

Gestiunea bazei de date a circuitului profesionist de tenis masculin

1.1 Descrierea

Baza de date va putea oferi informații despre jucătorii de tenis: informații despre antrenori, sponsori, stilul de joc, dar și despre rezultatele acestora de la turneele la care participă. Va putea oferi informații despre diferite tipuri de turnee.

Un sportiv poate să își angajeze unul sau mai mulți antrenori. Un sponsor poate să sponsorizeze unul sau mai mulți jucători.

Se va reține un istoric al tuturor turneelelor desfășurate.

Se va reține un istoric cu rezultatele jucătorilor la turneele la care a participat, care va include ediția(anul) turneului, rezultat obținut și nr de puncte câștigat.

Turneele sunt de mai multe feluri: se pot juca pe mai multe tipuri de suprafețe(ciment, zgură, iarba), acorda un număr de puncte diferit sau numărul de participanți.

Unii jucători participă doar la proba de simplu, alții doar la proba de dublu.

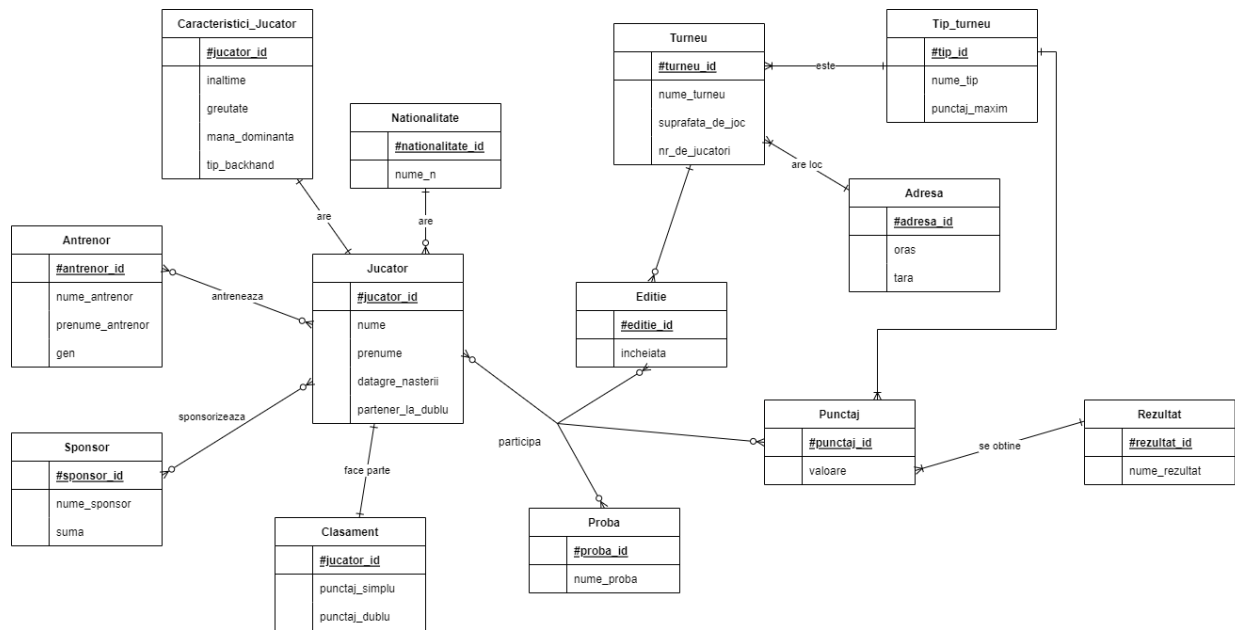
Vor fi clasamente formate pentru cele două tipuri de probe(simplu, dublu) formate din acumularea punctajelor obținute în decurs de un an. După ce s-a finalizat un turneu, clasamentul se va actualiza automat cu un trigger: vor fi eliminate punctele obținute de la ediția precedentă și vor fi adăugate punctele obținute de la ediția curentă.

Exemplu. Roger Federer are în prezent 2000 de puncte și a obținut anul trecut la Wimbledon 1000 puncte, iar anul acesta la Wimbledon a reușit să câștige doar 500 de puncte. Rezultă că la finalul turneului, după actualizarea clasamentului, va avea doar 1500 de puncte.

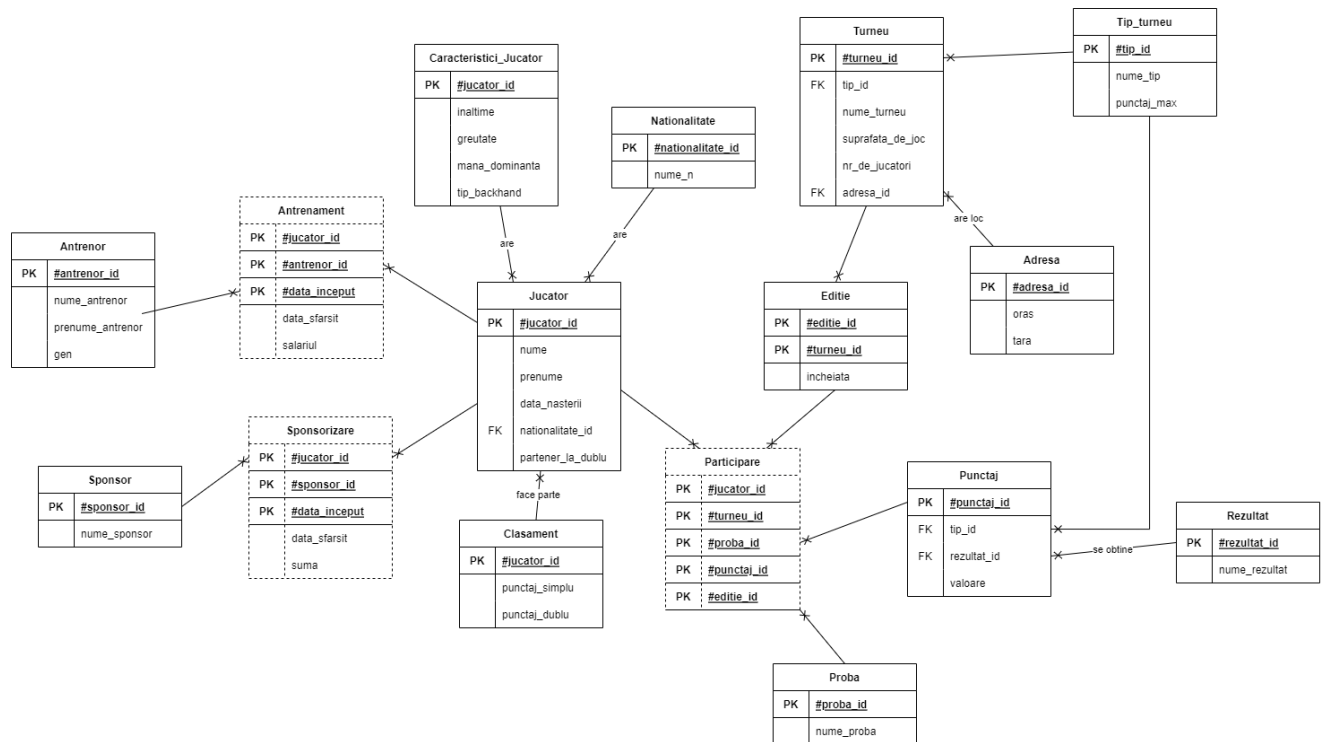
1.2 Restricții

- Un sportiv trebuie să aibă o singură naționalitate
- Un turneu trebuie să aibă loc la o singură adresă.

