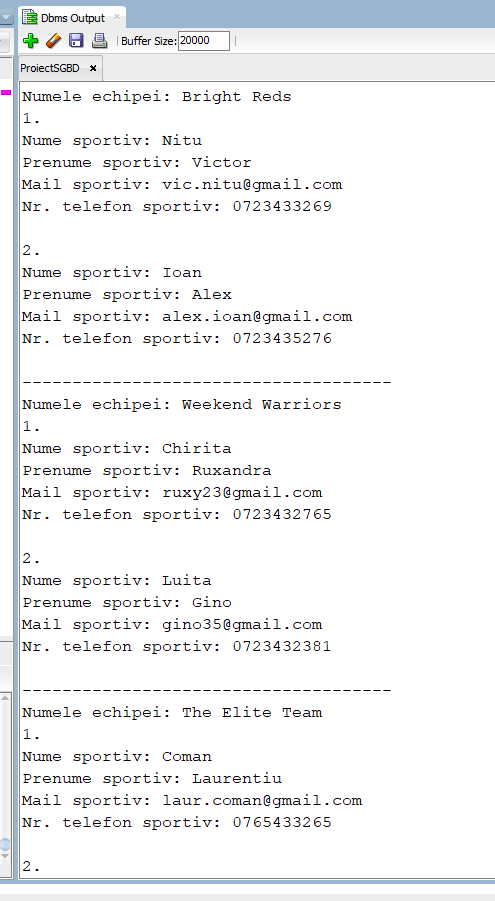


CREATE TABLE SPORTIV\_TITLU (id\_sportiv int, constraint id\_sportiv\_fk\_titlu FOREIGN KEY (id\_sportiv) references SPORTIV(id\_sportiv),  
 id\_titlu int, constraint id\_titlu\_fk\_titlu FOREIGN KEY (id\_titlu) references TITLU(id\_titlu),  
 data\_obtinere DATE NOT NULL,  
 CONSTRAINT pk\_sportiv\_titlu\_compus primary key(id\_sportiv,id\_titlu)  
 );  
   
INSERT INTO sportiv\_titlu  
VALUES(104,100,TO\_DATE('25-05-2021','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(104,100,TO\_DATE('26-05-2021','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(102,101,TO\_DATE('21-02-2014','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(103,101,TO\_DATE('22-06-2015','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(101,103,TO\_DATE('06-01-2013','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(102,101,TO\_DATE('17-02-2021','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(102,101,TO\_DATE('22-03-2022','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(101,100,TO\_DATE('22-03-2021','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(102,103,TO\_DATE('14-02-2023','dd-mm-yyyy'))  
INSERT INTO sportiv\_titlu  
VALUES(102,101,TO\_DATE('27-05-2021','dd-mm-yyyy'))  
   
SELECT \* FROM sportiv\_titlu



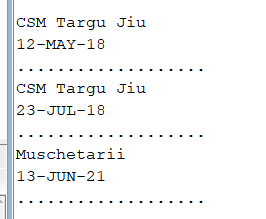
CREATE TABLE ECHIPA\_SPONSOR ( id\_sponsor int, constraint id\_sponsor\_fk\_echipa FOREIGN KEY (id\_sponsor) references SPONSOR(id\_sponsor),  
 id\_echipa int, constraint id\_echipa\_fk\_sponsor FOREIGN KEY (id\_echipa) references ECHIPA(id\_echipa),  
 suma\_acordata int NOT NULL,  
 data\_inceput DATE NOT NULL,  
 data\_final DATE NOT NULL,  
 CONSTRAINT pk\_echipa\_sponsor\_compus primary key(id\_sponsor,id\_echipa)  
 );  
INSERT INTO Echipa\_Sponsor  
VALUES(100,102,1540,TO\_DATE('23-05-2021','dd-mm-yyyy'),TO\_DATE('23-05-2023','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(103,101,1200,TO\_DATE('12-06-2029','dd-mm-yyyy'),TO\_DATE('12-06-2022','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(102,103,1350,TO\_DATE('14-06-2021','dd-mm-yyyy'),TO\_DATE('14-06-2025','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(104,101,1300,TO\_DATE('08-09-2019','dd-mm-yyyy'),TO\_DATE('08-09-2023','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(103,104,1600,TO\_DATE('24-06-2020','dd-mm-yyyy'),TO\_DATE('24-06-2026','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(102,104,2000,TO\_DATE('07-05-2021','dd-mm-yyyy'),TO\_DATE('07-05-2023','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(102,101,1300,TO\_DATE('14-03-2022','dd-mm-yyyy'),TO\_DATE('14-03-2023','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(101,102,1600,TO\_DATE('28-09-2019','dd-mm-yyyy'),TO\_DATE('28-09-2024','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(103,102,1200,TO\_DATE('06-05-2020','dd-mm-yyyy'),TO\_DATE('06-05-2023','dd-mm-yyyy'))  
INSERT INTO Echipa\_Sponsor  
VALUES(104,100,1540,TO\_DATE('25-05-2021','dd-mm-yyyy'),TO\_DATE('25-05-2023','dd-mm-yyyy'))  
   
SELECT \* FROM echipa\_sponsor

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

Pentru un Club dat, afisati toti sportivii(nume si DATE de contact) din echipele din cadrul clubului.   
   
