BAZA DE DATE A UNUI CLUB SPORTIV

COAJE FLORIAN-PETRUT

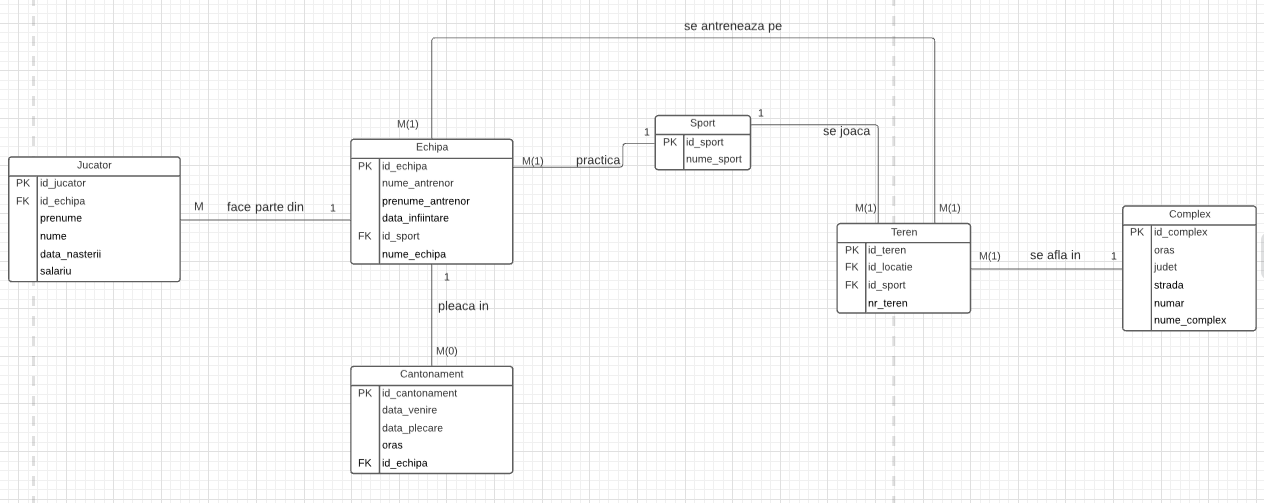
GRUPA 243

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

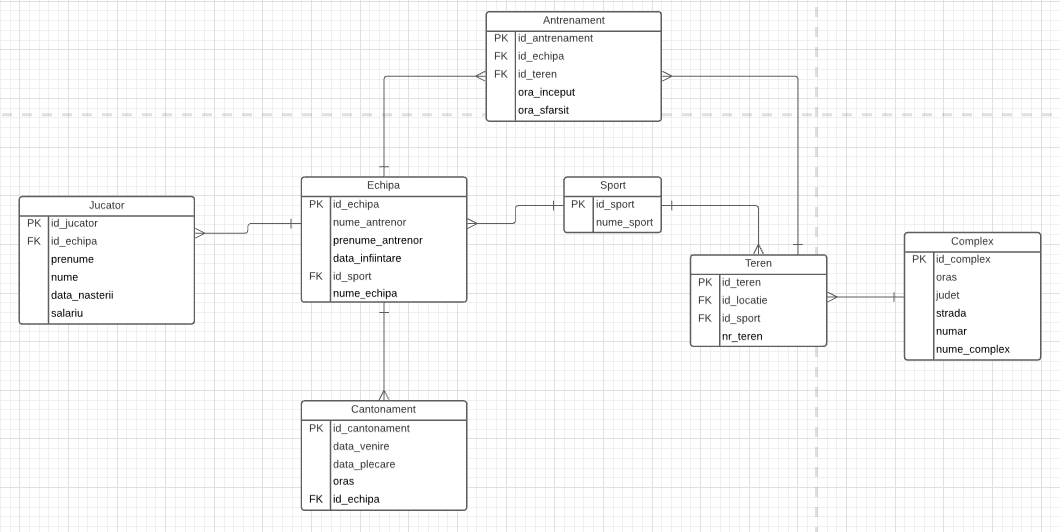
Aceasta baza de date are ca scop gestionarea activitatii unui club sportiv : planificarea antrenamentelor pe diferite terenuri din cadrul unor complexuri sportive, planificarea cantonamentelor echipelor. De asemenea, baza de date prezinta si informatii despre echipele clubului sportiv (data infiintarii, numele antrenorului) si despre jucatorii acestora (nume, data nasterii, salariu).

Tabela Complex retine informatii despre complexurile sportive ale orasului. Un complex are unul sau mai multe terenuri, fiecare teren fiind amenajat pentru un singur tip de sport.