2. Diagrama entitate-relație



3. Diagrama conceptuală



4. Crearea tabelor

create table ANTRENOR

```
(antrenor_id number(5),  
nume_antrenor varchar2(25) constraint antrenor_nume_nn not null,  
prenume_antrenor varchar2(25) constraint antrenor_prenume_nn not null,  
gen char(1) constraint antrenor_gen_nn not null,  
constraint antrenor_genul check(gen = 'M' or gen = 'F'),  
constraint antrenor_id_pk primary key(antrenor_id));
```

```
create table SPONSOR
```

```
(sponsor_id number(5),  
nume_sponsor varchar2(50) constraint sponsor_nume_nn not null,  
constraint sponsor_id_pk primary key(sponsor_id));
```

```
create table NATIONALITATE
```

```
(nationalitate_id number(3),  
nume_n varchar2(20) constraint nationalitate_nume_nn not null,  
constraint nationalitate_id_pk primary key(nationalitate_id));
```

```
create table JUCATOR
```

```
(jucator_id number(5),  
nume varchar2(25) constraint jucator_nume_nn not null,  
prenume varchar2(25) constraint jucator_prenume_nn not null,  
data_nasterii date constraint jucator_data_nasterii_nn not null,  
nationalitate_id number(5),  
partener_la_dublu number(5),  
constraint jucator_id_pk primary key(jucator_id),  
constraint jucator_nationalitate_id_fk foreign key(nationalitate_id)  
references NATIONALITATE(nationalitate_id));
```

```
create table ANTRENAMENT
(jucator_id number(5),
antrenor_id number(5),
data_inceput date,
data_sfarsit date,
salariul number(10),
constraint antrenament_pk primary key(jucator_id, antrenor_id, data_inceput),
constraint antrenament_jucator_id_fk foreign key(jucator_id)
references JUCATOR(jucator_id),
constraint antrenament_antrenor_id_fk foreign key(antrenor_id)
references ANTRENOR(antrenor_id));
```

```
create table SPONSORIZARE
(jucator_id number(5),
sponsor_id number(5),
data_inceput date,
data_sfarsit date,
suma number(10) constraint sponsorizare_suma_nn not null,
constraint sponsorizare_pk primary key(jucator_id, sponsor_id, data_inceput),
constraint sponsorizare_jucator_id_fk foreign key(jucator_id)
references JUCATOR(jucator_id),
constraint sponsorizare_sponsor_id_fk foreign key(sponsor_id)
references SPONSOR(sponsor_id));
```

```
create table CARACTERISTICI_JUCATOR
(jucator_id number(5),
inaltime number(3) constraint inaltime_nn not null,
greutate number(3) constraint greutate_nn not null,
```

```
mana_dominanta varchar2(10) constraint mana_dominanta_nn not null,  
tip_backhand varchar2(20) constraint tip_backhand_nn not null,  
constraint cj_jucator_id_pk primary key(jucator_id),  
constraint cj_jucator_id_fk foreign key(jucator_id)  
references JUCATOR(jucator_id));
```

```
create table CLASAMENT  
(jucator_id number(5),  
punctaj_simplu number(5),  
punctaj_dublu number(5),  
constraint c_jucator_id_pk primary key(jucator_id),  
constraint c_jucator_id_fk foreign key(jucator_id)  
references JUCATOR(jucator_id));
```

```
create table ADRESA  
(adresa_id number(5),  
oras varchar2(25) constraint adresa_oras_nn not null,  
tara varchar2(25) constraint tara_oras_nn not null,  
constraint adresa_id_pk primary key(adresa_id));
```

```
create table TIP_TURNEU  
(tip_id number(2),  
nume_tip varchar2(25) constraint tt_nume_nn not null,  
punctaj_max number(4) constraint tt_punctaj_max_nn not null,  
constraint tip_turneu_pk primary key(tip_id));
```

```
create table TURNEU  
(turneu_id number(5),
```

```
tip_id number(2),
nume_turneu varchar2(25) constraint turneu_nume_nn not null,
suprafata_de_joc varchar2(6) constraint turneu_sj_nn not null,
nr_de_jucatori number(3) constraint turneu_nj_nn not null,
adresa_id number(5),
constraint turneu_id_pk primary key(turneu_id),
constraint turneu_tip_id_fk foreign key(tip_id)
references TIP_TURNEU(tip_id),
constraint turneu_adresa_id_fk foreign key(adresa_id)
references ADRESA(adresa_id));
```

```
create table EDITIE
(editie_id number(4),
turneu_id number(4),
incheiata number(1) constraint incheiata_nn not null,
constraint editie_pk primary key(editie_id, turneu_id),
constraint editie_turneu_id_fk foreign key(turneu_id)
references TURNEU(turneu_id));
```

```
create table PROBA
(proba_id number(1),
nume_proba varchar2(6) constraint proba_nume_nn not null,
constraint proba_pk primary key(proba_id),
constraint probe check(nume_proba = 'simplu' or nume_proba = 'dublu'));
```

```
create table REZULTAT
(rezultat_id number(1),
nume_rezultat varchar2(20) constraint rezultat_nume_nn not null,
```

```
constraint rezultat_pk primary key(rezultat_id));
```

```
create table PUNCTAJ  
(punctaj_id number(2),  
tip_id number(2),  
rezultat_id number(1),  
valoare number(4) constraint punctaj_valoare_nn not null,  
constraint punctaj_pk primary key(punctaj_id),  
constraint punctaj_tip_id_fk foreign key(tip_id)  
references TIP_TURNEU(tip_id),  
constraint punctaj_rezultat_id_fk foreign key(rezultat_id)  
references REZULTAT(rezultat_id));
```

```
create table PARTICIPARE  
(jucator_id number(5),  
turneu_id number(5),  
proba_id number(1),  
punctaj_id number(2),  
editie_id number(4),  
constraint participare_pk primary key(jucator_id, turneu_id, proba_id, punctaj_id, editie_id),  
constraint participare_jucator_id_fk foreign key(jucator_id)  
references JUCATOR(jucator_id),  
constraint participare_editie_fk foreign key(editie_id, turneu_id)  
references EDITIE(editie_id, turneu_id),  
constraint participare_proba_id_fk foreign key(proba_id)  
references PROBA(proba_id),  
constraint participare_punctaj_id_fk foreign key(punctaj_id)  
references PUNCTAJ(punctaj_id));
```