CREATE OR REPLACE PROCEDURE cerinta6 (numeclub CLUB.nume%TYPE)  
AS  
 TYPE tablou\_indexat IS TABLE OF echipa%ROWTYPE INDEX BY PLS\_INTEGER;  
 echipe tablou\_indexat;  
 TYPE tablou\_imbricat IS TABLE OF sportiv%ROWTYPE;  
 sportivi tablou\_imbricat:= tablou\_imbricat();  
 cnt NUMBER(4);  
BEGIN  
 SELECT e.id\_echipa, e.id\_departament, e.nume  
 BULK COLLECT INTO echipe  
 FROM club c, departament d, echipa e  
 WHERE e.id\_departament=d.id\_departament AND d.id\_club = c.id\_club AND c.nume=numeclub;  
 FOR i IN 1..echipe.COUNT LOOP  
 DBMS\_OUTPUT.put\_line('-------------------------------------');  
 DBMS\_OUTPUT.put\_line('Numele echipei: '||TO\_CHAR(echipe(i).nume));  
 cnt:=0;  
 SELECT s.id\_sportiv, s.id\_echipa, s.nume, s.prenume, s.mail, s.nr\_telefon, s.nationalitate, s.data\_nasterii  
 BULK COLLECT INTO sportivi  
 FROM sportiv s  
 WHERE s.id\_echipa = echipe(i).id\_echipa;  
 FOR j IN 1..sportivi.COUNT LOOP  
 cnt:=cnt+1;   
 DBMS\_OUTPUT.put\_line(TO\_CHAR(cnt)||'.');  
 DBMS\_OUTPUT.put\_line('Nume sportiv: '||sportivi(j).nume);  
 DBMS\_OUTPUT.put\_line('Prenume sportiv: '||sportivi(j).prenume);  
 DBMS\_OUTPUT.put\_line('Mail sportiv: '||sportivi(j).mail);  
 DBMS\_OUTPUT.put\_line('Nr. telefon sportiv: '||sportivi(j).nr\_telefon);  
 DBMS\_OUTPUT.put\_line('');  
 END LOOP;  
 END LOOP;   
END;  
/  
   
BEGIN  
cerinta6('CSM Targu Jiu');  
END;  
/



7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.

Pentru un antenor dat(prin nume) afisati pentru competitiile la care a luat parte   
impreuna cu echipa, clubul din care echipa cu un punctaj satisfacator face parte si  
data desfasurarii competitiei. Daca exista mai multi antrenori cu acelasi nume se va alege primul.  
   
   
CREATE OR REPLACE PROCEDURE cerinta7 (nume\_antrenor antrenor.nume%TYPE)  
AS  
 id int;  
 CURSOR compet( id antrenor.id\_antrenor%TYPE) IS  
 SELECT c.id\_echipa, c.data\_desfasurarii  
 FROM competitie c, proba p   
 WHERE c.id\_antrenor = id AND c.puctaj\_obtinut>=p.puctaj\_minim AND c.id\_proba=p.id\_proba ;  
 id\_echipe competitie.id\_echipa%TYPE;  
 data competitie.data\_desfasurarii%TYPE;  
 is\_found\_rec BOOLEAN := FALSE;   
BEGIN  
 SELECT id\_antrenor  
 INTO id  
 FROM antrenor  
 WHERE nume=nume\_antrenor AND ROWNUM=1;  
   
 OPEN compet(id);  
 LOOP  
 FETCH compet INTO id\_echipe, data;  
 EXIT WHEN compet%NOTFOUND;  
 is\_found\_rec := TRUE;  
 FOR i IN (SELECT c.nume  
 FROM club c, departament d, echipa e  
 WHERE e.id\_departament = d.id\_departament AND d.id\_club=c.id\_club AND e.id\_echipa=id\_echipe  
 ) LOOP  
 DBMS\_OUTPUT.put\_line(i.nume);  
 END LOOP;  
 DBMS\_OUTPUT.put\_line(TO\_CHAR(data));  
 DBMS\_OUTPUT.put\_line('...................');  
 END LOOP;  
 CLOSE compet;  
 IF NOT is\_found\_rec THEN   
 DBMS\_OUTPUT.put\_line('Nu exista echipe');  
 END IF;  
END;  
/

BEGIN

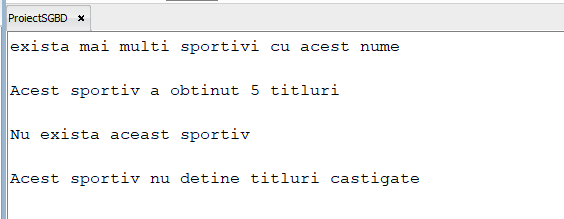
cerinta7('Luiza');

END;

