**Gestiunea unei herghelii**

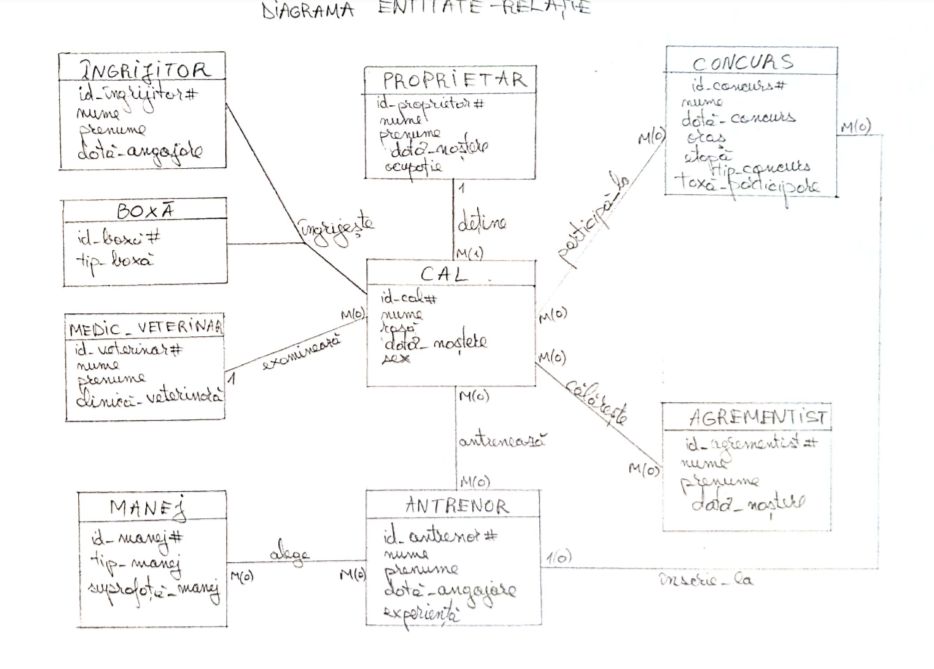
1. **Descrierea modelului real**

Acest model ofera posibilitatea iubitorilor de cai sa isi tina animalul in cele mai bune conditii, fiind hranit, antrenat si ingrijit de oameni interesati continuu de starea acestuia.

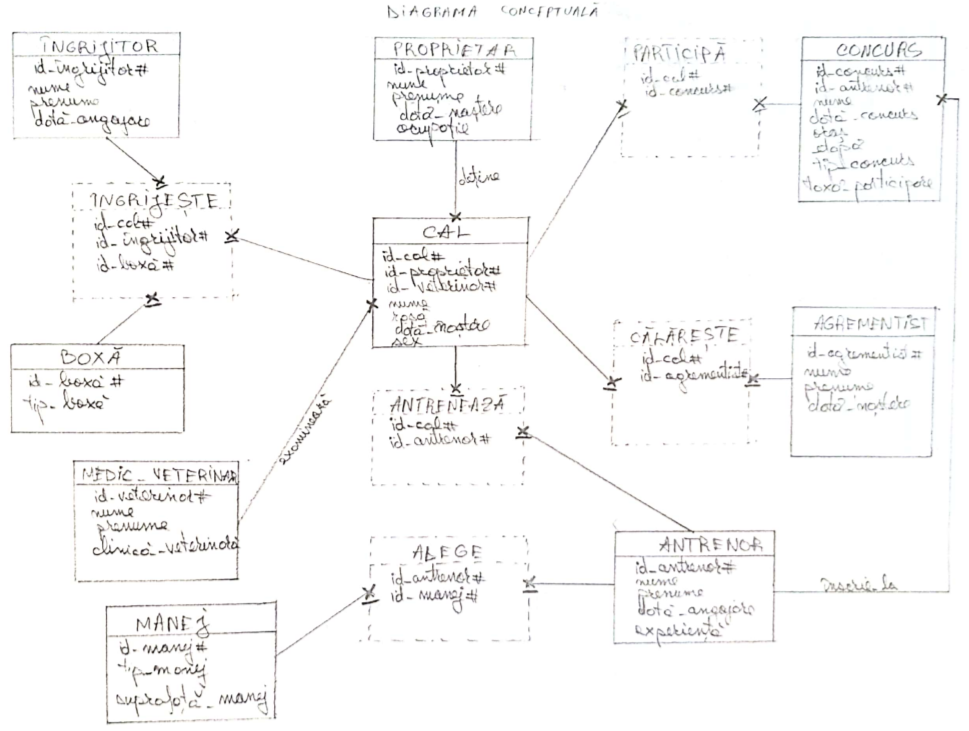
Fiecare cal are in programul lui zilnic ore de calarie/antrenament in locurile de manej, ore de agrement pentru oameni cu experienta in echitatie, dar care nu-si permit intretinerea unui asemenea animal, ore de odihna in padocurile din preajma herghelie, precum si orele garantate de odihna in boxe. Cu acordul proprietarilor, animalele pot fi inscrise in anumite concursuri de frumusete, in curse de viteza, de tractiune, curse de galop, sarituri de obstacole si in sporturi de echipa precum pollo. Fiecare cabalina va fi consultata de doua ori pe saptamana de catre un medic veterinar pentru a analiza problemele cauzate de oboseala si varsta, oferindu-i in cele din urma tratamentul de care are nevoie.

De asemenea, reproducerea se va face doar intre doi cai de aceeasi rasa in statiunile de monta amenajate de catre coordonatorul hergheliei.

1. **Diagrama entitate-relatie**



1. **Diagrama conceptuala**



1. **Definirea tabelelor**

create table **PROPRIETAR**(

id\_proprietar number(4) primary key,

nume varchar2(25) not null,

prenume varchar2(25) not null,

data\_nastere date,

ocupatie varchar2(20)

)

create table **AGREMENTIST**(

id\_agrementist number(5) primary key,

nume varchar2(25) not null,

prenume varchar2(25) not null,

data\_nastere date

)

create table **ANTRENOR**(

id\_antrenor number(4) primary key,

nume varchar2(25) not null,

prenume varchar2(25) not null,

data\_angajare date,

experienta number(3)

)

create table **MANEJ**(

id\_manej number(3) primary key,

tip\_manej varchar2(10) not null,

suprafata\_manej number(5)

)

create table **MEDIC\_VETERINAR**(

id\_veterinar number(5) primary key,

nume varchar(25) not null,

prenume varchar(25) not null,

clinica\_veterinara varchar(25)

)

create table **BOXA**(

id\_boxa number(4) primary key,

tip\_boxa varchar2(10)

)

create table **INGRIJITOR**(

id\_ingrijitor number(4) primary key,

nume varchar2(25) not null,

prenume varchar2(25) not null,

data\_angajare date

)

create table **CONCURS**(

id\_concurs number(4) primary key,

id\_antrenor number(4),

nume varchar2(25) not null,

data\_concurs date,

oras varchar2(20),

etapa varchar2(20),

tip\_concurs varchar2(20) not null,

taxa\_participare number(6)

);

alter table concurs

add constraint fk\_con\_ant foreign key(id\_antrenor) references antrenor(id\_antrenor);

create table **CAL**(

id\_cal number(4) primary key,

id\_proprietar number(4),

id\_veterinar number(5),

nume varchar2(25) not null,

rasa varchar2(25),

data\_nastere date,

sex varchar2(10) not null

)

alter table cal

add constraint fk\_prop\_cal foreign key(id\_proprietar) references proprietar(id\_proprietar);

alter table cal

add constraint fk\_vet\_cal foreign key(id\_veterinar) references medic\_veterinar(id\_veterinar);

create table **ALEGE**(

id\_antrenor number(4) references antrenor(id\_antrenor),

id\_manej number(3) references manej(id\_manej),

primary key(id\_antrenor, id\_manej)

)

create table **ANTRENEAZA**(

id\_cal number(4) references cal(id\_cal),

id\_antrenor number(4) references antrenor(id\_antrenor),

primary key(id\_cal, id\_antrenor)

)

create table **CALARESTE**(

id\_cal number(4) references cal(id\_cal),

id\_agrementist number(5) references agrementist(id\_agrementist) on delete cascade,

primary key (id\_cal, id\_agrementist)

)

create table **PARTICIPA**(

id\_cal number(4) references cal(id\_cal),

id\_concurs number(4) references concurs(id\_concurs),

primary key (id\_cal, id\_concurs)

)

create table **INGRIJESTE**(

id\_cal number(4) references cal(id\_cal),

id\_ingrijitor number(4) references ingrijitor(id\_ingrijitor),

id\_boxa number(4) references boxa(id\_boxa),

primary key(id\_cal, id\_ingrijitor, id\_boxa)

)

