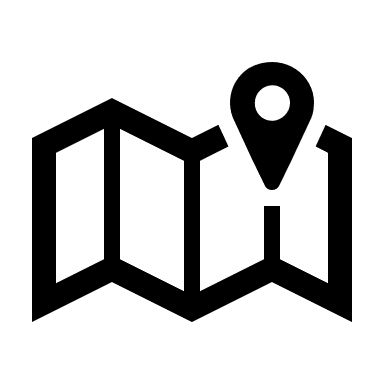
**Proiect SGBD**

**Gestiunea unei agenții de turism**

****

**Barbu Robert-Gabriel – Grupa 234**

# **Cerința 1**



În zilele noastre tendința populație este spre a-și petrece concediul în destinații menite relaxării și deconectării de viața cotidiană. Mai mult de atât, în rândul tinerilor, din ce în ce mai mulți doresc să aibă parte de vacanțe în zone exotice sau cu număr de atracții turistice ridicat.

Agenția de turism asupra căreia am creat baza de date oferă pachete de vacanță și are mai mulți angajați. Un pachet de vacanță oferă transport și hotel într-o anumită locație, și orice tip de pachet poate fi achiziționat de către un cumpărător prin semnarea unui contract de către acesta și un angajat de tip agent. În contract se află id\_ul cumpărătorului, al agentului și al tipului de pachet pe care dorește să îl cumpere, plus alte detalii cu legătură la plată și data plecării și a întoarcerii din vacanță.

**Restricții și reguli:**

1. un cumparator poate achizitiona doar cate un pachet pe rand deoarece e vorba de servicii diferite

2. un contract este semnat intre un agent si un cumparator

3. pachetul contine cate o locatie si un hotel

4. un pachet include un singur hotel

5. toti agentii au un salariu de baza, iar peste salariul de baza obtin cate un comision din contracte. Acest comision nu e identic pentru toti agentii(bazat pe cel negociat la inceperea contractului de munca)

6. tipurile de transport sunt doar cele folosite

7. hotelurile inregistrate sunt doar cele care se folosesc in pachete

8. locatiile inregistrate sunt doar cele care se folosesc in pachete

9. id-ul unui contract se schimba la fiecare contract nou, insemnand ca un alt contract poate avea acelasi cumparator sau acelasi tip de pachet inclus

Constrângeri:

Modelul relațional va impune constrângeri de tip:

- cheie primară (PK)

- cheie străină (FK)

CONTRACT(#id\_contract) - entitate ce va conține informații despre părțile care semnează contractul, cât și detalii despre pachetul oferit

CUMPARATOR(#id\_cumparator) - entitate ce va conține informații despre cumpărătorul unui pachet

ANGAJATI(#id\_angajat) - entitate ce va conține informații despre angajații agenției și rolul lor

AGENT - subentitatea AGENT, subentitatea supraentității ANGAJAȚI preia toate valoriile specificie unui angajat, plus comisionul care este specific doar angajaților de tip AGENT. Apare separat de entitatea ANGAJAȚI în diagrame deoarece un contract poate fi semnat doar de către un angajat de tip agent.

PACHET(#id\_pachet) - entitate ce va conține informații despre pachetul pe care un cumpărător ar vrea să îl achiziționeze

TRANSPORT(#id\_transport) - entitate ce va conține informații despre tipurile de transport

HOTEL(#id\_hotel) - entitate ce va conține informații despre hotel și locația unde se află

LOCATIE(#id\_locatie) - entitate ce va conține informații despre țara și orașul unde se află o locație

**Relații:**

1. CUMPARATOR-CONTRACT

- un cumpărător poate semna mai multe contracte, dar un contract este semnat de către un singur cumpărător

- tip relație: one-many

- cardinalitate: 1-M(1)

2. AGENT-CONTRACT

- un agent poate semna mai multe contracte, dar un contract este semnat de către un singur agent

- tip relație: one-many

- cardinalitate: 1-M(0)

3. ANGAJATI-AGENT

- un angajat poate fi agent, dar un agent sigur face parte dintre angajați

-tip relație: one-one

cardinalitate: 1-1(0)

4. PACHET-CONTRACT

- un pachet poate fi achiziționat și să apară pe mai multe contracte, dar un contract conține un singur pachet

- tip relație: one-one

- cardinalitate: 1-M(0)

5. PACHET\_TRANSPORT

- un pachet poate avea mai multe tipuri de transport diferite, iar tipurile de transport pot corespunde mai multor pachete

- relație: many-many

- cardinalitate: M(1)-M(1)

6. PACHET-HOTEL

- un pachet include un singur hotel, iar un hotel poate fi prezent intr-un singur tip de pachet de vacanta