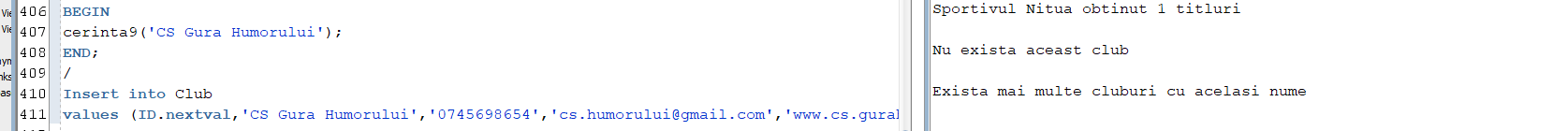
/

# 8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Pentru un sportiv dat dupa nume calculati numarul de titluri castigate  
CREATE OR REPLACE FUNCTION cerinta8 (parametru sportiv.nume%TYPE) RETURN NUMBER AS  
 titluri NUMBER;  
 TYPE tabel\_id IS TABLE OF sportiv.id\_sportiv%TYPE INDEX BY PLS\_INTEGER;  
 id\_uri tabel\_id;  
 Nu\_exista\_sportivi EXCEPTION;  
 Nume\_repetat EXCEPTION;  
 nu\_exista\_titluri EXCEPTION;  
BEGIN  
 SELECT id\_sportiv  
 BULK COLLECT INTO id\_uri  
 FROM sportiv  
 WHERE nume = parametru;  
 IF id\_uri.COUNT = 0 THEN   
 RAISE Nu\_exista\_sportivi;  
 END IF;  
 IF id\_uri.COUNT > 1 THEN   
 RAISE Nume\_repetat;  
 END IF;  
 SELECT COUNT (s.id\_sportiv)  
 INTO titluri  
 FROM sportiv s  
 join sportiv\_titlu st ON (s.id\_sportiv = st.id\_sportiv)  
 join titlu t  
 ON t.id\_titlu = st.id\_titlu  
 WHERE s.id\_sportiv = id\_uri(1);  
 IF titluri = 0 THEN   
 RAISE nu\_exista\_titluri;  
 ELSE  
 RETURN titluri;  
 END IF;  
EXCEPTION   
 WHEN nu\_exista\_sportivi THEN   
 DBMS\_OUTPUT.put\_line('Nu exista aceast sportiv');   
 RETURN -1;  
 WHEN Nume\_repetat THEN   
 DBMS\_OUTPUT.put\_line('exista mai multi sportivi cu acest nume');  
 RETURN -1;  
 WHEN nu\_exista\_titluri THEN   
 DBMS\_OUTPUT.put\_line('Acest sportiv nu detine titluri castigate');   
 RETURN -1;  
END;  
/  
   
 DECLARE   
 nr NUMBER;  
 BEGIN   
 nr:=cerinta8('Ioan');  
 IF(nr>=1) THEN  
 DBMS\_OUTPUT.put\_line('Acest sportiv a obtinut '|| TO\_CHAR(nr) || ' titluri');  
 END IF;  
 END;  
   
 DECLARE   
 nr NUMBER;  
 BEGIN   
 nr:=cerinta8('Ion');  
 IF(nr>=1) THEN  
 DBMS\_OUTPUT.put\_line('Acest sportiv a obtinut '|| TO\_CHAR(nr) || ' titluri');  
 END IF;  
 END;  
   
 DECLARE   
 nr NUMBER;  
 BEGIN   
 nr:=cerinta8('Maria');  
 IF(nr>=1) THEN  
 DBMS\_OUTPUT.put\_line('Acest sportiv a obtinut '|| TO\_CHAR(nr) || ' titluri');  
 END IF;  
 END;  
   
 DECLARE   
 nr NUMBER;  
 BEGIN   
 nr:=cerinta8('Luita');  
 IF(nr>=1) THEN  
 DBMS\_OUTPUT.put\_line('Acest sportiv a obtinut '|| TO\_CHAR(nr) || ' titluri');  
 END IF;  
 END;

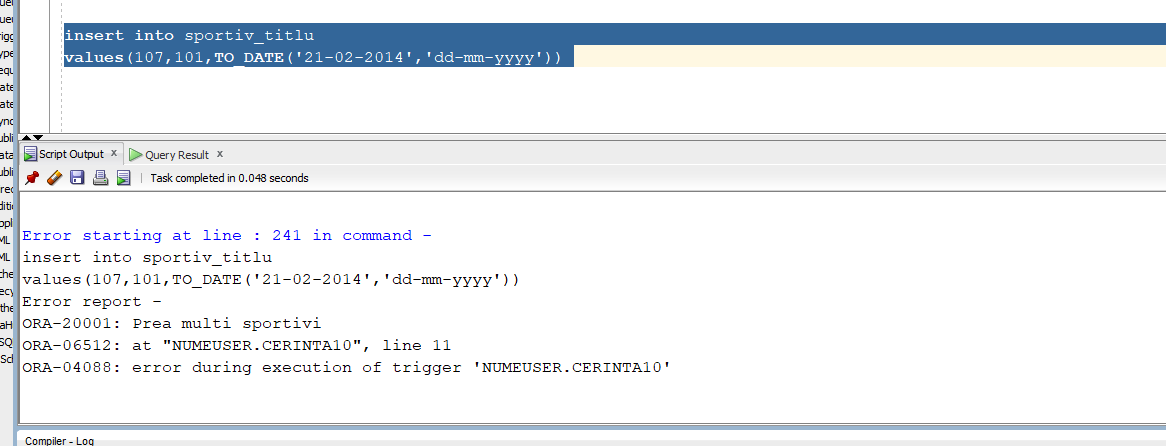


# 9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandăSQL5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Pentru un club afisati numele sportivilor si nr de titluri dobandite  
   
