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

Trebuie sa cream o baza de date pentru gestiune unei echipe de fotbal. Trebuie memorate date despre sponsorii echipei, contractele pe care le au sponsorii cu echipa noastra , meciurile pe care le disputa, atat in campionatul intern cat si in celelalte ligi (champions league, europa league, cupa , cupa ligii, etc.)

De asemenea, se memoreaza date despre fiecare jucator si fiecare antrenament pe care acestia il realizeaza. Antrenamentele sunt coordonate de membri ai staffului tehnic. Stafful tehnic este constituit din mai multe posturi: antrenor principal ,antrenor secund, manager, preparator fizic, medic, antrenor de portari, maseuri,etc.

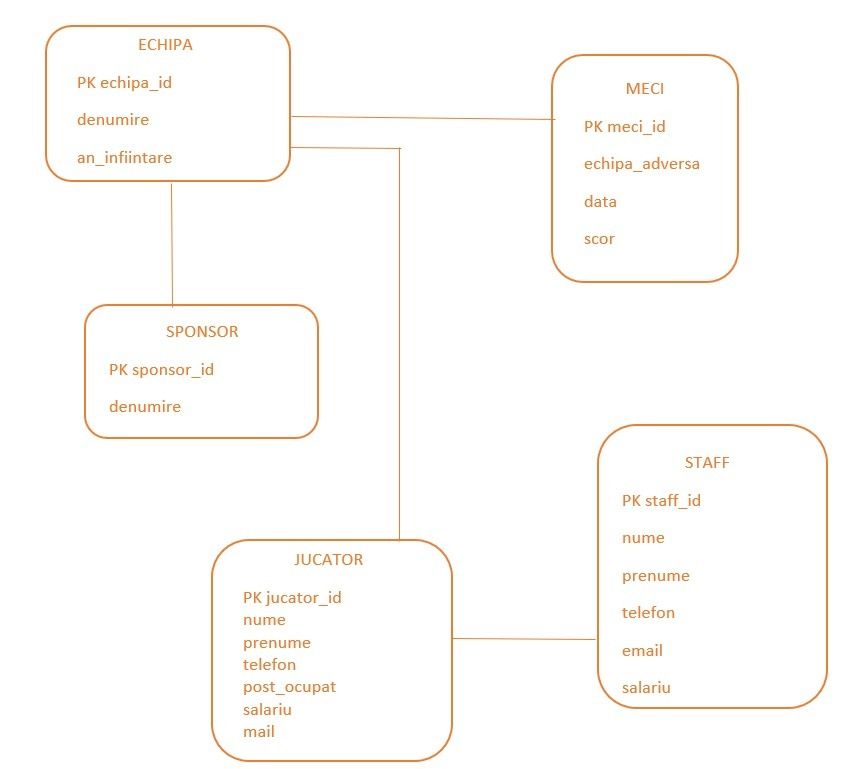
Fiind vorba despre o echipa de fotbal, adică un organism foarte mare şi complex, este nevoie de o bază de date uşor de utilizat pentru a păstra informaţii despre:

* Stafful tehnic
* Jucatorii detinuti de echipa.

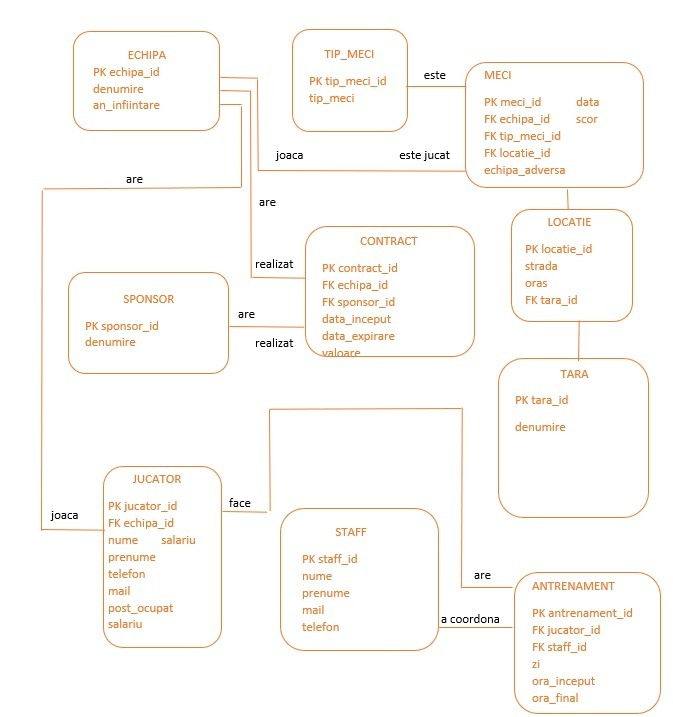
Baza de date trebuie sa fie îndeajuns de flexibilă pentru a permite adăugarea de:

* Noi jucatori
* Noi angajati in stafful tehnic
* Meciuri ce urmeaza a fi disputate
* Locatia si Tara in care urmeaza sa fie disputat meciul
* Antrenamente

1. Realizați diagrama entitate-relație(ERD).



1. Pornind de la diagrama entitate-relație realizați diagrama conceptualăa modelului propus, integrând 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 ECHIPA (

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

denumire varchar2(30) constraint null\_nume not null,

an\_infiintare date default sysdate constraint null\_an not null

);

create table TIP\_MECI(

tip\_meci\_id number(4) constraint pk\_tip\_meci\_id primary key,

tip\_meci varchar2(30) constraint null\_tip\_meci not null

);

create table TARA(

tara\_id NUMBER(4) constraint pk\_tara primary key,

denumire varchar2(20) constraint null\_denumire not null

);

create table LOCATIE(

locatie\_id number(4) constraint pk\_locatie\_id primary key,

strada varchar2(40),

oras varchar2(20),

tara\_id number(4),

constraint fk\_tara\_id foreign key(tara\_id) references TARA(tara\_id)

);

create table MECI(

meci\_id number(4) constraint pk\_meci\_id primary key,

echipa\_adversa varchar2(30) constraint null\_echipa\_adversa not null,

data\_joc date default sysdate constraint null\_data\_joc not null,

scor varchar2(10) constraint null\_scor not null,

echipa\_id number(4),

tip\_meci\_id number(4),

locatie\_id number(4),

constraint fk\_echipa\_id foreign key(echipa\_id) references ECHIPA(echipa\_id),

constraint fk\_tip\_meci\_id foreign key(tip\_meci\_id) references TIP\_MECI(tip\_meci\_id),

constraint fk\_locatie\_id foreign key(locatie\_id) references LOCATIE(locatie\_id)

);

create table JUCATOR(

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

echipa\_id number(4),

nume varchar2(30) constraint null\_num not null,

prenume varchar2(30) constraint null\_prenume not null,

telefon varchar2(11) constraint null\_telefon not null,

varsta number(2) constraint null\_varsta not null,

post\_ocupat varchar2(20),

salariu number(8,2) constraint null\_salariu not null,

mail varchar2(20),

constraint unq\_telefon unique(telefon),

constraint unq\_mail unique(mail),

constraint fk\_echipa\_id\_jucator foreign key(echipa\_id) references ECHIPA(echipa\_id)

);

create table STAFF(

staff\_id number(4) constraint pk\_staff\_id primary key,

nume varchar2(30) constraint null\_nume\_staff not null,

prenume varchar2(30) constraint null\_prenume\_staff not null,

varsat number(2) constraint null\_varsta\_staff not null,

telefon varchar2(11) constraint null\_telefon\_staff not null,

email varchar2(30),

salariu number(8,2),

constraint unq\_telefon\_staff unique(telefon),

constraint unq\_email unique(email)

);

create TABLE ANTRENAMENT(

antrenement\_id number(4) constraint pk\_antrenemant\_id primary key,

jucator\_id number(4),

staff\_id number(4),

zi varchar2(30) constraint null\_zi not null,

ora\_inceput varchar2(15) constraint null\_ora\_inceput not null,

ora\_sfarsit varchar2(15),

data\_antrenament date default sysdate constraint null\_data\_antrenament not null,

constraint fk\_jucator\_id foreign key(jucator\_id) references JUCATOR(jucator\_id),

constraint fk\_staff\_id foreign key(staff\_id) references STAFF(staff\_id)