1. **Inserarea datelor in tabele**

**Proprietar:**

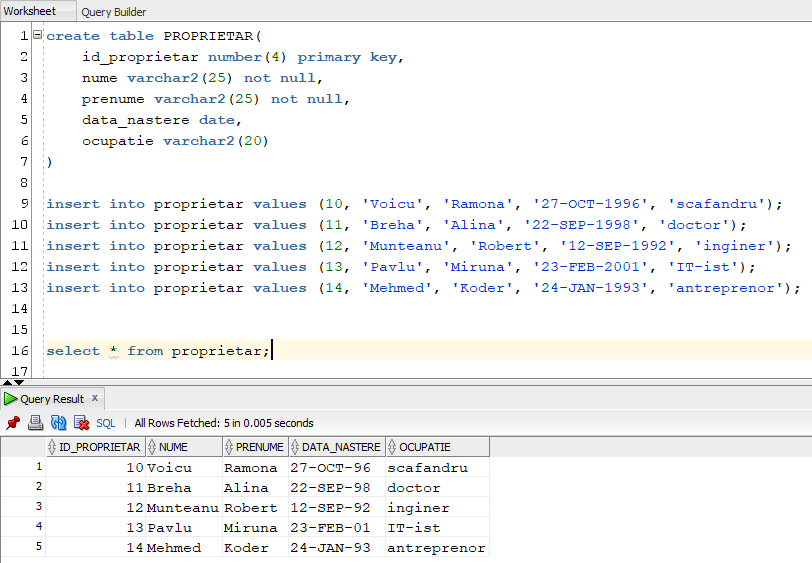
insert into proprietar values (10, 'Voicu', 'Ramona', '27-OCT-1996', 'scafandru');

insert into proprietar values (11, 'Breha', 'Alina', '22-SEP-1998', 'doctor');

insert into proprietar values (12, 'Munteanu', 'Robert', '12-SEP-1992', 'inginer');

insert into proprietar values (13, 'Pavlu', 'Miruna', '23-FEB-2001', 'IT-ist');

insert into proprietar values (14, 'Mehmed', 'Koder', '24-JAN-1993', 'antreprenor');



**Agrementist:**

insert into agrementist values (200, 'Nastase', 'Ramon', '23-OCT-2012');

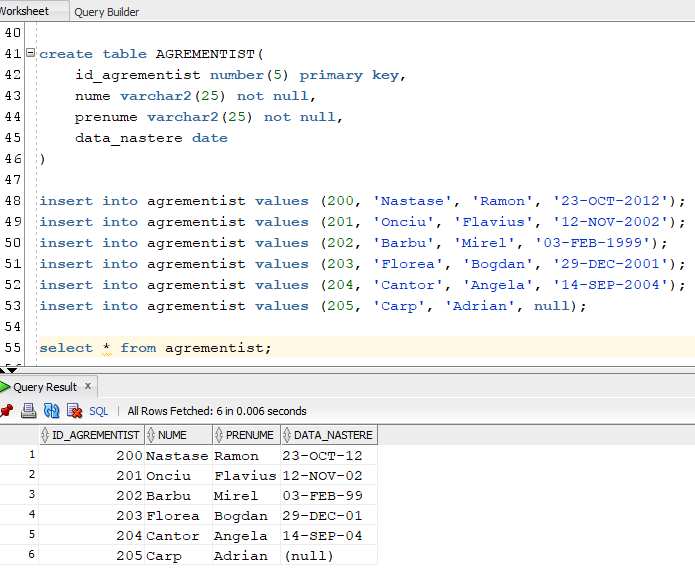
insert into agrementist values (201, 'Onciu', 'Flavius', '12-NOV-2002');

insert into agrementist values (202, 'Barbu', 'Mirel', '03-FEB-1999');

insert into agrementist values (203, 'Florea', 'Bogdan', '29-DEC-2001');

insert into agrementist values (204, 'Cantor', 'Angela', '14-SEP-2004');

insert into agrementist values (205, 'Carp', 'Adrian', null);



**Antrenor:**

insert into antrenor values (50, 'Mirunache', 'Angel', '07-SEP-2000', 14);

insert into antrenor values (51, 'Gavrila', 'Ion', '29-OCT-2004', 8);

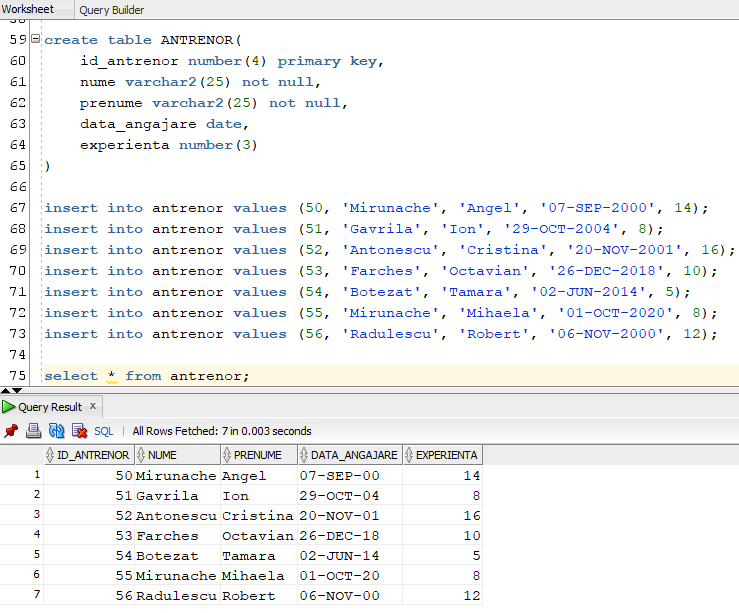
insert into antrenor values (52, 'Antonescu', 'Cristina', '20-NOV-2001', 16);

insert into antrenor values (53, 'Farches', 'Octavian', '26-DEC-2018', 10);

insert into antrenor values (54, 'Botezat', 'Tamara', '02-JUN-2014', 5);

insert into antrenor values (55, 'Mirunache', 'Mihaela', '01-OCT-2020', 8);

insert into antrenor values (56, 'Radulescu', 'Robert', '06-NOV-2000', 12);



**Manej:**

insert into manej values (20, 'obstacole', 200);

insert into manej values (21, 'liber', 160);

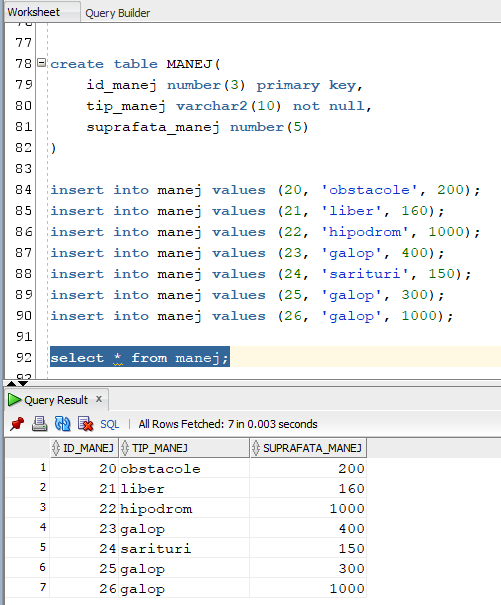
insert into manej values (22, 'hipodrom', 1000);

insert into manej values (23, 'galop', 400);

insert into manej values (24, 'sarituri', 150);

insert into manej values (25, 'galop', 300);

insert into manej values (26, 'galop', 1000);



**Medic\_veterinar:**

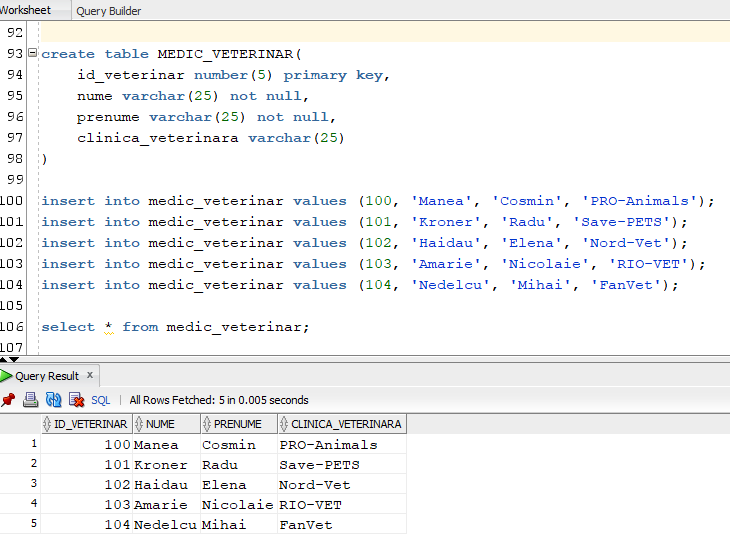
insert into medic\_veterinar values (100, 'Manea', 'Cosmin', 'PRO-Animals');

insert into medic\_veterinar values (101, 'Kroner', 'Radu', 'Save-PETS');

insert into medic\_veterinar values (102, 'Haidau', 'Elena', 'Nord-Vet');

insert into medic\_veterinar values (103, 'Amarie', 'Nicolaie', 'RIO-VET');

insert into medic\_veterinar values (104, 'Nedelcu', 'Mihai', 'FanVet');



**Boxa:**

insert into boxa values (1, 'mobila');