- relație: one-one

- cardinalitate: 1-1

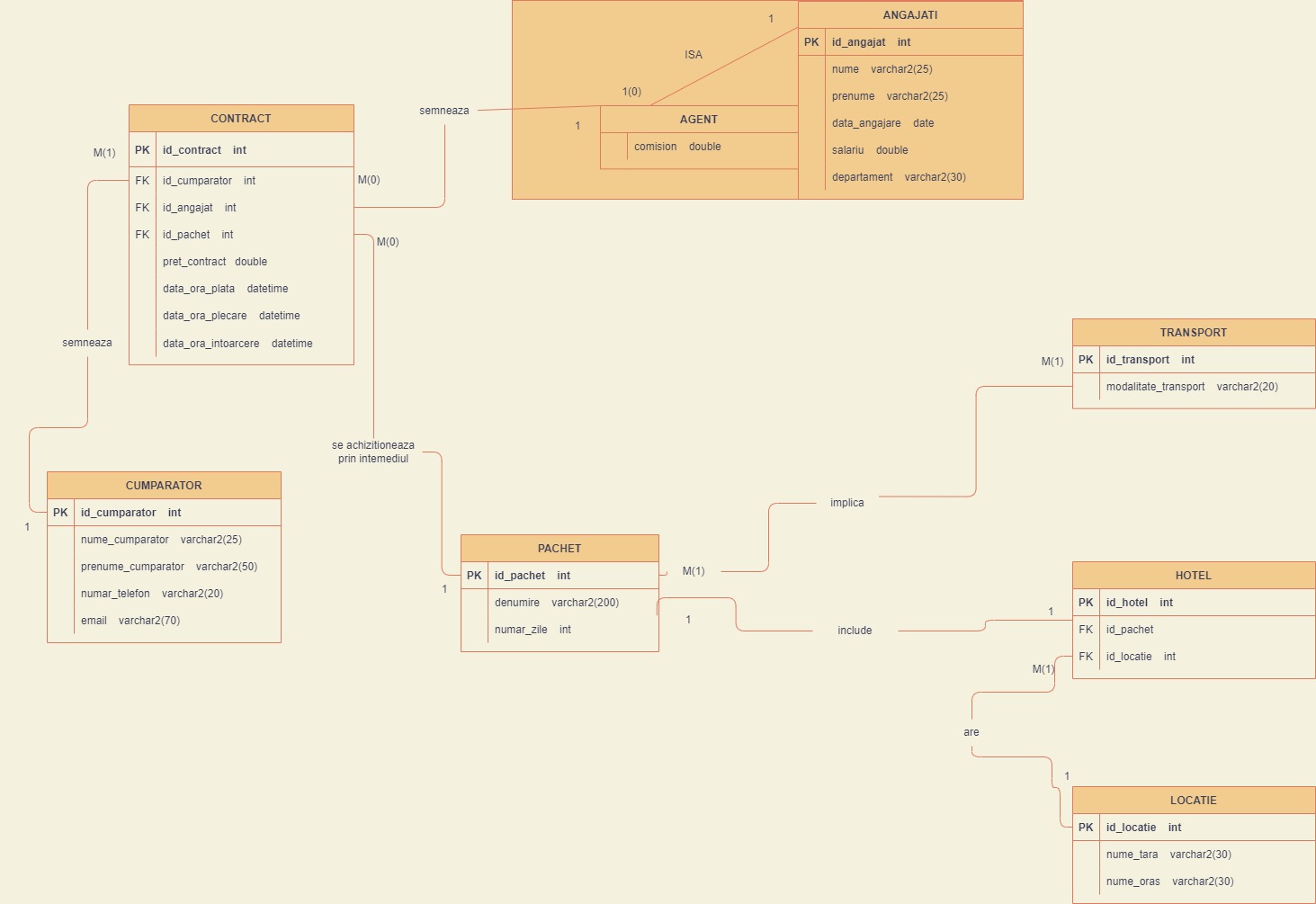
7. LOCATIE-HOTEL

- o locație oferă mai multe hotele, dar un hotel se află într-o singură locație

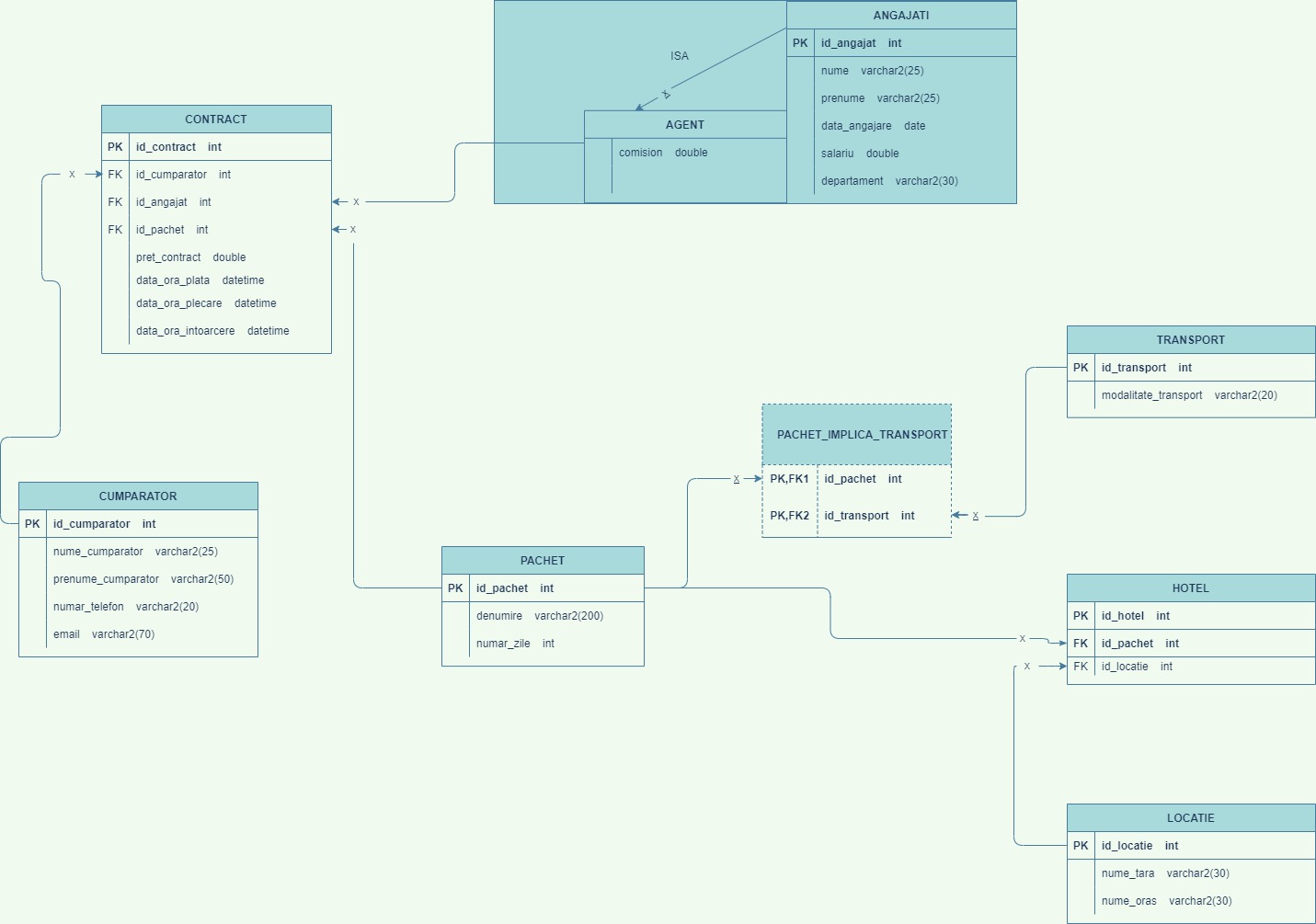
- relație: one-many

- cardinalitate: 1-M(1)

# **Cerința 2**

****

# **Cerința 3**

****

# **Cerința 4**

create table pachet(

id\_pachet int not null primary key,

denumire varchar2(200) not null,

numar\_zile int not null);

create table locatie(

id\_locatie int not null primary key,

nume\_tara varchar2(30) not null,

nume\_oras varchar2(30) not null);

create table hotel(

id\_hotel int not null primary key,

id\_locatie int not null,

id\_pachet int not null,

foreign key (id\_locatie) references locatie(id\_locatie),

foreign key (id\_pachet) references pachet(id\_pachet));

create table cumparator(

id\_cumparator int not null primary key,

nume\_cumparator varchar2(25) not null,

prenume\_cumparator varchar2(50) not null,

numar\_telefon varchar2(11),

email varchar2(70));

create table angajati(

id\_angajat int not null primary key,

nume varchar2(25) not null,

prenume varchar2(25) not null,

data\_angajare date not null,

salariu decimal(10,2),

departament varchar2(30) not null,

comision decimal(10,2));

create table contract(

id\_contract int not null primary key,

id\_cumparator int not null,

id\_angajat int not null,

id\_pachet int not null,

pret\_contract decimal(10,2) not null,

data\_plata date not null,

data\_plecare date not null,

data\_intoarcere date not null,

foreign key (id\_cumparator) references cumparator(id\_cumparator),

foreign key (id\_angajat) references angajati(id\_angajat),

foreign key (id\_pachet) references pachet(id\_pachet));

create table transport(

id\_transport int not null primary key,

modalitate\_transport varchar2(20) not null);

create table pachet\_implica\_transport(

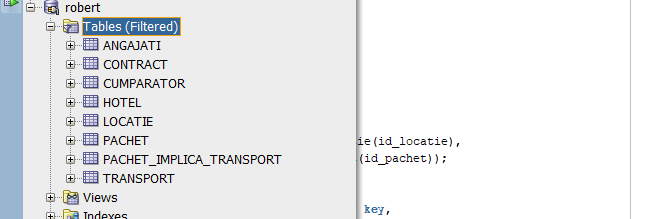
id\_pachet int not null,

id\_transport int not null,

constraint pk\_pachet\_transport primary key(id\_pachet, id\_transport));

alter table pachet\_implica\_transport add foreign key(id\_pachet) references pachet(id\_pachet);

alter table pachet\_implica\_transport add foreign key(id\_transport) references transport(id\_transport);

****

# **Cerința 5**

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(100, 'Ionescu', 'Raluca', to\_date('05/13/2021', 'MM/DD/RRRR'), 2000, 'AGENT', 15);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(101, 'Popescu', 'Mihai', to\_date('05/08/2021', 'MM/DD/RRRR'), 2000, 'AGENT', 17);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(102, 'Ionescu', 'Stefania', to\_date('06/18/2021', 'MM/DD/RRRR'), 2000, 'AGENT', 20);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(103, 'Barbu', 'Maria',to\_date('05/18/2011', 'MM/DD/RRRR'), 2000, 'AGENT', 20.5);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(104, 'Preda', 'Roxana', to\_date('05/18/2012', 'MM/DD/RRRR'), 2000, 'AGENT', 10);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(105, 'Stefanescu', 'Simona', to\_date('05/18/2020', 'MM/DD/RRRR'), 2500, 'IT', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(106, 'Andrei', 'Alexandru', to\_date('05/10/2021', 'MM/DD/RRRR'), 2500, 'IT', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(107, 'Ion', 'Raul', to\_date('09/18/2021', 'MM/DD/RRRR'), 1700, 'MARKETING', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

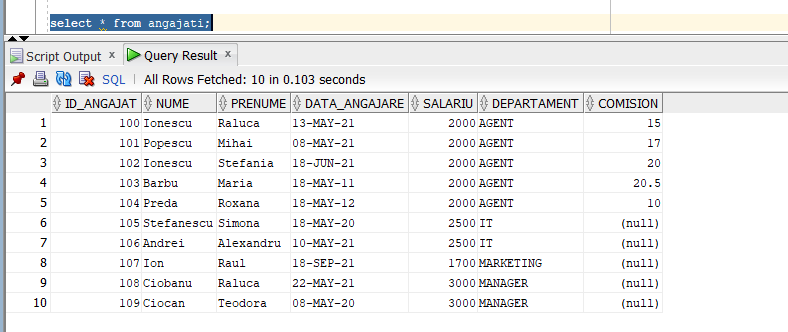
values

(108, 'Ciobanu', 'Raluca', to\_date('05/22/2021', 'MM/DD/RRRR'), 3000, 'MANAGER', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(109, 'Ciocan', 'Teodora', to\_date('05/08/2020', 'MM/DD/RRRR'), 3000, 'MANAGER', null);

****

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(200, 'Ion', 'Ionescu');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(201, 'Irina', 'Alina');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(202, 'Paun', 'Ionela');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

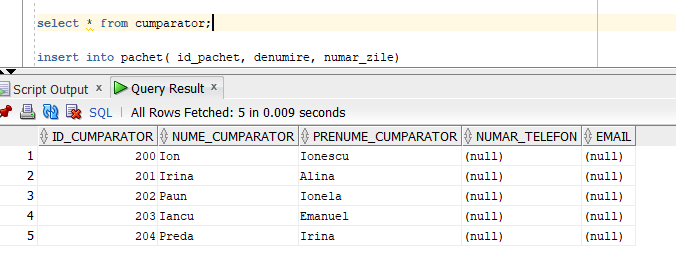
values

(203, 'Iancu', 'Emanuel');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(204, 'Preda', 'Irina');



insert into pachet( id\_pachet, denumire, numar\_zile)

values

(300, 'Paris', 4);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(301, 'Londra', 5);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(302, 'Brasov', 7);

insert into pachet( id\_pachet, denumire, numar\_zile)

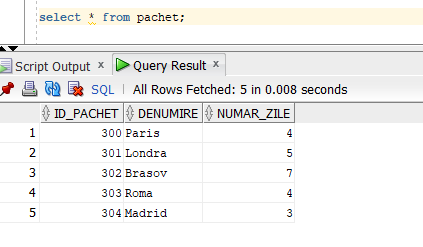
values

(303, 'Roma', 4);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(304, 'Madrid', 3);



insert into transport(id\_transport, modalitate\_transport)

values

(500, 'avion');

insert into transport(id\_transport, modalitate\_transport)

values

(501, 'tren');

insert into transport(id\_transport, modalitate\_transport)

values

(502, 'autobuz');

insert into transport(id\_transport, modalitate\_transport)

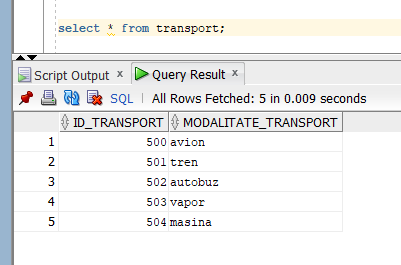
values

(503, 'vapor');

insert into transport(id\_transport, modalitate\_transport)

values

(504, 'masina');



insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(300, 500);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(301, 501);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(302, 502);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(303, 503);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(304, 504);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(300, 504);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(301, 503);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(302, 504);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

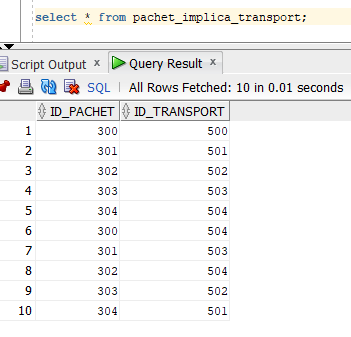
values

(303, 502);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(304, 501);

****

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( 400, 200, 100, 300, 1900,to\_date('05/08/2020', 'MM/DD/RRRR'),to\_date('05/10/2020', 'MM/DD/RRRR'),to\_date('05/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( 401, 201, 101, 301, 1500,to\_date('06/08/2020', 'MM/DD/RRRR'),to\_date('06/10/2020', 'MM/DD/RRRR'),to\_date('06/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

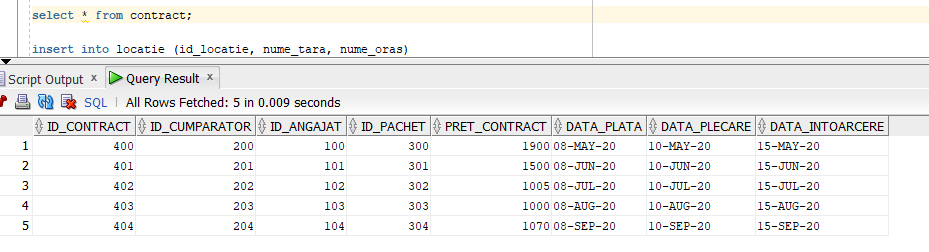
values ( 402, 202, 102, 302, 1005,to\_date('07/08/2020', 'MM/DD/RRRR'),to\_date('07/10/2020', 'MM/DD/RRRR'),to\_date('07/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( 403, 203, 103, 303, 1000,to\_date('08/08/2020', 'MM/DD/RRRR'),to\_date('08/10/2020', 'MM/DD/RRRR'),to\_date('08/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( 404, 204, 104, 304, 1070,to\_date('09/08/2020', 'MM/DD/RRRR'),to\_date('09/10/2020', 'MM/DD/RRRR'),to\_date('09/15/2020', 'MM/DD/RRRR'));