5. Inserare de date coerente

create sequence seq_jucator

increment by 1

start with 1

maxvalue 99999

nocycle

nocache;

insert into ANTRENOR

values(1, 'Ferrero', 'Juan Carlos', 'M');

insert into ANTRENOR

values(2, 'Becker', 'Boris', 'M');

insert into ANTRENOR

values(3, 'Ivanisevic', 'Goran', 'M');

insert into ANTRENOR

values(4, 'Ljubicic', 'Ivan', 'M');

insert into ANTRENOR

values(5, 'Nadal', 'Antonio', 'M');

insert into SPONSOR

values(1, 'Nike');

insert into SPONSOR

values(2, 'Adidas');

insert into SPONSOR

values(3, 'Puma');

insert into SPONSOR

values(4, 'Lacoste');

insert into SPONSOR

values(5, 'Polo');

insert into NATIONALITATE

values(1, 'Român');

insert into NATIONALITATE

values(2, 'Spaniol');

insert into NATIONALITATE

values(3, 'Elve?ian');

insert into NATIONALITATE

values(4, 'Brazilian');

insert into NATIONALITATE

values(5, 'Sârb');

insert into JUCATOR

values(seq_jucator.nextval, 'Federer', 'Roger', to_date('08-08-1981', 'dd-mm-yyyy'), 3, null);

insert into JUCATOR

values(seq_jucator.nextval, 'Nadal', 'Rafael', to_date('03-06-1986', 'dd-mm-yyyy'), 2, 5);

insert into JUCATOR

values(seq_jucator.nextval, 'Djokovic', 'Novak', to_date('22-05-1987', 'dd-mm-yyyy'), 5, null);

insert into JUCATOR

values(seq_jucator.nextval, 'Wawrinka', 'Stan', to_date('28-03-1985', 'dd-mm-yyyy'), 3, null);

insert into JUCATOR

values(seq_jucator.nextval, 'Alcaraz', 'Carlos', to_date('05-05-2003', 'dd-mm-yyyy'), 2, 2);

insert into CARACTERISTICI_JUCATOR

values(1, 185, 85, 'dreapta', 'cu o mana');

insert into CARACTERISTICI_JUCATOR

```
values(2, 185, 85, 'stanga', 'cu doua maini');  
insert into CARACTERISTICI_JUCATOR  
values(3, 188, 77, 'dreapta', 'cu doua maini');  
insert into CARACTERISTICI_JUCATOR  
values(4, 183, 81, 'dreapta', 'cu o mana');  
insert into CARACTERISTICI_JUCATOR  
values(5, 185, 72, 'dreapta', 'cu doua maini');
```

```
insert into CLASAMENT  
values(1, 4000, null);  
insert into CLASAMENT  
values(2, 5000, 2500);  
insert into CLASAMENT  
values(3, 10000, null);  
insert into CLASAMENT  
values(4, 2000, null);  
insert into CLASAMENT  
values(5, 3500, 3500);
```

```
insert into PROBA  
values(1, 'simplu');  
insert into PROBA  
values(2, 'dublu');
```

```
insert into TIP_TURNEU  
values(1, 'Grand Slam', 2000);  
insert into TIP_TURNEU  
values(2, 'ATP Masters', 1000);
```

```
insert into TIP_TURNUEU
values(3, 'ATP 500 Series', 500);
insert into TIP_TURNUEU
values(4, 'ATP 250 Series', 250);
insert into TIP_TURNUEU
values(5, 'ATP Finals', 1500);
```

```
insert into ADRESA
values(1, 'Paris', 'Franta');
insert into ADRESA
values(2, 'Londra', 'Anglia');
insert into ADRESA
values(3, 'Roma', 'Italia');
insert into ADRESA
values(4, 'Madrid', 'Spania');
insert into ADRESA
values(5, 'Dubai', 'Emiratele Arabe Unite');
```

```
insert into TURNEU
values(1, 1, 'Roland Garros', 'zgura', 128, 1);
insert into TURNEU
values(2, 1, 'Wimbledon', 'iarba', 128, 2);
insert into TURNEU
values(3, 2, 'Italian Open', 'zgura', 56, 3);
insert into TURNEU
values(4, 2, 'Madrid Open', 'zgura', 56, 4);
insert into TURNEU
values(5, 3, 'Dubai Open', 'ciment', 32, 5);
```

```
insert into REZULTAT
values(1, 'castigator');
insert into REZULTAT
values(2, 'finalist');
insert into REZULTAT
values(3, 'semifinalist');
insert into REZULTAT
values(4, 'sfertfinalist');
insert into REZULTAT
values(5, 'runda 1');
```

```
insert into PUNCTAJ
values(1, 1, 1, 2000);
insert into PUNCTAJ
values(2, 1, 2, 1200);
insert into PUNCTAJ
values(3, 2, 1, 1000);
insert into PUNCTAJ
values(4, 2, 2, 600);
insert into PUNCTAJ
values(5, 3, 1, 500);
insert into PUNCTAJ
values(6, 3, 2, 300);
```

```
insert into ANTRENAMENT
values(1, 4, to_date('01-01-2015', 'dd-mm-yyyy'), null, 100000);
insert into ANTRENAMENT
```

```
values(1, 1, to_date('01-01-2014', 'dd-mm-yyyy'), to_date('31-12-2014', 'dd-mm-yyyy'), 80000);
insert into ANTRENAMENT
values(1, 3, to_date('01-01-2013', 'dd-mm-yyyy'), to_date('31-12-2013', 'dd-mm-yyyy'), 70000);
insert into ANTRENAMENT
values(2, 5, to_date('01-01-2008', 'dd-mm-yyyy'), null, null);
insert into ANTRENAMENT
values(2, 2, to_date('01-01-2010', 'dd-mm-yyyy'), to_date('31-12-2012', 'dd-mm-yyyy'), 55000);
insert into ANTRENAMENT
values(3, 2, to_date('01-01-2013', 'dd-mm-yyyy'), null, 120000);
insert into ANTRENAMENT
values(3, 3, to_date('01-01-2014', 'dd-mm-yyyy'), null, 120000);
insert into ANTRENAMENT
values(4, 4, to_date('01-01-2010', 'dd-mm-yyyy'), to_date('31-12-2014', 'dd-mm-yyyy'), 50000);
insert into ANTRENAMENT
values(5, 1, to_date('01-01-2015', 'dd-mm-yyyy'), null, 100000);
insert into ANTRENAMENT
values(5, 1, to_date('01-01-2013', 'dd-mm-yyyy'), to_date('31-12-2013', 'dd-mm-yyyy'), 20000);

insert into SPONSORIZARE
values(1, 1, to_date('01-01-2010', 'dd-mm-yyyy'), to_date('31-12-2020', 'dd-mm-yyyy'), 50000);
insert into SPONSORIZARE
values(1, 2, to_date('01-01-2021', 'dd-mm-yyyy'), null, 60000);
insert into SPONSORIZARE
values(2, 4, to_date('01-01-2010', 'dd-mm-yyyy'), null, 75000);
insert into SPONSORIZARE
values(2, 5, to_date('01-01-2015', 'dd-mm-yyyy'), null, 65000);
insert into SPONSORIZARE
values(3, 1, to_date('01-01-2010', 'dd-mm-yyyy'), to_date('31-12-2014', 'dd-mm-yyyy'), 40000);
```

```
insert into SPONSORIZARE
values(3, 3, to_date('01-01-2015', 'dd-mm-yyyy'), null, 80000);
insert into SPONSORIZARE
values(4, 1, to_date('01-01-2010', 'dd-mm-yyyy'), to_date('31-12-2014', 'dd-mm-yyyy'), 50000);
insert into SPONSORIZARE
values(4, 3, to_date('01-01-2015', 'dd-mm-yyyy'), null, 80000);
insert into SPONSORIZARE
values(5, 2, to_date('01-01-2020', 'dd-mm-yyyy'), null, 20000);
insert into SPONSORIZARE
values(5, 5, to_date('01-01-2020', 'dd-mm-yyyy'), null, 35000);
```

```
insert into EDITIE
values(2020, 2, 1);
insert into EDITIE
values(2021, 2, 1);
insert into EDITIE
values(2021, 1, 1);
insert into EDITIE
values(2021, 3, 1);
insert into EDITIE
values(2021, 4, 1);
insert into EDITIE
values(2022, 4, 1);
```

```
insert into PARTICIPARE
values(1, 2, 1, 2, 2020);
insert into PARTICIPARE
```

```

values(1, 2, 1, 1, 2021);
insert into PARTICIPARE
values(3, 2, 1, 1, 2020);
insert into PARTICIPARE
values(3, 1, 1, 2, 2021);
insert into PARTICIPARE
values(2, 1, 1, 1, 2021);
insert into PARTICIPARE
values(2, 1, 2, 1, 2021);
insert into PARTICIPARE
values(4, 3, 1, 4, 2021);
insert into PARTICIPARE
values(4, 4, 1, 3, 2021);
insert into PARTICIPARE
values(5, 4, 1, 3, 2022);
insert into PARTICIPARE
values(5, 1, 2, 1, 2021);

commit;

