**PROIECT SGBD**

Realizat de Stoica Elias-Valeriu

Facultatea de Matematica si Informatica Bucuresti

Grupa 251

1. **Descrierea modelului real, a utilității acestuia si a regulilor de funcționare**

Ligile din Europa pun la start echipe din diferite colțuri ale țărilor. Fiecare echipa are un stadion în orașul de provenienta, unde disputa meciurile de pe propriul teren. Stadioanele se clasifica după capacitate, dupa numarul de stele oferit de UEFA, depinzand de cat de modern este. O echipa este alcatuita dintr-un lot de jucători. Acestea sunt pregatiti de un antrenor principal.

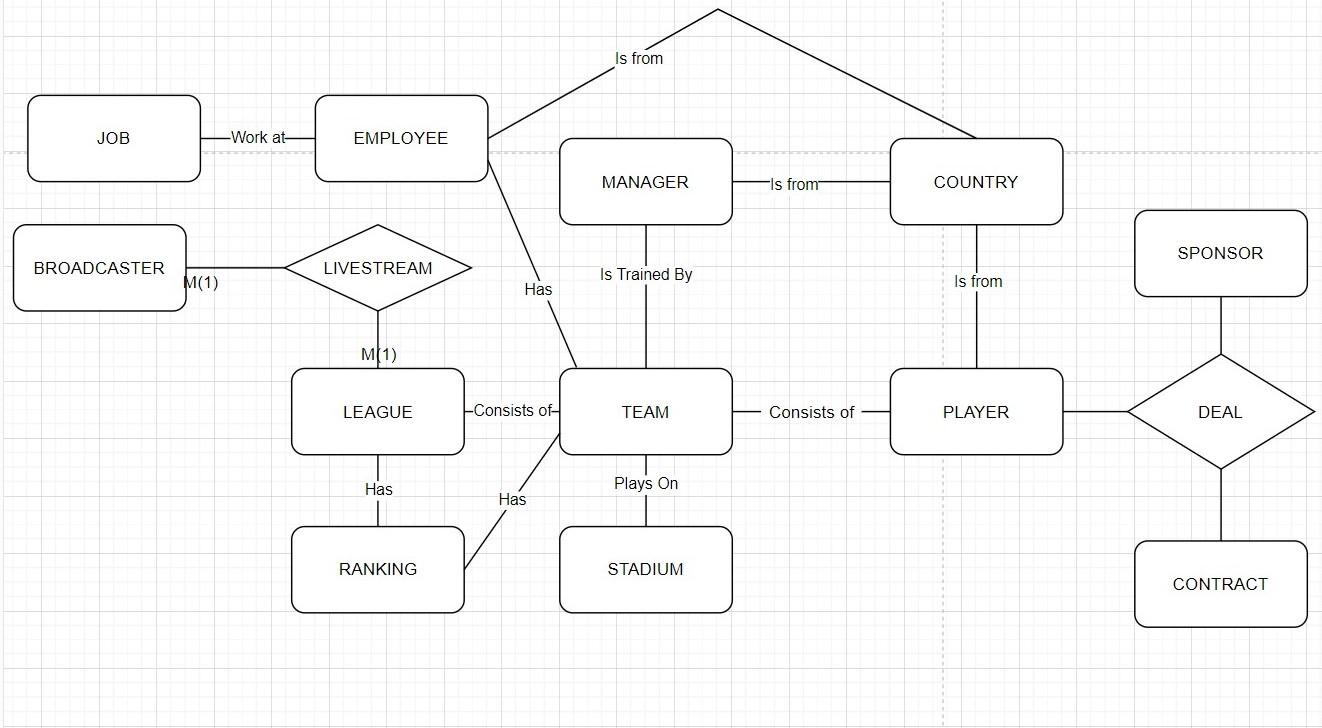
Fiecare echipa are în spate un număr mare de angajați. Aceștia primesc un salariu atractiv, în funcție de job. Pentru fiecare membru din club se cunoaște țara de proveniență.

Jucătorii sunt sponsorizati de anumite companii. Acestia au un contract bine stabilit si nu pot avea mai multe sponsorizări în același timp. Sponsorizarile constau în produse de marca, adidasi, echipament etc. Acest contract poate avea și o clauza de reziliere, in cazul in care exista, jucătorul poate renunța oricând la sponsorul sau.