****

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(500, 'Franta', 'Paris');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(503, 'Italia', 'Roma');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

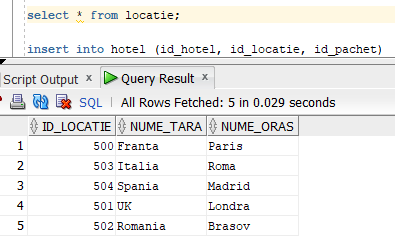
values(504, 'Spania', 'Madrid');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(501, 'UK', 'Londra');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(502, 'Romania', 'Brasov');



insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(600, 500, 300);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(601, 501, 301);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(602, 502, 302);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

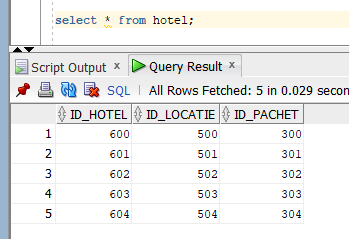
values

(603, 503, 303);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(604, 504, 304);



# **Cerința 6**

--Cate pachete se folosesc de un anumit tip de transport ( id\_transport ) si care sunt id-urile acestora? Afisati apoi si numele pachetelor

create or replace procedure cerinta6 (idtransport transport.id\_transport%type)

as

type tablou\_indexat is table of pachet\_implica\_transport.id\_pachet%type index by pls\_integer;

t tablou\_indexat;

type tablou\_imbricat is table of pachet.denumire%type;

t2 tablou\_imbricat;

begin

select id\_pachet bulk collect into t

from pachet\_implica\_transport

where id\_transport = idtransport;

dbms\_output.put\_line('Sunt ' || t.count || ' pachete care se folosesc de transportul de tip ' || idtransport || ':');

for i in t.first..t.last

loop

dbms\_output.put\_line(t(i) || ' ');

end loop;

select p.denumire bulk collect into t2

from pachet p join pachet\_implica\_transport pi on p.id\_pachet = pi.id\_pachet

join transport t on pi.id\_transport = t.id\_transport

where t.id\_transport=idtransport;

dbms\_output.put\_line('Denumirile pachetelor sunt: ' || ' ');

for i in reverse t2.first..t2.last

loop

dbms\_output.put\_line(t2(i) || ' ');

end loop;

end;

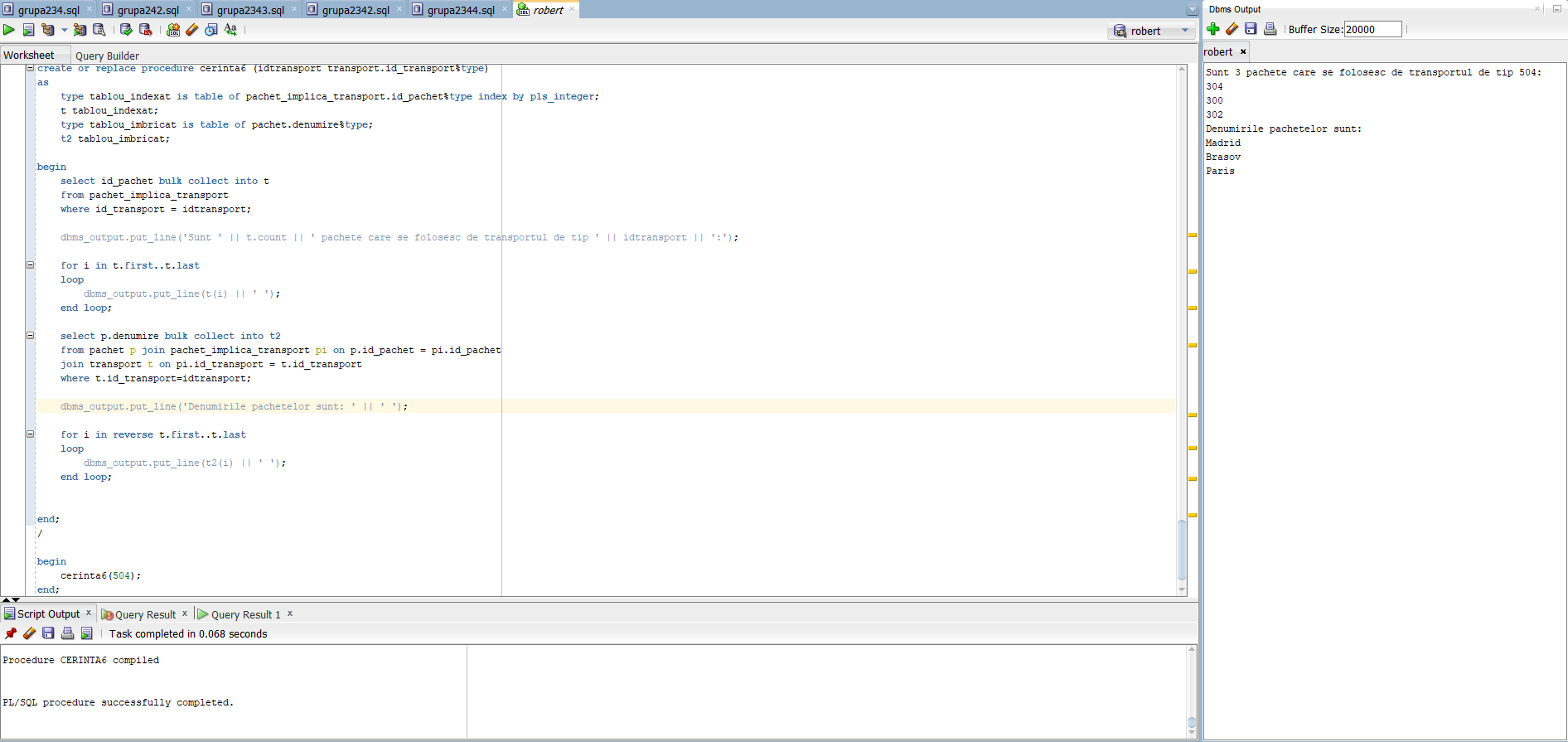
/

begin

cerinta6(504);

end;

/



# **Cerința 7**

Obtineti numarul angajatilor din fiecare departament si numele departamentului. Afisati apoi numele tuturor angajatilor cu salariu mai mic decat 2500.

create or replace procedure cerinta7

as

v\_nr number(4);

v\_dep angajati.departament%type;

cursor c is

select count(id\_angajat) , departament

from angajati

group by departament;

v\_nume angajati%rowtype;

cursor c1(salariu\_max number) is

select \*

from angajati

where salariu < salariu\_max;

begin

open c;

loop

fetch c into v\_nr, v\_dep;

exit when c%NOTFOUND;

if v\_nr =0 then

dbms\_output.put\_line('Niciun angajat nu lucreaza in departamentul ' || v\_dep);

elsif v\_nr =1 then

dbms\_output.put\_line( 'Un angajat lucreaza in departamentul ' || v\_dep);

else

dbms\_output.put\_line( v\_nr || ' angajati lucreaza in departamentul ' || v\_dep);

end if;

end loop;

close c;

open c1(2500);

loop

fetch c1 into v\_nume;

exit when c1%NOTFOUND;

dbms\_output.put\_line ( 'Angajatul ' || v\_nume.prenume || ' ' || v\_nume.nume || ' are salariul mai mic decat 2500');

end loop;

close c1;

end;