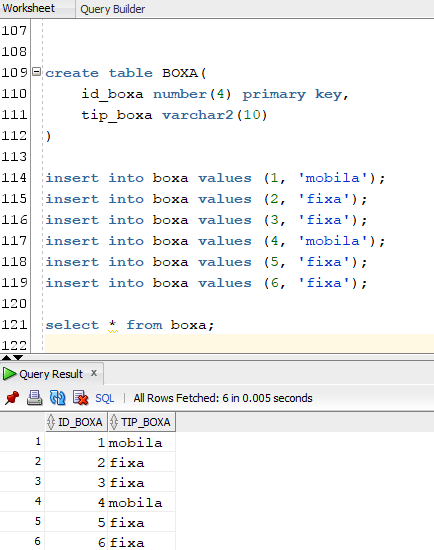
insert into boxa values (2, 'fixa');

insert into boxa values (3, 'fixa');

insert into boxa values (4, 'mobila');

insert into boxa values (5, 'fixa');

insert into boxa values (6, 'fixa');



**Ingrijitor:**

insert into ingrijitor values (30, 'Zargan', 'Florin', '23-MAR-2004');

insert into ingrijitor values (31, 'Culea', 'Sebastian', '12-APR-2010');

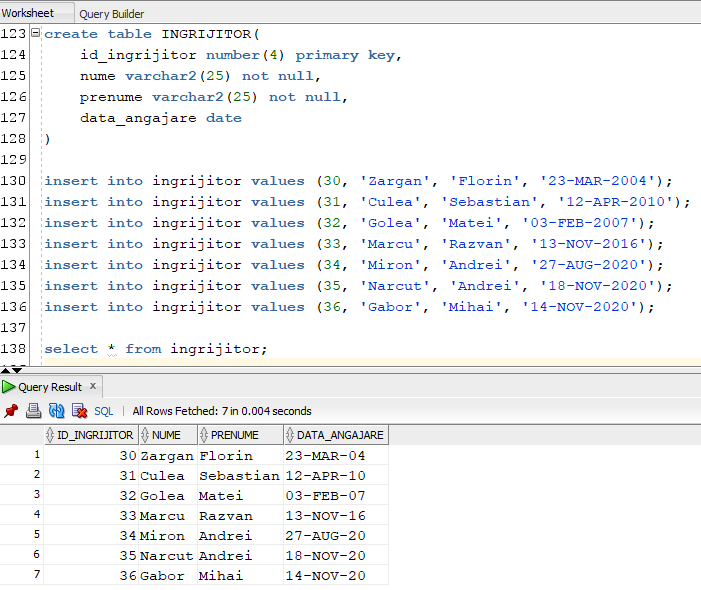
insert into ingrijitor values (32, 'Golea', 'Matei', '03-FEB-2007');

insert into ingrijitor values (33, 'Marcu', 'Razvan', '13-NOV-2016');

insert into ingrijitor values (34, 'Miron', 'Andrei', '27-AUG-2020');

insert into ingrijitor values (35, 'Narcut', 'Andrei', '18-NOV-2020');

insert into ingrijitor values (36, 'Gabor', 'Mihai', '14-NOV-2020');



**Concurs:**

insert into concurs values (1, 50, 'Carpatii sub copite', '23-AUG-2012', 'Brasov', 'nationala', 'tractiune', 500);

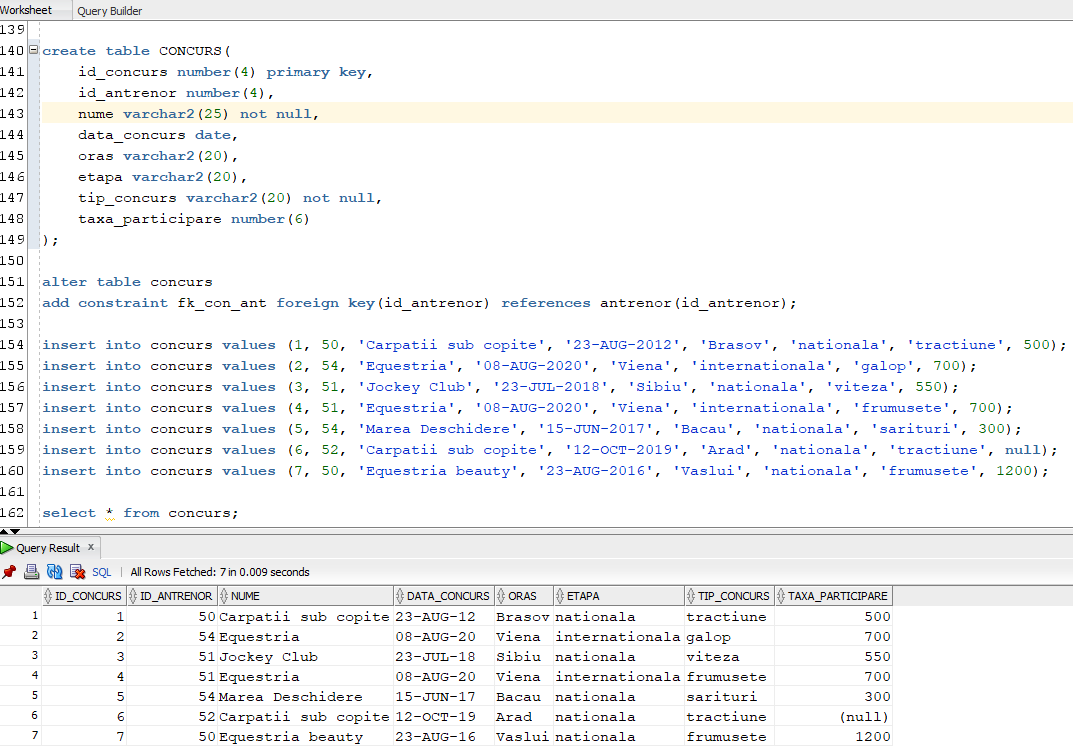
insert into concurs values (2, 54, 'Equestria', '08-AUG-2020', 'Viena', 'internationala', 'galop', 700);

insert into concurs values (3, 51, 'Jockey Club', '23-JUL-2018', 'Sibiu', 'nationala', 'viteza', 550);

insert into concurs values (4, 51, 'Equestria', '08-AUG-2020', 'Viena', 'internationala', 'frumusete', 700);

insert into concurs values (5, 54, 'Marea Deschidere', '15-JUN-2017', 'Bacau', 'nationala', 'sarituri', 300);

insert into concurs values (6, 52, 'Carpatii sub copite', '12-OCT-2019', 'Arad', 'nationala', 'tractiune', null);

insert into concurs values (7, 50, 'Equestria beauty', '23-AUG-2016', 'Vaslui', 'nationala', 'frumusete', 1200);

**Cal:**

insert into cal values (1, 11, 100, 'Cenai', 'Thoroughbred', '23-SEP-2015', 'femela');

insert into cal values (2, 14, 103, 'Russell', 'Cal Arab', '02-JAN-2001', 'mascul');

insert into cal values (3, 13, 102, 'Kira', 'Frizon','30-DEC-2012', 'femela');

insert into cal values (4, 14, 104, 'Maolo', 'Appaloosa', '19-NOV-2014', 'mascul');

insert into cal values (5, 10, 100, 'Pilak', 'Shire', '27-JUL-2015', 'mascul');

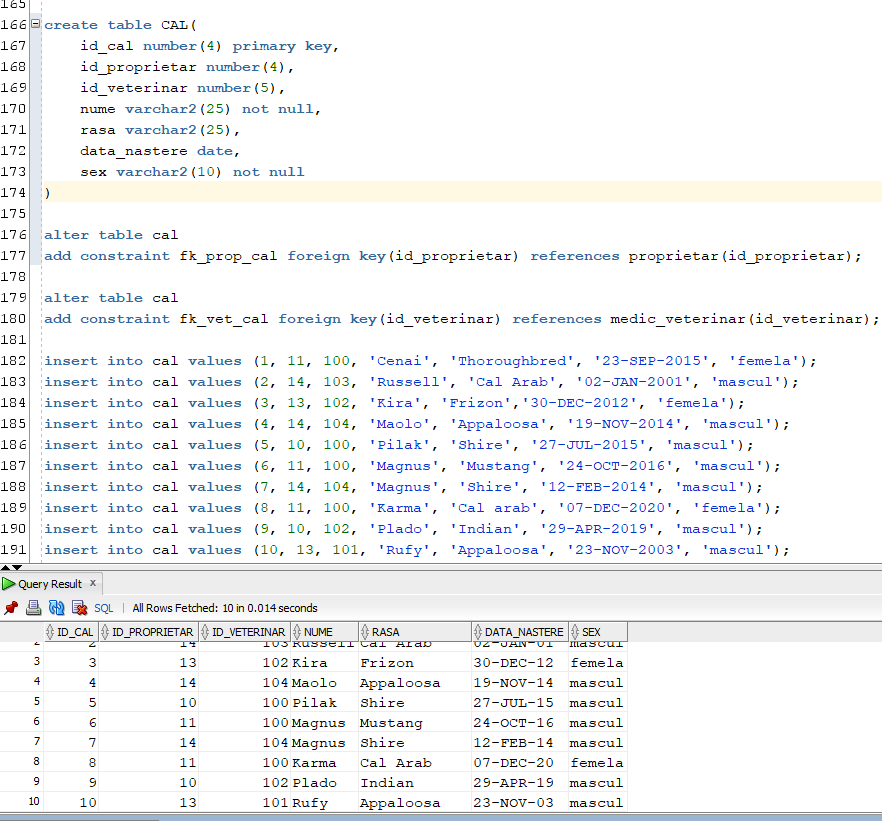
insert into cal values (6, 11, 100, 'Magnus', 'Mustang', '24-OCT-2016', 'mascul');

insert into cal values (7, 14, 104, 'Magnus', 'Shire', '12-FEB-2014', 'mascul');

insert into cal values (8, 11, 100, 'Karma', 'Cal arab', '07-DEC-2020', 'femela');

insert into cal values (9, 10, 102, 'Plado', 'Indian', '29-APR-2019', 'mascul');

insert into cal values (10, 13, 101, 'Rufy', 'Appaloosa', '23-NOV-2003', 'mascul');