```

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

Pentru ediția unui turneu afișați toți antrenorii și sponsorii pe care i-a avut câștigătorul de la proba de simplu.

```

procedure      afisare_antrenori_sponsori(cod_turneu      editie.turneu_id%TYPE,      an
editie.editie_id%TYPE)

is

type vector_antrenor is varray(10) of antrenor%rowtype;

```

```

v_ant vector_antrenor := vector_antrenor();
type tablou_imbricat_sponsor is table of sponsor%rowtype;
ti_spo tablou_imbricat_sponsor := tablou_imbricat_sponsor();

cod_jucator jucator.jucator_id%TYPE;
begin
    select jucator_id into cod_jucator
    from participare natural join punctaj natural join rezultat
    where turneu_id = cod_turneu and editie_id = an and proba_id = 1 and nume_rezultat =
'castigator';
--
    select distinct antrenor_id, nume_antrenor, prenume_antrenor, gen
    bulk collect into v_ant
    from antrenament natural join antrenor
    where jucator_id = cod_jucator
    group by antrenor_id, nume_antrenor, prenume_antrenor, gen;

    dbms_output.put_line('Nr de antrenori: ' || v_ant.count);
    if v_ant.count != 0 then
        for i in 1..v_ant.count loop
            dbms_output.put_line(v_ant(i).prenume_antrenor || ' ' || v_ant(i).nume_antrenor);
        end loop;
    end if;

    dbms_output.new_line;

    select sponsor_id, nume_sponsor
    bulk collect into ti_spo
    from sponsorizare natural join sponsor

```



```
where jucator_id = cod_jucator
group by sponsor_id, nume_sponsor;

dbms_output.put_line('Nr de sponsori: ' || ti_spo.count);
if ti_spo.count != 0 then
    for i in ti_spo.first..ti_spo.last loop
        dbms_output.put_line(ti_spo(i).nume_sponsor);
    end loop;
end if;
exception
when no_data_found then
    dbms_output.put_line('Inca nu s-a jucat editia aceasta!');
when others then
    dbms_output.put_line('Alta eroare!');
end afisare_antrenori_sponsori;
```

```
124
125 begin
126     proiect.afisare_antrenori_sponsori(2, 2021);
127     proiect.afisare_antrenori_sponsori(1, 2025);
128 end;
129 /
```

Script Output x Query Result x

Task completed in 0.036 seconds

Nr de antrenori: 3
Juan Carlos Ferrero
Goran Ivanisevic
Ivan Ljubicic

Nr de sponsori: 2
Nike
Adidas
Inca nu s-a jucat editia aceasta!

PL/SQL procedure successfully completed.

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

Modificați suma cu care sunt sponsorizați jucătorii. Jucătorilor care au înălțimea peste 1.85 li se mărește sponsorizarea cu 10%, celorlalți cu 5%.

```
procedure modifica_sponsorizare
is
    type informatii_jucator is record(
        cod_jucator jucator.jucator_id%TYPE,
        inaltime caracteristici_jucator.inaltime%TYPE);

    cursor c1
    return informatii_jucator is
```

```

select jucator_id, inaltime
from jucator natural join caracteristici_jucator;

cursor c2(cod_jucator jucator.jucator_id%TYPE)
is
select *
from sponsorizare
where data_sfarsit is null and jucator_id = cod_jucator
for update of suma nowait;
begin
for i in c1 loop
for j in c2(i.cod_jucator) loop
if i.inaltime > 185 then
update sponsorizare
set suma = suma * 1.1
where current of c2;
else
update sponsorizare
set suma = suma * 1.05
where current of c2;
end if;
end loop;
end loop;
end modifica_sponsorizare;

```

```

161
162 select prenume, nume, suma, inaltime
163 from sponsorizare natural join jucator natural join caracteristici_jucator
164 where data_sfarsit is null;
165
166 execute proiect.modifica_sponsorizare;
167
168 rollback;

```

Script Output x Query Result x

SQL | All Rows Fetched: 7 in 0 seconds

	PRENUME	NUME	SUMA	INALTIME
1	Roger	Federer	60000	185
2	Rafael	Nadal	75000	185
3	Rafael	Nadal	65000	185
4	Novak	Djokovic	80000	188
5	Stan	Wawrinka	80000	183
6	Carlos	Alcaraz	20000	185
7	Carlos	Alcaraz	35000	185

```

165
166 execute proiect.modifica_sponsorizare;
167
168 rollback;

```

Script Output x Query Result x

Task completed in 0.043 seconds

Rollback complete.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.