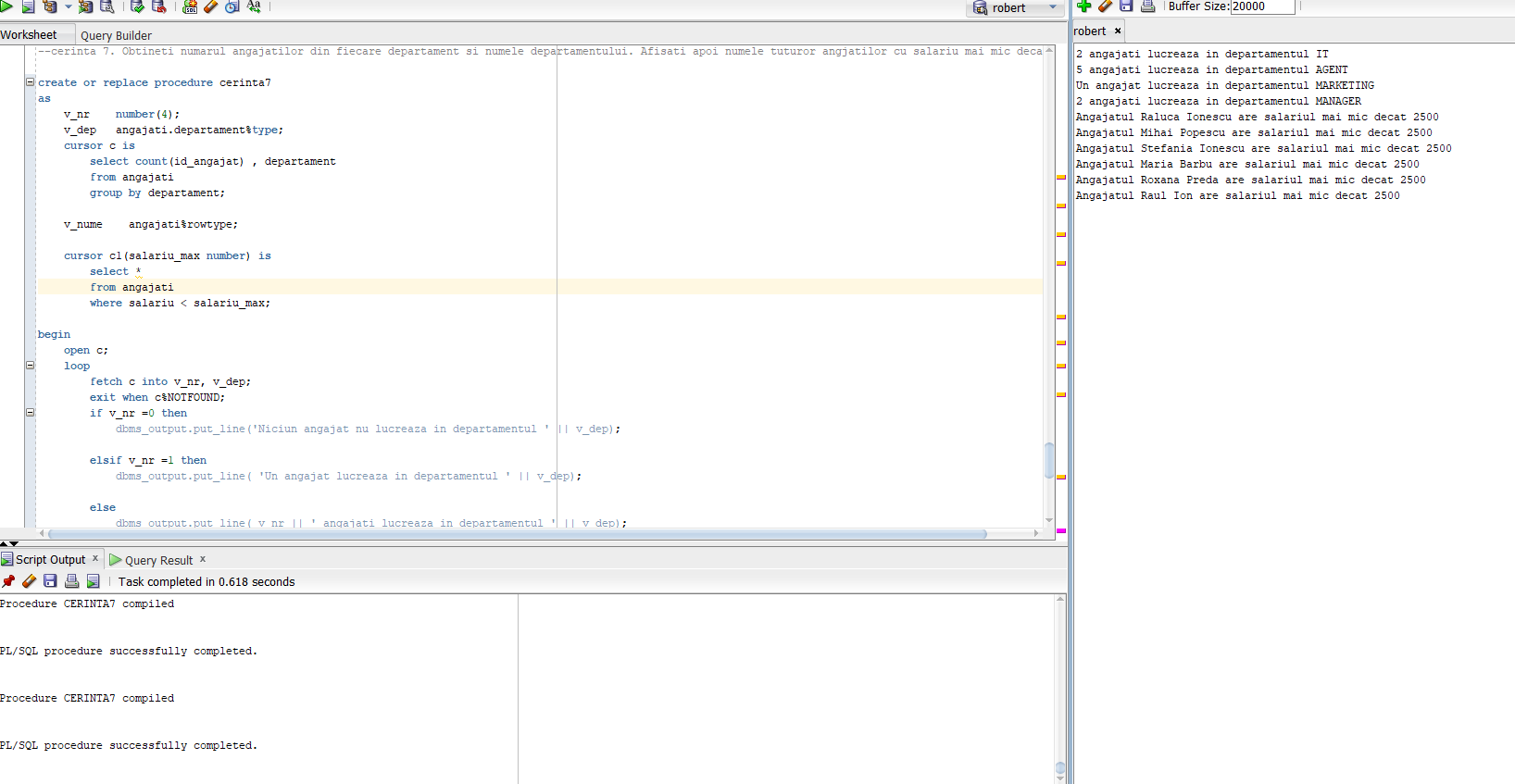
/

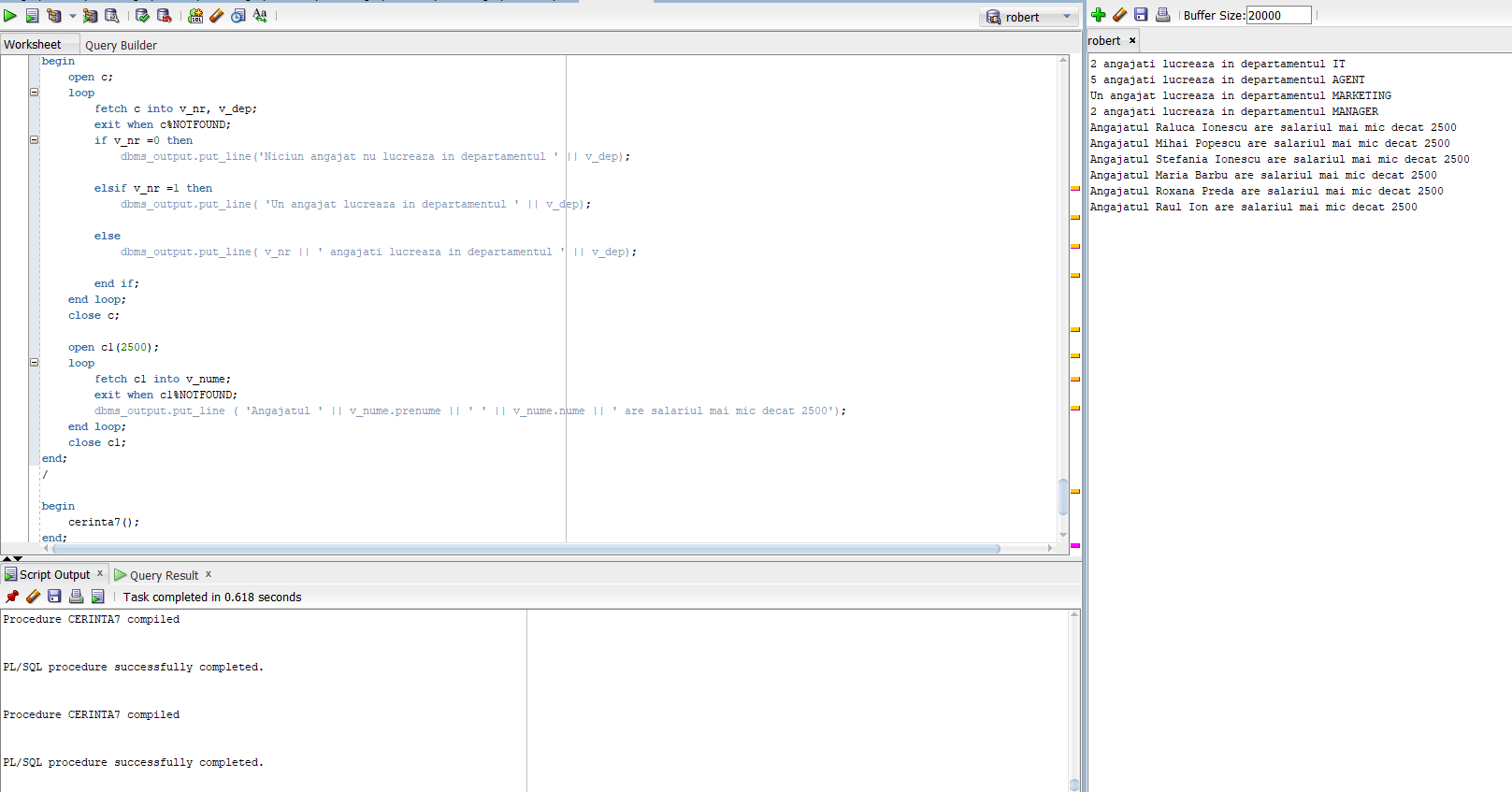
begin

cerinta7();

end;

/

****

****

# **Cerința 8**

Cate pachete contin hoteluri dintr-o tara data?

Exceptiile care pot aparea sunt : niciun pachet nu contine hotel in tara data; tara nu exista in tabelul de locatii; alte exceptii posibile

create or replace function cerinta8(tara locatie.nume\_tara%type) return number

is

numar\_rez number;

type tabel\_indexat is table of locatie%rowtype index by pls\_integer;

t tabel\_indexat;

exceptie1 exception;

exceptie2 exception;

begin

select \* bulk collect into t

from locatie

where upper(nume\_tara) like upper(tara);

if SQL%NOTFOUND then

raise exceptie1;

end if;

select count(p.id\_pachet) into numar\_rez

from pachet p join hotel h on p.id\_pachet = h.id\_pachet

join locatie l on h.id\_locatie = l.id\_locatie

where upper(l.nume\_tara) like upper(tara);

if numar\_rez = 0 then

raise exceptie2;

else

return numar\_rez;

end if;

exception

when exceptie1 then

dbms\_output.put\_line('Nu exista tara ' || tara || ' in tabelul de locatii');

return -1;

when exceptie2 then

dbms\_output.put\_line('Nu exista pachet care sa contina un hotel din tara ' || tara);

return -1;

when others then

dbms\_output.put\_line ( SQLCODE || ' ' || SQLERRM);

return -1;

end;

/

declare

exemplu locatie.nume\_tara%type;

begin

exemplu := cerinta8(-99);