1. Realizați diagrama entitate-relație (ERD).



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



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

create table sport

( id\_sport number(4) constraint pk\_id\_sport primary key,

nume\_sport varchar2(30) constraint nl\_nume\_sport not null

);

create table complex

(

id\_complex number(4) constraint pk\_id\_complex primary key,

oras varchar2(30),

judet varchar2(20),

strada varchar2(50),

numar number(4)

);

create table echipa

(

id\_echipa number(4) constraint pk\_id\_echipa primary key,

nume\_antrenor varchar2(30),

prenume\_antrenor varchar2(30),

data\_infiintare date default sysdate,

id\_sport number(4) constraint fk\_id\_sport references sport(id\_sport)

nume\_echipa varchar2(40);

);

create table teren

(

id\_teren number(4) constraint pk\_id\_teren primary key,

id\_complex number(4) constraint fk\_id\_complex references complex(id\_complex),

id\_sport number(4) constraint fk\_id\_sport\_teren references sport(id\_sport),

nr\_teren number(2)

);

create table jucator

(

id\_jucator number(4) constraint pk\_id\_jucator primary key,

id\_echipa number(4) constraint fk\_id\_echipa references echipa(id\_echipa),

prenume varchar2(20),

nume varchar2(20),

data\_nasterii date,

salariu number(9, 2)

);

create table antrenament

(

id\_antrenament number(4) constraint pk\_id\_antrenament primary key,

id\_echipa number(4) constraint fk\_id\_echipa\_antr references echipa(id\_echipa),

id\_teren number(4) constraint fk\_id\_teren references teren(id\_teren),

ora\_inceput timestamp(0),

ora\_sfarsit timestamp(0)

);

create table cantonament

(

id\_cantonament number(4) constraint pk\_id\_cantonament primary key,

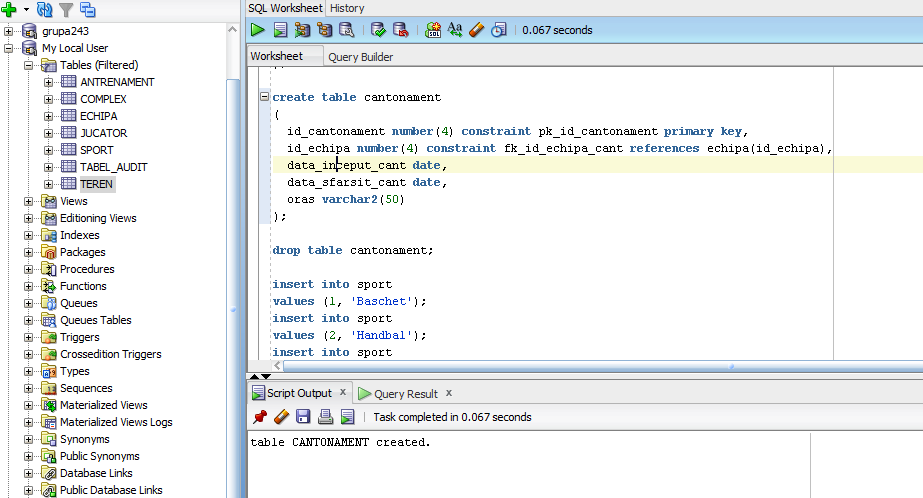
id\_echipa number(4) constraint fk\_id\_echipa\_cant references echipa(id\_echipa),

data\_inceput\_cant date,

data\_sfarsit\_cant date,

oras varchar2(50)

);



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

insert into sport

values (1, 'Baschet');

insert into sport

values (2, 'Handbal');

insert into sport

values (3, 'Volei');

insert into sport

values (4, 'Fotbal');



insert into complex

values (1, 'Craiova', 'Dolj', 'Bucovat', 25, 'Centrul Sportiv de Antrenament');

insert into complex

values (2, 'Craiova', 'Dolj', 'Ilie Balaci', 6, 'Centrul Sportiv Vointa');

insert into complex

values (3, 'Craiova', 'Dolj', 'Brestei', 20, 'Sala sporturilor Ion Constantinescu');

insert into complex

values (4, 'Craiova', 'Dolj', 'Decebal', 43, 'Centrul Sportiv Banie');

insert into teren

values (1, 1, 1, 1);

insert into teren

values (2, 1, 4, 2);

insert into teren

values (3, 2, 1, 1);

insert into teren

values (4, 2, 3, 2);

insert into teren

values (5, 3, 1, 1);

insert into teren

values (6, 3, 2, 2);

insert into teren

values (7, 3, 3, 3);