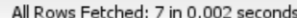
```

162 select prenume, nume, suma, inaltime
163 from sponsorizare natural join jucator natural join caracteristici_jucator
164 where data_sfarsit is null;
165
166 execute proiect.modifica_sponsorizare;
167
168 rollback;

```

Script Output x

Query Result x

	PRENUME	NUME	SUMA	INALTIME
1	Roger	Federer	63000	185
2	Rafael	Nadal	78750	185
3	Rafael	Nadal	68250	185
4	Novak	Djokovic	88000	188
5	Stan	Wawrinka	84000	183
6	Carlos	Alcaraz	21000	185
7	Carlos	Alcaraz	36750	185

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 SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Afișați numărul de turnee câștigate într-un sezon de către jucători de o naționalitate dată.

```
function turnee_castigate(an editie.editie_id%TYPE, nat nationalitate.nume_n%TYPE)
```

```
return number is
```

```
    cursor jucatori(cod_nat nationalitate.nationalitate_id%TYPE)
```

```
    return jucator%rowtype is
```

```
select *
```

```
from jucator
```

```
where nationalitate_id = cod_nat;
```

```
exceptie1 exception;
```

```
exceptie2 exception;
```

```
test1 number;
```

```
test2 number;
```

```

cod_nat nationalitate.nationalitate_id%TYPE;
raspuns number := 0;
cnt number := 0;
begin
    select count(*) into test1
    from participare
    where editie_id = an;
    if test1 = 0 then
        raise exceptie1;
    end if;

    select count(*) into test2
    from nationalitate
    where nume_n = nat;
    if test2 = 0 then
        raise exceptie2;
    end if;

    select nationalitate_id into cod_nat
    from nationalitate
    where nume_n = nat;

    for i in jucatori(cod_nat) loop
        select count(*) into cnt
        from participare natural join punctaj natural join rezultat
        where jucator_id = i.jucator_id and editie_id = an and nume_rezultat = 'castigator';
        raspuns := raspuns + cnt;
    end loop;

```

```

return raspuns;

exception

when exceptie1 then

    dbms_output.put_line('Nu s-au jucat turnee in acest an!');

    return -1;

when exceptie2 then

    dbms_output.put_line('Nationalitatea este gresita!');

    return -2;

when others then

    return -3;

end turnee_castigate;

```

```

124
125 begin
126     dbms_output.put_line(proiect.turnee_castigate(2021, 'Spaniol'));
127     dbms_output.put_line(proiect.turnee_castigate(2025, 'Spaniol'));
128     dbms_output.put_line(proiect.turnee_castigate(2021, 'cefaci'));
129 end;
130 /

```

Script Output x Query Result x

Task completed in 0.036 seconds

```

3
Nu s-au jucat turnee in acest an!
-1
Nationalitatea este gresita!
-2

PL/SQL procedure successfully completed.

```

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 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 afișeze ultimul an când un stângaci a câștigat un turneu de un tip dat.

```
function ultimul_an(tip tip_turneu.nume_tip%TYPE)
```

return number is

type vector_ani is varray(10) of editie.editie_id%TYPE;

ani vector_ani := vector_ani();

type vector_turnee is varray(10) of turneu.turneu_id%TYPE;

turnee vector_turnee := vector_turnee();

type vector_stangaci is varray(10) of jucator.jucator_id%TYPE;

stangaci vector_stangaci := vector_stangaci();

exceptie1 exception;

exceptie2 exception;

cod_tip tip_turneu.tip_id%TYPE;

cnt number;

begin

 select tip_id into cod_tip

 from tip_turneu

 where nume_tip = tip;

 select distinct(editie_id)

 bulk collect into ani

 from participare

 order by editie_id desc;

 if ani.count = 0 then

 raise exceptie1;

 end if;

 select jucator_id


```

bulk collect into stangaci
from caracteristici_jucator
where mana_dominanta = 'stanga';

if stangaci.count = 0 then
    raise exceptie2;
end if;

for i in 1..ani.count loop
    select distinct(turneu_id)
    bulk collect into turnee
    from participare natural join turneu
    where tip_id = cod_tip and editie_id = ani(i);

    if turnee.count != 0 then
        for j in 1..turnee.count loop
            for k in 1..stangaci.count loop
                select count(*)
                into cnt
                from participare natural join punctaj natural join rezultat
                where jucator_id = stangaci(k) and turneu_id = turnee(j) and editie_id = ani(i) and
nume_rezultat = 'castigator';

                if cnt > 0 then
                    return ani(i);
                end if;
            end loop;
        end loop;
    end loop;
end if;

```

```
end loop;

dbms_output.put_line('Niciun stangaci nu a castigat tipul asta de turneu!');
return -1;
exception
when no_data_found then
    dbms_output.put_line('Tipul turneului este gresit!');
    return -2;
when too_many_rows then
    dbms_output.put_line('Nu cred ca poate sa apara in procedura aceasta!');
    return -3;
when exceptie1 then
    dbms_output.put_line('Aparent nu s-o jucat tenis niciodata!');
    return -4;
when exceptie2 then
    dbms_output.put_line('Nu exista jucatori stangaci!');
    return -5;
when others then
    dbms_output.put_line('N-am prevazut aceasta eroare!');
end;
```

```
243 begin
244     dbms_output.put_line(proiect.ultimul_an('Grand Slam'));
245     dbms_output.put_line(proiect.ultimul_an('ATP Finals'));
246     dbms_output.put_line(proiect.ultimul_an('Vocea Romaniei'));
247 end;
248 /
```

Script Output x Query Result x

Task completed in 0.025 seconds

2021
Niciun stangaci nu a castigat tipul asta de turneu!
-1
Tipul turneului este gresit!
-2

PL/SQL procedure successfully completed.

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

În tabelul tip_turneu pot fi introduse maxim 5 înregistrări(există doar 5 tipuri de turnee oficiale).

Vom adăuga un trigger care să controleze acest lucru.

```
CREATE OR REPLACE TRIGGER max_tip
```

```
    BEFORE INSERT ON tip_turneu
```

```
DECLARE
```

```
    cnt number;
```

```
BEGIN
```

```
    select count(*)
```

```
    into cnt
```

```
    from tip_turneu;
```

```
    if cnt >= 5 then
```

```
        raise_application_error(-20001, 'Exista doar 5 tipuri de turnee oficiale!');
```

```
    end if;
```

```
END;
```

```
265
266 insert into tip_turneu
267 values(6, 'Turneu neoficial', 100);
268
```

Script Output x Query Result x

Task completed in 0.048 seconds

Error report -

ORA-20001: Exista doar 5 tipuri de turnee oficiale!