);

create table SPONSOR (

sponsor\_id number(4) constraint pk\_sponsor\_id primary key,

denumire varchar2(30)

);

create table CONTRACT (

contract\_id number(4) constraint pk\_contract\_id primary key,

echipa\_id number(4),

sponsor\_id number(4),

nume\_contract varchar2(30) constraint null\_nume\_contract not null,

data\_inceput date default sysdate constraint null\_data\_inceput\_contract not null,

data\_expirare date default sysdate constraint null\_data\_expirare\_contract not null,

valoare number(8,2),

constraint fk\_echipa\_id\_contract foreign key(echipa\_id) references ECHIPA(echipa\_id),

constraint fk\_sponsor\_id\_contract foreign key(sponsor\_id) references SPONSOR(sponsor\_id)

);

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

ECHIPA :

insert into ECHIPA

values (1 ,'Manchester United', to\_date('01-05-1878', 'dd-mm-yyyy'));

insert into echipa

values (2, 'Manchester United U23',to\_date('23-07-1975', 'dd-mm-yyyy'));

insert into echipa

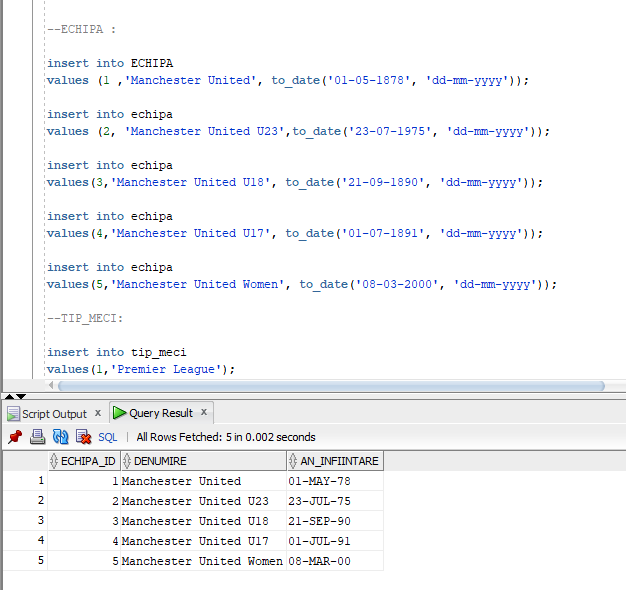
values(3,'Manchester United U18', to\_date('21-09-1890', 'dd-mm-yyyy'));

insert into echipa

values(4,'Manchester United U17', to\_date('01-07-1891', 'dd-mm-yyyy'));

insert into echipa

values(5,'Manchester United Women', to\_date('08-03-2000', 'dd-mm-yyyy'));



--TIP\_MECI:

insert into tip\_meci

values(1,'Premier League');

delete from tip\_meci where tip\_meci\_id = 1;

insert into tip\_meci

values(2,'FA CUP');

insert into tip\_meci

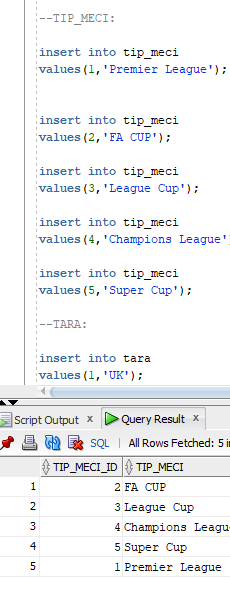
values(3,'League Cup');

insert into tip\_meci

values(4,'Champions League');

insert into tip\_meci

values(5,'Super Cup');



--TARA:

insert into tara

values(1,'UK');

insert into tara

values(2,'Spania');

insert into tara

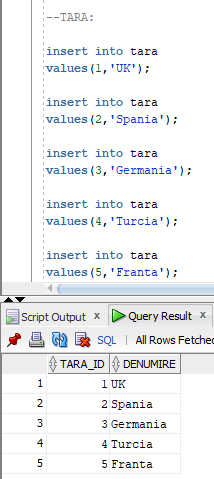
values(3,'Germania');

insert into tara

values(4,'Turcia');

insert into tara

values(5,'Franta');



--locatie:

insert into locatie

values(1,'Old Trafford','Manchester',1);

insert into locatie

values(2,'Anfield','Liverpool',1);

insert into locatie

values(3,'Emirates','Londra',1);

insert into locatie

values(4,'Wembley','Londra',1);

insert into locatie

values(5,'Camp Nou','Barcelona',2);

insert into locatie

values(6,'Allianz','Dortmund',3);

insert into locatie

values(7,'Basakesir','Istanbul',4);

insert into locatie

values(8,'Parc des Princes','Paris',5);



--meci:

insert into meci

values(1, 'Spurs', to\_date('30-07-2020', 'dd-mm-yyyy'), '5-1', 1, 1, 1);

insert into meci

values(2, 'Liverpool', to\_date('09-08-2020', 'dd-mm-yyyy'), '2-2', 1, 1, 2);

insert into meci

values(3, 'Manchester City', to\_date('16-08-2020', 'dd-mm-yyyy'), '0-3', 1, 2, 1);

insert into meci

values(4, 'New Castle', to\_date('20-08-2020', 'dd-mm-yyyy'), '1-0', 1, 3, 4);

insert into meci

values(5, 'Barcelona', to\_date('15-09-2020', 'dd-mm-yyyy'), '2-1', 1, 4, 5);

insert into meci

values(6, 'Bayern Munchen', to\_date('29-09-2020', 'dd-mm-yyyy'), '0-2', 1, 4, 6);

insert into meci

values(7, 'BB Istambul', to\_date('13-10-2020', 'dd-mm-yyyy'), '5-0', 1, 4, 7);

insert into meci

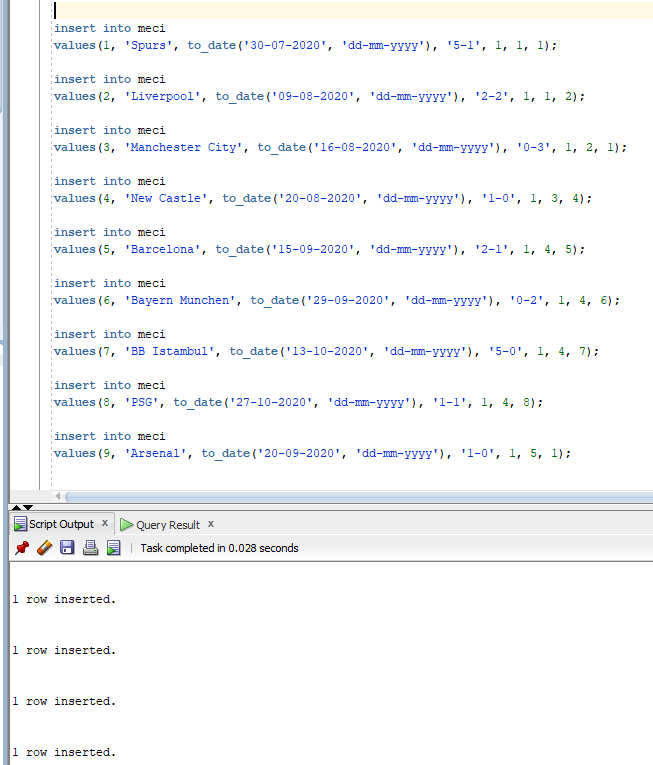
values(8, 'PSG', to\_date('27-10-2020', 'dd-mm-yyyy'), '1-1', 1, 4, 8);

insert into meci

values(9, 'Arsenal', to\_date('20-09-2020', 'dd-mm-yyyy'), '1-0', 1, 5, 1);

insert into meci

values(10, 'West Ham United', to\_date('27-09-2020', 'dd-mm-yyyy'), '2-1', 1, 2, 1);