**Alege:**

insert into alege values (50,20);

insert into alege values (52,23);

insert into alege values (54,24);

insert into alege values (51,22);

insert into alege values (53,20);

insert into alege values (53,23);

insert into alege values (50,24);

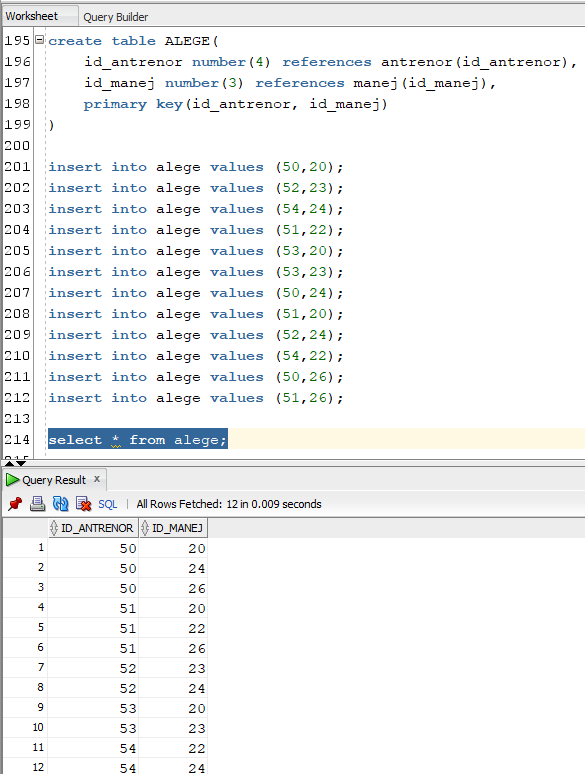
insert into alege values (51,20);

insert into alege values (52,24);

insert into alege values (54,22);

insert into alege values (50,26);

insert into alege values (51,26);



**Antreneaza:**

insert into antreneaza values (1,51);

insert into antreneaza values (2,54);

insert into antreneaza values (3,51);

insert into antreneaza values (1,54);

insert into antreneaza values (3,54);

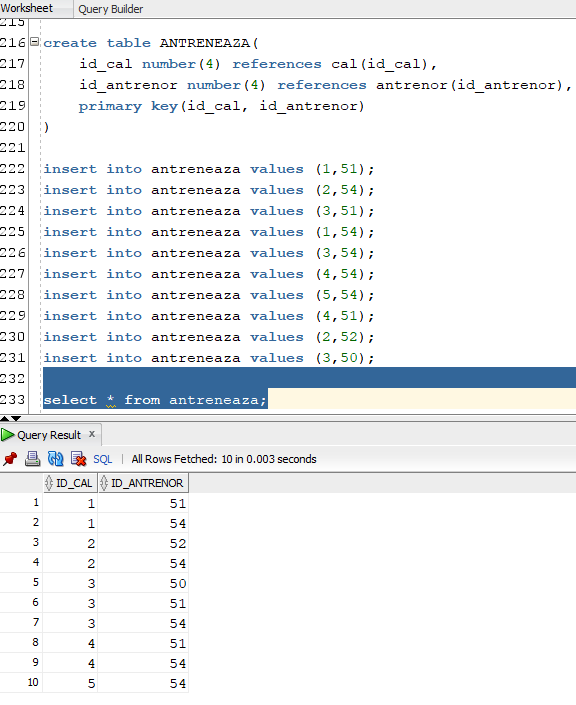
insert into antreneaza values (4,54);

insert into antreneaza values (5,54);

insert into antreneaza values (4,51);

insert into antreneaza values (2,52);

insert into antreneaza values (3,50);



**Participa:**

insert into participa values (1,4);

insert into participa values (2,3);

insert into participa values (3,4);

insert into participa values (1,5);

insert into participa values (3,2);

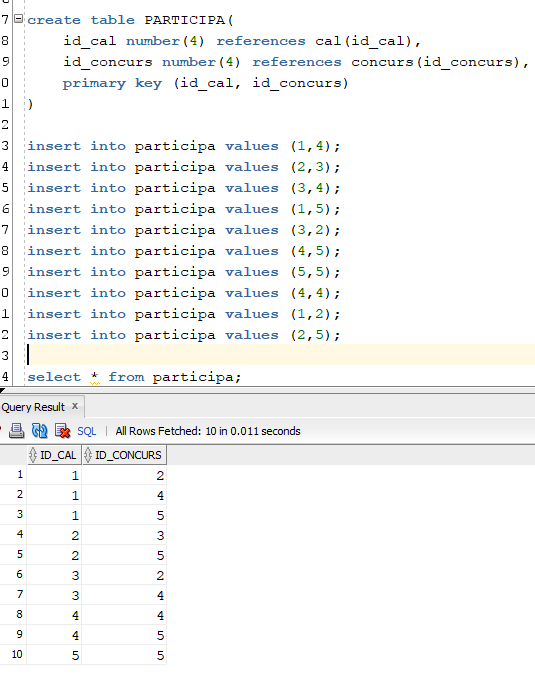
insert into participa values (4,5);

insert into participa values (5,5);

insert into participa values (4,4);

insert into participa values (1,2);

insert into participa values (2,5);



**Calareste:**

insert into calareste values (1,200);

insert into calareste values (2,204);

insert into calareste values (3,200);

insert into calareste values (4,201);

insert into calareste values (5,203);

insert into calareste values (1,203);

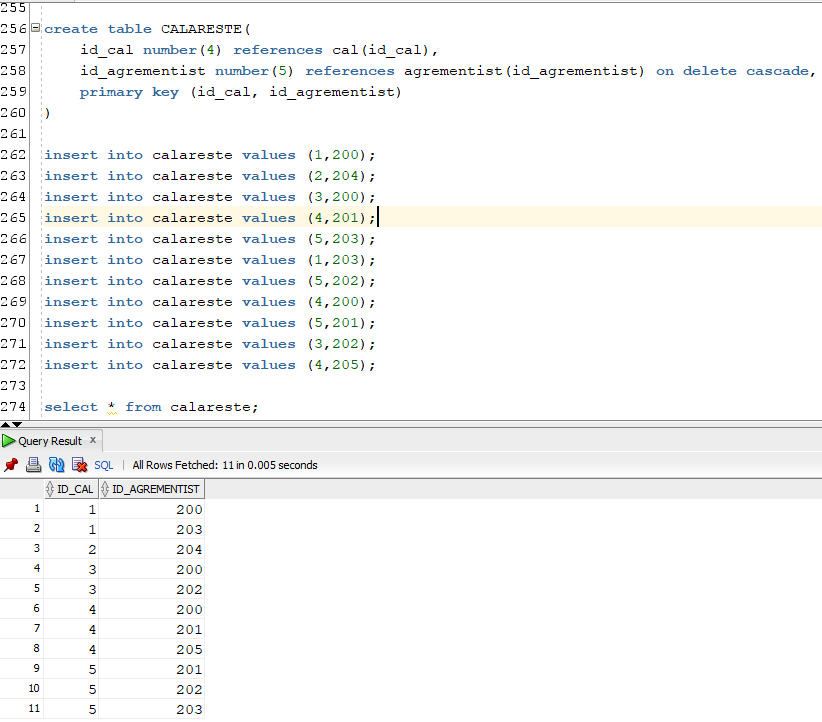
insert into calareste values (5,202);

insert into calareste values (4,200);

insert into calareste values (5,201);

insert into calareste values (3,202);

insert into calareste values (4,205);



**Ingrijeste:**

insert into ingrijeste values (1,32,2);

insert into ingrijeste values (2,33,3);

insert into ingrijeste values (3,34,1);

insert into ingrijeste values (4,31,4);

insert into ingrijeste values (5,30,5);

insert into ingrijeste values (1,33,2);

insert into ingrijeste values (2,30,3);