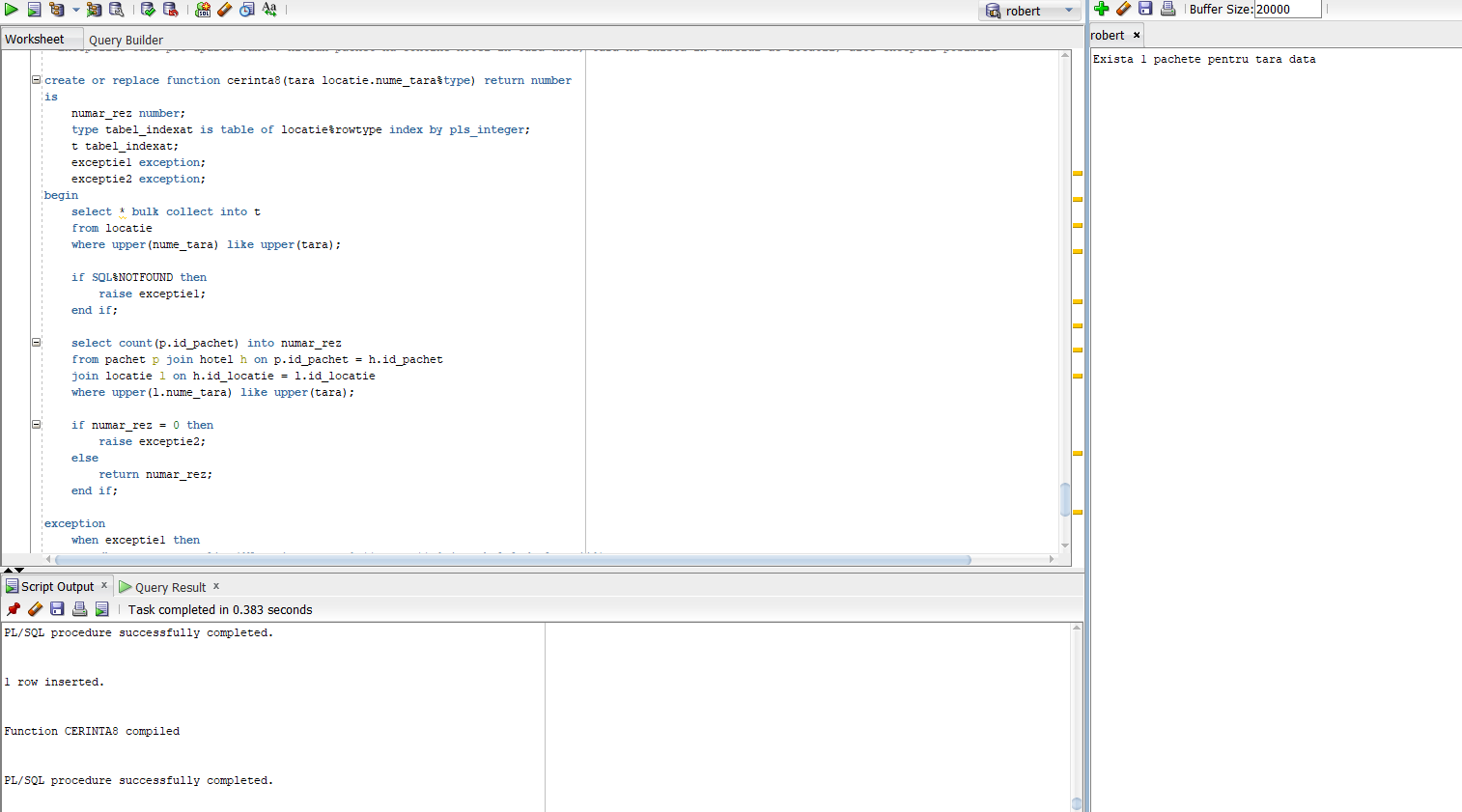
if exemplu> -1 then

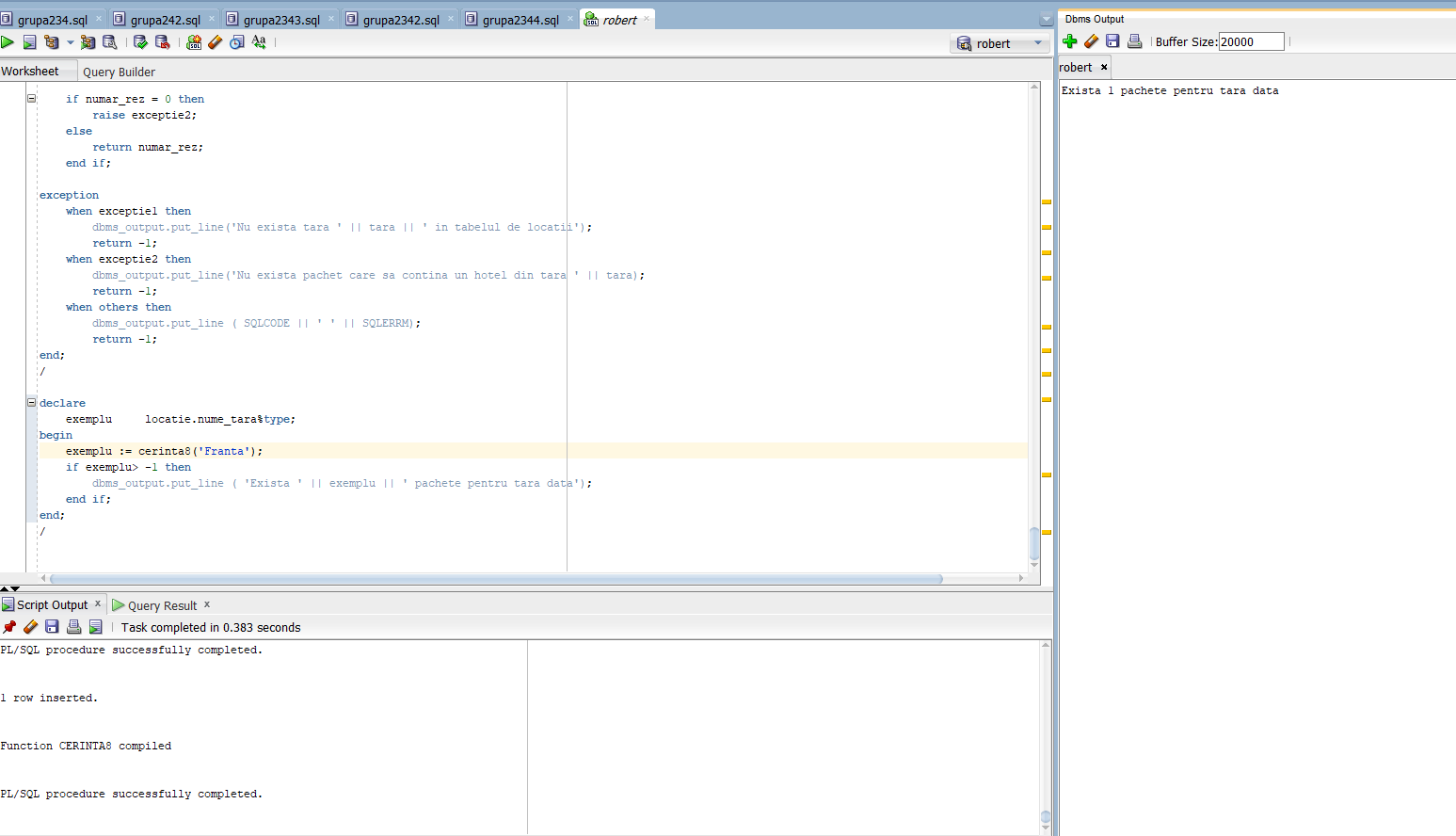
dbms\_output.put\_line ( 'Exista ' || exemplu || ' pachete pentru tara data');

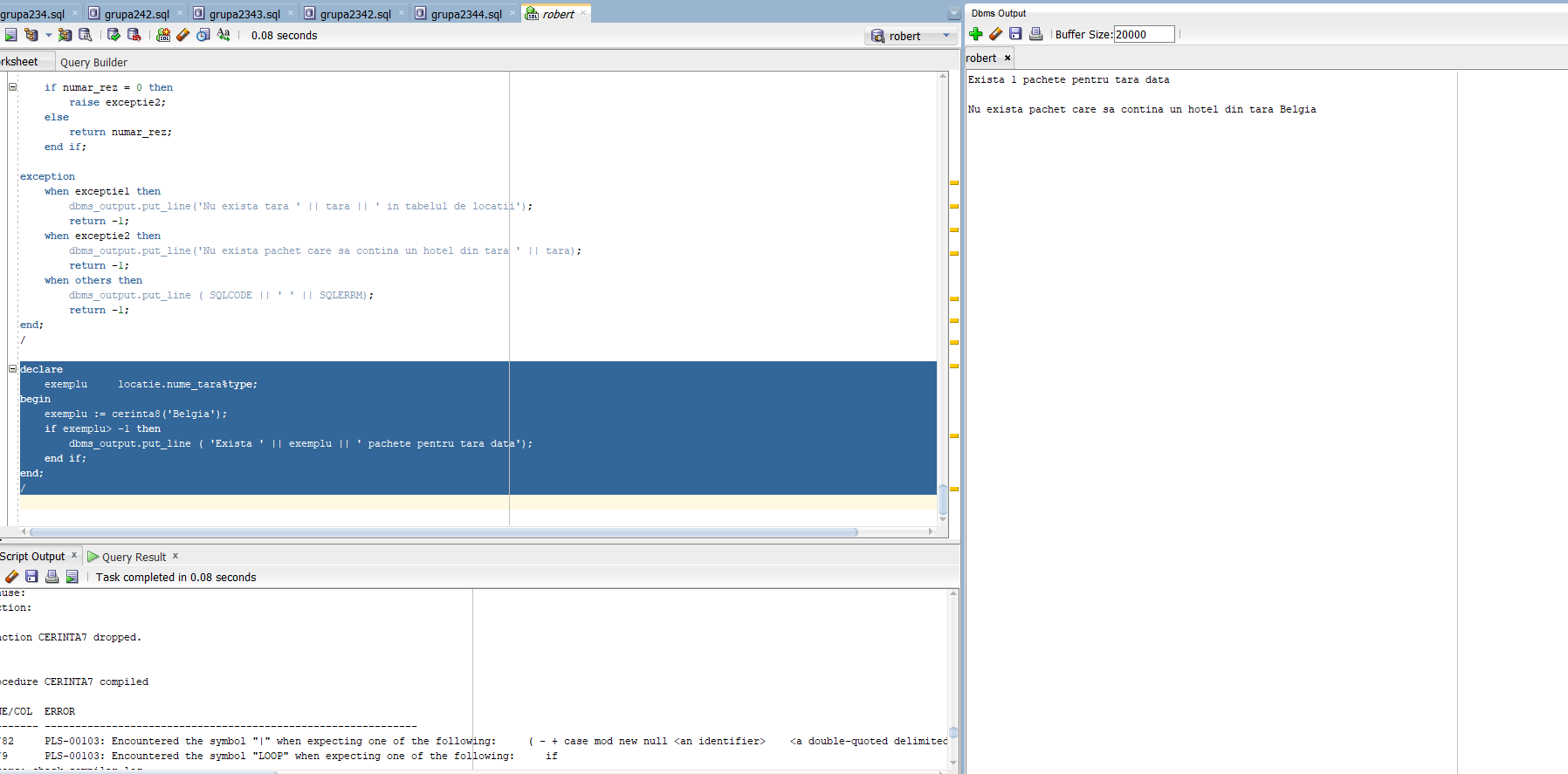
end if;

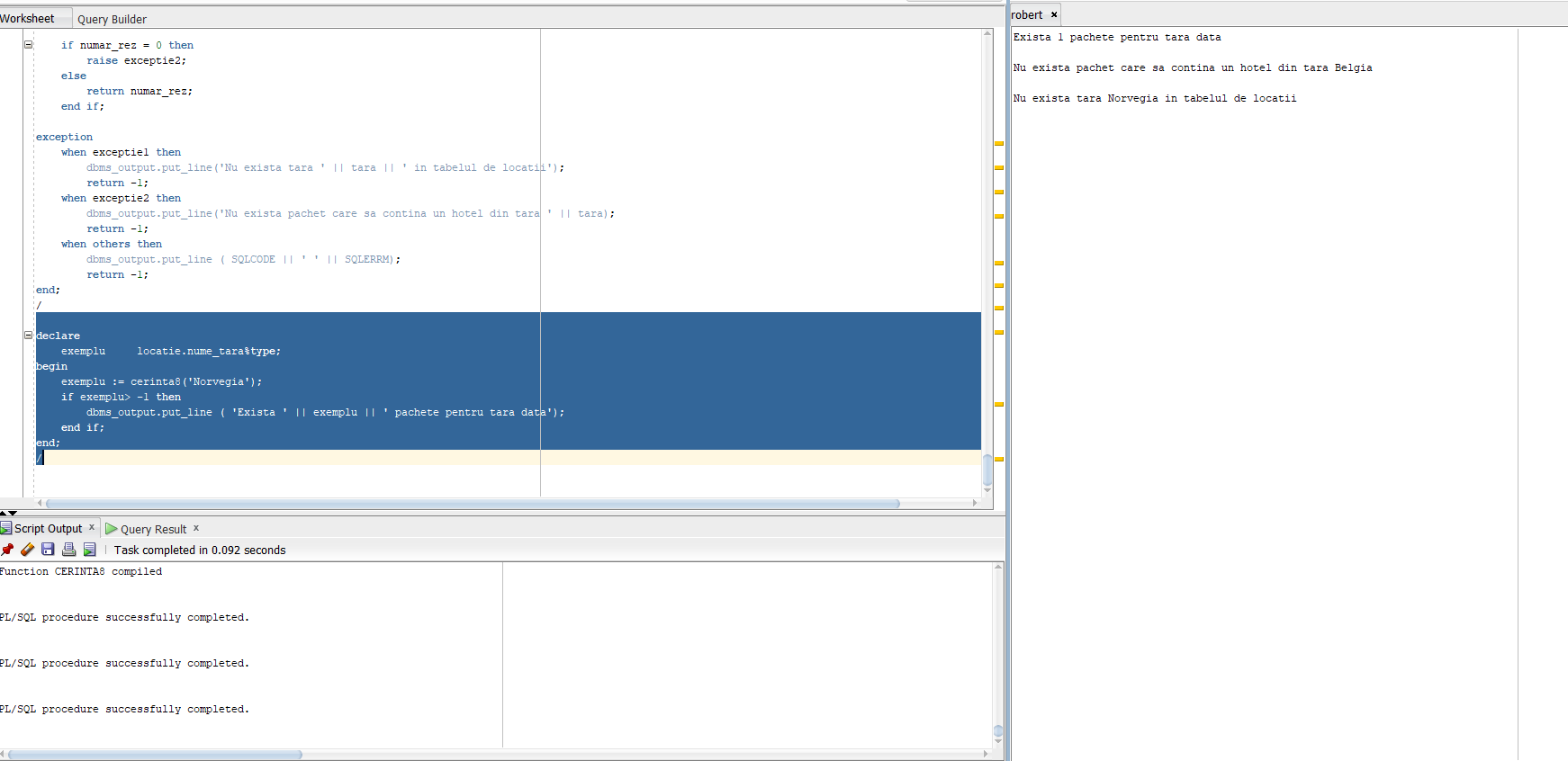
end;

/









# **Cerința 9**

Care este prenumele cumparatorului care a facut un contract pentru un pachet de vacanta intr-o tara data?