 CREATE OR REPLACE PROCEDURE cerinta9 (nume\_club club.nume%TYPE) AS  
 id\_club\_cautat int;  
 TYPE tabel IS RECORD(  
 id\_sportiv\_club sportiv.id\_sportiv%TYPE,  
 nr\_titluri NUMBER  
 );  
 TYPE sportiv\_titlu IS TABLE OF tabel;  
 sportiv\_titluri sportiv\_titlu;  
 nume\_sportiv VARCHAR(20);  
 nu\_exista EXCEPTION;  
BEGIN  
 SELECT id\_club  
 INTO id\_club\_cautat  
 FROM club  
 WHERE nume = nume\_club;  
   
 SELECT s.id\_sportiv, COUNT(st.id\_titlu)  
 BULK COLLECT INTO sportiv\_titluri  
 FROM sportiv s   
 join sportiv\_titlu st ON(st.id\_sportiv= s.id\_sportiv)  
 join echipa e ON(s.id\_echipa = e.id\_echipa)  
 join departament d ON (e.id\_departament = d.id\_departament)  
 join club c ON (c.id\_club = d.id\_club)  
 WHERE c.id\_club = id\_club\_cautat  
 GROUP BY s.id\_sportiv;  
 IF sportiv\_titluri.COUNT=0 THEN  
 RAISE nu\_exista;  
 END IF;  
 FOR i IN 1..sportiv\_titluri.COUNT LOOP  
 SELECT nume  
 INTO nume\_sportiv  
 FROM sportiv   
 WHERE id\_sportiv = sportiv\_titluri(i).id\_sportiv\_club ;  
 DBMS\_OUTPUT.put\_line('Sportivul '|| nume\_sportiv || 'a obtinut ' || sportiv\_titluri(i).nr\_titluri || ' titluri' );  
 END LOOP;   
EXCEPTION   
   
 WHEN NO\_DATA\_FOUND THEN   
 DBMS\_OUTPUT.put\_line('Nu exista aceast club');   
 WHEN TOO\_MANY\_ROWS THEN   
 DBMS\_OUTPUT.put\_line('Exista mai multe cluburi cu acelasi nume');   
 WHEN nu\_exista THEN   
 DBMS\_OUTPUT.put\_line('Clubul nu are sportivi');   
END;

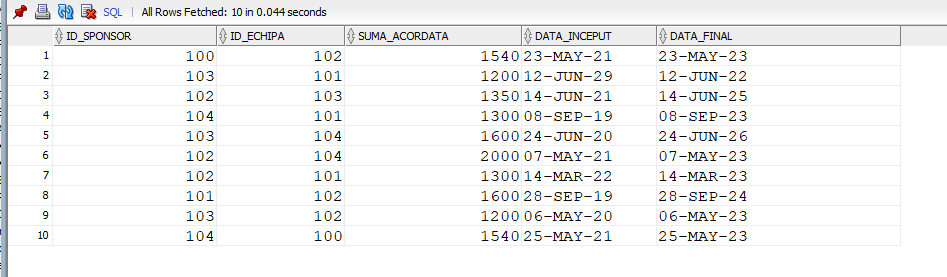


# 10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

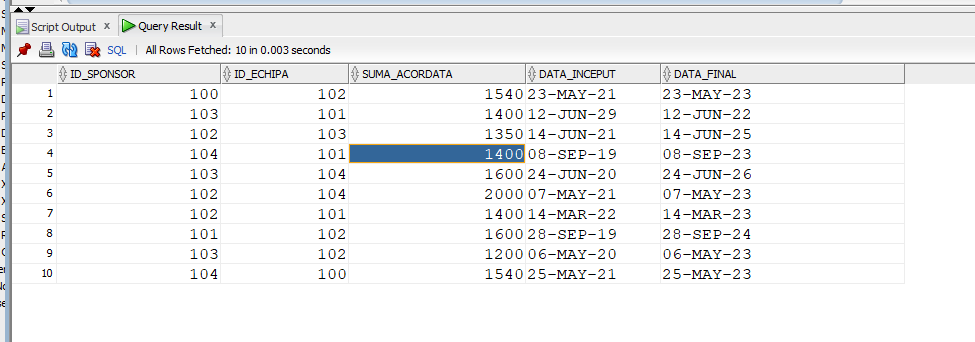
IN tabela sportiv\_titlu pot fi inserati maxim 6 sportivi diferiti  
   