ORA-06512: at "NFJJUNIOR.MAX_TIP", line 9

ORA-04088: error during execution of trigger 'NFJJUNIOR.MAX_TIP'

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

Când se va termina un turneu, clasamentul la simplu se va actualiza automat: va scădea punctele câștigate anul trecut și v-a adăuga punctele câștigate anul acesta.

```
CREATE OR REPLACE TRIGGER actualizare_clasament
```

```
    AFTER INSERT OR UPDATE ON editie
```

```
    FOR EACH ROW
```

```
DECLARE
```

```
    type info is record(
```

```
        cod_jucator jucator.jucator_id%TYPE,
```

```
        valoare punctaj.valoare%TYPE);
```

```
    type vector is varray(10) of info;
```

```
    vechi vector := vector();
```

```
    nou vector := vector();
```

```
BEGIN
```

```
    IF :new.incheiata = 1 THEN
```

```
        select jucator_id, valoare
```

```
        bulk collect into vechi
```

```
        from participare natural join punctaj
```

```
        where editie_id = :new.editie_id - 1 and turneu_id = :new.turneu_id and proba_id = 1;
```

```

if vechi.count != 0 then
    for i in 1..vechi.count loop
        update clasament
        set clasament.punctaj_simplu = clasament.punctaj_simplu - vechi(i).valoare
        where jucator_id = vechi(i).cod_jucator;
    end loop;
end if;

select jucator_id, valoare
bulk collect into nou
from participare natural join punctaj
where editie_id = :new.editie_id and turneu_id = :new.turneu_id and proba_id = 1;
if nou.count != 0 then
    for i in 1..nou.count loop
        update clasament
        set clasament.punctaj_simplu = nvl(clasament.punctaj_simplu, 0) + nou(i).valoare
        where jucator_id = nou(i).cod_jucator;
    end loop;
end if;
END IF;
END;

```

```

289
290 select *
291 from clasament natural join jucator;
292
293 insert into editie
294 values(2022, 1, 0);
295
296 insert into participare
297 values(3, 1, 1, 1, 2022);
298
299 update editie
300 set incheiata = 1
301 where editie_id = 2022 and turneu_id = 1;
302
303 rollback;
304
305
306

```

Script Output x Query Result x

SQL All Rows Fetched: 5 in 0.002 seconds

JUCATOR_ID	PUNCTAJ_SIMPLU	PUNCTAJ_DUBLU	NUME	PRENUME	DATA_NASTERII	NATIONALITATE_ID	PARTENER_LA_DUBLU
1	1	4000	(null) Federer	Roger	08-AUG-81	3	(null)
2	2	5000	2500 Nadal	Rafael	03-JUN-86	2	5
3	3	10000	(null) Djokovic	Novak	22-MAY-87	5	(null)
4	4	2000	(null) Wawrinka	Stan	28-MAR-85	3	(null)
5	5	3500	3500 Alcaraz	Carlos	05-MAY-03	2	2

```

289
290 select *
291 from clasament natural join jucator;
292
293 insert into editie
294 values(2022, 1, 0);
295
296 insert into participare
297 values(3, 1, 1, 1, 2022);
298
299 update editie
300 set incheiata = 1
301 where editie_id = 2022 and turneu_id = 1;
302
303 rollback;
304
305
306

```

Script Output x Query Result x

SQL All Rows Fetched: 5 in 0.001 seconds

JUCATOR_ID	PUNCTAJ_SIMPLU	PUNCTAJ_DUBLU	NUME	PRENUME	DATA_NASTERII	NATIONALITATE_ID	PARTENER_LA_DUBLU
1	1	4000	(null) Federer	Roger	08-AUG-81	3	(null)
2	2	3000	2500 Nadal	Rafael	03-JUN-86	2	5
3	3	10800	(null) Djokovic	Novak	22-MAY-87	5	(null)
4	4	2000	(null) Wawrinka	Stan	28-MAR-85	3	(null)
5	5	3500	3500 Alcaraz	Carlos	05-MAY-03	2	2

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

Doar utilizatorul NFJJUNIOR poate să modifice schema.

Vom salva modificările în tabela istoric_modificări.

CREATE TABLE istoric_modificari

(utilizator varchar2(30),

nume_bd varchar2(50),

eveniment varchar2(20),

nume_obiect varchar2(30),

```
data date);
```

```
CREATE OR REPLACE TRIGGER adauga_istoric
```

```
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
```

```
BEGIN
```

```
    if user != 'NFJJUNIOR' then
```

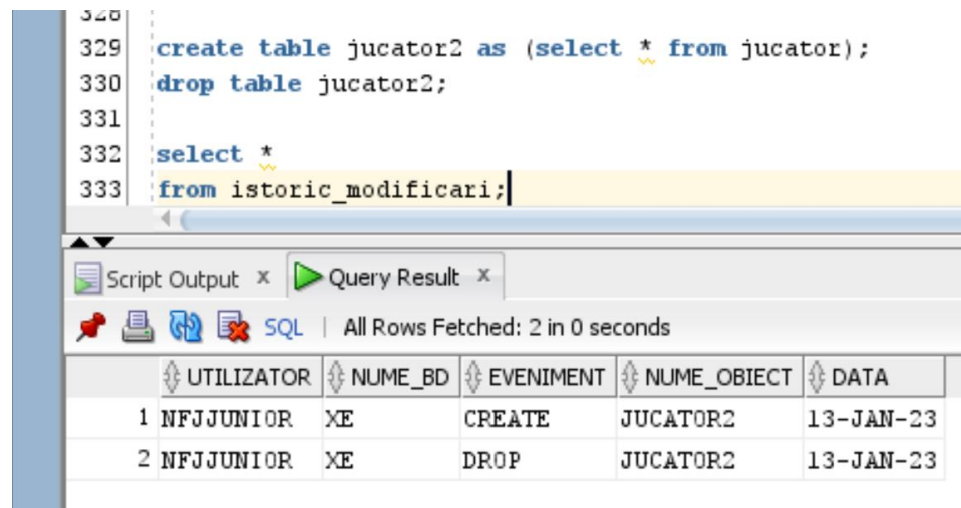
```
        raise_application_error(-20002, 'Nu esti NFJJUNIOR!');
```

```
    end if;
```

```
insert into istoric_modificari
```

```
values(sys.login_user, sys.database_name, sys.sysevent, sys.dictionary_obj_name, sysdate);
```

```
END;
```



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

```
328
329 create table jucator2 as (select * from jucator);
330 drop table jucator2;
331
332 select *
333 from istoric_modificari;
```

Below the script, the 'Query Result' tab is active, displaying the results of the last query. The status bar indicates 'All Rows Fetched: 2 in 0 seconds'.