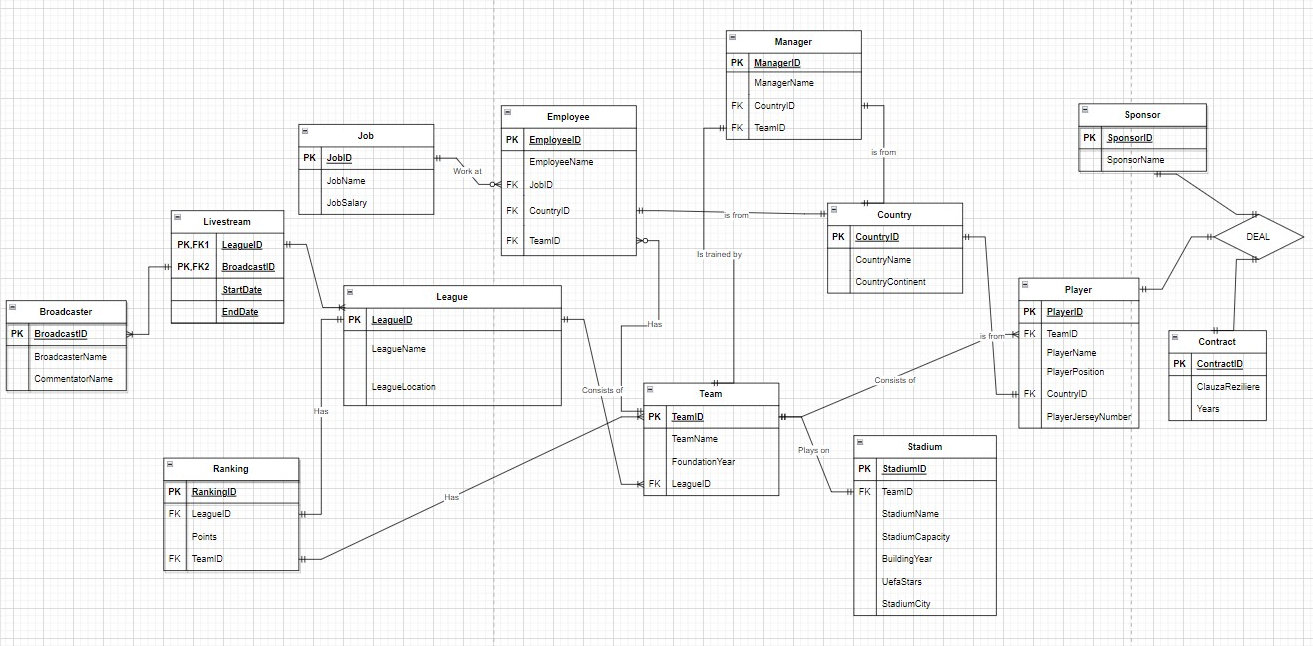
Campionatul de fotbal are mai multe contracte de livestreaming cu diferite televiziuni. Acestea sunt făcute pe o perioada bine definită.

La finalul sezonului, echipele își află locul în clasament. În funcție de cate puncte acumulează în timpul sezonului, se pot bate pentru competitiile europene.

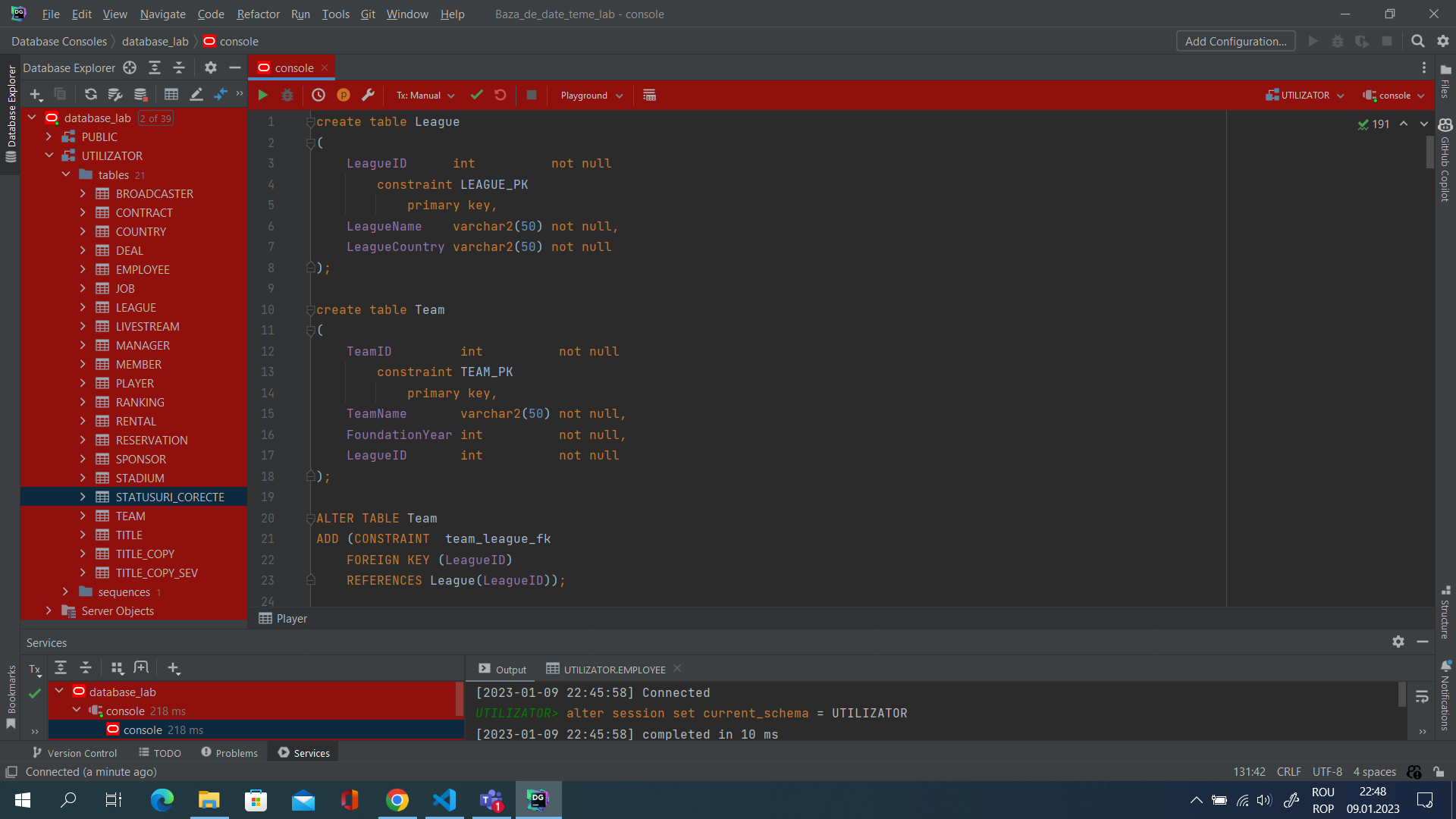
1. **Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.**