insert into teren

values (8, 4, 4, 1);

insert into echipa

values (1, 'Todorov', 'Aco', TO\_DATE('17/09/2007', 'DD/MM/YYYY'), 1, 'SCM U Craiova Baschet');

insert into echipa

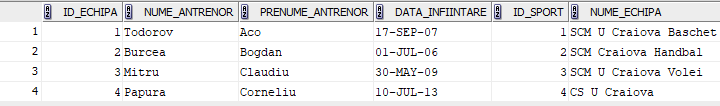
values (2, 'Burcea', 'Bogdan', TO\_DATE('01/07/2006', 'DD/MM/YYYY'), 2, 'SCM Craiova Handbal');

insert into echipa

values (3, 'Mitru', 'Claudiu', TO\_DATE('30/05/2009', 'DD/MM/YYYY'), 3, 'SCM U Craiova Volei');

insert into echipa

values (4, 'Papura', 'Corneliu', TO\_DATE('10/07/2013', 'DD/MM/YYYY'), 4, 'CS U Craiova');



insert into jucator

values (1, 1, 'Dragos' , 'Diculescu' , TO\_DATE('25/08/1999', 'DD/MM/YYYY'), 5000);

insert into jucator

values (2, 1, 'Monyea' , 'Pratt', TO\_DATE('11/07/1985', 'DD/MM/YYYY'), 10000);

insert into jucator

values (3, 1, 'Goran' , 'Filipovic', TO\_DATE('26/11/1996', 'DD/MM/YYYY'), 8000);

insert into jucator

values (4, 1, 'Mate' , 'Vucic' , TO\_DATE('06/04/1994', 'DD/MM/YYYY'), 7000);

insert into jucator

values (5, 1, 'Bogdan' , 'Tibarna', TO\_DATE('15/02/1990', 'DD/MM/YYYY'), 2500);

insert into jucator

values (6, 2, 'Cristina' , 'Zamfir' , TO\_DATE('29/09/1989', 'DD/MM/YYYY'), 4000);

insert into jucator

values (7, 2, 'Ana-Maria' , 'Ticu' , TO\_DATE('23/02/1992', 'DD/MM/YYYY'), 3000);

insert into jucator

values (8, 2, 'Ana-Maria' , 'Savu' , TO\_DATE('24/02/1990', 'DD/MM/YYYY'), 2000);

insert into jucator

values (9, 3, 'Igor' , 'Jovanovic' , TO\_DATE('21/06/1986', 'DD/MM/YYYY'), 2500);

insert into jucator

values (10, 3, 'Robert' , 'Irinel' , TO\_DATE('12/09/1993', 'DD/MM/YYYY'), 3000);

insert into jucator

values (11, 3, 'Razvan' , 'Olteanu' , TO\_DATE('07/02/1997', 'DD/MM/YYYY'), 5000);

insert into jucator

values (12, 3, 'Laurentiu' , 'Lica' , TO\_DATE('02/03/2000', 'DD/MM/YYYY'), 1000);

insert into jucator

values (13, 4, 'Andrei' , 'Ivan' , TO\_DATE('04/01/1996', 'DD/MM/YYYY'), 10000);

insert into jucator

values (14, 4, 'Ivan' , 'Mamut' , TO\_DATE('30/04/1997', 'DD/MM/YYYY'), 7000);

insert into jucator

values (15, 4, 'Alexandru', 'Cicaldau', TO\_DATE('08/07/1997', 'DD/MM/YYYY'), 20000);

insert into jucator

values (16, 4, 'Dan', 'Nistor', TO\_DATE('08/10/1988', 'DD/MM/YYYY'), 12000);

insert into jucator

values (17, 2, 'Cristina', 'Neagu', TO\_DATE('03/09/1998', 'DD/MM/YYYY'), 5000);

insert into jucator

values (18, 4, 'Elvir', 'Koljic', TO\_DATE('02/06/1996', 'DD/MM/YYYY'), 20000);

insert into cantonament

values (1, 1, TO\_DATE('25/01/2021', 'DD/MM/YYYY'), TO\_DATE('02/02/2021', 'DD/MM/YYYY'), 'Viena');

insert into cantonament

values (2, 2, TO\_DATE('24/01/2021', 'DD/MM/YYYY'), TO\_DATE('01/02/2021', 'DD/MM/YYYY'), 'Poiana Brasov');