 CREATE OR REPLACE TRIGGER cerinta10   
before INSERT ON sportiv\_titlu  
DECLARE   
 sportivi NUMBER;  
BEGIN   
 SELECT COUNT (DISTINCT (s.id\_sportiv))  
 INTO sportivi  
 FROM sportiv s  
 join sportiv\_titlu st ON (s.id\_sportiv = st.id\_sportiv)  
 join titlu t  
 ON t.id\_titlu = st.id\_titlu;  
 IF sportivi > 5 THEN  
 RAISE\_APPLICATION\_ERROR(-20001,'Prea multi sportivi');  
 END IF;  
END;

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

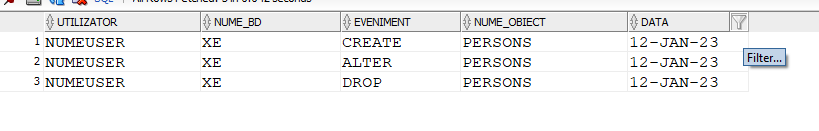
Atunci cand un sportiv isi schimba echipa, suma acordata de sponsori va creste cu 100 pentru acea echipa  
   
CREATE OR REPLACE TRIGGER cerinta11   
 after UPDATE OF id\_echipa ON sportiv  
 FOR each ROW  
DECLARE  
 exceptie EXCEPTION;  
 TYPE suma IS TABLE OF echipa\_sponsor.suma\_acordata%TYPE INDEX BY PLS\_INTEGER;  
 sume suma;  
BEGIN  
 IF(:old.id\_echipa=:NEW.id\_echipa)  
 THEN RAISE exceptie;  
 END IF;  
 SELECT suma\_acordata  
 BULK COLLECT INTO sume  
 FROM echipa\_sponsor  
 WHERE id\_echipa=(:NEW.id\_echipa);  
 FOR i IN 1..sume.COUNT LOOP  
 UPDATE echipa\_sponsor  
 SET suma\_acordata = sume(i)+100  
 WHERE id\_echipa = (:NEW.id\_echipa);  
 END LOOP;  
   
EXCEPTION   
 WHEN exceptie THEN  
 RAISE\_APPLICATION\_ERROR (-20003, 'Sportivul nu s-a mutat din echipa');   
 END;   
   
   
 UPDATE sportiv  
 SET id\_echipa=101  
 WHERE id\_sportiv=100;

Inainte

Dupa



12. Definiți un trigger de tip LDD. Declanșați trigger-ul.  
   
CREATE TABLE istoric  
 (utilizator VARCHAR2(30),  
 nume\_bd VARCHAR2(50),  
 eveniment VARCHAR2(20),  
 nume\_obiect VARCHAR2(30),  
 data DATE);  
   
CREATE OR REPLACE TRIGGER cerinta12  
AFTER CREATE OR ALTER OR DROP ON SCHEMA  
BEGIN  
 INSERT INTO istoric  
 VALUES (SYS.LOGIN\_USER, SYS.DATABASE\_NAME, SYS.SYSEVENT,SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);  
END;  
   
   
 CREATE TABLE Persons (  
 PersonID int,  
 LastName VARCHAR(255),  
 FirstName VARCHAR(255),  
 Address VARCHAR(255),  
 City VARCHAR(255)  
);   
   
ALTER TABLE Persons DROP column City;  
   
DROP TABLE Persons;  
   
SELECT \* FROM istoric;