--jucator:

insert into jucator

values(10,1,'de Gea','David','077394109',30,'GK',15000,'davd\_gea@yahoo.com');

insert into jucator

values(11,1,'Henderson','Dean','0743983487',24,'GK',6000,'dean\_hend@yahoo.com');

insert into jucator

values(12,1,'Maguire','Harry','0773942009',28,'fundas',12000,'harry5@yahoo.com');

insert into jucator

values(13,1,'Lindelof','Victor','077394100',26,'fundas',7500,'victorlind@yahoo.com');

insert into jucator

values(14,1,'Pogba','Paul','0773941234',28,'mijlocas',16000,'paullabile@yahoo.com');

insert into jucator

values(15,1,'Fernandes','Bruno','077394209',24,'mijlocas',3000,'brunof@yahoo.com');

insert into jucator

values(16,1,'Rashford','Marcus','072394109',24,'atacant',14000,'marcusrsh@yahoo.com');

insert into jucator

values(17,1,'Cavani','Edison','073394109',35,'atacant',14000,'cavani09@yahoo.com');

insert into jucator

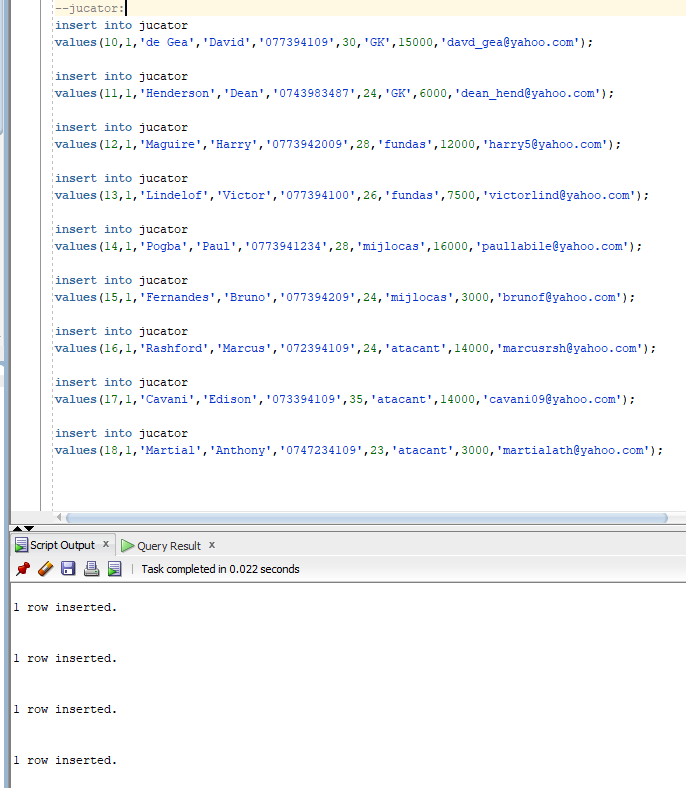
values(18,1,'Martial','Anthony','0747234109',23,'atacant',3000,'martialath@yahoo.com');

insert into jucator

values(20,1,'Greenwood','Mason','072456330',33,null,9000,'greenmason@yahoo.com');

insert into jucator

values(22,1,'McTominay','Scott','0724783343',22,null,9000,null);



--staff:

insert into staff

values(1,'Neubert','Thomas', 53, '0778453692',null,9000);

insert into staff

values(2,'Davis','Michael', 47, '0778453609','davidmih@yahoo.com',7000);

insert into staff

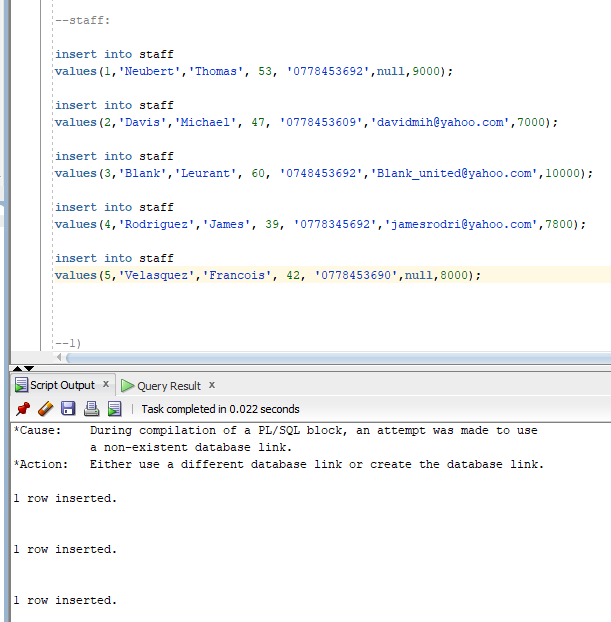
values(3,'Blank','Leurant', 60, '0748453692','Blank\_united@yahoo.com',10000);

insert into staff

values(4,'Rodriguez','James', 39, '0778345692','jamesrodri@yahoo.com',7800);

insert into staff

values(5,’Velasquez’, ’Francois’, 42, '0778453690',null,8000);



--antrenament

insert into antrenament

values(1,10,1,'antrenment fizic','09:00','11:00',to\_date('29-09-2020', 'dd-mm-yyyy'));

insert into antrenament

values(2,11,1,'antrenment fizic','09:00','11:00',to\_date('29-09-2020', 'dd-mm-yyyy'));

insert into antrenament

values(3,12,1,'antrenment fizic','09:00','11:00',to\_date('29-09-2020', 'dd-mm-yyyy'));

insert into antrenament

values(4,13,2,'antrenament rezistenta','14:00','15:30',to\_date('29-09-2020', 'dd-mm-yyyy'));

insert into antrenament

values(5,14,2,'antrenment fizic','14:00','15:30',to\_date('01-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(6,15,3,'fizicul cu mingea','18:00','20:00',to\_date('02-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(7,16,3,'fizicul cu mingea','18:00','20:00',to\_date('03-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(8,17,4,'coordonare si finalizare','18:00','20:00',to\_date('04-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(9,18,4,'coordonare si finalizare','18:00','20:00',to\_date('04-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(10,10,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(11,11,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(12,12,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(13,13,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(14,14,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(15,15,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

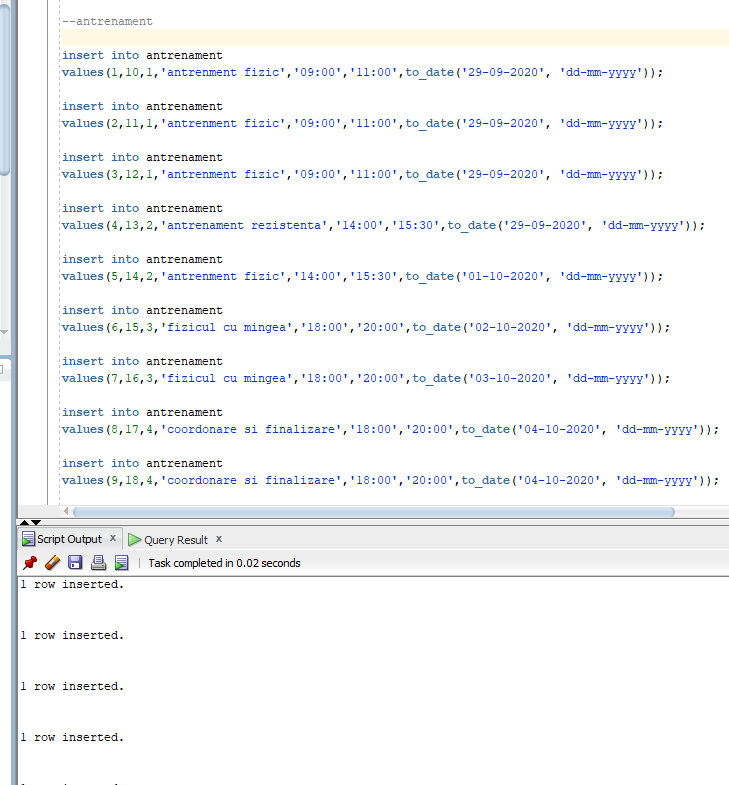
values(16,16,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(17,17,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));

insert into antrenament

values(18,18,5,'recuperare','15:00','16:00',to\_date('05-10-2020', 'dd-mm-yyyy'));