1. **Realizarea diagramei conceptuale corespunzatoare diagramei entitate-relatie proiectate la punctul 6.**



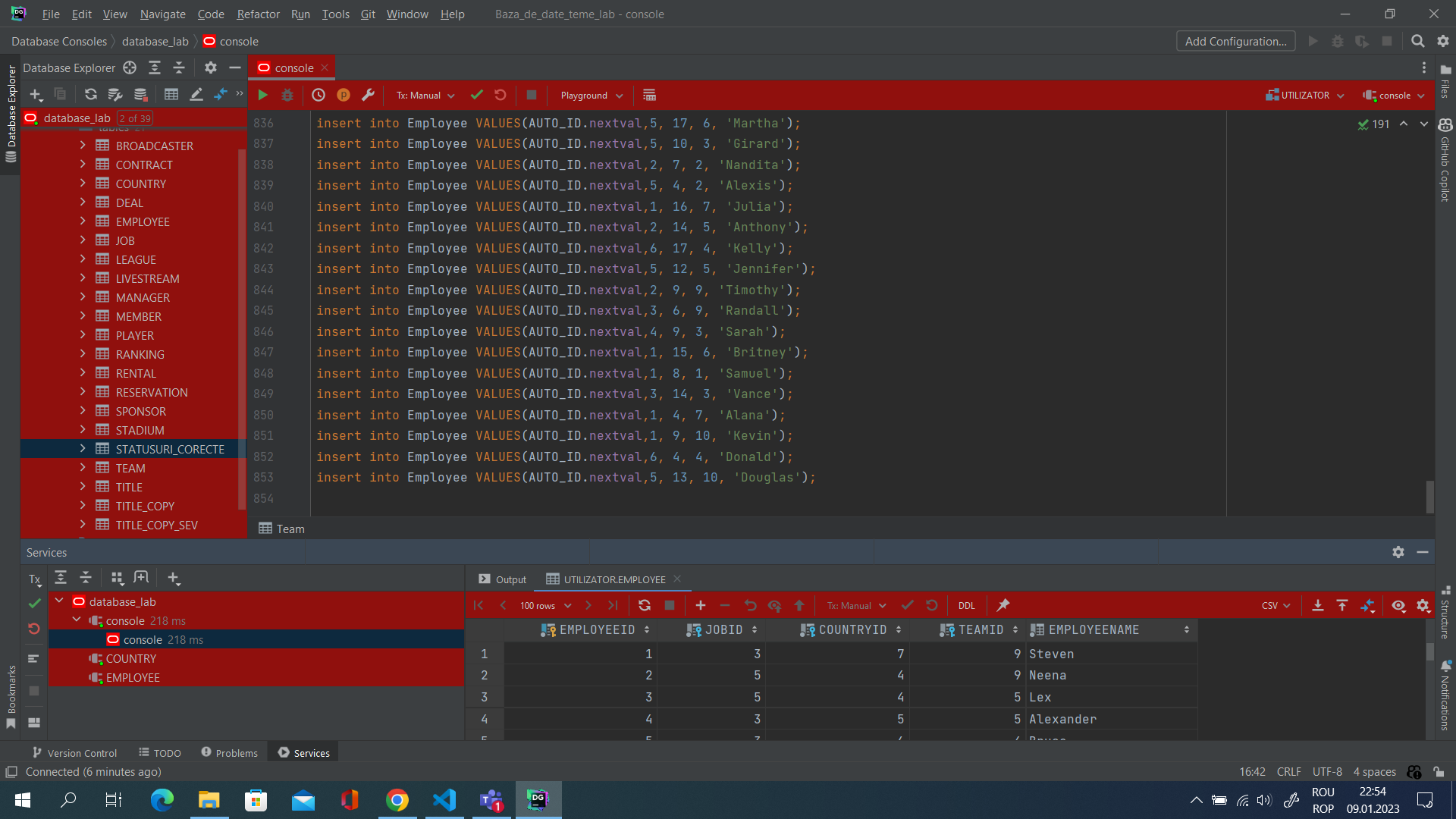
1. **Implementati in Oracle diagrama conceptuala realizata definiti toate tabelele, implementand toate constrangerile de integritate necesare (chei primare, chei externe etc.).**



Fisierele de creare sunt intr-un repository pe github, la adresa urmatoare:

[**https://github.com/wood11nho/Proiect-SGBD**](https://github.com/wood11nho/Proiect-SGBD)

1. **Adaugati informatii coerente in tabelele create (minim 5 inregistrari pentru fiecare entitate independenta; minim 10 inregistrari pentru tabela asociativa).**



Fisierele cu comenzile de inserare se gasesc pe urmatorul repository de pe github:

[**https://github.com/wood11nho/Proiect-SGBD**](https://github.com/wood11nho/Proiect-SGBD)

1. **Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze doua tipuri de colectie studiate. Apelati subprogramul.**

**--Pentru un jucator de fotbal, caruia i se va da ID-ul de la tastatura, salvati si afisati numele tuturor coechipierilor sai, iar pentru fiecare coechipier sa se salveze si sponsorul acestuia.**

create or replace procedure *afisare\_coechipieri*

(id\_jucator\_tast Player.PlayerID%type)

AS

TYPE tablou\_indexat is TABLE OF Player.PlayerName%type INDEX BY PLS\_INTEGER;

nume\_coechipieri tablou\_indexat;