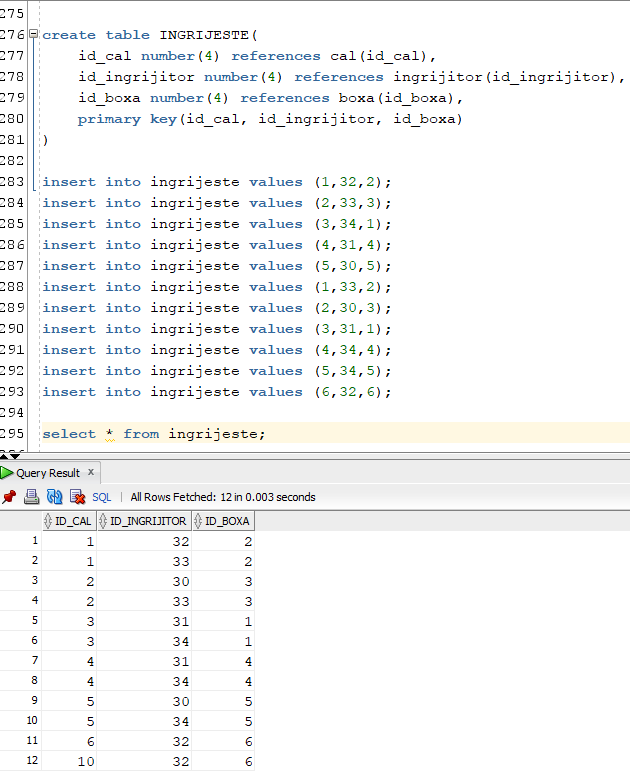
insert into ingrijeste values (3,31,1);

insert into ingrijeste values (4,34,4);

insert into ingrijeste values (5,34,5);

insert into ingrijeste values (6,32,6);

insert into ingrijeste values (10, 32, 6);



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.**

-- 6. Cerinta propusa:

-- Pentru fiecare antrenor din tabelul ANTRENOR,

-- afisati lista cailor (id-ul calului si numele) pe care ii antreneaza

-- pentru a participa la concurs.

set serveroutput on;

create or replace procedure ex6

is

type tabel\_antrenori is table of antrenor%ROWTYPE;

t\_antrenori tabel\_antrenori := tabel\_antrenori();

type tabel\_cai is table of cal%rowtype index by pls\_integer;

t\_cai tabel\_cai;

begin

select distinct ant.id\_antrenor, ant.nume, ant.prenume, ant.data\_angajare, ant.experienta

bulk collect into t\_antrenori

from antrenor ant, concurs co

where ant.id\_antrenor = co.id\_antrenor;

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

dbms\_output.put\_line('Id antrenor: ' || t\_antrenori(i).id\_antrenor || ' --- '

|| ' Nume antrenor: ' || t\_antrenori(i).nume || ' '

|| t\_antrenori(i).prenume);

dbms\_output.put\_line(' ');

select distinct c.id\_cal, c.id\_proprietar, c.id\_veterinar, c.nume, c.rasa, c.data\_nastere, c.sex

bulk collect into t\_cai

from cal c, participa p, concurs co

where c.id\_cal = p.id\_cal and

p.id\_concurs = co.id\_concurs and

co.id\_antrenor = t\_antrenori(i).id\_antrenor;

if t\_cai.count > 0 then

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

dbms\_output.put\_line('Id cal: ' || t\_cai(i).id\_cal

|| ' -> Nume cal: ' || t\_cai(i).nume);

end loop;

t\_cai.delete;

else

dbms\_output.put\_line('Nu antreneaza niciun cal.');

end if;

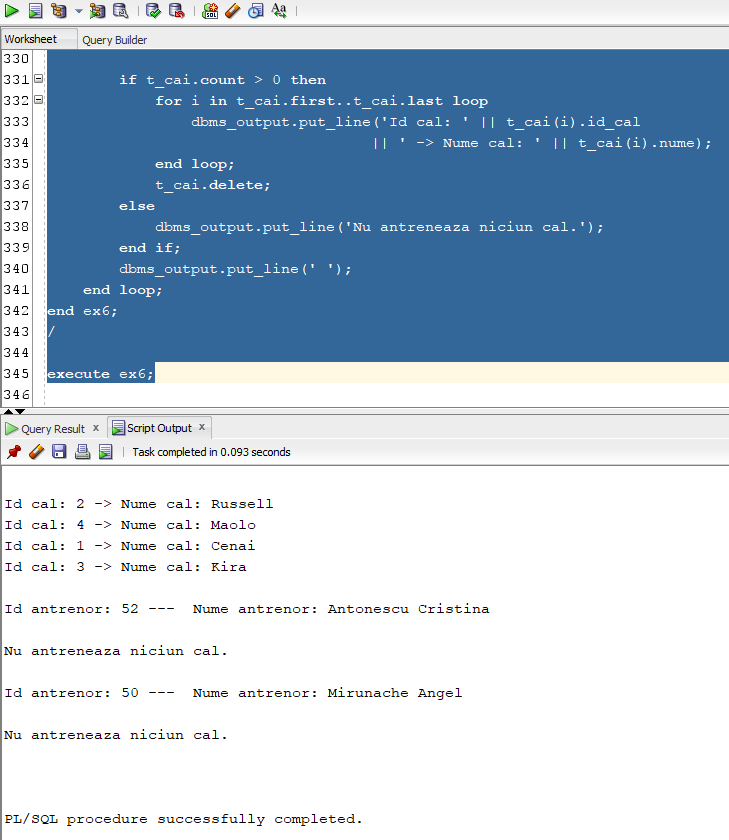
dbms\_output.put\_line(' ');

end loop;

end ex6;

/

execute 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.**

-- 7. Cerinta propusa:

-- Pentru fiecare cal existent in tabel,

-- specificati numarul de agrementisti care l-au calarit.

create or replace procedure ex7

is

type t\_nume\_cai is table of cal.nume%type;

type t\_numar\_calareti is table of number;

v\_nume t\_nume\_cai;

v\_numar t\_numar\_calareti;

cursor curs is

select c.nume nume\_cal, count(agr.id\_agrementist) numar\_calareti

from cal c, calareste calar, agrementist agr

where calar.id\_cal = c.id\_cal and calar.id\_agrementist = agr.id\_agrementist

group by c.nume;

begin

open curs;

fetch curs bulk collect into v\_nume, v\_numar;

close curs;

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

if v\_numar(i) = 0 then

DBMS\_OUTPUT.PUT\_LINE('Calul ' || v\_nume(i)

|| ' nu a fost calarit inca de niciun agrementist.');

elsif v\_numar(i) = 1 then

DBMS\_OUTPUT.PUT\_LINE('Calul ' || v\_nume(i)

|| ' a fost calarit de un singur agrementist.');

else

DBMS\_OUTPUT.PUT\_LINE('Calul ' || v\_nume(i) || ' a fost calarit de '

|| v\_numar(i) || ' agrementisti.' );

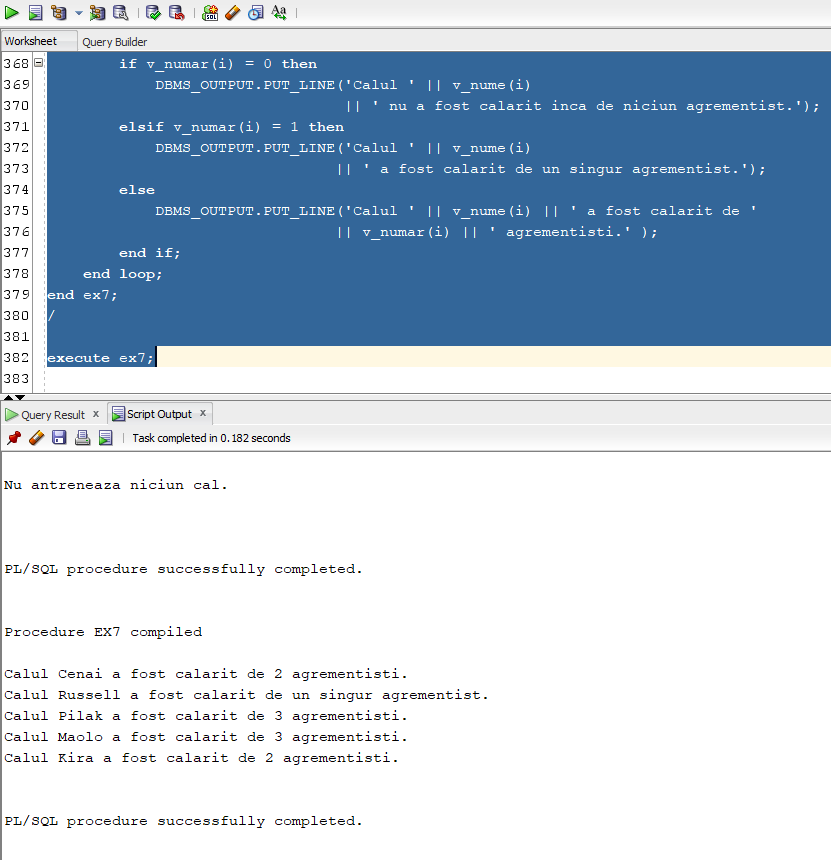
end if;

end loop;

end ex7;

/

execute ex7;



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.**

-- 8. Cerinta propusa:

-- Pentru un cal cu numele specificat, sa se gaseasca unicul ingrijitor al acestuia.