--sponsor:

insert into sponsor

values (1,'Chevrolet');

insert into sponsor

values (2,'Adidas');

insert into sponsor

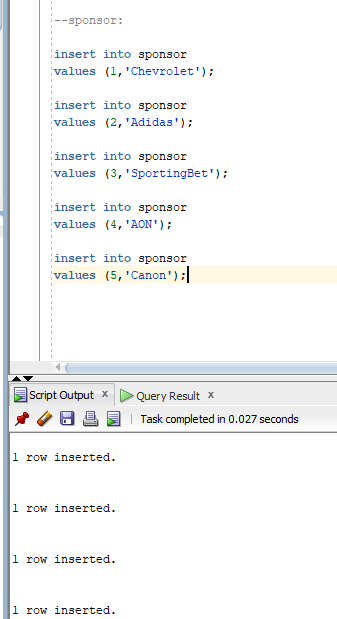
values (3,'SportingBet');

insert into sponsor

values (4,'AON');

insert into sponsor

values (5,'Canon');



--contract:

insert into contract

values(1,1,1,'Principal Partner of ManUnited',to\_date('14-05-2014', 'dd-mm-yyyy'),to\_date('14-05-2025', 'dd-mm-yyyy'),500000);

insert into contract

values(2,1,2,'Official Carrier of ManUnited',to\_date('14-05-2019', 'dd-mm-yyyy'),to\_date('14-05-2022', 'dd-mm-yyyy'),300000);

insert into contract

values(4,1,2,'Official Kit Supplier',to\_date('09-04-2015', 'dd-mm-yyyy'),to\_date('18-05-2024', 'dd-mm-yyyy'),500000);

insert into contract

values(5,1,3,'Official Tyre Partner',to\_date('23-09-2018', 'dd-mm-yyyy'),to\_date('20-10-2021', 'dd-mm-yyyy'),650000);

insert into contract

values(6,1,4,'Principal Partner',to\_date('10-12-2013', 'dd-mm-yyyy'),to\_date('04-11-2022', 'dd-mm-yyyy'),650000);

insert into contract

values(7,1,5,'Official MedicalSystem Partner',to\_date('02-01-2017', 'dd-mm-yyyy'),to\_date('24-12-2021', 'dd-mm-yyyy'),240000);

insert into contract

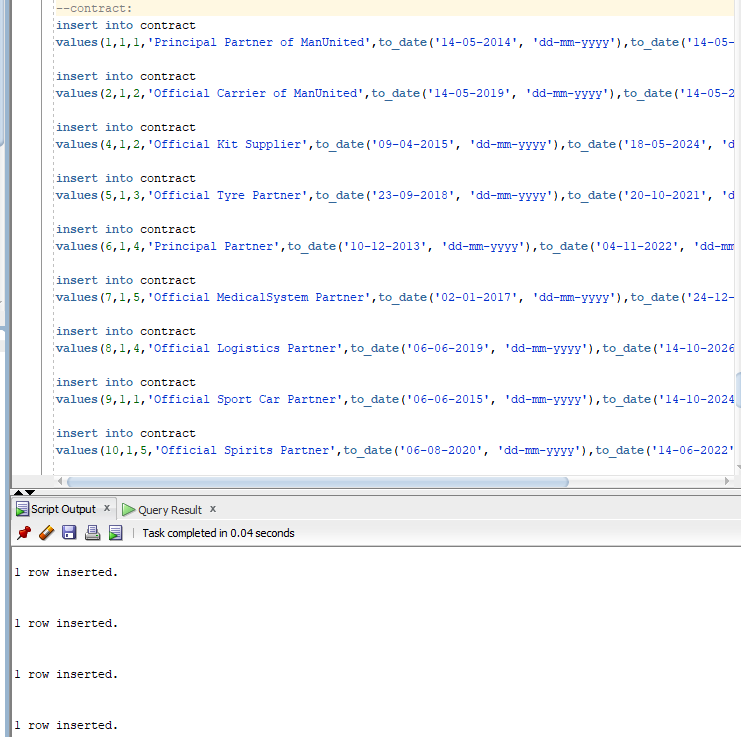
values(8,1,4,'Official Logistics Partner',to\_date('06-06-2019', 'dd-mm-yyyy'),to\_date('14-10-2026', 'dd-mm-yyyy'),500000);

insert into contract

values(9,1,1,'Official Sport Car Partner',to\_date('06-06-2015', 'dd-mm-yyyy'),to\_date('14-10-2024', 'dd-mm-yyyy'),650000);

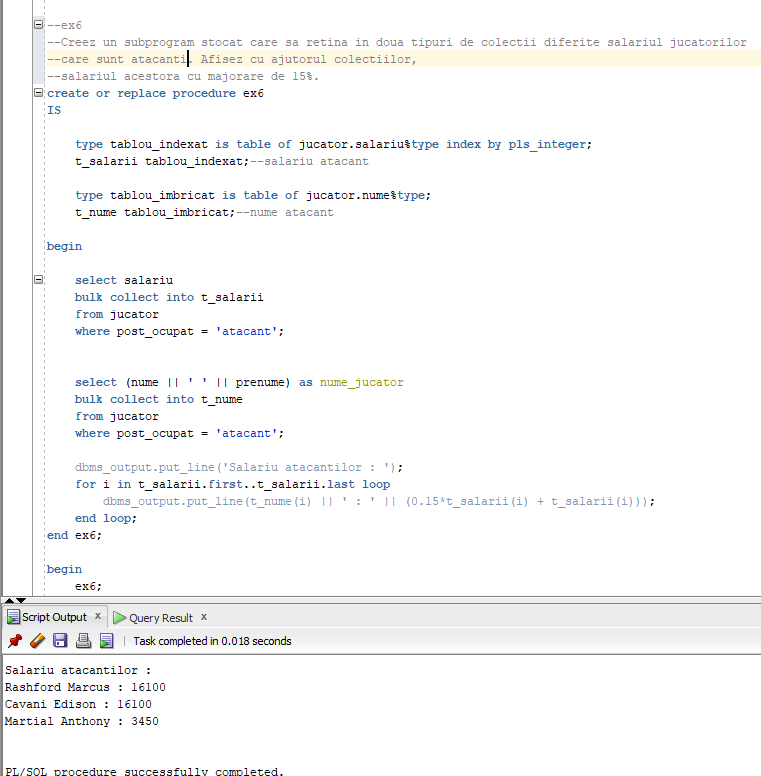
insert into contract

values(10,1,5,'Official Spirits Partner',to\_date('06-08-2020', 'dd-mm-yyyy'),to\_date('14-06-2022', 'dd-mm-yyyy'),150000);



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

Creez un subprogram stocat care sa retina in doua tipuri de colectii diferite salariul jucatorilor care sunt atacanti si numele lor. Afisez cu ajutorul colectiilor, numele si salariul atacantilor cu majorare de 15%.



create or replace procedure ex6

IS

type tablou\_indexat is table of jucator.salariu%type index by pls\_integer;

t\_salarii tablou\_indexat;--salariu atacant

type tablou\_imbricat is table of jucator.nume%type;

t\_nume tablou\_imbricat;--nume atacant

begin

select salariu

bulk collect into t\_salarii

from jucator

where post\_ocupat = 'atacant';

select (nume || ' ' || prenume) as nume\_jucator

bulk collect into t\_nume

from jucator

where post\_ocupat = 'atacant';

dbms\_output.put\_line('Salariu atacantilor : ');

for i in t\_salarii.first..t\_salarii.last loop

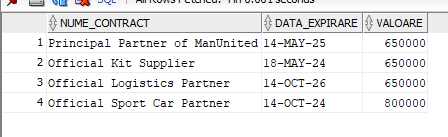
dbms\_output.put\_line(t\_nume(i) || ' : ' || (0.15\*t\_salarii(i) + t\_salarii(i)));