Exceptii: nu exista niciun cumparator pentru tara data (NO\_DATA\_FOUND), exista mai multi cumparatori cu acelasi prenume pentru tara data( TOO\_MANY\_ROWS ), alte exceptii ce pot aparea

create or replace procedure cerinta9( tara locatie.nume\_tara%type)

as

prenume\_c cumparator.prenume\_cumparator%type;

begin

select c.prenume\_cumparator into prenume\_c

from cumparator c join contract ct on c.id\_cumparator = ct.id\_cumparator

join pachet p on ct.id\_pachet = p.id\_pachet

join hotel h on p.id\_pachet = h.id\_pachet

join locatie l on h.id\_locatie = l.id\_locatie

where upper(l.nume\_tara) like upper(tara);

dbms\_output.put\_line('Prenumele cumparatorului care a achizitionat un pachet de vanta in tara ' || tara || ' este ' || prenume\_c);

exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('Nu exista niciun cumparator pentru un pachet de vacanta cu locatia in tara mentionata');

when TOO\_MANY\_ROWS then

dbms\_output.put\_line('Exista mai multi cumparatori cu prenumele ' || prenume\_c || ' care au achizitionat un pachet cu locatia in ' || tara);

when others then

dbms\_output.put\_line( SQLCODE || ' ' || SQLERRM);

end;

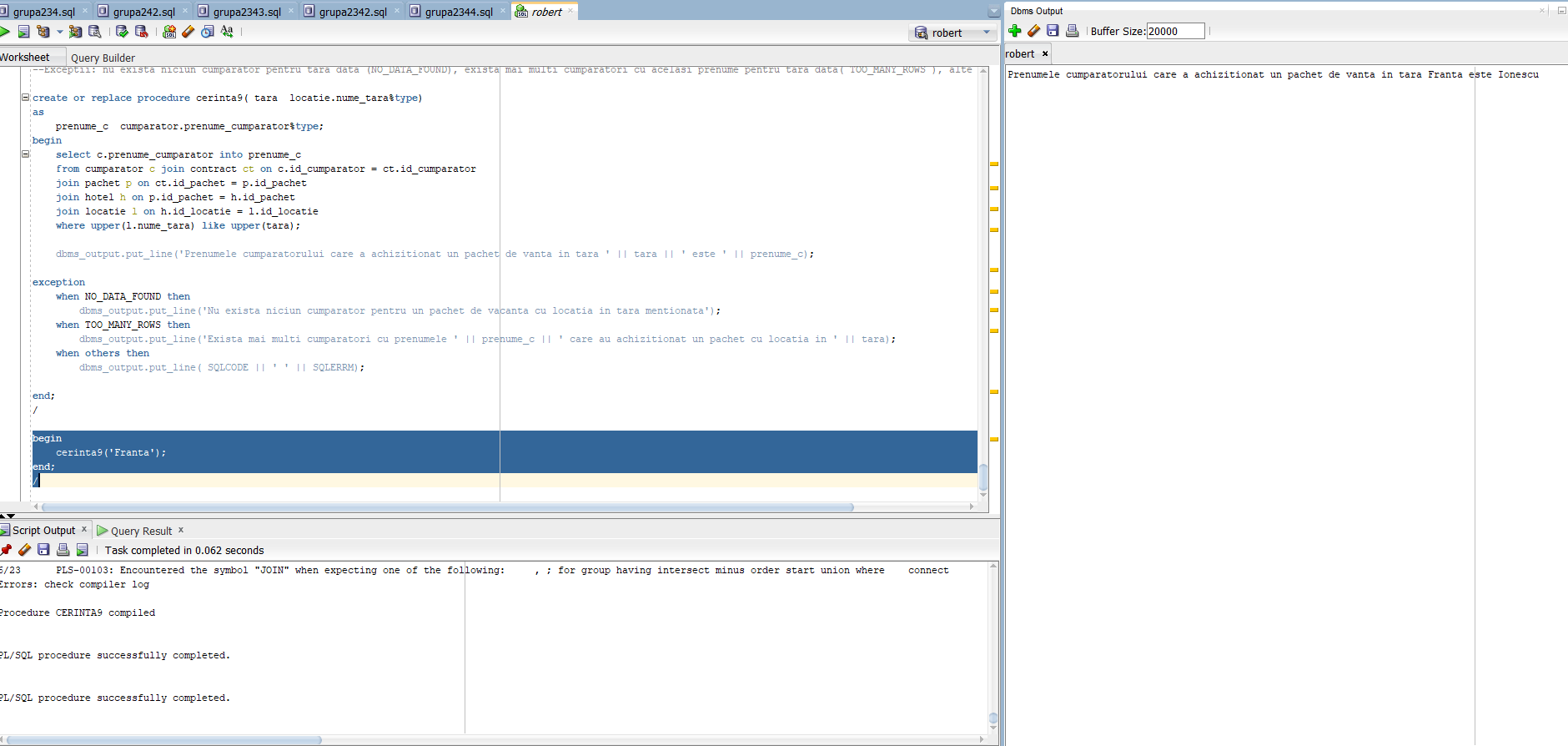
/

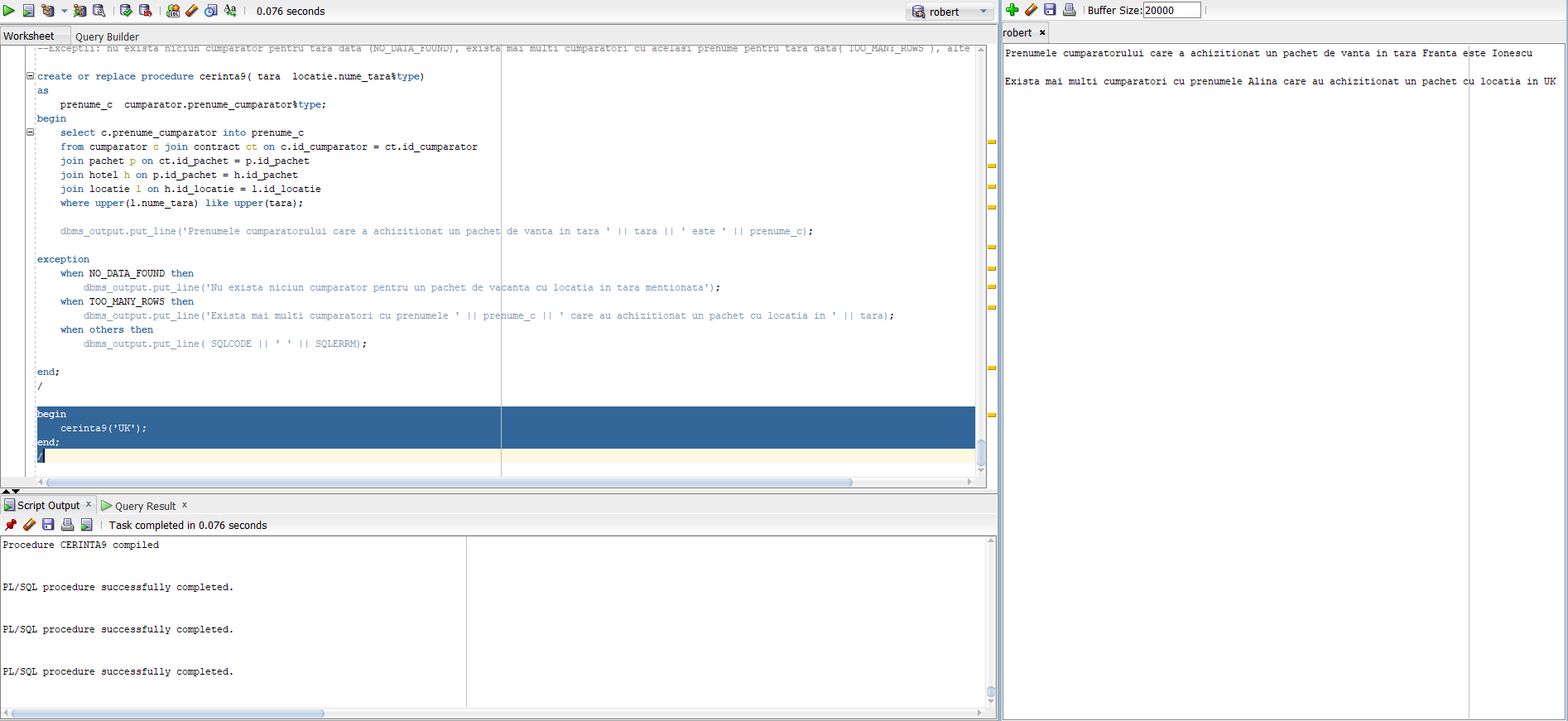
begin

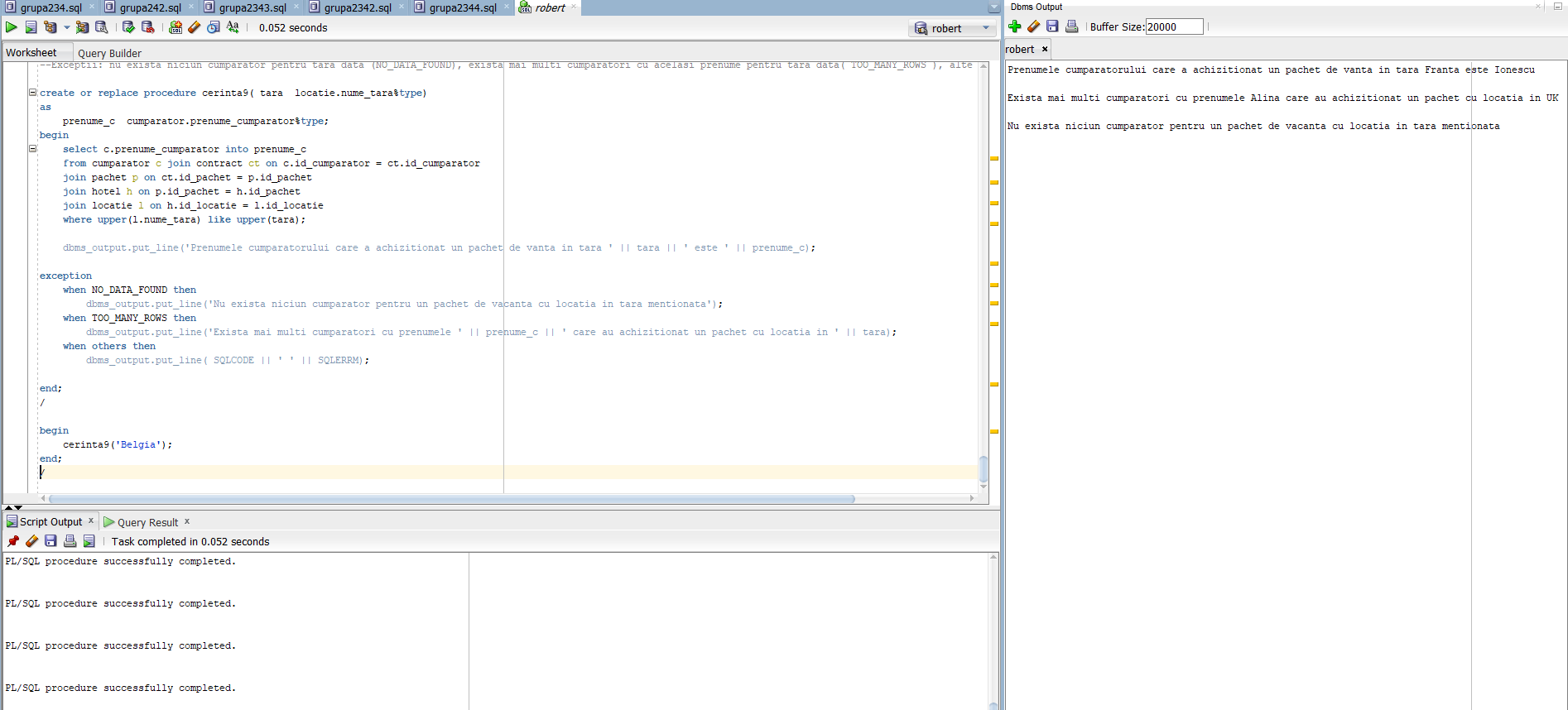
cerinta9('Belgia');

end;