TYPE tablou\_imbricat is TABLE OF Sponsor.SponsorName%type;

sponsor\_jucatori tablou\_imbricat := tablou\_imbricat();

i NUMBER;

id\_echipa\_jucator Player.TeamID%type;

id\_echipa\_coechiper Player.TeamID%type;

nume\_jucator Player.PlayerName%type;

id\_coechipier Player.PlayerID%type;

nume\_coechipier Player.PlayerName%type;

nume\_sponsor Sponsor.SponsorName%type;

numar\_jucatori NUMBER;

BEGIN

i := 0;

select TeamID, PlayerName into id\_echipa\_jucator, nume\_jucator from Player where PlayerID = id\_jucator\_tast;

select *count*(\*) into numar\_jucatori from Player;

for id\_jucator in 1..numar\_jucatori

loop

select TeamID into id\_echipa\_coechiper from Player where PlayerID = id\_jucator;

if id\_echipa\_coechiper = id\_echipa\_jucator and id\_jucator\_tast <> id\_jucator then

i := i + 1;

select PlayerName, PlayerID into nume\_coechipier, id\_coechipier from Player where PlayerID = id\_jucator;

nume\_coechipieri(i) := nume\_coechipier;

sponsor\_jucatori.extend;

select SponsorName into nume\_sponsor from Sponsor where SponsorID = (select SponsorID from Deal where PlayerID = id\_coechipier);

sponsor\_jucatori(i) := nume\_sponsor;

end if;

end loop;

DBMS\_OUTPUT.*PUT\_LINE*('Jucatorul ' || nume\_jucator || ' are urmatorii coechipieri:');

FOR j IN 1..nume\_coechipieri.COUNT LOOP

DBMS\_OUTPUT.*PUT\_LINE*(nume\_coechipieri(j) || ' - ' || sponsor\_jucatori(j));

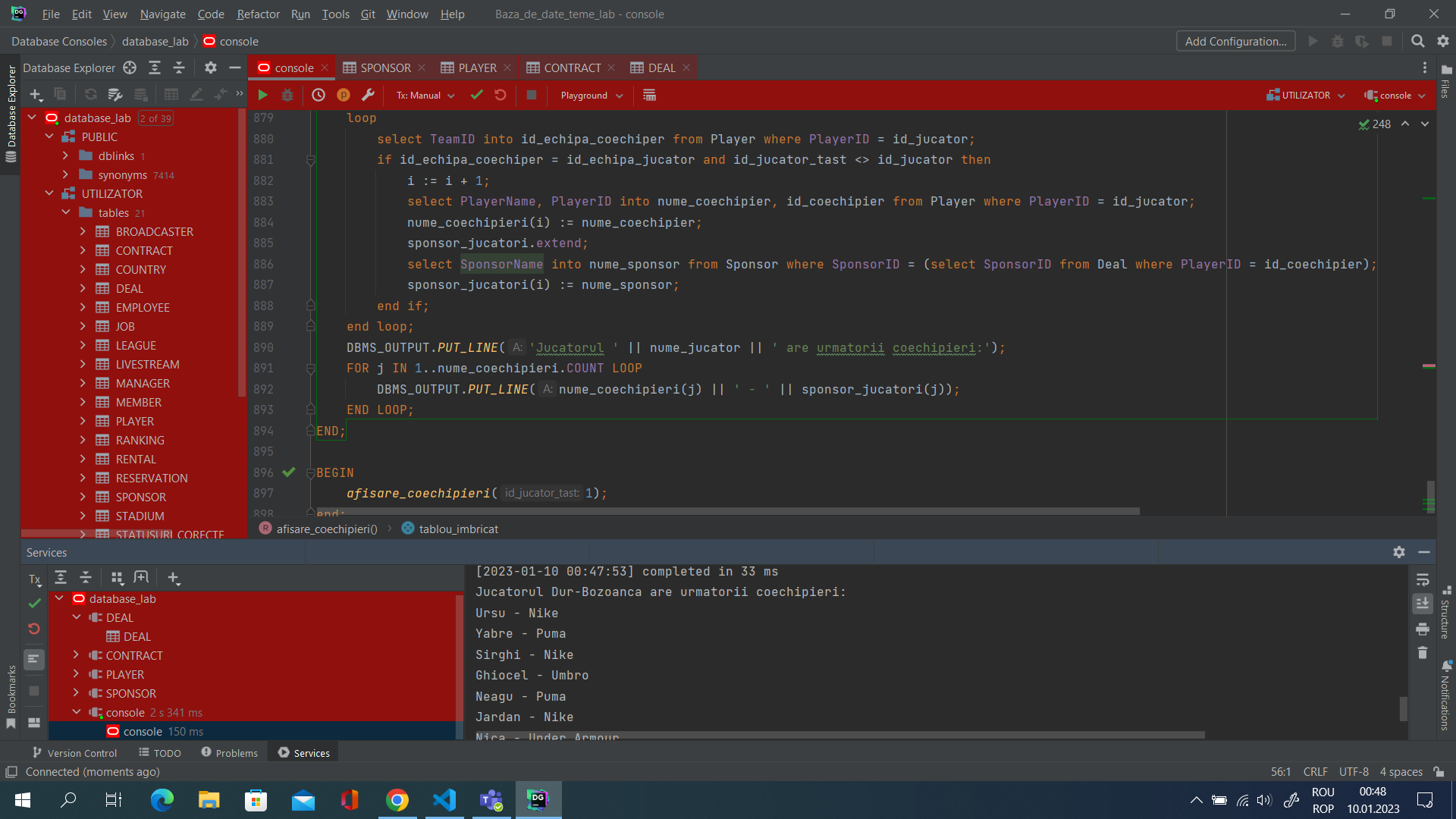
END LOOP;

END;

BEGIN

*afisare\_coechipieri*(1);

end;



1. **Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze 2 tipuri de cursoare studiate, unul dintre acestea fiind parametrizat. Apelati subprogramul.**