-- Se vor evidentia toate cazurile tratate. Rezolvarea va trebui sa foloseasca un subprogram

-- stocat de tip functie care sa utilizeze

-- intr-o singura comanda SQL 3 dintre tabelele definite.

create or replace function ex8

(v\_nume cal.nume%type)

return varchar2

is

v\_nume\_ingrijitor varchar2(30) := null;

v\_prenume\_ingrijitor varchar2(30) := null;

v\_id number := null;

begin

select id\_cal into v\_id

from cal

where upper(v\_nume) = upper(nume);

select ing.nume, ing.prenume into v\_nume\_ingrijitor, v\_prenume\_ingrijitor

from ingrijitor ing, ingrijeste ingrij, cal c

where upper(c.nume) = upper(v\_nume) and c.id\_cal = ingrij.id\_cal and ingrij.id\_ingrijitor = ing.id\_ingrijitor;

return 'Unicul ingrijitor al calului cu numele ' || v\_nume

|| ' este ' || v\_nume\_ingrijitor || ' ' || v\_prenume\_ingrijitor || '.';

exception

when no\_data\_found then

if v\_id is null then

return 'Calul cu numele ' || v\_nume || ' nu a fost inregistrat.';

end if;

if v\_nume\_ingrijitor is null then

return 'Nu sunt ingrijitori pentru calul cu numele ' || v\_nume || '.';

end if;

when too\_many\_rows then

if v\_nume\_ingrijitor is not null then

return 'Sunt mai multi ingrijitori pentru calul cu numele ' || v\_nume || '.';

end if;

if v\_id is not null then

return 'Sunt mai multi cai cu numele ' || v\_nume || '.';

end if;

when others then

return 'Alta eroare';

end ex8;

/

begin

dbms\_output.put\_line(ex8('Karma'));

dbms\_output.put\_line(ex8('Cenai'));

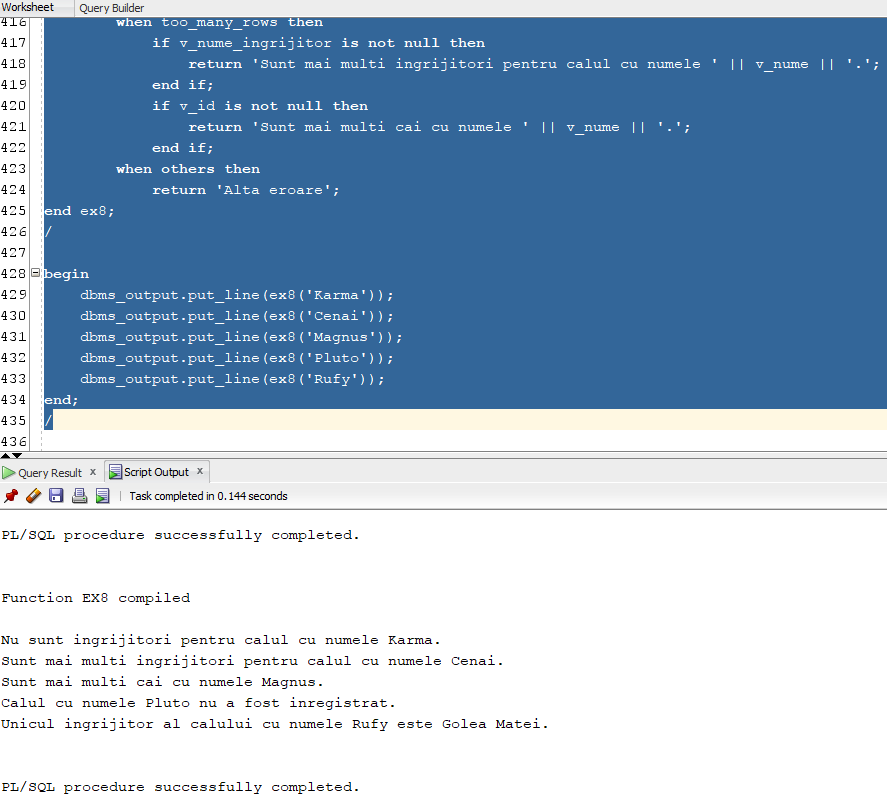
dbms\_output.put\_line(ex8('Magnus'));

dbms\_output.put\_line(ex8('Pluto'));

dbms\_output.put\_line(ex8('Rufy'));

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.**

-- 9. Cerinta propusa:

-- Pentru un antrenor cu numele dat ca parametru unui subprogram stocat de tip

-- procedură, sa se afiseze toti caii pe care ii antreneaza, id-ul manejului unde sunt antrenati

-- si aria acestuia, specificandu-se daca antrenamentele au loc si pe hipodrom sau inca nu.

-- Sa se foloseasca intr-o singură comandă SQL 5 dintre tabelele definite

-- si sa se evidentieze toate cazurile tratate, incluzand excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS.

create or replace procedure ex9

(v\_nume\_ant antrenor.nume%type)

is

v\_id\_ant antrenor.id\_antrenor%type;

type info is record(

nume\_antrenor antrenor.nume%type,

prenume\_antrenor antrenor.prenume%type,

nume\_cal cal.nume%type,

v\_id\_manej manej.id\_manej%type,

v\_suprafata manej.suprafata\_manej%type);

type t\_tablou is table of info index by pls\_integer;

v\_tablou t\_tablou;

exceptie exception;

v\_tip\_manej manej.tip\_manej%type;

nr number := 0;

begin

select id\_antrenor into v\_id\_ant

from antrenor

where upper(v\_nume\_ant) = upper(nume);

select ant.nume, ant.prenume, c.nume, m.id\_manej, m.suprafata\_manej

bulk collect into v\_tablou

from manej m, alege a, antrenor ant, antreneaza antren, cal c

where upper(v\_nume\_ant) = upper(ant.nume) and

m.id\_manej = a.id\_manej and

a.id\_antrenor = ant.id\_antrenor and

ant.id\_antrenor = antren.id\_antrenor and

antren.id\_cal = c.id\_cal;

if v\_tablou.count = 0 then

v\_id\_ant := -1;

raise no\_data\_found;

else

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

dbms\_output.put\_line(v\_tablou(i).nume\_antrenor || ' '

|| v\_tablou(i).prenume\_antrenor

|| ' antreneaza calul '

|| v\_tablou(i).nume\_cal || ' la manejul cu id-ul '

|| v\_tablou(i).v\_id\_manej || ' cu suprafata de '

|| v\_tablou(i).v\_suprafata || ' metri patrati.');

end loop;

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

select tip\_manej into v\_tip\_manej

from manej

where id\_manej = v\_tablou(i).v\_id\_manej;

if lower(v\_tip\_manej) = 'hipodrom' then

nr := nr + 1;

end if;

end loop;

if nr = 0 then

raise exceptie;

else

dbms\_output.put\_line(' -> Caii sunt antrenati si pe hipodrom.');

end if;

end if;

exception

when no\_data\_found then

if v\_id\_ant is null then

dbms\_output.put\_line('Nu exista antrenorul cu numele ' || v\_nume\_ant || '.');

else

dbms\_output.put\_line('Antrenorul cu numele ' || v\_nume\_ant

|| ' nu antreneaza niciun cal din herghelie.');

end if;

when too\_many\_rows then

dbms\_output.put\_line('Sunt mai multi antrenori cu numele ' || v\_nume\_ant || '.');

when exceptie then

dbms\_output.put\_line(' -> Caii nu sunt antrenati inca pe hipodrom.');

end ex9;

/

begin

dbms\_output.put\_line(' -- Test 1: ');

ex9('Radulescu');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 2: ');

ex9('Gavrila');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 3: ');

ex9('Mirunache');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 4: ');

ex9('Filip');