/







# **Cerința 10**

Trigger pentru a nu permite inserarea unui angajat in cazul in care sunt deja 10 sau mai multi angajati;

create or replace trigger cerinta10

before insert on angajati

declare

v\_nr number(4);

begin

select count(id\_angajat) into v\_nr

from angajati;

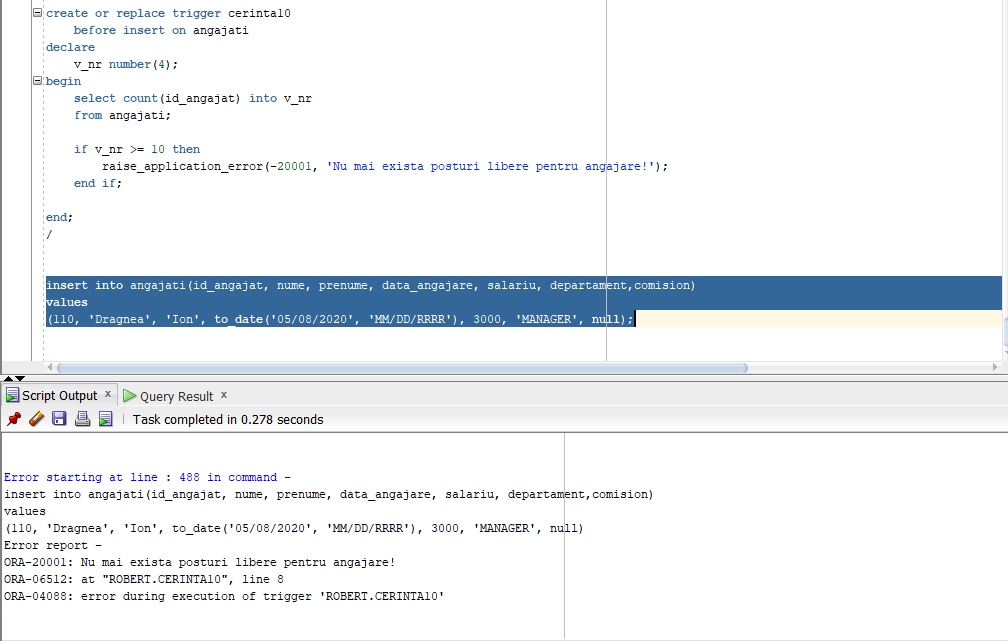
if v\_nr >= 10 then

raise\_application\_error(-20001, 'Nu mai exista posturi libere pentru angajare!');

end if;

end;

/



# **Cerința 11**

Trigger care sa nu permita cresterea salariului cu peste 2500 RON, intrucat este o crestere prea mare

create or replace trigger cerinta11

before update of salariu on angajati

for each row

begin

if( :NEW.salariu > :OLD.salariu + 2500 ) then

raise\_application\_error(-20002, 'Cresterea in salariu este prea mare');

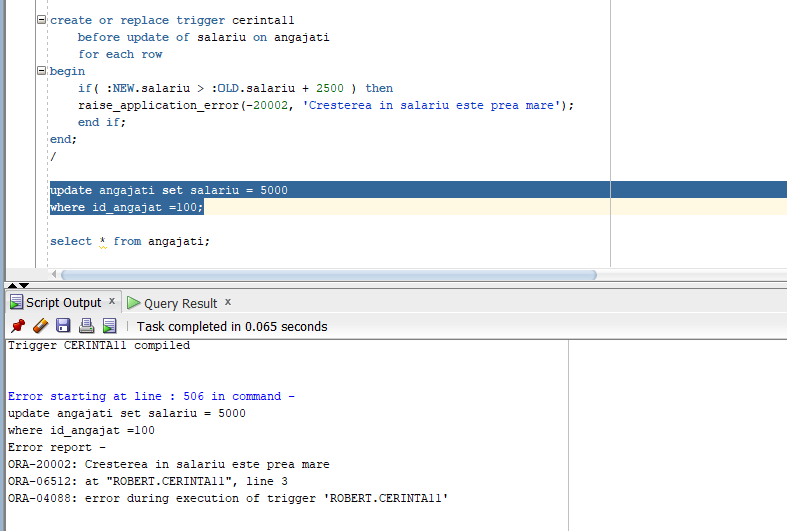
end if;

end;

/

update angajati set salariu = 5000

where id\_angajat =100;

****

# **Cerința 12**

Trigger care se declanseaza la fiecare operatie de tip LDD si care insereaza in tabela toate modificarile facute.

create table informatii (

baza\_date varchar2(40),

nume\_utilizator varchar2(40),

modificare varchar2(40),

data\_modificare date);

create or replace trigger cerinta12

after create or drop or alter on schema

begin

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

end;

/

create table temp (t1 varchar(20));

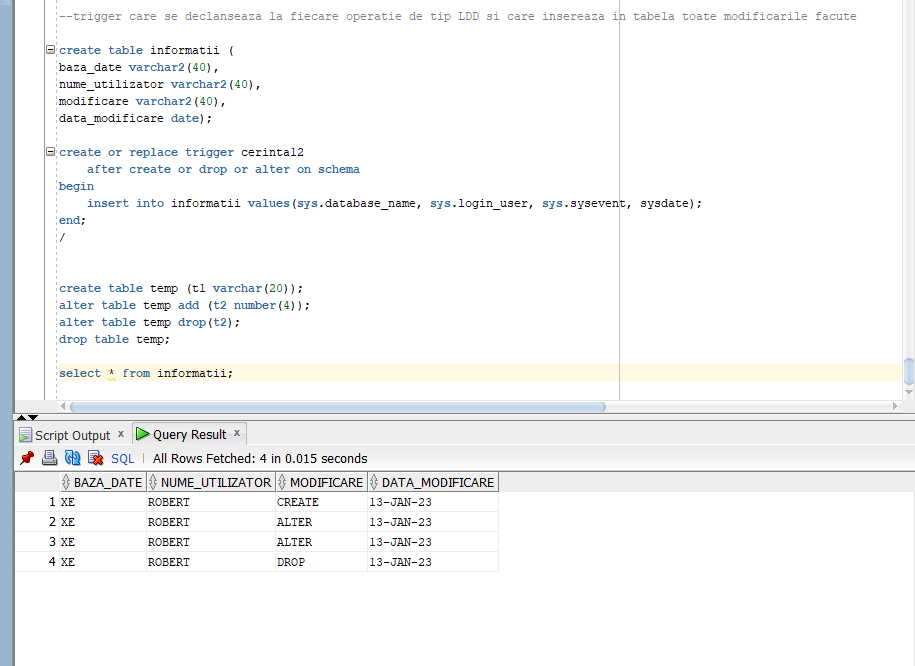
alter table temp add (t2 number(4));

alter table temp drop(t2);

drop table temp;

select \* from informatii;

drop trigger cerinta12;

****

# **Cerința 13**

create or replace package cerinta13

as

procedure cerinta6 (idtransport transport.id\_transport%type);

procedure cerinta7;

function cerinta8(tara locatie.nume\_tara%type) return number;

procedure cerinta9( tara locatie.nume\_tara%type);

end cerinta13;

/

create or replace package body cerinta13

as

procedure cerinta6 (idtransport transport.id\_transport%type)

as

type tablou\_indexat is table of pachet\_implica\_transport.id\_pachet%type index by pls\_integer;

t tablou\_indexat;

type tablou\_imbricat is table of pachet.denumire%type;

t2 tablou\_imbricat;

begin

select id\_pachet bulk collect into t

from pachet\_implica\_transport

where id\_transport = idtransport;

dbms\_output.put\_line('Sunt ' || t.count || ' pachete care se folosesc de transportul de tip ' || idtransport || ':');

for i in t.first..t.last

loop

dbms\_output.put\_line(t(i) || ' ');

end loop;

select p.denumire bulk collect into t2

from pachet p join pachet\_implica\_transport pi on p.id\_pachet = pi.id\_pachet

join transport t on pi.id\_transport = t.id\_transport

where t.id\_transport=idtransport;

dbms\_output.put\_line('Denumirile pachetelor sunt: ' || ' ');

for i in reverse t2.first..t2.last

loop

dbms\_output.put\_line(t2(i) || ' ');

end loop;

end;

procedure cerinta7

as

v\_nr number(4);

v\_dep angajati.departament%type;

cursor c is

select count(id\_angajat) , departament

from angajati

group by departament;

v\_nume angajati%rowtype;

cursor c1(salariu\_max number) is

select \*

from angajati

where salariu < salariu\_max;

begin

open c;

loop

fetch c into v\_nr, v\_dep;

exit when c%NOTFOUND;

if v\_nr =0 then

dbms\_output.put\_line('Niciun angajat nu lucreaza in departamentul ' || v\_dep);

elsif v\_nr =1 then

dbms\_output.put\_line( 'Un angajat lucreaza in departamentul ' || v\_dep);

else

dbms\_output.put\_line( v\_nr || ' angajati lucreaza in departamentul ' || v\_dep);

end if;

end loop;

close c;

open c1(2500);

loop

fetch c1 into v\_nume;

exit when c1%NOTFOUND;

dbms\_output.put\_line ( 'Angajatul ' || v\_nume.prenume || ' ' || v\_nume.nume || ' are salariul mai mic decat 2500');

end loop;

close c1;

end;

function cerinta8(tara locatie.nume\_tara%type) return number

is

numar\_rez number;