**--Sa se afiseze pentru fiecare echipa numarul de jucatori. In cazul in care echipa nu are jucatori inscrisi, se va afisa un mesaj corespunzator. Pentru fiecare jucator cu numarul de pe tricou mai mare decat un parametru dat, sa se afiseze pozitia, numarul de pe tricou si tara de provenienta.**

CREATE OR REPLACE FUNCTION *filtrare\_jucatori\_dupa\_numar*

(numar\_echipa Player.TeamID%TYPE, numar\_tricou Player.JerseyNumber%TYPE)

RETURN NUMBER is

nr\_jucatori\_filtrati NUMBER;

BEGIN

select *count*(\*) into nr\_jucatori\_filtrati

from Player

where JerseyNumber > numar\_tricou and TeamID = numar\_echipa;

return nr\_jucatori\_filtrati;

end;

CREATE OR REPLACE PROCEDURE *afisare\_detalii\_echipe* AS

CURSOR c1 is

SELECT t.TeamID id\_echipa, TeamName nume\_echipa, *COUNT*(PlayerID) numar\_jucatori

FROM Team t, Player p

WHERE t.TeamID = p.TeamID(+)

GROUP BY t.TeamID, TeamName;

CURSOR c2(parametru NUMBER, echipa\_id Team.TeamID%TYPE) is

SELECT PlayerName nume\_jucator, PlayerPosition pozitie\_jucator, CountryName tara\_jucator, JerseyNumber numar\_jucator

FROM Player p, Country c

WHERE p.CountryID = c.CountryID(+) and JerseyNumber > parametru and TeamID = echipa\_id;

TYPE tip\_detalii\_echipa IS RECORD(

id\_echipa Team.TeamID%TYPE,

nume\_echipa Team.TeamName%TYPE,

numar\_jucatori NUMBER

);

detalii\_echipa tip\_detalii\_echipa;

count\_echipe NUMBER;

count\_jucatori NUMBER;

numar\_jucator\_tastatura NUMBER := &numar\_tast;

nr\_jucatori\_filtrati NUMBER;

BEGIN

DBMS\_OUTPUT.*PUT\_LINE*('Echipe si jucatori: ');

count\_echipe := 0;

OPEN c1;

LOOP

FETCH c1 INTO detalii\_echipa;

count\_echipe := count\_echipe + 1;

EXIT WHEN c1%NOTFOUND;

IF detalii\_echipa.numar\_jucatori = 0 THEN

DBMS\_OUTPUT.*PUT\_LINE*(count\_echipe || '. ' || detalii\_echipa.nume\_echipa || ' nu are niciun jucator semnat.');

ELSE

nr\_jucatori\_filtrati:=*filtrare\_jucatori\_dupa\_numar*(detalii\_echipa.id\_echipa,numar\_jucator\_tastatura);

DBMS\_OUTPUT.*PUT\_LINE*(count\_echipe || '. ' || detalii\_echipa.nume\_echipa || ' are ' || detalii\_echipa.numar\_jucatori || ' jucatori semnati iar cei ' || nr\_jucatori\_filtrati || ' cu numarul mai mare decat ' || numar\_jucator\_tastatura || ' sunt: ');

count\_jucatori := 0;

FOR i in c2(numar\_jucator\_tastatura, detalii\_echipa.id\_echipa) LOOP

count\_jucatori := count\_jucatori + 1;

DBMS\_OUTPUT.*PUT\_LINE*(count\_jucatori || '. ' || i.nume\_jucator || ', numarul ' || i.numar\_jucator || ', joaca pe pozitia ' || i.pozitie\_jucator || ' si este din ' || i.tara\_jucator);

end loop;

END IF;