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

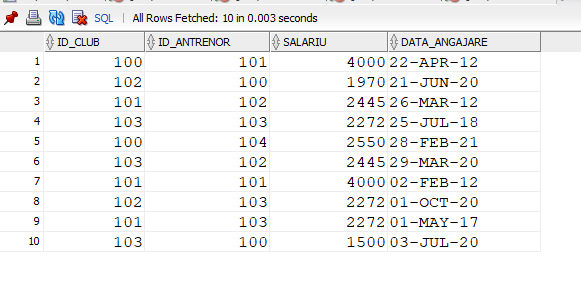
CREATE OR REPLACE PACKAGE cerinta13 AS   
 PROCEDURE cerinta6 (numeclub CLUB.nume%TYPE);  
 PROCEDURE cerinta7 (nume\_antrenor antrenor.nume%TYPE);  
 FUNCTION cerinta8 (parametru sportiv.nume%TYPE) RETURN NUMBER;  
 PROCEDURE cerinta9 (nume\_club club.nume%TYPE);  
END cerinta13;  
/  
CREATE OR REPLACE PACKAGE BODY cerinta13  
AS  
 PROCEDURE cerinta6 (numeclub CLUB.nume%TYPE)  
AS  
 TYPE tablou\_indexat IS TABLE OF echipa%ROWTYPE INDEX BY PLS\_INTEGER;  
 echipe tablou\_indexat;  
 TYPE tablou\_imbricat IS TABLE OF sportiv%ROWTYPE;  
 sportivi tablou\_imbricat:= tablou\_imbricat();  
 cnt NUMBER(4);  
BEGIN  
 SELECT e.id\_echipa, e.id\_departament, e.nume  
 BULK COLLECT INTO echipe  
 FROM club c, departament d, echipa e  
 WHERE e.id\_departament=d.id\_departament AND d.id\_club = c.id\_club AND c.nume=numeclub;  
 FOR i IN 1..echipe.COUNT LOOP  
 DBMS\_OUTPUT.put\_line('-------------------------------------');  
 DBMS\_OUTPUT.put\_line('Numele echipei: '||TO\_CHAR(echipe(i).nume));  
 cnt:=0;  
 SELECT s.id\_sportiv, s.id\_echipa, s.nume, s.prenume, s.mail, s.nr\_telefon, s.nationalitate, s.data\_nasterii  
 BULK COLLECT INTO sportivi  
 FROM sportiv s  
 WHERE s.id\_echipa = echipe(i).id\_echipa;  
 FOR j IN 1..sportivi.COUNT LOOP  
 cnt:=cnt+1;   
 DBMS\_OUTPUT.put\_line(TO\_CHAR(cnt)||'.');  
 DBMS\_OUTPUT.put\_line('Nume sportiv: '||sportivi(j).nume);  
 DBMS\_OUTPUT.put\_line('Prenume sportiv: '||sportivi(j).prenume);  
 DBMS\_OUTPUT.put\_line('Mail sportiv: '||sportivi(j).mail);  
 DBMS\_OUTPUT.put\_line('Nr. telefon sportiv: '||sportivi(j).nr\_telefon);  
 DBMS\_OUTPUT.put\_line('');  
 END LOOP;  
 END LOOP;   
END cerinta6;  
 PROCEDURE cerinta7 (nume\_antrenor antrenor.nume%TYPE)  
AS  
 id int;  
 CURSOR compet( id antrenor.id\_antrenor%TYPE) IS  
 SELECT c.id\_echipa, c.data\_desfasurarii  
 FROM competitie c, proba p   
 WHERE c.id\_antrenor = id AND c.puctaj\_obtinut>=p.puctaj\_minim AND c.id\_proba=p.id\_proba ;  
 id\_echipe competitie.id\_echipa%TYPE;  
 data competitie.data\_desfasurarii%TYPE;  
 is\_found\_rec BOOLEAN := FALSE;   
BEGIN  
 SELECT id\_antrenor  
 INTO id  
 FROM antrenor  
 WHERE nume=nume\_antrenor AND ROWNUM=1;  
   
 OPEN compet(id);  
 LOOP  
 FETCH compet INTO id\_echipe, data;  
 EXIT WHEN compet%NOTFOUND;  
 is\_found\_rec := TRUE;  
 FOR i IN (SELECT c.nume  
 FROM club c, departament d, echipa e  
 WHERE e.id\_departament = d.id\_departament AND d.id\_club=c.id\_club AND e.id\_echipa=id\_echipe  
 ) LOOP  
 DBMS\_OUTPUT.put\_line(i.nume);  
 END LOOP;  
 DBMS\_OUTPUT.put\_line(TO\_CHAR(data));  
 DBMS\_OUTPUT.put\_line('...................');  
 END LOOP;  
 CLOSE compet;  
 IF NOT is\_found\_rec THEN   
 DBMS\_OUTPUT.put\_line('Nu exista echipe');  
 END IF;  
END cerinta7;  
 FUNCTION cerinta8 (parametru sportiv.nume%TYPE) RETURN NUMBER AS  
 titluri NUMBER;  
 TYPE tabel\_id IS TABLE OF sportiv.id\_sportiv%TYPE INDEX BY PLS\_INTEGER;  
 id\_uri tabel\_id;  
 Nu\_exista\_sportivi EXCEPTION;  
 Nume\_repetat EXCEPTION;  
 nu\_exista\_titluri EXCEPTION;  
BEGIN  
 SELECT id\_sportiv  
 BULK COLLECT INTO id\_uri  
 FROM sportiv  
 WHERE nume = parametru;  
 IF id\_uri.COUNT = 0 THEN   
 RAISE Nu\_exista\_sportivi;  
 END IF;  
 IF id\_uri.COUNT > 1 THEN   
 RAISE Nume\_repetat;  
 END IF;  
 SELECT COUNT (s.id\_sportiv)  
 INTO titluri  
 FROM sportiv s  
 join sportiv\_titlu st ON (s.id\_sportiv = st.id\_sportiv)  
 join titlu t  
 ON t.id\_titlu = st.id\_titlu  
 WHERE s.id\_sportiv = id\_uri(1);  
 IF titluri = 0 THEN   
 RAISE nu\_exista\_titluri;  
 ELSE  
 RETURN titluri;  
 END IF;  
EXCEPTION   
 WHEN nu\_exista\_sportivi THEN   
 DBMS\_OUTPUT.put\_line('Nu exista aceast sportiv');   
 RETURN -1;  
 WHEN Nume\_repetat THEN   
 DBMS\_OUTPUT.put\_line('exista mai multi sportivi cu acest nume');  
 RETURN -1;  
 WHEN nu\_exista\_titluri THEN   
 DBMS\_OUTPUT.put\_line('Acest sportiv nu detine titluri castigate');   
 RETURN -1;  
END cerinta8;  
 PROCEDURE cerinta9 (nume\_club club.nume%TYPE) AS  
 id\_club\_cautat int;  
 TYPE tabel IS RECORD(  
 id\_sportiv\_club sportiv.id\_sportiv%TYPE,  
 nr\_titluri NUMBER  
 );  
 TYPE sportiv\_titlu IS TABLE OF tabel;  
 sportiv\_titluri sportiv\_titlu;  
 nume\_sportiv VARCHAR(20);  
 nu\_exista EXCEPTION;  
BEGIN  
 SELECT id\_club  
 INTO id\_club\_cautat  
 FROM club  
 WHERE nume = nume\_club;  
   