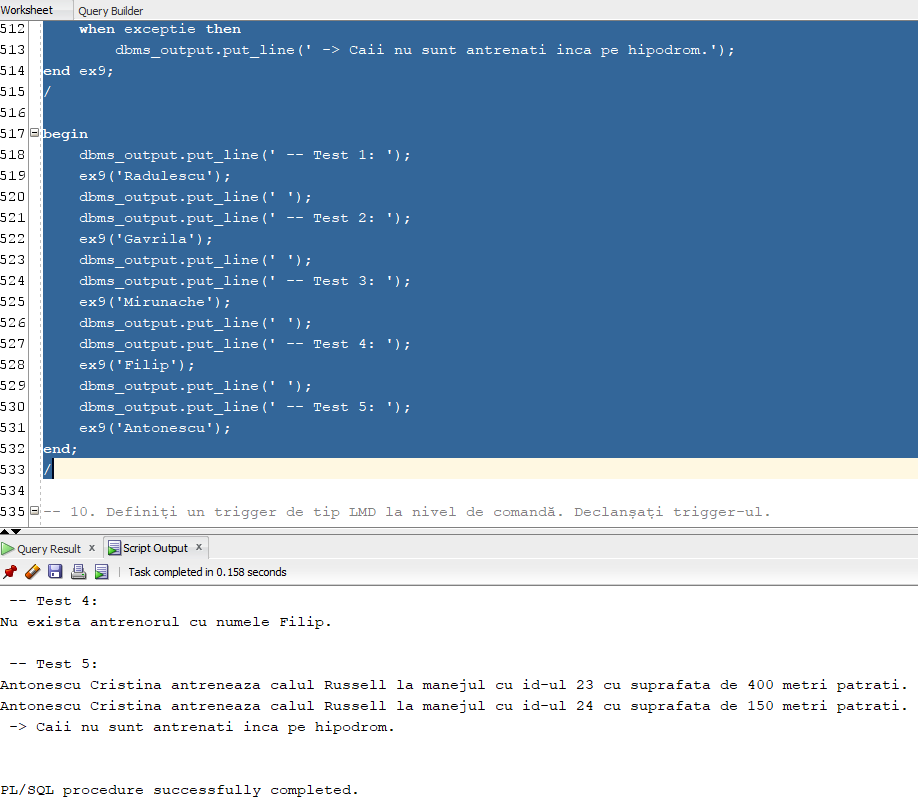
dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 5: ');

ex9('Antonescu');

end;

/



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

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

-- Trigger-ul se va declansa la inserarea unui cal in tabela CAL atunci cand

-- numarul boxelor nu permite tinerea calului in herghelie. Se considera ca

-- intr-o boxa incap maxim 4 cai.

create or replace trigger trigger10

before insert on cal

declare

nr\_cai number;

nr\_boxe number;

exceptie exception;

begin

select count(\*) into nr\_cai

from cal;

select count(\*) into nr\_boxe

from boxa;

if nr\_cai > 4 \* nr\_boxe then

raise exceptie;

end if;

exception

when exceptie then

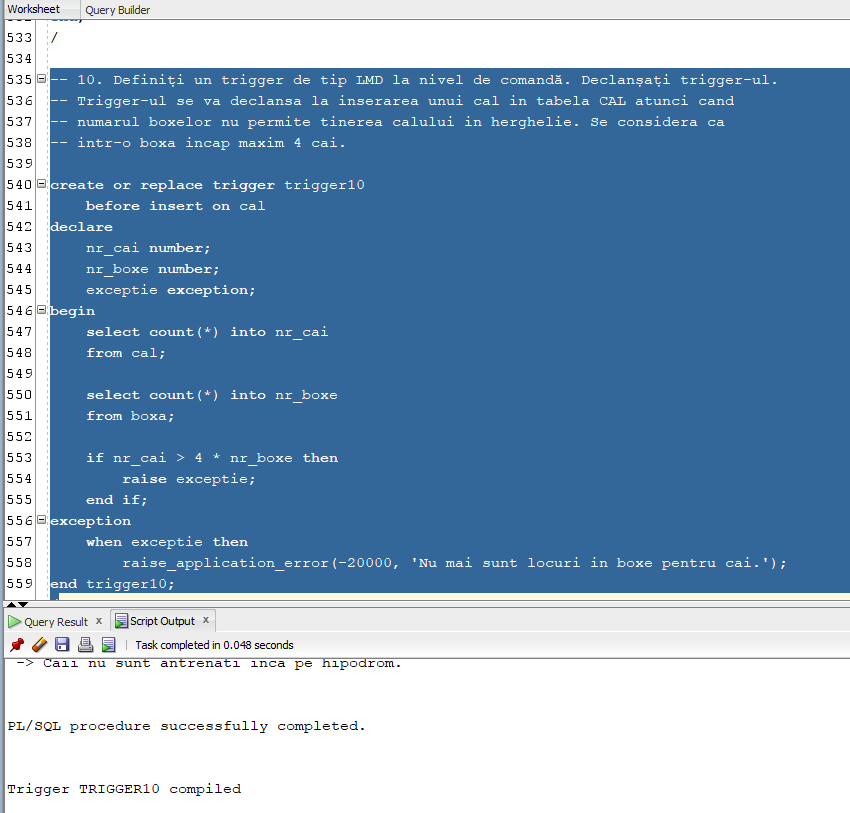
raise\_application\_error(-20000, 'Nu mai sunt locuri in boxe pentru cai.');

end trigger10;

/

insert into cal values (10, 10, 102, 'Plado', 'Indian', '29-APR-19', 'mascul');

drop trigger trigger10;



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

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

-- Trigger-ul se va declansa la modificarea suprafetei manejului atunci cand

-- caii antrenati la cel putin un manej nu vor avea o arie minima de desfasurare

-- de 30 de metri patrati fiecare.

create or replace trigger trigger11

before update of suprafata\_manej on manej

for each row

declare

numar\_cai\_antr number;

supra\_min\_per\_cal number := 30; -- 30 m^2

exceptie exception;

begin

select count(id\_manej)

into numar\_cai\_antr

from alege

where id\_manej = :old.id\_manej;

if :new.suprafata\_manej < supra\_min\_per\_cal \* numar\_cai\_antr then

raise exceptie;

end if;

exception

when exceptie then

raise\_application\_error(-20001, 'Nu se pot face modificari la ariile manejelor.');

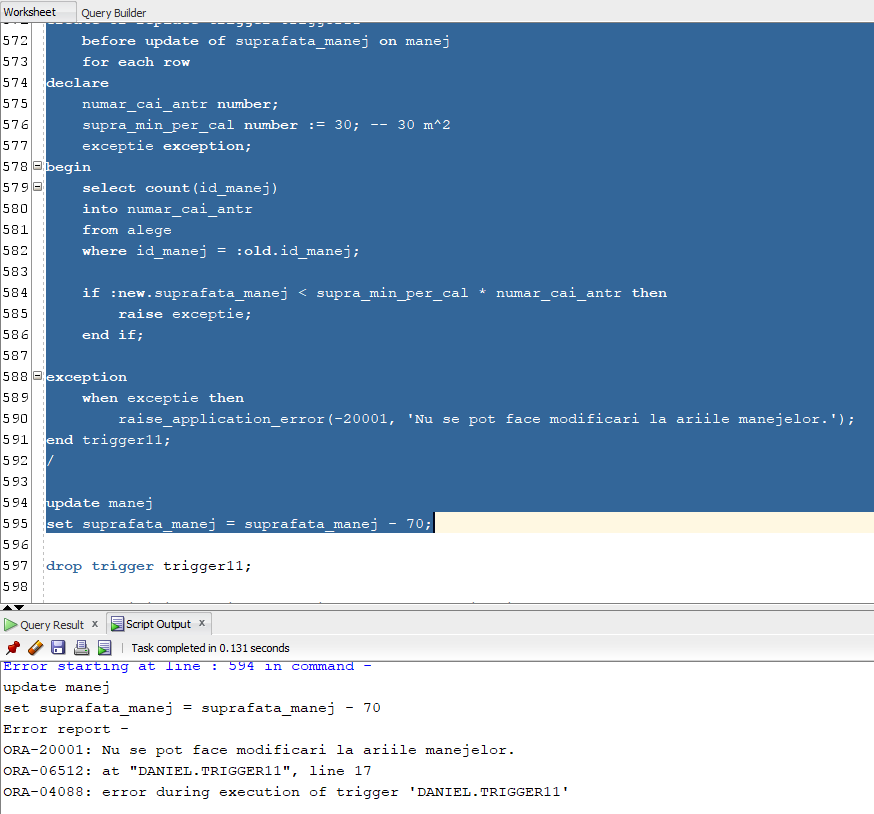
end trigger11;

/

update manej

set suprafata\_manej = suprafata\_manej - 70;

drop trigger trigger11;



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

create table schema\_audit(

data date,

utilizator varchar2(15),

obiect\_creat varchar2(15),

nume\_obiect varchar2(15),

operatie varchar2(15)

);

create or replace trigger trigger12

after ddl on schema

begin

insert into schema\_audit values(

sysdate,

sys\_context('USERENV','CURRENT\_USER'),

ora\_dict\_obj\_type,

ora\_dict\_obj\_name,

ora\_sysevent

);

end trigger12;

/

create table club\_echitatie( nume\_club varchar2(30));

insert into club\_echitatie values('Club1');