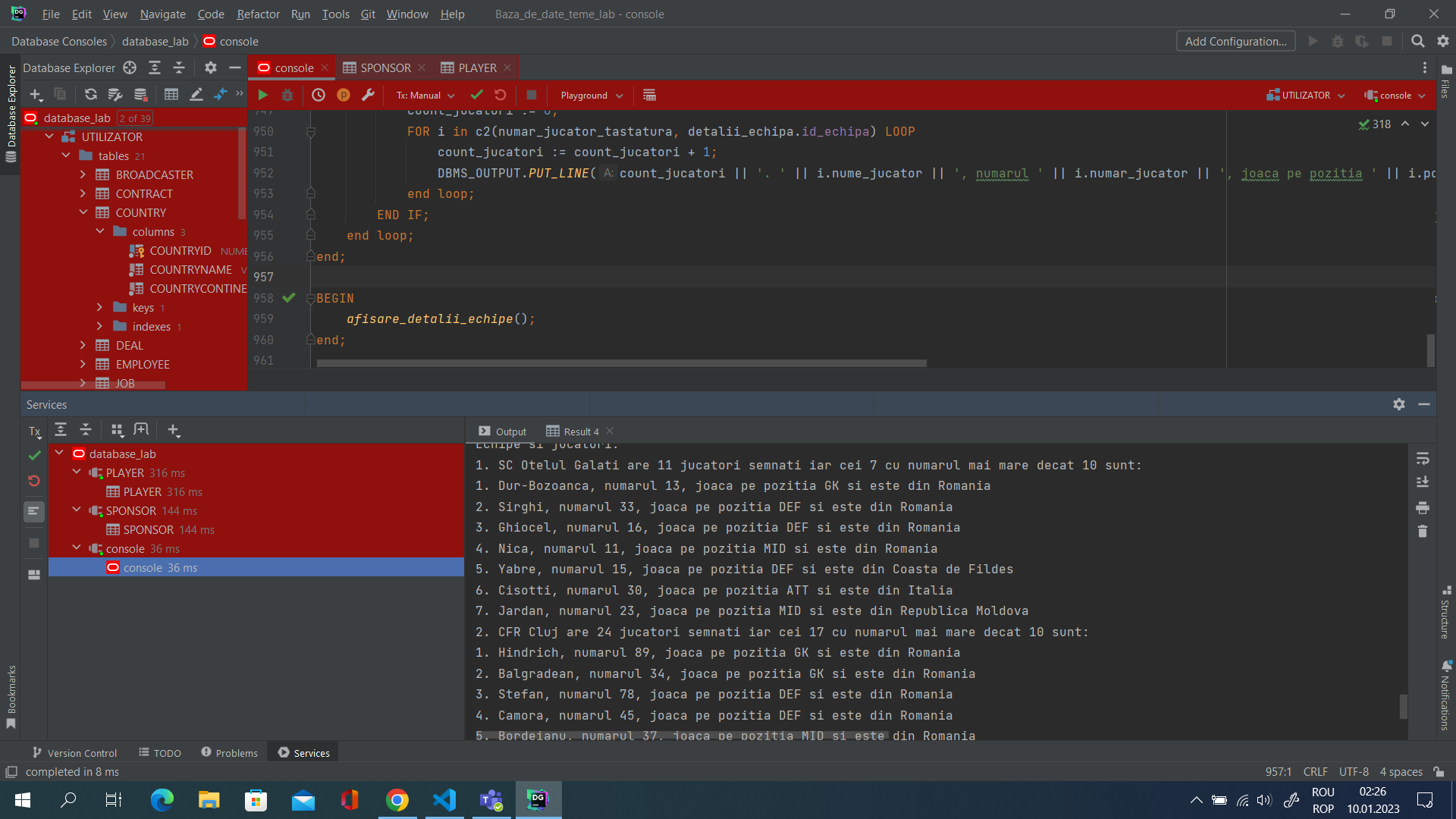
end loop;

end;

BEGIN

*afisare\_detalii\_echipe*();

end;



1. **Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip functie care sa utilizeze intr-o singura comanda SQL 3 tabele diferite. Definiti minim 2 exceptii.**

**Scrie o functie care va afisa stadionul pe care evolueaza echipa unui jucator, al carui nume va fi dat de la tastatura.**

CREATE OR REPLACE FUNCTION *stadion\_jucator*

(nume\_jucator Player.PlayerName%TYPE)

RETURN Stadium.StadiumName%TYPE IS

id\_jucator Player.PlayerID%TYPE;

nume\_stadion Stadium.StadiumName%TYPE;

BEGIN

select PlayerID

into id\_jucator

from Player

where PlayerName = nume\_jucator;

select StadiumName

into nume\_stadion

from Stadium s

join Team t on t.TeamID = s.TeamID

join Player p on p.TeamID = t.TeamID

where p.PlayerID = id\_jucator;

return nume\_stadion;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

*RAISE\_APPLICATION\_ERROR*(-20000, 'Nu exista jucator cu numele ' || nume\_jucator);

WHEN OTHERS THEN

*RAISE\_APPLICATION\_ERROR*(-20002, 'Alta eroare');

end;

BEGIN

DBMS\_OUTPUT.*PUT\_LINE*('Jucatorul cu numele Sefer evolueaza pe ' || *stadion\_jucator*('Sefer'));

--Raspuns:

--[2023-01-10 03:17:42] completed in 15 ms

--Jucatorul cu numele Sefer evolueaza pe stadionul Stadionul Giulesti

DBMS\_OUTPUT.*PUT\_LINE*('Jucatorul cu numele Elias evolueaza pe ' || *stadion\_jucator*('Elias'));