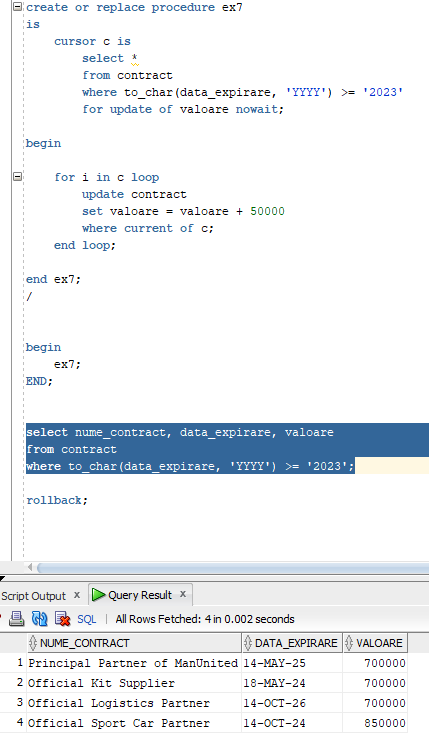
end loop;

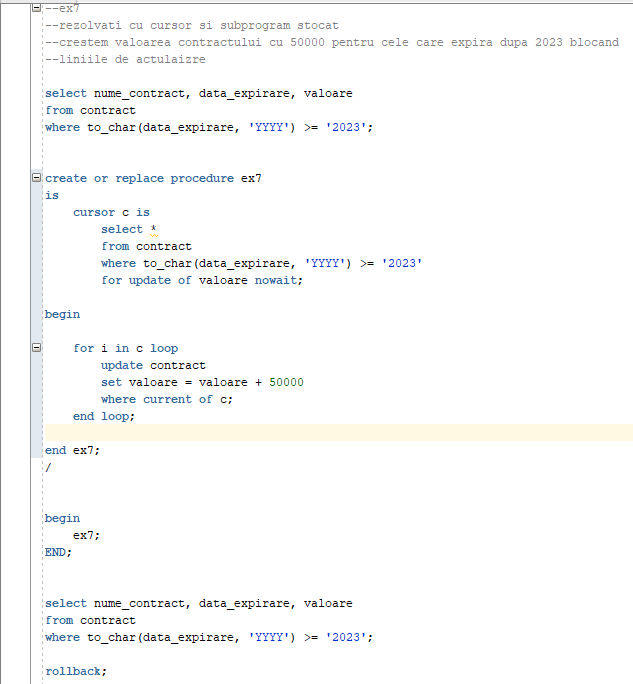
end ex6;

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

Creem un subprogram stocat care sa contina un cursor ce realizeaza cresterea valoarii contractule cu 50000 pentru cele care expira dupa 2023 cu un sponsor, blocand liniile inainte de actualizare.







select nume\_contract, data\_expirare, valoare

from contract

where to\_char(data\_expirare, 'YYYY') >= '2023';

create or replace procedure ex7

is

cursor c is

select \*

from contract

where to\_char(data\_expirare, 'YYYY') >= '2023'

for update of valoare nowait;

begin

for i in c loop

update contract

set valoare = valoare + 50000

where current of c;

end loop;

end ex7;

/

begin

ex7;

END;

select nume\_contract, data\_expirare, valoare

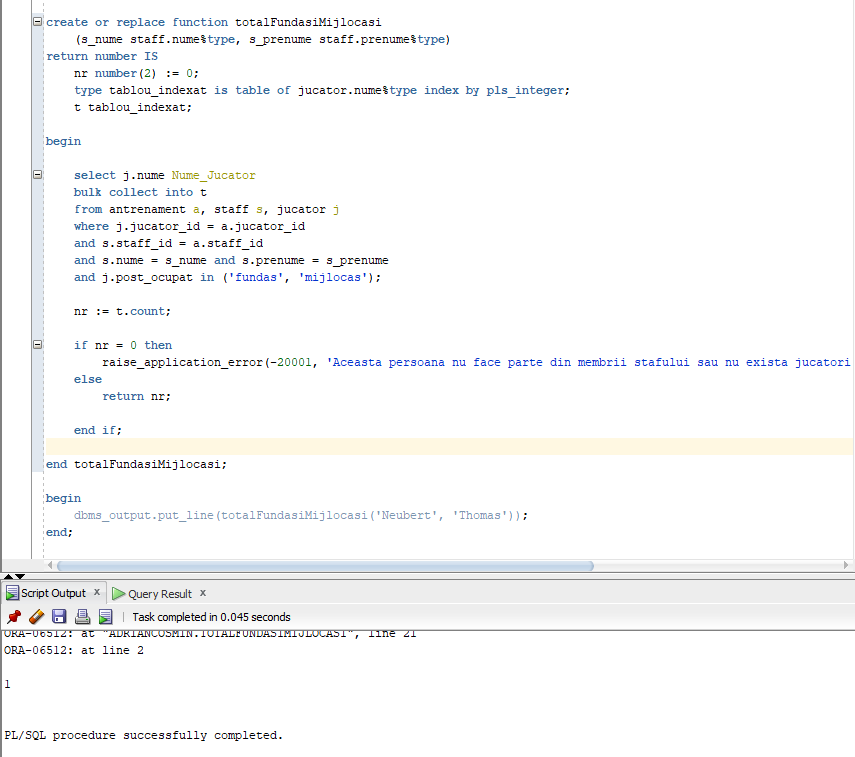
from contract

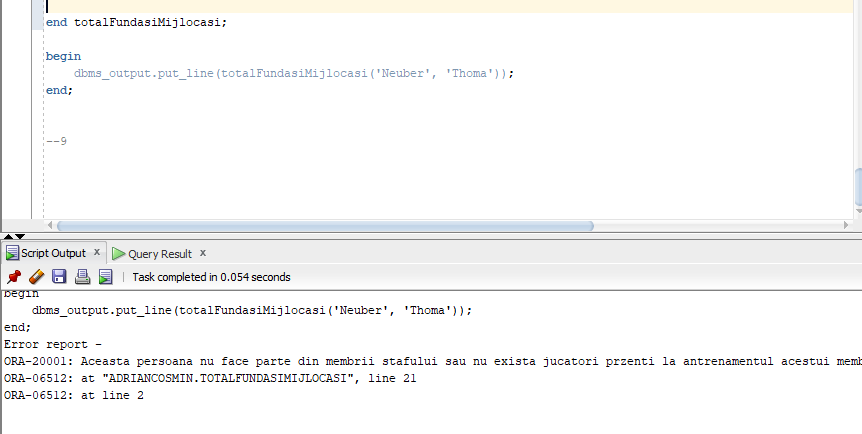
where to\_char(data\_expirare, 'YYYY') >= '2023';

rollback;

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

O functie care returneaza cati fundasi si mijlocasi participa la antrenamentele unui anumit membru din staff ale carui nume si prenume sunt date ca parametrii. (Erorile pot aparea in cazul in care este dat ca parametru o persoana care nu exista in baza de date a staff-ului sau daca se introduc datele unui membru din staff la care nu participa nici un jucator).





create or replace function totalFundasiMijlocasi

(s\_nume staff.nume%type, s\_prenume staff.prenume%type)

return number IS

nr number(2) := 0;

type tablou\_indexat is table of jucator.nume%type index by pls\_integer;

t tablou\_indexat;

begin

select j.nume Nume\_Jucator

bulk collect into t

from antrenament a, staff s, jucator j

where j.jucator\_id = a.jucator\_id

and s.staff\_id = a.staff\_id

and s.nume = s\_nume and s.prenume = s\_prenume

and j.post\_ocupat in ('fundas', 'mijlocas');

nr := t.count;

if nr = 0 then

raise\_application\_error(-20001, 'Aceasta persoana nu face parte din membrii stafului sau nu exista jucatori przenti la antrenamentul acestui membru din staff');

else

return nr;

end if;

end totalFundasiMijlocasi;