insert into cantonament

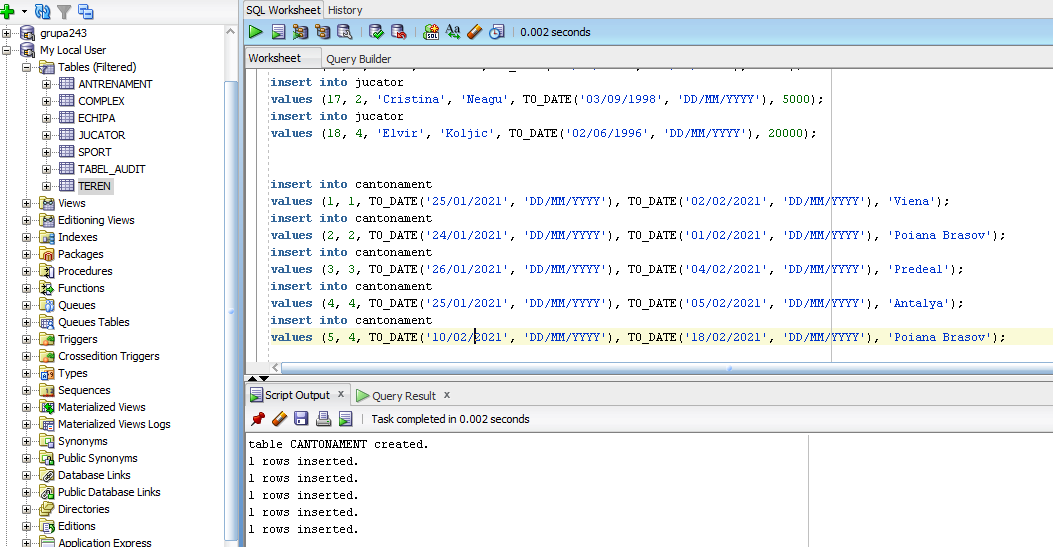
values (3, 3, TO\_DATE('26/01/2021', 'DD/MM/YYYY'), TO\_DATE('04/02/2021', 'DD/MM/YYYY'), 'Predeal');

insert into cantonament

values (4, 4, TO\_DATE('25/01/2021', 'DD/MM/YYYY'), TO\_DATE('05/02/2021', 'DD/MM/YYYY'), 'Antalya');

insert into cantonament

values (5, 4, TO\_DATE('10/02/2021', 'DD/MM/YYYY'), TO\_DATE('18/02/2021', 'DD/MM/YYYY'), 'Poiana Brasov');



insert into antrenament

values (1, 1, 1, TO\_TIMESTAMP('01/12/2020 09:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('01/12/2020 12:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (2, 1, 1, TO\_TIMESTAMP('05/12/2020 16:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('05/12/2020 18:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (3, 1, 1, TO\_TIMESTAMP('11/12/2020 08:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('11/12/2020 10:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (4, 1, 1, TO\_TIMESTAMP('08/12/2020 17:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('08/12/2020 20:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (5, 2, 6, TO\_TIMESTAMP('02/12/2020 09:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('02/12/2020 12:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (6, 2, 6, TO\_TIMESTAMP('06/12/2020 16:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('06/12/2020 18:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (7, 2, 6, TO\_TIMESTAMP('09/12/2020 12:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('09/12/2020 13:30', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (8, 2, 6, TO\_TIMESTAMP('11/12/2020 13:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('11/12/2020 15:30', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (9, 3, 4, TO\_TIMESTAMP('02/12/2020 08:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('02/12/2020 10:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (10, 3, 7, TO\_TIMESTAMP('04/12/2020 09:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('04/12/2020 11:30', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (11, 3, 4, TO\_TIMESTAMP('06/12/2020 17:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('06/12/2020 19:30', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (12, 3, 7, TO\_TIMESTAMP('08/12/2020 18:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('08/12/2020 20:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (13, 4, 2, TO\_TIMESTAMP('04/12/2020 08:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('04/12/2020 11:00', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

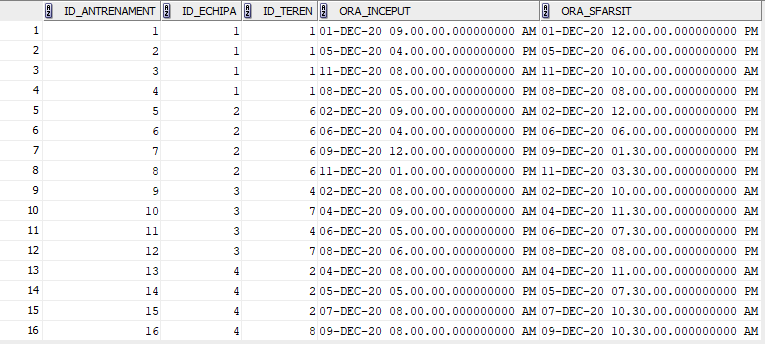
values (14, 4, 2, TO\_TIMESTAMP('05/12/2020 17:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('05/12/2020 19:30', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (15, 4, 2, TO\_TIMESTAMP('07/12/2020 08:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('07/12/2020 10:30', 'DD/MM/YYYY HH24:MI'));

insert into antrenament

values (16, 4, 8, TO\_TIMESTAMP('09/12/2020 08:00', 'DD/MM/YYYY HH24:MI'),TO\_TIMESTAMP('09/12/2020 10:30', 'DD/MM/YYYY HH24:MI'));