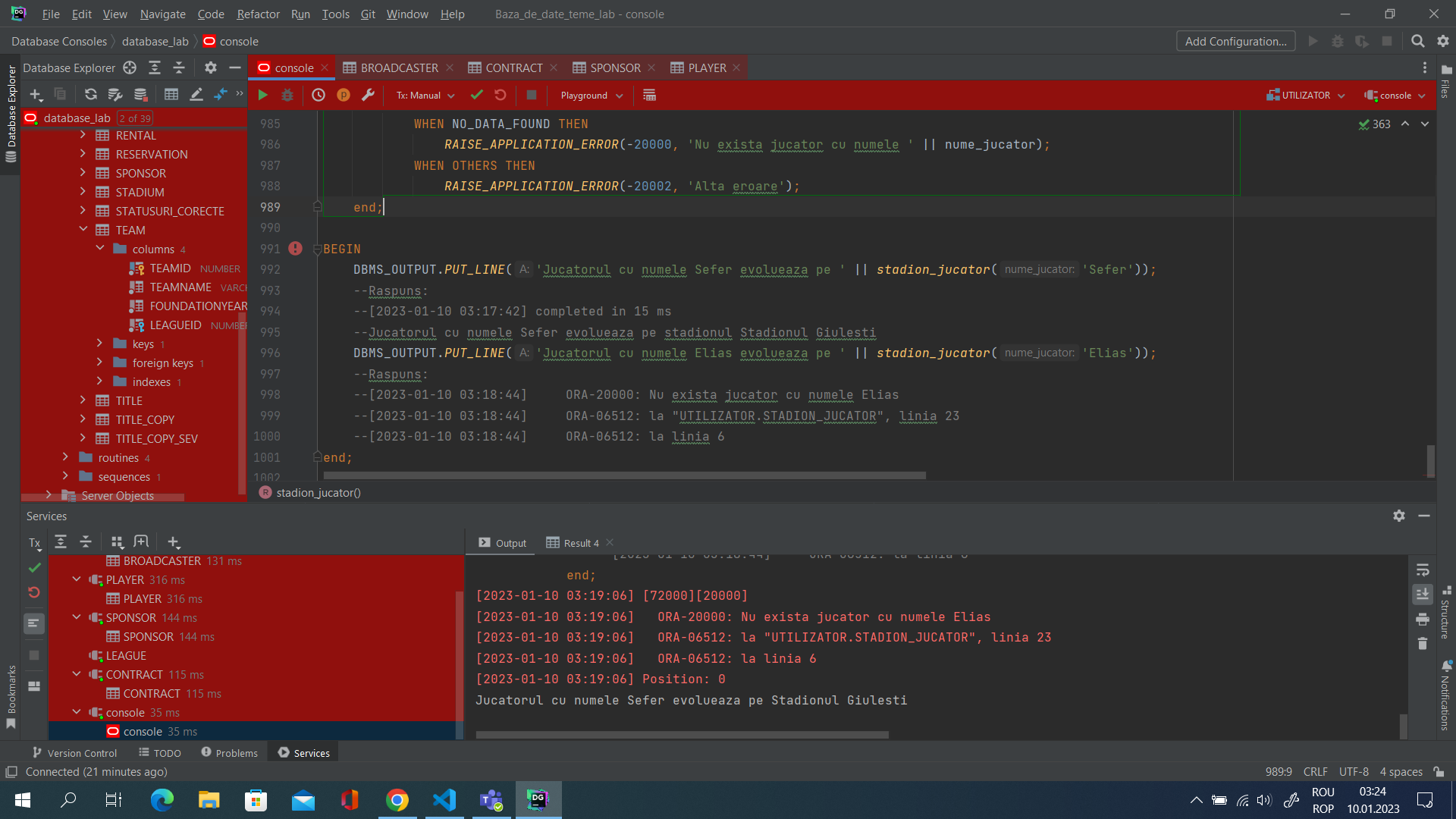
--Raspuns:

--[2023-01-10 03:18:44] ORA-20000: Nu exista jucator cu numele Elias

--[2023-01-10 03:18:44] ORA-06512: la "UTILIZATOR.STADION\_JUCATOR", linia 23

--[2023-01-10 03:18:44] ORA-06512: la linia 6

end;



1. **Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip procedura care sa utilizeze intr-o singura comanda SQL 5 tabele diferite. Tratati toate exceptiile caare pot aparea.**

**--Sa se afiseze numele, echipa, numarul de pe tricou si sponsorul unui jucator dat ca parametru, doar daca acesta a terminat pe locul 1 in campionatul din Romania si daca are o clauza de reziliere activa in contractul cu sponsorii sai.**

CREATE OR REPLACE PROCEDURE *afisare\_jucator\_exc*

(nume\_jucator Player.PlayerName%TYPE)

IS

CURSOR c1 is

select t.TeamID id\_echipa, t.TeamName nume\_echipa, Points puncte

from Team t, Ranking r

where t.TeamID = r.TeamID;

TYPE tip\_detalii\_echipa IS RECORD (

id\_echipa Team.TeamID%TYPE,

nume\_echipa Team.TeamName%TYPE,

puncte\_echipa Ranking.Points%TYPE );

TYPE tip\_detalii\_jucator IS RECORD(

id\_jucator Player.PlayerID%TYPE,

nume\_jucator Player.PlayerName%TYPE,

echipa\_jucator Team.TeamName%TYPE,

numar\_jucator Player.JerseyNumber%TYPE,

sponsor\_jucator Sponsor.SponsorName%TYPE,

clauza\_reziliere Contract.ClauzaReziliere%TYPE);

jucator\_par tip\_detalii\_jucator;

echipa\_castigatoare tip\_detalii\_echipa;

nr\_maxim\_puncte Ranking.Points%TYPE;

exceptie\_clauza EXCEPTION;

exceptie\_echipa EXCEPTION;

BEGIN

select *MAX*(Points)

into nr\_maxim\_puncte

from Ranking;

open c1;

LOOP

FETCH c1 into echipa\_castigatoare;

EXIT WHEN echipa\_castigatoare.puncte\_echipa = nr\_maxim\_puncte;

end loop;

select p.PlayerID, p.PlayerName, t.TeamName, p.JerseyNumber, s.SponsorName, c.ClauzaReziliere

into jucator\_par

from Player p

join Team t on (t.TeamID = p.TeamID)

join Deal d on (d.PlayerID = p.PlayerID)

join Contract c on (c.ContractID = d.ContractID)

join Sponsor s on (s.SponsorID = d.SponsorID)

where p.PlayerName = nume\_jucator;

if jucator\_par.clauza\_reziliere = 0 then

raise exceptie\_clauza;

end if;

if jucator\_par.echipa\_jucator <> echipa\_castigatoare.nume\_echipa then

raise exceptie\_echipa;

end if;

DBMS\_OUTPUT.*PUT\_LINE*('Jucatorul ' || jucator\_par.nume\_jucator || ' evolueaza pentru echipa ' || jucator\_par.echipa\_jucator || ', campioana Romaniei, poarta numarul ' || jucator\_par.numar\_jucator || ' si este sponsorizat de ' || jucator\_par.sponsor\_jucator);

EXCEPTION

WHEN exceptie\_clauza THEN

*RAISE\_APPLICATION\_ERROR*(-20003, 'Jucatorul ' || nume\_jucator || ' nu are o clauza activa.');

WHEN exceptie\_echipa THEN

*RAISE\_APPLICATION\_ERROR*(-20003, 'Jucatorul ' || nume\_jucator || ' nu joaca pentru echipa castigatoare din Romania.');

WHEN NO\_DATA\_FOUND THEN

*RAISE\_APPLICATION\_ERROR*(-20000, 'Nu exista jucator cu numele ' || nume\_jucator);

WHEN TOO\_MANY\_ROWS THEN

*RAISE\_APPLICATION\_ERROR*(-20001, 'Sunt mai multi jucatori cu numele ' || nume\_jucator);

WHEN OTHERS THEN

*RAISE\_APPLICATION\_ERROR*(-20002, 'Alta eroare');

end;

BEGIN

*AFISARE\_JUCATOR\_EXC*('Susic');

--[2023-01-10 04:37:24] ORA-20003: Jucatorul Susic nu are o clauza activa.