 SELECT s.id\_sportiv, COUNT(st.id\_titlu)  
 BULK COLLECT INTO sportiv\_titluri  
 FROM sportiv s   
 join sportiv\_titlu st ON(st.id\_sportiv= s.id\_sportiv)  
 join echipa e ON(s.id\_echipa = e.id\_echipa)  
 join departament d ON (e.id\_departament = d.id\_departament)  
 join club c ON (c.id\_club = d.id\_club)  
 WHERE c.id\_club = id\_club\_cautat  
 GROUP BY s.id\_sportiv;  
 IF sportiv\_titluri.COUNT=0 THEN  
 RAISE nu\_exista;  
 END IF;  
 FOR i IN 1..sportiv\_titluri.COUNT LOOP  
 SELECT nume  
 INTO nume\_sportiv  
 FROM sportiv   
 WHERE id\_sportiv = sportiv\_titluri(i).id\_sportiv\_club ;  
 DBMS\_OUTPUT.put\_line('Sportivul '|| nume\_sportiv || 'a obtinut ' || sportiv\_titluri(i).nr\_titluri || ' titluri' );  
 END LOOP;   
EXCEPTION   
   
 WHEN NO\_DATA\_FOUND THEN   
 DBMS\_OUTPUT.put\_line('Nu exista aceast club');   
 WHEN TOO\_MANY\_ROWS THEN   
 DBMS\_OUTPUT.put\_line('Exista mai multe cluburi cu acelasi nume');   
 WHEN nu\_exista THEN   
 DBMS\_OUTPUT.put\_line('Clubul nu are sportivi');   
END cerinta9;  
END cerinta13;  
/  
   
EXECUTE cerinta13.cerinta6('CSM Targu Jiu');

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

CREATE TYPE antrenori AS OBJECT (  
 id int,  
 nume\_antrenor VARCHAR(20)  
);  
CREATE OR REPLACE TYPE antrenori\_tip AS TABLE OF antrenori ;   
   
CREATE OR REPLACE PACKAGE cerinta14 AS  
   
 FUNCTION AntrenoriPremii RETURN antrenori\_tip;  
 PROCEDURE GetInformatiiClub (numeclub CLUB.nume%TYPE);  
 PROCEDURE UpdateSalariu ;  
 FUNCTION Suma\_Sponsori (numeclub CLUB.nume%TYPE) RETURN NUMBER;  
 END cerinta14;  
 /

CREATE OR REPLACE PACKAGE BODY cerinta14 AS  
 FUNCTION AntrenoriPremii RETURN antrenori\_tip AS  
 antrenori\_returnare antrenori\_tip := antrenori\_tip();  
 TYPE tabel IS RECORD(  
 id\_antrenor\_club antrenor.id\_antrenor%TYPE,  
 nr\_competitii NUMBER  
 );  
 TYPE antrenor\_cometitie IS TABLE OF tabel;  
 antrenor\_competitii antrenor\_cometitie;  
 no\_data\_found1 EXCEPTION;  
 nume\_antrenor VARCHAR(20);  
 BEGIN   
 SELECT id\_antrenor, COUNT(id\_antrenor)  
 BULK COLLECT INTO antrenor\_competitii  
 FROM competitie  
 GROUP BY id\_antrenor;  
   
 IF antrenor\_competitii.COUNT = 0  
 THEN RAISE no\_data\_found1;  
 END IF;  
 FOR i IN 1..antrenor\_competitii.COUNT LOOP  
 IF antrenor\_competitii(i).nr\_competitii>0 THEN  
 SELECT nume  
 INTO nume\_antrenor  
 FROM antrenor  
 WHERE id\_antrenor = antrenor\_competitii(i).id\_antrenor\_club;  
 antrenori\_returnare.extend;  
 antrenori\_returnare(antrenori\_returnare.COUNT) := antrenori(antrenor\_competitii(i).id\_antrenor\_club,nume\_antrenor);  
 END IF;  
 END LOOP;  
 RETURN antrenori\_returnare;  
 EXCEPTION   
 WHEN no\_data\_found1 THEN   
 DBMS\_OUTPUT.put\_line('Nu exista antrenori');  
 END AntrenoriPremii;  
 PROCEDURE GetInformatiiClub (numeclub CLUB.nume%TYPE) AS  
 id int;  
 nr\_locatii NUMBER;  
 nr\_antrenori NUMBER;  
 nr\_echipe NUMBER;  
 TYPE tablou\_indexat IS TABLE OF echipa%ROWTYPE INDEX BY PLS\_INTEGER;  
 echipe tablou\_indexat;  
 BEGIN  
 SELECT id\_club  
 INTO id  
 FROM club  
 WHERE nume = numeclub;  
   