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

Definiti o procedura stocata care micsoreaza cu 5% salariile jucatorilor de volei cu salariul initial >=2500;

create or replace type nested\_table is table of number;

/

create or replace procedure micsorare\_salarii

is

cnt\_rez NUMBER := 0;

id\_volei NUMBER := 3;

ids nested\_table := nested\_table();

BEGIN

SELECT id\_jucator

bulk collect into ids

FROM jucator j, echipa e

WHERE j.salariu >= 2500

and j.id\_echipa = e.id\_echipa and id\_volei = e.id\_sport;

IF ids is null THEN

RAISE NO\_DATA\_FOUND;

END IF;

for i in ids.first..ids.last loop

UPDATE jucator

SET salariu = salariu \* 0.95

WHERE id\_jucator = ids(i);

end loop;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista jucatori de volei care sa corespunda cerintei');

end micsorare\_salarii;

/

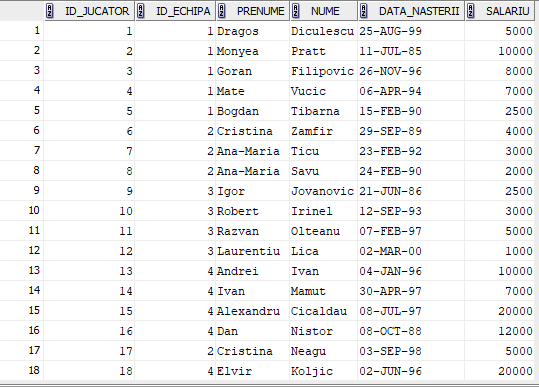
begin

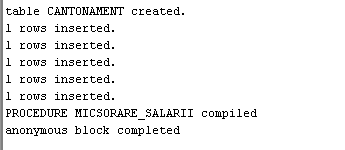
micsorare\_salarii;

end;

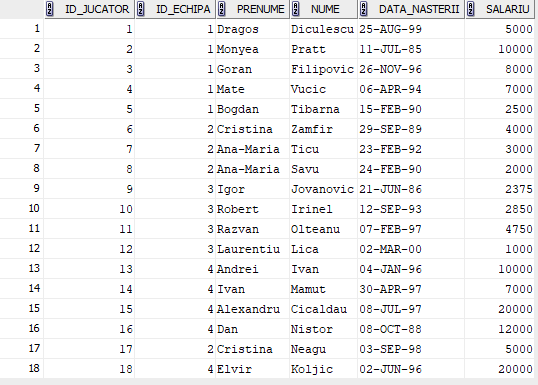
/

Datele din tabela jucatori inainte de rularea procedurii:





Datele din tabelul jucatori dupa rularea procedurii:



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

Definiti o procedura cu ajutorul careia sa se calculeze ce procent din bugetul lunar al clubului sportiv este cheltuit de fiecare echipa pentru salarii. Se vor afisa numele echipei, numele antrenorului si procentul.

create or replace procedure proc (v\_buget in number default 100000)

is

v\_suma number(7);

v\_nume\_echipa echipa.nume\_echipa%type;

v\_nume\_antrenor echipa.nume\_antrenor%type;

cursor crs is

select nume\_echipa , nume\_antrenor, sum(salariu) suma

from jucator j, echipa e

where j.id\_echipa = e.id\_echipa

group by e.nume\_echipa, e.nume\_antrenor

order by suma desc;

begin

open crs;

loop

fetch crs into v\_nume\_echipa, v\_nume\_antrenor, v\_suma;

exit when crs%notfound;

DBMS\_OUTPUT.PUT\_LINE('Echipa '|| v\_nume\_echipa || ' antrenata de ' || v\_nume\_antrenor || ' cheltuie ' || round(v\_suma/v\_buget \*100, 2) || '% din bugetul total al clubului sportiv');

end loop;

close crs;

end proc;