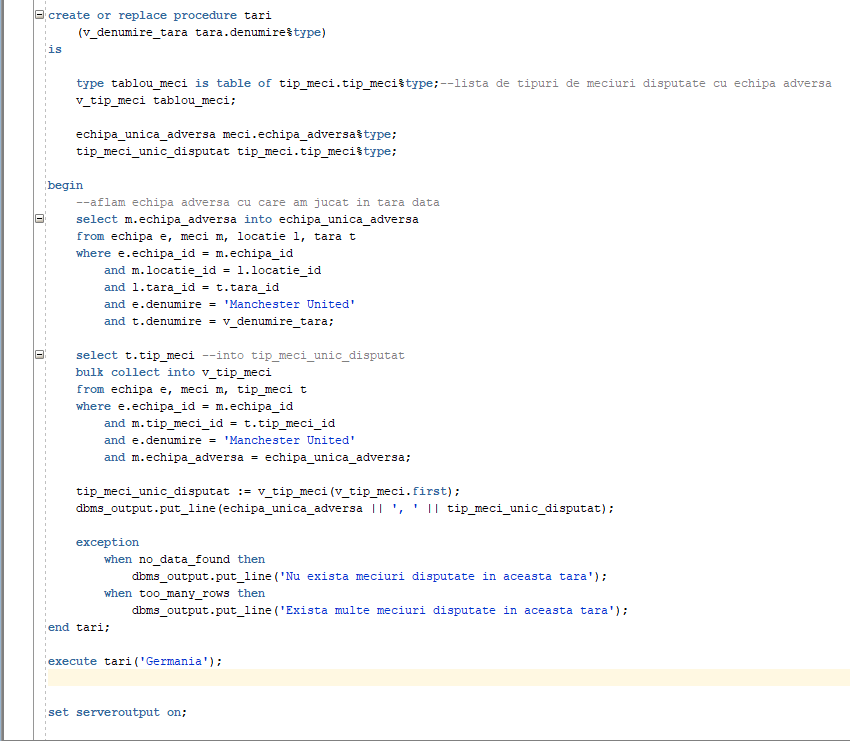
begin

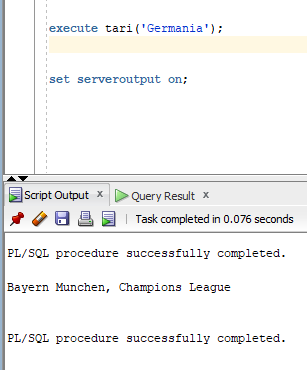
dbms\_output.put\_line(totalFundasiMijlocasi('Neuber', 'Thoma'));

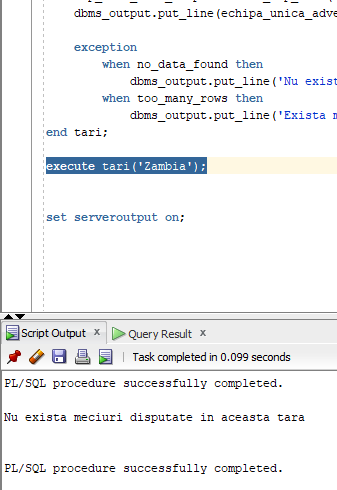
end;

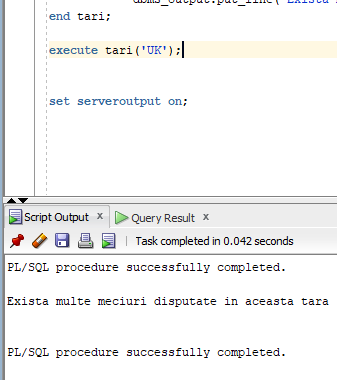
1. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un **subprogram stocat de tip procedură** care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Sa se creeze o functie care primeste ca parametru o tara, si pentru echipa noastra cu numele de ‘Manchester United’ , sa se spuna echipa adversa cu care disputam meciul in tara respectiva si tipul de meci, daca sunt mai multe meciuri cu aceeasi echipa dar in competitii diferite, se va afisa doar primul. Vom trata erorile urmatoare, cazul in care se da o tara in care se disputa mai multe meciuri cu echipe diferite si daca se da o tara in care nu s-a disputant nici un meci.









create or replace procedure tari

(v\_denumire\_tara tara.denumire%type)

is

type tablou\_meci is table of tip\_meci.tip\_meci%type;--lista de tipuri de meciuri disputate cu echipa adversa

v\_tip\_meci tablou\_meci;

echipa\_unica\_adversa meci.echipa\_adversa%type;

tip\_meci\_unic\_disputat tip\_meci.tip\_meci%type;

begin

--aflam echipa adversa cu care am jucat in tara data

select m.echipa\_adversa into echipa\_unica\_adversa

from echipa e, meci m, locatie l, tara t

where e.echipa\_id = m.echipa\_id

and m.locatie\_id = l.locatie\_id

and l.tara\_id = t.tara\_id

and e.denumire = 'Manchester United'

and t.denumire = v\_denumire\_tara;

select t.tip\_meci --into tip\_meci\_unic\_disputat

bulk collect into v\_tip\_meci

from echipa e, meci m, tip\_meci t

where e.echipa\_id = m.echipa\_id

and m.tip\_meci\_id = t.tip\_meci\_id

and e.denumire = 'Manchester United'

and m.echipa\_adversa = echipa\_unica\_adversa;

tip\_meci\_unic\_disputat := v\_tip\_meci(v\_tip\_meci.first);

dbms\_output.put\_line(echipa\_unica\_adversa || ', ' || tip\_meci\_unic\_disputat);

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista meciuri disputate in aceasta tara');

when too\_many\_rows then

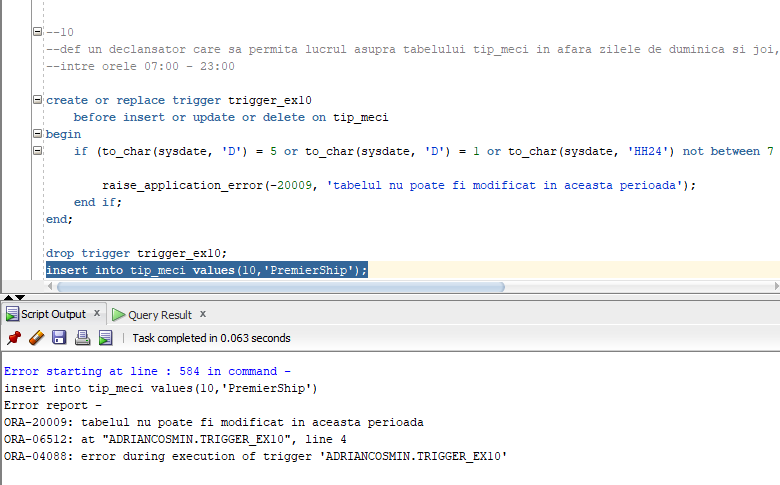
dbms\_output.put\_line('Exista multe meciuri disputate in aceasta tara');

end tari;

execute tari('UK');

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

Definim un declansator care sa permita actualizari asupra tabelului tip\_meci tot timpul intre orele 07:00 – 23:00, in afara de zilele de joi si duminica.



create or replace trigger trigger\_ex10

before insert or update or delete on tip\_meci

begin

if (to\_char(sysdate, 'D') = 5 or to\_char(sysdate, 'D') = 1 or to\_char(sysdate, 'HH24') not between 7 and 23) then

raise\_application\_error(-20009, 'tabelul nu poate fi modificat in aceasta perioada');

end if;