 SELECT COUNT ( c.id\_club)  
 INTO nr\_locatii  
 FROM club c  
 join club\_locatie cl ON (c.id\_club = cl.id\_club)  
 WHERE c.id\_club=id;  
 SELECT COUNT ( c.id\_club)  
 INTO nr\_antrenori  
 FROM club c  
 join antrenor\_club cl ON (c.id\_club = cl.id\_club)  
 WHERE c.id\_club=id;  
   
 SELECT e.id\_echipa, e.id\_departament, e.nume  
 BULK COLLECT INTO echipe  
 FROM club c, departament d, echipa e  
 WHERE e.id\_departament=d.id\_departament AND d.id\_club = c.id\_club AND c.nume=numeclub;  
   
 nr\_echipe := echipe.COUNT;  
   
 DBMS\_OUTPUT.put\_line( 'Nr de locatii '|| TO\_CHAR(nr\_locatii));  
 DBMS\_OUTPUT.put\_line( 'Nr de antrenori '|| TO\_CHAR(nr\_antrenori));  
 DBMS\_OUTPUT.put\_line( 'Nr de echipe '|| TO\_CHAR(nr\_echipe));   
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN   
 DBMS\_OUTPUT.put\_line('Nu exista aceast club');   
 WHEN TOO\_MANY\_ROWS THEN   
 DBMS\_OUTPUT.put\_line('Exista mai multe cluburi cu acelasi nume');   
END GetInformatiiClub;  
 PROCEDURE UpdateSalariu AS  
 TYPE tablou\_indexat IS TABLE OF NUMBER INDEX BY PLS\_INTEGER;  
 suma tablou\_indexat;  
 BEGIN  
 FOR i IN ( SELECT id FROM TABLE(cerinta14.AntrenoriPremii)) LOOP  
 SELECT salariu   
 BULK COLLECT INTO suma  
 FROM antrenor\_club  
 WHERE id\_antrenor=i.id;  
 FOR j IN 1..suma.COUNT LOOP  
 FOR k IN (SELECT \* FROM antrenor\_club) LOOP  
 UPDATE antrenor\_club  
 SET salariu = suma(j)+100  
 WHERE id\_antrenor = i.id;  
 END LOOP;  
 END LOOP;  
 END LOOP;   
EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN   
 DBMS\_OUTPUT.put\_line('Nu exista antrenori');   
END UpdateSalariu;  
 FUNCTION Suma\_Sponsori(numeclub CLUB.nume%TYPE) RETURN NUMBER AS  
 TYPE tablou\_indexat IS TABLE OF echipa%ROWTYPE INDEX BY PLS\_INTEGER;  
 TYPE tablou\_indexat\_numere IS TABLE OF NUMBER INDEX BY PLS\_INTEGER;  
 echipe tablou\_indexat;  
 suma\_totala NUMBER;  
 suma\_curenta tablou\_indexat\_numere;  
 exceptie EXCEPTION;  
 aux NUMBER;  
 BEGIN  
   
 SELECT e.id\_echipa, e.id\_departament, e.nume  
 BULK COLLECT INTO echipe  
 FROM club c, departament d, echipa e  
 WHERE e.id\_departament=d.id\_departament AND d.id\_club = c.id\_club AND c.nume=numeclub;  
   
 IF echipe.COUNT=0 THEN  
 RAISE NO\_DATA\_FOUND;  
 END IF;  
 suma\_totala:=0;  
 FOR i IN 1..echipe.COUNT LOOP  
 SELECT suma\_acordata  
 BULK COLLECT INTO suma\_curenta  
 FROM echipa\_sponsor  
 WHERE id\_echipa=echipe(i).id\_echipa;  
 FOR j IN 1..suma\_curenta.COUNT LOOP  
 aux:=suma\_totala;  
 suma\_totala := suma\_curenta(j)+aux;  
 END LOOP;  
 END LOOP;  
   
 IF suma\_totala=0 THEN  
 RAISE exceptie;  
 END IF;  
 RETURN suma\_totala;   
EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN   
 DBMS\_OUTPUT.put\_line('Nu exista antrenori');   
 WHEN exceptie THEN   
 DBMS\_OUTPUT.put\_line('Clubul nu beneficiaza de sponsori');   
END Suma\_Sponsori;  
END cerinta14;  
/

EXECUTE cerinta14.getinformatiiclub('CSM Targu Jiu')   
EXECUTE cerinta14.updatesalariu;   
   
DECLARE   
   
 a NUMBER;   
   
BEGIN   
   
 a:=cerinta14.suma\_sponsori('CSM Targu Jiu');   
   
 DBMS\_OUTPUT.put\_line('Suma totala a clubului ' || a);   
   
END;

Inainte

Dupa