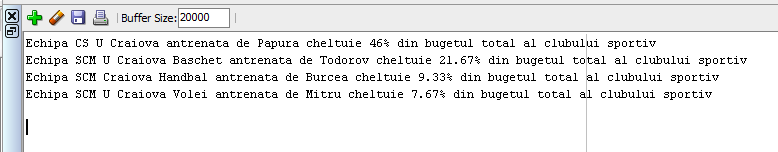
/

begin

proc(150000);

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 afiseze numele complexului cu cele mai multe antrenamente dintr-o zi data ca parametru.

create or replace function func (v\_data date)

return varchar2 is

nume\_com complex.nume\_complex%type;

v\_nr\_antr number;

nume\_com2 complex.nume\_complex%type;

v\_nr\_antr2 number;

exceptie exception;

cursor crs is

select nume\_complex, count(an.id\_teren) nr\_antr

from teren t, complex com, antrenament an

where an.id\_teren = t.id\_teren and com.id\_complex = t.id\_complex

and to\_char(cast(ora\_inceput as date)) = to\_char(v\_data)

group by nume\_complex

order by nr\_antr desc;

begin

open crs;

fetch crs into nume\_com, v\_nr\_antr;

if crs%notfound then

raise NO\_DATA\_FOUND;

end if;

fetch crs into nume\_com2, v\_nr\_antr2;

if crs%found then

if v\_nr\_antr = v\_nr\_antr2 then

raise exceptie;

end if;

end if;

close crs;

return nume\_com;

exception

when NO\_DATA\_FOUND then

RAISE\_APPLICATION\_ERROR(-20005, 'In aceasta zi nu s-a desfasurat niciun antrenament');

WHEN exceptie THEN

RAISE\_APPLICATION\_ERROR(-20006, 'Exista mai multe complexe in care s-a desfasurat acelasi numar de antrenamente');

end func;

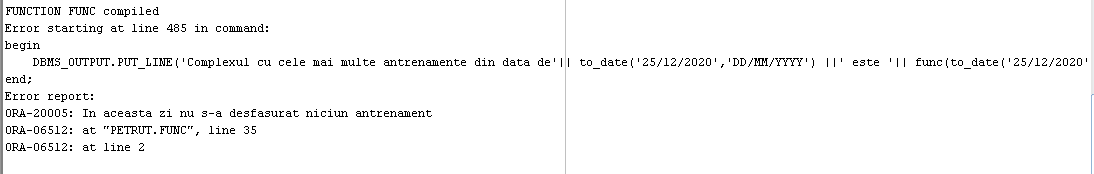
/

begin

DBMS\_OUTPUT.PUT\_LINE('Complexul cu cele mai multe antrenamente din data de'|| to\_date('25/12/2020','DD/MM/YYYY') ||' este '|| func(to\_date('25/12/2020','DD/MM/YYYY')));

end;

/

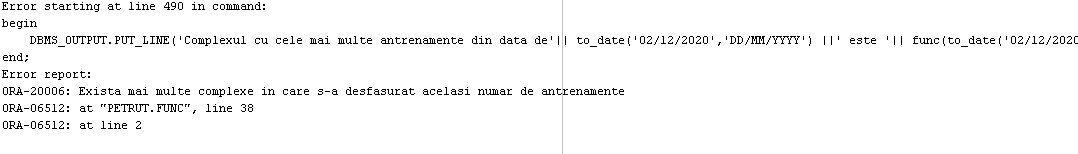


begin

DBMS\_OUTPUT.PUT\_LINE('Complexul cu cele mai multe antrenamente din data de'|| to\_date('02/12/2020','DD/MM/YYYY') ||' este '|| func(to\_date('02/12/2020','DD/MM/YYYY')));

end;

/

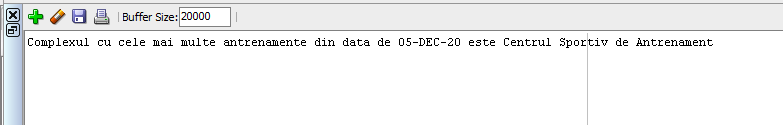


begin

DBMS\_OUTPUT.PUT\_LINE('Complexul cu cele mai multe antrenamente din data de'|| to\_date('05/12/2020','DD/MM/YYYY') ||' este '|| func(to\_date('05/12/2020','DD/MM/YYYY')));

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.