--[2023-01-10 04:37:24] ORA-06512: la "UTILIZATOR.AFISARE\_JUCATOR\_EXC", linia 58

*AFISARE\_JUCATOR\_EXC*('Manea');

--Jucatorul Manea evolueaza pentru echipa CFR Cluj, campioana Romaniei, poarta numarul 4 si este sponsorizat de Umbro

*AFISARE\_JUCATOR\_EXC*('Elias');

--[2023-01-10 04:37:54] ORA-20000: Nu exista jucator cu numele Elias

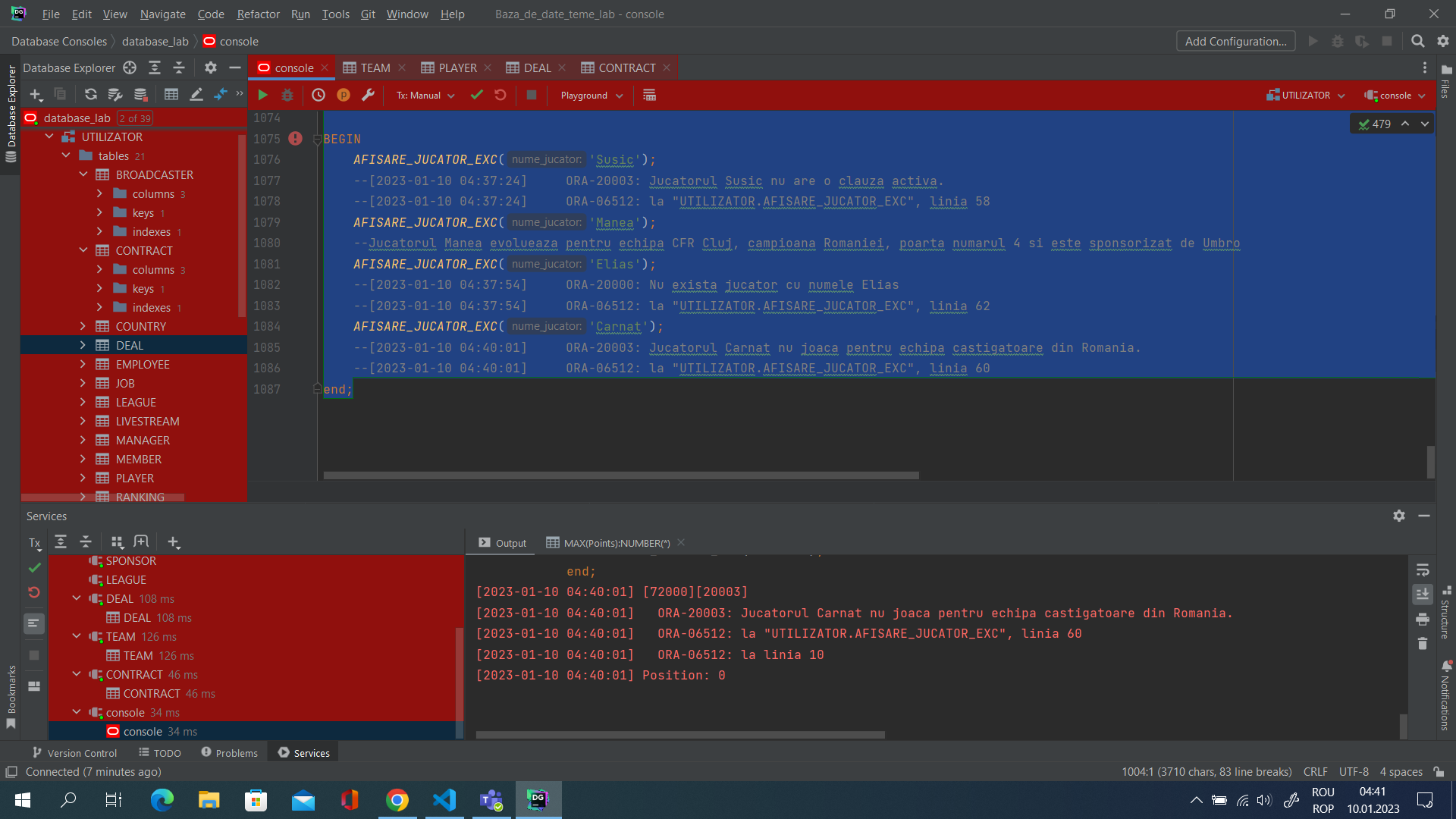
--[2023-01-10 04:37:54] ORA-06512: la "UTILIZATOR.AFISARE\_JUCATOR\_EXC", linia 62

*AFISARE\_JUCATOR\_EXC*('Carnat');

--[2023-01-10 04:40:01] ORA-20003: Jucatorul Carnat nu joaca pentru echipa castigatoare din Romania.

--[2023-01-10 04:40:01] ORA-06512: la "UTILIZATOR.AFISARE\_JUCATOR\_EXC", linia 60

end;

****