	UTILIZATOR	NUME_BD	EVENIMENT	NUME_OBIECT	DATA
1	NFJJUNIOR	XE	CREATE	JUCATOR2	13-JAN-23
2	NFJJUNIOR	XE	DROP	JUCATOR2	13-JAN-23

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

Toate funcțiile pe care le-am folosit pana acum le-am integrat de la început într-un pachet.

```
CREATE OR REPLACE PACKAGE proiect IS
```

```
-- cerinta 6
```

```
    procedure    afisare_antrenori_sponsori(cod_turneu    editie.turneu_id%TYPE,    an
editie.editie_id%TYPE);
```

-- cerinta 7

procedure modifica_sponsorizare;

-- cerinta 8

function turnee_castigate(an editie.editie_id%TYPE, nat nationalitate.num_e_n%TYPE)

return number;

-- cerinta 9

function ultimul_an(tip tip_turneu.num_e_tip%TYPE)

return number;

END proiect;

/

-- Corpul pachetului proiect

CREATE OR REPLACE PACKAGE BODY proiect IS

-- cerinta 6

procedure afisare_antrenori_sponsori(cod_turneu editie.turneu_id%TYPE, an editie.editie_id%TYPE)

is

type vector_antrenor is varray(10) of antrenor%rowtype;

v_ant vector_antrenor := vector_antrenor();

type tablou_imbricat_sponsor is table of sponsor%rowtype;

ti_spo tablou_imbricat_sponsor := tablou_imbricat_sponsor();

cod_jucator jucator.jucator_id%TYPE;

begin

select jucator_id into cod_jucator

from participare natural join punctaj natural join rezultat

where turneu_id = cod_turneu and editie_id = an and proba_id = 1 and nume_rezultat = 'castigator';

```
select distinct antrenor_id, nume_antrenor, prenume_antrenor, gen  
bulk collect into v_ant  
from antrenament natural join antrenor  
where jucator_id = cod_jucator  
group by antrenor_id, nume_antrenor, prenume_antrenor, gen;
```

```
dbms_output.put_line('Nr de antrenori: ' || v_ant.count);  
if v_ant.count != 0 then  
    for i in 1..v_ant.count loop  
        dbms_output.put_line(v_ant(i).prenume_antrenor || ' ' || v_ant(i).nume_antrenor);  
    end loop;  
end if;
```

```
dbms_output.new_line;
```

```
select sponsor_id, nume_sponsor  
bulk collect into ti_spo  
from sponsorizare natural join sponsor  
where jucator_id = cod_jucator  
group by sponsor_id, nume_sponsor;
```

```
dbms_output.put_line('Nr de sponsori: ' || ti_spo.count);  
if ti_spo.count != 0 then  
    for i in ti_spo.first..ti_spo.last loop  
        dbms_output.put_line(ti_spo(i).nume_sponsor);  
    end loop;
```

```

        end if;
exception
    when no_data_found then
        dbms_output.put_line('Inca nu s-a jucat editia aceasta!');
    when others then
        dbms_output.put_line('Alta eroare!');
end afisare_antrenori_sponsori;

```

--cerinta 7

```

procedure modifica_sponsorizare
is
    type informatii_jucator is record(
        cod_jucator jucator.jucator_id%TYPE,
        inaltime caracteristici_jucator.inaltime%TYPE);

    cursor c1
    return informatii_jucator is
    select jucator_id, inaltime
    from jucator natural join caracteristici_jucator;

    cursor c2(cod_jucator jucator.jucator_id%TYPE)
    is
    select *
    from sponsorizare
    where data_sfarsit is null and jucator_id = cod_jucator
    for update of suma nowait;
begin
    for i in c1 loop

```

```

for j in c2(i.cod_jucator) loop
    if i.inaltime > 185 then
        update sponsorizare
        set suma = suma * 1.1
        where current of c2;
    else
        update sponsorizare
        set suma = suma * 1.05
        where current of c2;
    end if;
end loop;
end loop;
end modifica_sponsorizare;

```

-- cerinta 8

```

function turnee_castigate(an editie.editie_id%TYPE, nat nationalitate.nume_n%TYPE)

```

return number is

```

    cursor jucatori(cod_nat nationalitate.nationalitate_id%TYPE)

```

```

    return jucator%rowtype is

```

```

    select *

```

```

    from jucator

```

```

    where nationalitate_id = cod_nat;

```

```

exceptie1 exception;

```

```

exceptie2 exception;

```

```

test1 number;

```

```

test2 number;

```

```

cod_nat nationalitate.nationalitate_id%TYPE;
raspuns number := 0;
cnt number := 0;
begin
    select count(*) into test1
    from participare
    where editie_id = an;
    if test1 = 0 then
        raise exceptie1;
    end if;

    select count(*) into test2
    from nationalitate
    where nume_n = nat;
    if test2 = 0 then
        raise exceptie2;
    end if;

    select nationalitate_id into cod_nat
    from nationalitate
    where nume_n = nat;

    for i in jucatori(cod_nat) loop
        select count(*) into cnt
        from participare natural join punctaj natural join rezultat
        where jucator_id = i.jucator_id and editie_id = an and nume_rezultat = 'castigator';
        raspuns := raspuns + cnt;
    end loop;

```

```

return raspuns;
exception
    when exceptie1 then
        dbms_output.put_line('Nu s-au jucat turnee in acest an!');
        return -1;
    when exceptie2 then
        dbms_output.put_line('Nationalitatea este gresita!');
        return -2;
    when others then
        return -3;
end turnee_castigate;

-- cerinta 9
function ultimul_an(tip tip_turneu.nume_tip%TYPE)
return number is
    type vector_ani is varray(10) of editie.editie_id%TYPE;
    ani vector_ani := vector_ani();
    type vector_turnee is varray(10) of turneu.turneu_id%TYPE;
    turnee vector_turnee := vector_turnee();
    type vector_stangaci is varray(10) of jucator.jucator_id%TYPE;
    stangaci vector_stangaci := vector_stangaci();

    exceptie1 exception;
    exceptie2 exception;

    cod_tip tip_turneu.tip_id%TYPE;
    cnt number;

```

```
begin
    select tip_id into cod_tip
    from tip_turneu
    where nume_tip = tip;

    select distinct(editie_id)
    bulk collect into ani
    from participare
    order by editie_id desc;

    if ani.count = 0 then
        raise exceptie1;
    end if;

    select jucator_id
    bulk collect into stangaci
    from caracteristici_jucator
    where mana_dominanta = 'stanga';

    if stangaci.count = 0 then
        raise exceptie2;
    end if;

    for i in 1..ani.count loop
        select distinct(turneu_id)
        bulk collect into turnee
        from participare natural join turneu
        where tip_id = cod_tip and editie_id = ani(i);
```

```

if turnee.count != 0 then
    for j in 1..turnee.count loop
        for k in 1..stangaci.count loop
            select count(*)
            into cnt
            from participare natural join punctaj natural join rezultat
            where jucator_id = stangaci(k) and turneu_id = turnee(j) and editie_id = ani(i) and
nume_rezultat = 'castigator';

            if cnt > 0 then
                return ani(i);
            end if;
        end loop;
    end loop;
end if;
end loop;

dbms_output.put_line('Niciun stangaci nu a castigat tipul asta de turneu!');
return -1;
exception
when no_data_found then
    dbms_output.put_line('Tipul turneului este gresit!');
    return -2;
when too_many_rows then
    dbms_output.put_line('Nu cred ca poate sa apara in procedura aceasta!');
    return -3;
when exceptie1 then
    dbms_output.put_line('Aparent nu s-o jucat tenis niciodata!');