Sa se afiseze jucatorul cu cel mai mare salariu ce se antreneaza in perioada 08.12.2020-11.12.2020, intr-un complex al carui id e dat ca parametru al procedurii.

create or replace procedure proc\_ex9 (v\_nr\_complex in complex.id\_complex%type)

is

v\_max\_sal jucator.salariu%type;

v\_id\_echipa echipa.id\_echipa%type;

v\_nume\_juc jucator.nume%type;

v\_nume\_echipa echipa.nume\_echipa%type;

v\_max\_sal\_fin number := 0;

v\_id\_echipa\_fin number := 0;

contor number(4) :=0;

exceptie exception;

cursor crs is

select max(salariu),j.id\_echipa

from teren t, complex com, antrenament an, echipa e, jucator j

where an.id\_teren = t.id\_teren and com.id\_complex = t.id\_complex

and an.id\_echipa = e.id\_echipa and j.id\_echipa = e.id\_echipa

and extract( day from ora\_inceput) <=11 and extract( day from ora\_inceput) >=8

and extract ( month from ora\_inceput) = 12

and extract( year from ora\_inceput) = 2020

and com.id\_complex = v\_nr\_complex

group by j.id\_echipa;

begin

open crs;

loop

fetch crs into v\_max\_sal, v\_id\_echipa;

exit when crs%notfound;

contor := contor + 1;

if v\_max\_sal > v\_max\_sal\_fin then

v\_max\_sal\_fin := v\_max\_sal;

v\_id\_echipa\_fin := v\_id\_echipa;

else if v\_max\_sal = v\_max\_sal\_fin then

raise exceptie;

end if;

end if;

end loop;

close crs;

if contor = 0 then

raise NO\_DATA\_FOUND;

end if;

select nume , nume\_echipa

into v\_nume\_juc, v\_nume\_echipa

from echipa e, jucator j

where salariu = v\_max\_sal\_fin and j.id\_echipa = v\_id\_echipa\_fin

and e.id\_echipa = v\_id\_echipa;

DBMS\_OUTPUT.PUT\_LINE( v\_nume\_juc || ' este jucatorul cel mai bine platit, de la echipa '

|| v\_nume\_echipa );

exception

when NO\_DATA\_FOUND then

RAISE\_APPLICATION\_ERROR(-20010, 'In perioada 08.12.2020-11.12.2020 nicio echipa nu a desfasurat antrenamente la complexul cu id-ul ' || v\_nr\_complex);

when exceptie then

RAISE\_APPLICATION\_ERROR(-20011, 'Exista mai multi jucatori cu salariul maxim, in echipe diferite' );

when TOO\_MANY\_ROWS then

RAISE\_APPLICATION\_ERROR(-20012, 'Exista mai multi jucatori cu salariul maxim, in cadrul aceleiasi echipe' );

end proc\_ex9;

/

declare

v\_id\_complex number(4) := &p\_id\_complex;

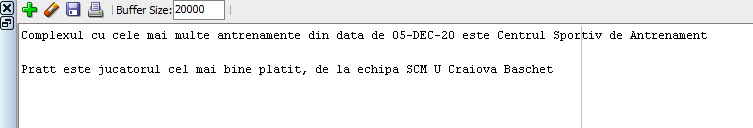
begin

proc\_ex9(v\_id\_complex);

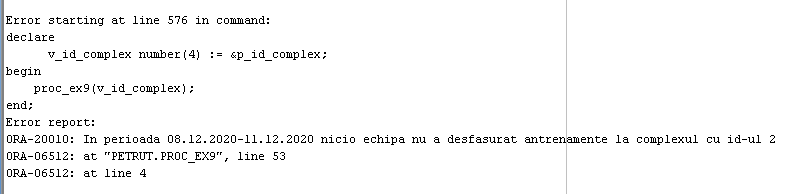
end;

/

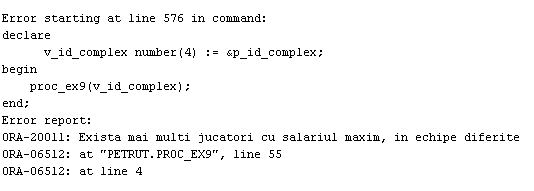
Pentru v\_id\_complex = 1 :



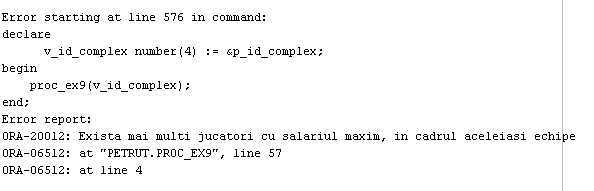
Pentru v\_id\_complex = 2 :



Pentru v\_id\_complex = 3 :



Pentru v\_id\_complex = 4 :



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

Definiti un declansator care sa nu permita planificarea unui antrenament in vacanta de iarna (1-22 ianuarie) . O planificare se face cu exact o saptamana inainte de ziua dorita.