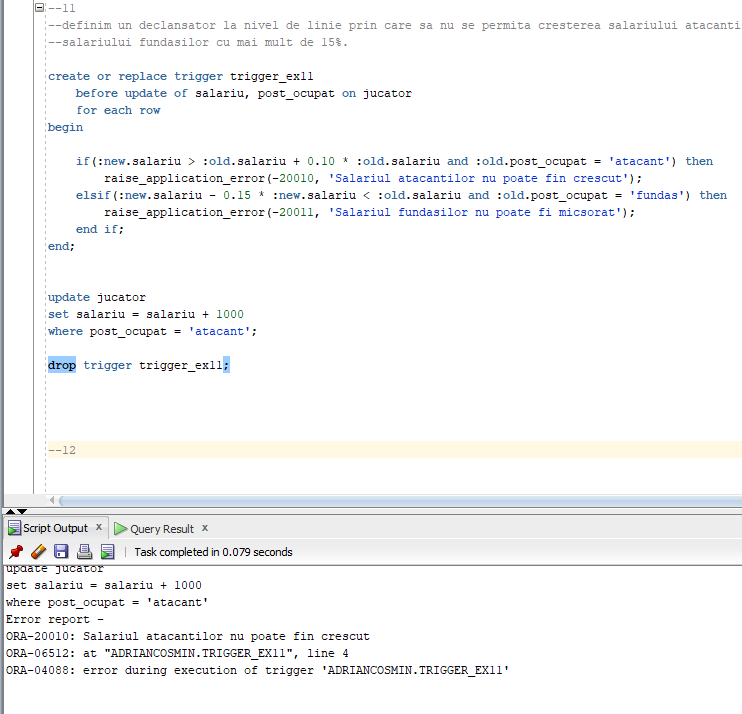
end;

drop trigger trigger\_ex10;

insert into tip\_meci values(10,'PremierShip');

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

Definim un declansator la nivel de linie prin care sa nu se permita cresterea salariului atacantilor cu mai mult de 10% si micsorarea salariului fundasiilor cu mai mult de 15%.



create or replace trigger trigger\_ex11

before update of salariu, post\_ocupat on jucator

for each row

begin

if(:new.salariu > :old.salariu + 0.10 \* :old.salariu and :old.post\_ocupat = 'atacant') then

raise\_application\_error(-20010, 'Salariul atacantilor nu poate fin crescut');

elsif(:new.salariu - 0.15 \* :new.salariu < :old.salariu and :old.post\_ocupat = 'fundas') then

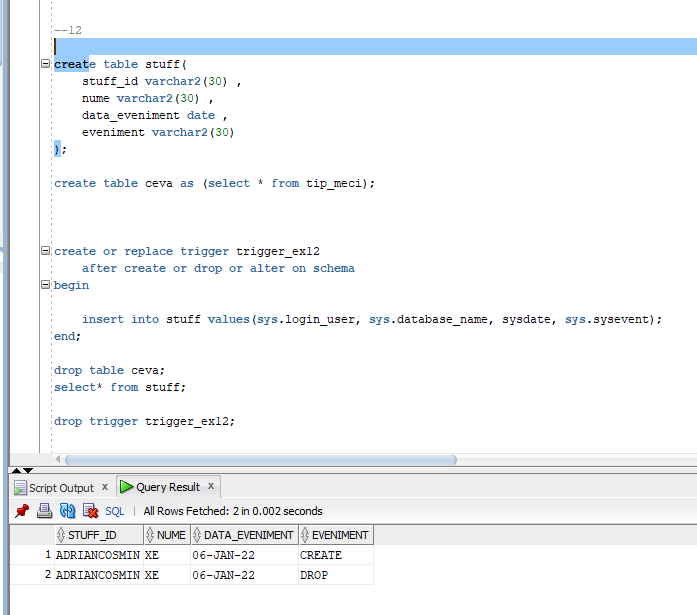
raise\_application\_error(-20011, 'Salariul fundasilor nu poate fi micsorat');

end if;

end;

1. Definiți un *trigger* de tip LDD. Declanșați *trigger*-ul.

Definim un declansator care sa introduca date in tabelul stuff dupa ce utilizatorul a folosit o comanda LDD.



create table stuff(

stuff\_id varchar2(30) ,

nume varchar2(30) ,

data\_eveniment date ,

eveniment varchar2(30)

);

create or replace trigger trigger\_ex12

after create or drop or alter on schema

begin

insert into stuff values(sys.login\_user, sys.database\_name, sysdate, sys.sysevent);

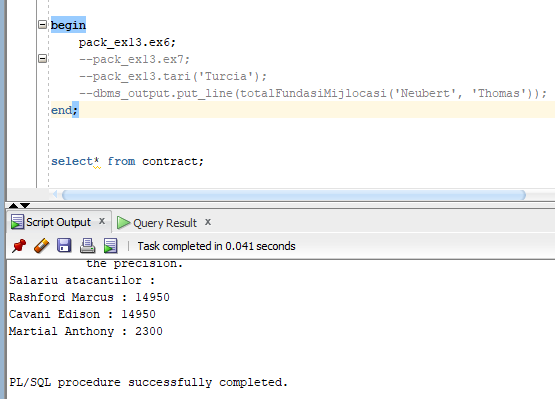
end;

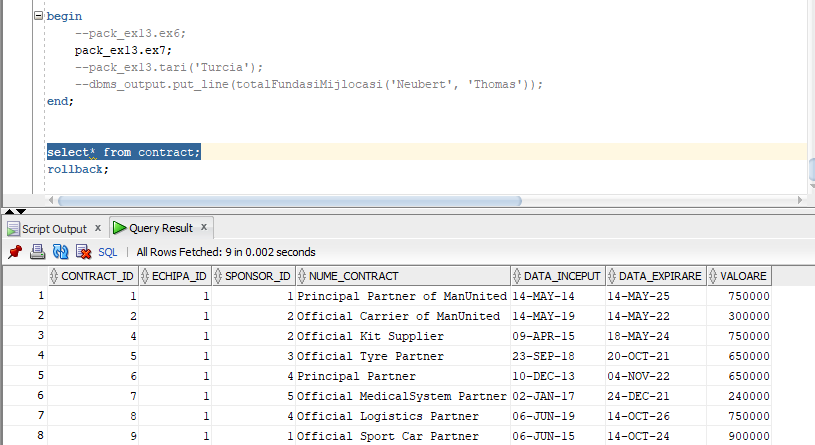
create table ceva as (select \* from tip\_meci);

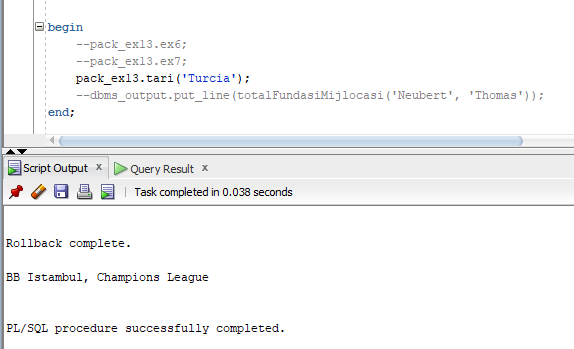
select\* from stuff;

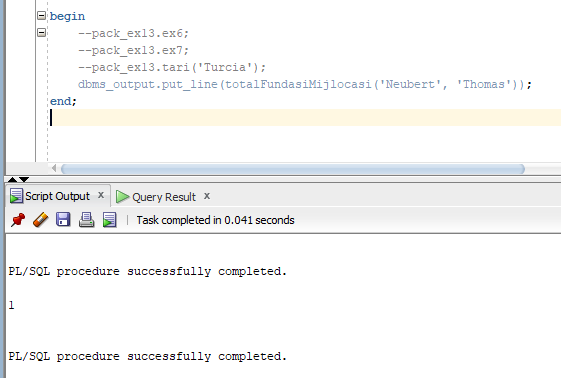
drop table ceva;