```

```

        return -4;
    when exceptie2 then
        dbms_output.put_line('Nu exista jucatori stangaci!');
        return -5;
    when others then
        dbms_output.put_line('N-am prevazut aceasta eroare!');
    end;
END proiect;
/

```

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 OR REPLACE PACKAGE bonus IS

```

    type juc is record(
        prenume jucator.prenume%TYPE,
        nume jucator.nume%TYPE);

    type finala is record(
        nume turneu.nume_turneu%TYPE,
        an editie.editie_id%TYPE,
        prenume_finalist1 jucator.prenume%TYPE,
        nume_finalist1 jucator.nume%TYPE,
        prenume_finalist2 jucator.prenume%TYPE,
        nume_finalist2 jucator.nume%TYPE);

    cursor c_jucatori return jucator%rowtype;

```



```

-- cine e primul in clasament
procedure primul;

-- cine a jucat finala la proba de simplu a unei editii de turneu
procedure finala_turneu(ume turneu.ume_turneu%TYPE, an editie.editie_id%TYPE);

-- cate turnee a castigat un jucator
function campion(cod_jucator jucator.jucator_id%TYPE)
return number;

-- cati bani primeste din sponsorizari un jucator
function bani_sponsorizare(cod_jucator jucator.jucator_id%TYPE)
return number;

END bonus;

/

```

CREATE OR REPLACE PACKAGE BODY bonus IS

```

cursor c_jucatori
return jucator%rowtype is
select *
from jucator;

procedure primul
is
n juc;
begin
select * into n from
(select prenume, nume
from jucator natural join clasament
order by punctaj_simplu desc)

```

```
where rownum = 1;
```

```
dbms_output.put_line(n.prenume || ' ' || n.num);
```

```
end primul;
```

```
procedure finala_turneu(num turneu.num_turneu%TYPE, an editie.editie_id%TYPE)
```

```
is
```

```
  f finala;
```

```
  cod_turneu turneu.turneu_id%TYPE;
```

```
begin
```

```
  f.num := num;
```

```
  f.an := an;
```

```
  select turneu_id
```

```
  into cod_turneu
```

```
  from turneu
```

```
  where num_turneu = num;
```

```
  select prenume, num
```

```
  into f.prenume_finalist1, f.num_finalist1
```

```
  from participare natural join punctaj natural join rezultat natural join jucator
```

```
  where turneu_id = cod_turneu and editie_id = an and proba_id = 1 and num_rezultat =  
'castigator';
```

```
  select prenume, num
```

```
  into f.prenume_finalist2, f.num_finalist2
```

```
  from participare natural join punctaj natural join rezultat natural join jucator
```

```
  where turneu_id = cod_turneu and editie_id = an and proba_id = 1 and num_rezultat =  
'finalist';
```

```

    dbms_output.put_line(f.nume || ' ' || f.an);
    dbms_output.put_line(f.prenume_finalist1 || ' ' || f.nume_finalist1);
    dbms_output.put_line(f.prenume_finalist2 || ' ' || f.nume_finalist2);
exception
    when no_data_found then
        dbms_output.put_line('Ori anul e gresit ori numele turneului e gresit ori nu s-au gasit cei
doi finaliști!');
    when others then
        dbms_output.put_line('N-am prevazut aceasta eroare!');
end finala_turneu;

function campion(cod_jucator jucator.jucator_id%TYPE)
return number is
    raspuns number;
begin
    select count(*)
    into raspuns
    from participare natural join punctaj natural join rezultat
    where jucator_id = cod_jucator and nume_rezultat = 'castigator';

    return raspuns;
end campion;

function bani_sponsorizare(cod_jucator jucator.jucator_id%TYPE)
return number is
    raspuns number := 0;
begin
    select sum(suma)

```

into raspuns

from sponsorizare natural join jucator

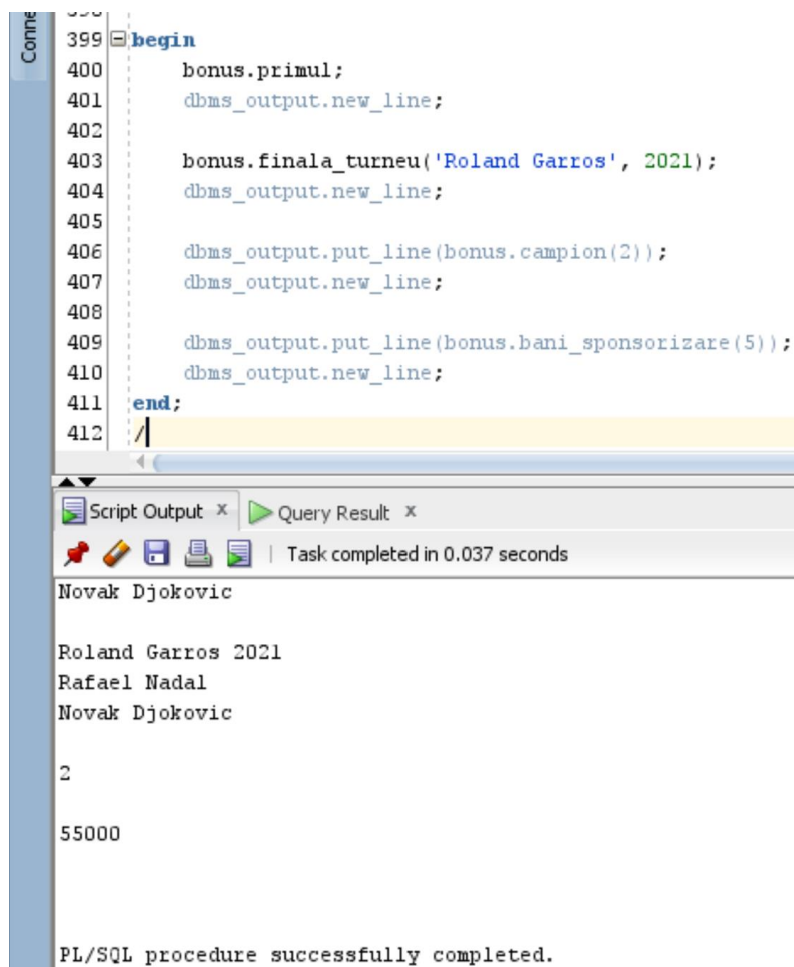
where jucator_id = cod_jucator and data_sfarsit is null;

return raspuns;

end bani_sponsorizare;

END bonus;

/



```
399 begin
400     bonus.primul;
401     dbms_output.new_line;
402
403     bonus.finala_turneu('Roland Garros', 2021);
404     dbms_output.new_line;
405
406     dbms_output.put_line(bonus.campion(2));
407     dbms_output.new_line;
408
409     dbms_output.put_line(bonus.bani_sponsorizare(5));
410     dbms_output.new_line;
411 end;
412 /
```

Script Output x Query Result x

Task completed in 0.037 seconds

Novak Djokovic

Roland Garros 2021

Rafael Nadal

Novak Djokovic

2

55000

PL/SQL procedure successfully completed.