create or replace trigger trig\_ex10

before insert or update on antrenament

begin

if(sysdate+7> to\_date('2021/01/01', 'yyyy/mm/dd')) or (sysdate+7 < to\_date('2021/01/22', 'yyyy/mm/dd')) then

raise\_application\_error(-20002, 'Nu se pot planifica antrenamente in vacanta');

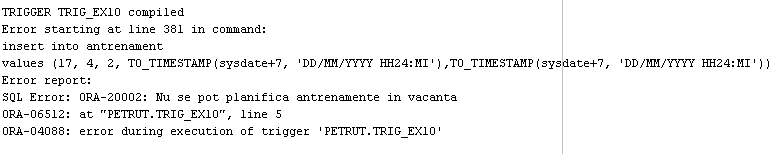
end if;

end;

/

insert into antrenament

values(16, 4, 2, TO\_TIMESTAMP( sysdate+7 , 'DD/MM/YYYY HH24:MI'), TO\_TIMESTAMP( sysdate + 7 , 'DD/MM/YYYY HH24:MI'));



drop trigger trig\_ex10;

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

Definiti un declansator prin care sa nu se permita adaugare unui jucator mai batran decat cel mai varstnic component al echipei respective.

CREATE OR REPLACE PACKAGE pachet

AS

TYPE tip\_rec IS RECORD

( echipa number(2),

d\_nastere jucator.data\_nasterii%type

);

TYPE tip\_ind IS TABLE OF tip\_rec INDEX BY PLS\_INTEGER;

t tip\_ind;

END;

/

CREATE OR REPLACE TRIGGER trig\_comanda

BEFORE INSERT ON jucator

BEGIN

select j.id\_echipa, min(j.data\_nasterii)

bulk collect into pachet.t

from jucator j, echipa e

where j.id\_echipa = e.id\_echipa

group by e.id\_echipa;

end;

/

CREATE OR REPLACE TRIGGER trig\_linie

BEFORE INSERT ON jucator

FOR EACH ROW

BEGIN

FOR i in 1..pachet.t.last LOOP

IF pachet.t(i).echipa = :NEW.id\_echipa

AND pachet.t(i).d\_nastere >= :NEW.data\_nasterii THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Jucatorul este prea batran');

END IF;

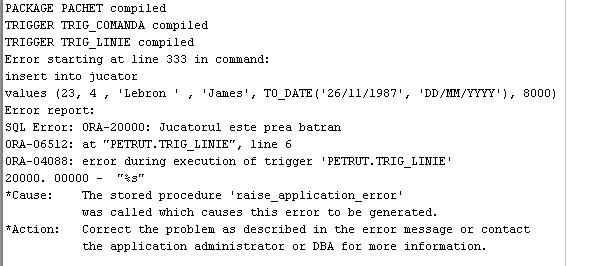
END LOOP;

END;

/

insert into jucator

values (23, 4 , 'Lebron ' , 'James', TO\_DATE('26/11/1987', 'DD/MM/YYYY'), 8000);



drop trigger trig\_linie;

drop trigger trig\_comanda;

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

Creați tabelul tabel\_audit cu următoarele câmpuri:  
- utilizator (numele utilizatorului);  
- nume\_bd (numele bazei de date);  
- eveniment (evenimentul sistem);  
- nume\_obiect (numele obiectului);  
- data (data producerii evenimentului).  
 Definiți un declanșator care să introducă date în acest tabel după ce utilizatorul a folosit o comandă LDD (declanșator sistem - la nivel de schemă).

CREATE TABLE tabel\_audit

(utilizator VARCHAR2(20),

nume\_bd VARCHAR2(50),

eveniment VARCHAR2(20),

nume\_obiect VARCHAR2(30),

data DATE);

CREATE OR REPLACE TRIGGER trig3

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO tabel\_audit

VALUES (SYS.LOGIN\_USER, SYS.DATABASE\_NAME, SYS.SYSEVENT,

SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);

END;

/

CREATE INDEX ind ON jucator(nume);

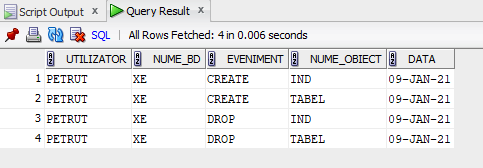
CREATE TABLE tabel (col\_1 number(2));

insert into tabel values(10);

DROP INDEX ind;

drop table tabel;

SELECT \* FROM tabel\_audit;



DROP TRIGGER trig3;

drop table tabel\_audit;