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









create or replace package pack\_ex13 as

procedure ex6;

procedure ex7;

function totalFundasiMijlocasi (s\_nume staff.nume%type, s\_prenume staff.prenume%type)

return number;

procedure tari (v\_denumire\_tara tara.denumire%type);

end pack\_ex13;

create or replace package body pack\_ex13 as

function totalFundasiMijlocasi (s\_nume staff.nume%type, s\_prenume staff.prenume%type)

return number IS

nr number(2) := 0;

type tablou\_indexat is table of jucator.nume%type index by pls\_integer;

t tablou\_indexat;

begin

select j.nume Nume\_Jucator

bulk collect into t

from antrenament a, staff s, jucator j

where j.jucator\_id = a.jucator\_id

and s.staff\_id = a.staff\_id

and s.nume = s\_nume and s.prenume = s\_prenume

and j.post\_ocupat in ('fundas', 'mijlocas');

nr := t.count;

if nr = 0 then

raise\_application\_error(-20001, 'Aceasta persoana nu face parte din membrii stafului sau nu exista jucatori przenti la antrenamentul acestui membru din staff');

else

return nr;

end if;

end totalFundasiMijlocasi;

procedure ex6 IS

type tablou\_indexat is table of jucator.salariu%type index by pls\_integer;

t\_salarii tablou\_indexat;--salariu atacant

type tablou\_imbricat is table of jucator.nume%type;

t\_nume tablou\_imbricat;--nume atacant

begin

select salariu

bulk collect into t\_salarii

from jucator

where post\_ocupat = 'atacant';

select (nume || ' ' || prenume) as nume\_jucator

bulk collect into t\_nume

from jucator

where post\_ocupat = 'atacant';

dbms\_output.put\_line('Salariu atacantilor : ');

for i in t\_salarii.first..t\_salarii.last loop

dbms\_output.put\_line(t\_nume(i) || ' : ' || (0.15\*t\_salarii(i) + t\_salarii(i)));

end loop;

end ex6;

procedure ex7 is

cursor c is

select \*

from contract

where to\_char(data\_expirare, 'YYYY') >= '2023'

for update of valoare nowait;

begin

for i in c loop

update contract

set valoare = valoare + 50000

where current of c;

end loop;

end ex7;

procedure tari (v\_denumire\_tara tara.denumire%type) is

type tablou\_meci is table of tip\_meci.tip\_meci%type;--lista de tipuri de meciuri disputate cu echipa adversa

v\_tip\_meci tablou\_meci;

echipa\_unica\_adversa meci.echipa\_adversa%type;

tip\_meci\_unic\_disputat tip\_meci.tip\_meci%type;

begin

--aflam echipa adversa cu care am jucat in tara data

select m.echipa\_adversa into echipa\_unica\_adversa

from echipa e, meci m, locatie l, tara t

where e.echipa\_id = m.echipa\_id

and m.locatie\_id = l.locatie\_id

and l.tara\_id = t.tara\_id

and e.denumire = 'Manchester United'

and t.denumire = v\_denumire\_tara;

select t.tip\_meci --into tip\_meci\_unic\_disputat

bulk collect into v\_tip\_meci

from echipa e, meci m, tip\_meci t

where e.echipa\_id = m.echipa\_id

and m.tip\_meci\_id = t.tip\_meci\_id

and e.denumire = 'Manchester United'

and m.echipa\_adversa = echipa\_unica\_adversa;

tip\_meci\_unic\_disputat := v\_tip\_meci(v\_tip\_meci.first);

dbms\_output.put\_line(echipa\_unica\_adversa || ', ' || tip\_meci\_unic\_disputat);

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista meciuri disputate in aceasta tara');

when too\_many\_rows then

dbms\_output.put\_line('Exista multe meciuri disputate in aceasta tara');

end tari;

end pack\_ex13;

begin

pack\_ex13.ex6;

end;

begin

pack\_ex13.ex7;

end;

begin

pack\_ex13.tari('Turcia');

end;

begin

dbms\_output.put\_line(totalFundasiMijlocasi('Neubert', 'Thomas'));

end;

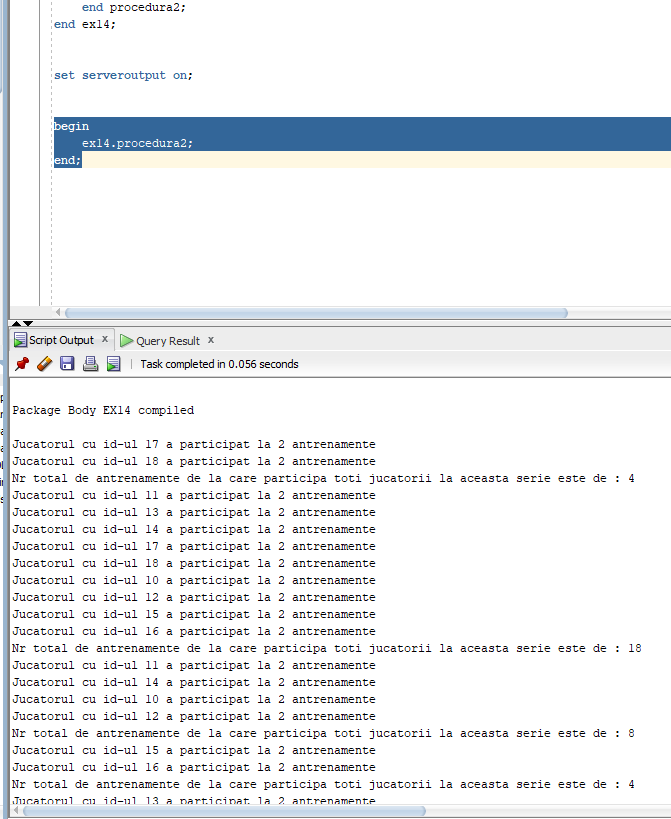
1. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

Functia nr\_antrenam\_jucator, primeste ca parametru un jucator\_id si returneaza nr de antrenamente la care a participat printr-un record care retine o variabila de tip number.

Functia lista\_jucatori\_by\_tip\_antrenam, primeste ca parametru un tip de antrenament si returneaza un tablou imbricat de id-uri de jucatori care au participat la acest antrenament.

Procedura1 primeste ca parametru un tip de antrenament si apoi apeleaza functia lista\_jucatori\_by\_tip\_antrenam, ca sa afle id-urile jucatorilor care participa la antrenament. Apoi apelam pentru fiecare jucator\_id sa vedem la cate antrenament participa acesta in total. Vom afisa si numarul total de antrenamente pentru fiecare serie.

Procedura2 apeleaza procedura1 pentru fiecare tip de antrenament cu ajutorul unui cursor



create or replace package ex14 as

type nr\_total\_antr is record(nr\_antr number(2));

type v\_jucatori\_id is table of jucator.jucator\_id%type;

function nr\_antrenam\_jucator(id\_j jucator.jucator\_id%type) return nr\_total\_antr;

function lista\_jucatori\_by\_tip\_antrenam(tip\_antr antrenament.zi%type) return v\_jucatori\_id;

procedure procedura1 (tip\_antr antrenament.zi%type);

procedure procedura2;

end ex14;

create or replace package body ex14 as

function nr\_antrenam\_jucator(id\_j jucator.jucator\_id%type) return nr\_total\_antr is

nta nr\_total\_antr;

begin

select distinct count(a.zi)

into nta.nr\_antr

from jucator j, antrenament a

where j.jucator\_id = a.jucator\_id

and j.jucator\_id = id\_j;

return nta;

exception

when no\_data\_found then

dbms\_output.put\_line('Jucatorul acesta nu exista sau nu a participat la nici un antrenament');

end nr\_antrenam\_jucator;

function lista\_jucatori\_by\_tip\_antrenam(tip\_antr antrenament.zi%type) return v\_jucatori\_id is

v v\_jucatori\_id := v\_jucatori\_id();

begin

select distinct j.jucator\_id

bulk collect into v

from jucator j, antrenament a

where j.jucator\_id = a.jucator\_id

and a.zi = tip\_antr;

return v;

end lista\_jucatori\_by\_tip\_antrenam;

procedure procedura1 (tip\_antr antrenament.zi%type) is

nta nr\_total\_antr;

v v\_jucatori\_id;

nr number(2) := 0;

begin

v := lista\_jucatori\_by\_tip\_antrenam(tip\_antr);

for i in v.first..v.last loop

nta := nr\_antrenam\_jucator(v(i));

nr := nr + nta.nr\_antr;

dbms\_output.put\_line('Jucatorul cu id-ul ' || v(i) || ' a participat la ' || nta.nr\_antr || ' antrenamente');

end loop;

dbms\_output.put\_line('Nr total de antrenamente de la care participa toti jucatorii la aceasta serie este de : ' || nr);

end procedura1;

procedure procedura2 is

cursor c is

select distinct zi z

from antrenament;

begin

for i in c loop

procedura1(i.z);

end loop;

end procedura2;

end ex14;

begin

ex14.procedura2;

end;