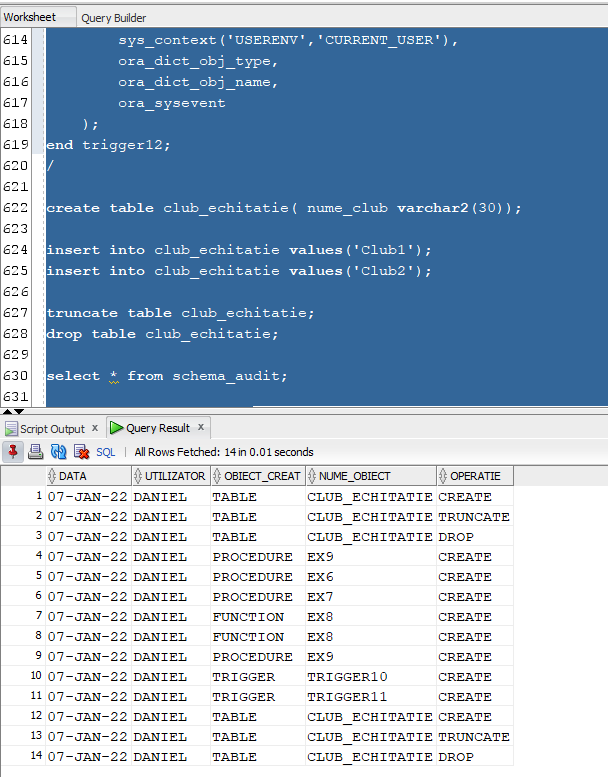
insert into club\_echitatie values('Club2');

truncate table club\_echitatie;

drop table club\_echitatie;

select \* from schema\_audit;

drop trigger trigger12;



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

create or replace package pachet13 as

-------

procedure ex6;

procedure ex7;

function ex8 (v\_nume cal.nume%type) return varchar2;

procedure ex9 (v\_nume\_ant antrenor.nume%type);

-------

end pachet13;

create or replace package body pachet13 as

-------

procedure ex6

is

type tabel\_antrenori is table of antrenor%ROWTYPE;

t\_antrenori tabel\_antrenori := tabel\_antrenori();

type tabel\_cai is table of cal%rowtype index by pls\_integer;

t\_cai tabel\_cai;

begin

select distinct ant.id\_antrenor, ant.nume, ant.prenume, ant.data\_angajare, ant.experienta

bulk collect into t\_antrenori

from antrenor ant, concurs co

where ant.id\_antrenor = co.id\_antrenor;

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

dbms\_output.put\_line('Id antrenor: ' || t\_antrenori(i).id\_antrenor || ' --- '

|| ' Nume antrenor: ' || t\_antrenori(i).nume || ' '

|| t\_antrenori(i).prenume);

dbms\_output.put\_line(' ');

select distinct c.id\_cal, c.id\_proprietar, c.id\_veterinar, c.nume, c.rasa, c.data\_nastere, c.sex

bulk collect into t\_cai

from cal c, participa p, concurs co

where c.id\_cal = p.id\_cal and

p.id\_concurs = co.id\_concurs and

co.id\_antrenor = t\_antrenori(i).id\_antrenor;

if t\_cai.count > 0 then

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

dbms\_output.put\_line('Id cal: ' || t\_cai(i).id\_cal

|| ' -> Nume cal: ' || t\_cai(i).nume);

end loop;

t\_cai.delete;

else

dbms\_output.put\_line('Nu antreneaza niciun cal.');

end if;

dbms\_output.put\_line(' ');

end loop;

end ex6;

-------

procedure ex7

is

type t\_nume\_cai is table of cal.nume%type;

type t\_numar\_calareti is table of number;

v\_nume t\_nume\_cai;

v\_numar t\_numar\_calareti;

cursor curs is

select c.nume nume\_cal, count(agr.id\_agrementist) numar\_calareti

from cal c, calareste calar, agrementist agr

where calar.id\_cal = c.id\_cal and calar.id\_agrementist = agr.id\_agrementist

group by c.nume;

begin

open curs;

fetch curs bulk collect into v\_nume, v\_numar;

close curs;

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

if v\_numar(i) = 0 then

DBMS\_OUTPUT.PUT\_LINE('Calul ' || v\_nume(i)

|| ' nu a fost calarit inca de niciun agrementist.');

elsif v\_numar(i) = 1 then

DBMS\_OUTPUT.PUT\_LINE('Calul ' || v\_nume(i)

|| ' a fost calarit de un singur agrementist.');

else

DBMS\_OUTPUT.PUT\_LINE('Calul ' || v\_nume(i) || ' a fost calarit de '

|| v\_numar(i) || ' agrementisti.' );

end if;

end loop;

end ex7;

-------

function ex8

(v\_nume cal.nume%type)

return varchar2

is

v\_nume\_ingrijitor varchar2(30) := null;

v\_prenume\_ingrijitor varchar2(30) := null;

v\_id number := null;

begin

select id\_cal into v\_id

from cal

where upper(v\_nume) = upper(nume);

select ing.nume, ing.prenume into v\_nume\_ingrijitor, v\_prenume\_ingrijitor

from ingrijitor ing, ingrijeste ingrij, cal c

where upper(c.nume) = upper(v\_nume) and c.id\_cal = ingrij.id\_cal and ingrij.id\_ingrijitor = ing.id\_ingrijitor;

return 'Unicul ingrijitor al calului cu numele ' || v\_nume

|| ' este ' || v\_nume\_ingrijitor || ' ' || v\_prenume\_ingrijitor || '.';

exception

when no\_data\_found then

if v\_id is null then

return 'Calul cu numele ' || v\_nume || ' nu a fost inregistrat.';

end if;

if v\_nume\_ingrijitor is null then

return 'Nu sunt ingrijitori pentru calul cu numele ' || v\_nume || '.';

end if;

when too\_many\_rows then

if v\_nume\_ingrijitor is not null then

return 'Sunt mai multi ingrijitori pentru calul cu numele ' || v\_nume || '.';

end if;

if v\_id is not null then

return 'Sunt mai multi cai cu numele ' || v\_nume || '.';

end if;

when others then

return 'Alta eroare';

end ex8;

-------

procedure ex9

(v\_nume\_ant antrenor.nume%type)

is

v\_id\_ant antrenor.id\_antrenor%type;

type info is record(

nume\_antrenor antrenor.nume%type,

prenume\_antrenor antrenor.prenume%type,

nume\_cal cal.nume%type,

v\_id\_manej manej.id\_manej%type,

v\_suprafata manej.suprafata\_manej%type);

type t\_tablou is table of info index by pls\_integer;

v\_tablou t\_tablou;

exceptie exception;

v\_tip\_manej manej.tip\_manej%type;

nr number := 0;

begin

select id\_antrenor into v\_id\_ant

from antrenor

where upper(v\_nume\_ant) = upper(nume);

select ant.nume, ant.prenume, c.nume, m.id\_manej, m.suprafata\_manej

bulk collect into v\_tablou

from manej m, alege a, antrenor ant, antreneaza antren, cal c

where upper(v\_nume\_ant) = upper(ant.nume) and

m.id\_manej = a.id\_manej and

a.id\_antrenor = ant.id\_antrenor and

ant.id\_antrenor = antren.id\_antrenor and

antren.id\_cal = c.id\_cal;

if v\_tablou.count = 0 then

v\_id\_ant := -1;

raise no\_data\_found;

else

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

dbms\_output.put\_line(v\_tablou(i).nume\_antrenor || ' '

|| v\_tablou(i).prenume\_antrenor

|| ' antreneaza calul '

|| v\_tablou(i).nume\_cal || ' la manejul cu id-ul '

|| v\_tablou(i).v\_id\_manej || ' cu suprafata de '

|| v\_tablou(i).v\_suprafata || ' metri patrati.');

end loop;

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

select tip\_manej into v\_tip\_manej

from manej

where id\_manej = v\_tablou(i).v\_id\_manej;

if lower(v\_tip\_manej) = 'hipodrom' then

nr := nr + 1;

end if;

end loop;

if nr = 0 then

raise exceptie;

else

dbms\_output.put\_line(' -> Caii sunt antrenati si pe hipodrom.');

end if;

end if;

exception

when no\_data\_found then

if v\_id\_ant is null then

dbms\_output.put\_line('Nu exista antrenorul cu numele ' || v\_nume\_ant || '.');

else

dbms\_output.put\_line('Antrenorul cu numele ' || v\_nume\_ant || ' nu antreneaza niciun cal din herghelie.');

end if;

when too\_many\_rows then

dbms\_output.put\_line('Sunt mai multi antrenori cu numele ' || v\_nume\_ant || '.');

when exceptie then

dbms\_output.put\_line(' -> Caii nu sunt antrenati inca pe hipodrom.');

end ex9;

-------

end pachet13;

begin

dbms\_output.put\_line('EX6: ');

pachet13.ex6;

dbms\_output.put\_line('EX7: ');

dbms\_output.put\_line(' ');

pachet13.ex7;

dbms\_output.put\_line(' ');

dbms\_output.put\_line('EX8: ');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(pachet13.ex8('Karma'));

dbms\_output.put\_line(pachet13.ex8('Cenai'));

dbms\_output.put\_line(pachet13.ex8('Magnus'));

dbms\_output.put\_line(pachet13.ex8('Pluto'));

dbms\_output.put\_line(pachet13.ex8('Rufy'));

dbms\_output.put\_line(' ');

dbms\_output.put\_line('EX9: ');

dbms\_output.put\_line(' -- Test 1: ');

pachet13.ex9('Radulescu');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 2: ');

pachet13.ex9('Gavrila');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 3: ');

pachet13.ex9('Mirunache');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 4: ');

pachet13.ex9('Filip');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' -- Test 5: ');

pachet13.ex9('Antonescu');

end;

/