type tabel\_indexat is table of locatie%rowtype index by pls\_integer;

t tabel\_indexat;

exceptie1 exception;

exceptie2 exception;

begin

select \* bulk collect into t

from locatie

where upper(nume\_tara) like upper(tara);

if SQL%NOTFOUND then

raise exceptie1;

end if;

select count(p.id\_pachet) into numar\_rez

from pachet p join hotel h on p.id\_pachet = h.id\_pachet

join locatie l on h.id\_locatie = l.id\_locatie

where upper(l.nume\_tara) like upper(tara);

if numar\_rez = 0 then

raise exceptie2;

else

return numar\_rez;

end if;

exception

when exceptie1 then

dbms\_output.put\_line('Nu exista tara ' || tara || ' in tabelul de locatii');

return -1;

when exceptie2 then

dbms\_output.put\_line('Nu exista pachet care sa contina un hotel din tara ' || tara);

return -1;

when others then

dbms\_output.put\_line ( SQLCODE || ' ' || SQLERRM);

return -1;

end;

procedure cerinta9( tara locatie.nume\_tara%type)

as

prenume\_c cumparator.prenume\_cumparator%type;

begin

select c.prenume\_cumparator into prenume\_c

from cumparator c join contract ct on c.id\_cumparator = ct.id\_cumparator

join pachet p on ct.id\_pachet = p.id\_pachet

join hotel h on p.id\_pachet = h.id\_pachet

join locatie l on h.id\_locatie = l.id\_locatie

where upper(l.nume\_tara) like upper(tara);

dbms\_output.put\_line('Prenumele cumparatorului care a achizitionat un pachet de vanta in tara ' || tara || ' este ' || prenume\_c);

exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('Nu exista niciun cumparator pentru un pachet de vacanta cu locatia in tara mentionata');

when TOO\_MANY\_ROWS then

dbms\_output.put\_line('Exista mai multi cumparatori cu prenumele ' || prenume\_c || ' care au achizitionat un pachet cu locatia in ' || tara);

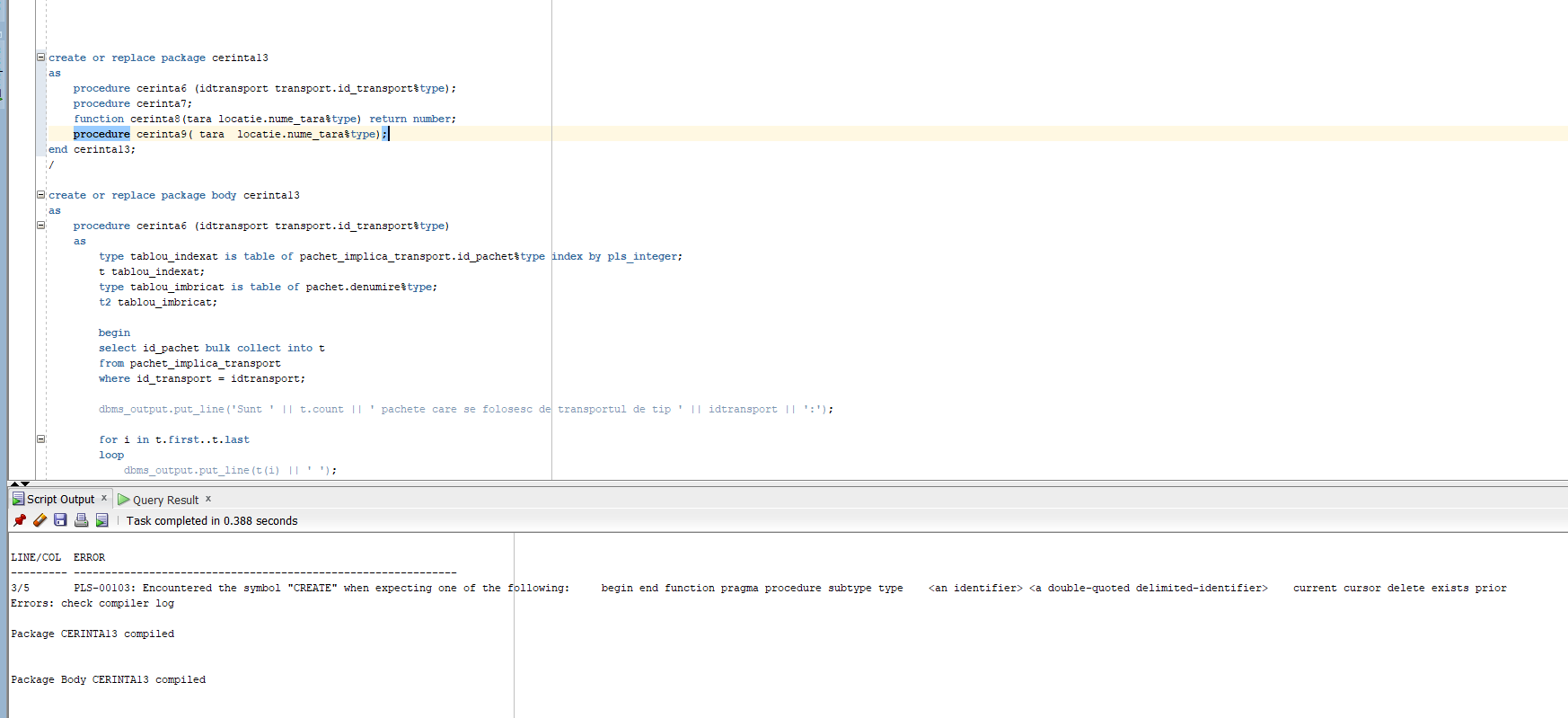
when others then

dbms\_output.put\_line( SQLCODE || ' ' || SQLERRM);

end;

end cerinta13;

/



# **Cerința 14**

create or replace package cerinta14

as

type tablou\_indexat is table of pachet\_implica\_transport.id\_pachet%type index by pls\_integer;

t tablou\_indexat;

cursor c is

select count(id\_angajat) , departament

from angajati

group by departament;

procedure cerinta6 (idtransport transport.id\_transport%type);

procedure cerinta7;

function cerinta8(tara locatie.nume\_tara%type) return number;

function cerinta91(tara locatie.nume\_tara%type)return cumparator.prenume\_cumparator%type;

procedure apelare (idtransport transport.id\_transport%type, tara locatie.nume\_tara%type);

end cerinta14;

/

create or replace package body cerinta14

as

procedure cerinta6 (idtransport transport.id\_transport%type)

as

type tablou\_imbricat is table of pachet.denumire%type;

t2 tablou\_imbricat;

begin

select id\_pachet bulk collect into t

from pachet\_implica\_transport

where id\_transport = idtransport;

dbms\_output.put\_line('Sunt ' || t.count || ' pachete care se folosesc de transportul de tip ' || idtransport || ':');

for i in t.first..t.last

loop

dbms\_output.put\_line(t(i) || ' ');

end loop;

select p.denumire bulk collect into t2

from pachet p join pachet\_implica\_transport pi on p.id\_pachet = pi.id\_pachet

join transport t on pi.id\_transport = t.id\_transport

where t.id\_transport=idtransport;

dbms\_output.put\_line('Denumirile pachetelor sunt: ' || ' ');

for i in reverse t2.first..t2.last

loop

dbms\_output.put\_line(t2(i) || ' ');

end loop;

end;

procedure cerinta7

as

v\_nr number(4);

v\_dep angajati.departament%type;

v\_nume angajati%rowtype;

cursor c1(salariu\_max number) is

select \*

from angajati

where salariu < salariu\_max;

begin

open c;

loop

fetch c into v\_nr, v\_dep;

exit when c%NOTFOUND;

if v\_nr =0 then

dbms\_output.put\_line('Niciun angajat nu lucreaza in departamentul ' || v\_dep);

elsif v\_nr =1 then

dbms\_output.put\_line( 'Un angajat lucreaza in departamentul ' || v\_dep);

else

dbms\_output.put\_line( v\_nr || ' angajati lucreaza in departamentul ' || v\_dep);

end if;

end loop;

close c;

open c1(2500);

loop

fetch c1 into v\_nume;

exit when c1%NOTFOUND;

dbms\_output.put\_line ( 'Angajatul ' || v\_nume.prenume || ' ' || v\_nume.nume || ' are salariul mai mic decat 2500');

end loop;

close c1;

end;

function cerinta8(tara locatie.nume\_tara%type) return number

is

numar\_rez number;

type tabel\_indexat is table of locatie%rowtype index by pls\_integer;

t tabel\_indexat;

exceptie1 exception;

exceptie2 exception;

begin

select \* bulk collect into t

from locatie

where upper(nume\_tara) like upper(tara);

if SQL%NOTFOUND then

raise exceptie1;

end if;

select count(p.id\_pachet) into numar\_rez

from pachet p join hotel h on p.id\_pachet = h.id\_pachet

join locatie l on h.id\_locatie = l.id\_locatie

where upper(l.nume\_tara) like upper(tara);

if numar\_rez = 0 then

raise exceptie2;

else

return numar\_rez;

end if;

exception

when exceptie1 then

dbms\_output.put\_line('Nu exista tara ' || tara || ' in tabelul de locatii');

return -1;

when exceptie2 then

dbms\_output.put\_line('Nu exista pachet care sa contina un hotel din tara ' || tara);

return -1;

when others then

dbms\_output.put\_line ( SQLCODE || ' ' || SQLERRM);

return -1;

end;

function cerinta91( tara locatie.nume\_tara%type) return cumparator.prenume\_cumparator%type

as

prenume\_c cumparator.prenume\_cumparator%type;

begin

select c.prenume\_cumparator into prenume\_c

from cumparator c join contract ct on c.id\_cumparator = ct.id\_cumparator

join pachet p on ct.id\_pachet = p.id\_pachet

join hotel h on p.id\_pachet = h.id\_pachet

join locatie l on h.id\_locatie = l.id\_locatie

where upper(l.nume\_tara) like upper(tara);

return prenume\_c;

exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('Nu exista niciun cumparator pentru un pachet de vacanta cu locatia in tara mentionata');

return -1;

when TOO\_MANY\_ROWS then

dbms\_output.put\_line('Exista mai multi cumparatori cu prenumele ' || prenume\_c || ' care au achizitionat un pachet cu locatia in ' || tara);

return -1;

when others then

dbms\_output.put\_line( SQLCODE || ' ' || SQLERRM);

return -1;

end;

procedure apelare(idtransport transport.id\_transport%type, tara locatie.nume\_tara%type)

as

exemplu locatie.nume\_tara%type;

exemplu2 cumparator.prenume\_cumparator%type;

begin

cerinta6(idtransport);

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

cerinta7();

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

exemplu := cerinta8(tara);

if exemplu> -1 then

dbms\_output.put\_line ( 'Exista ' || exemplu || ' pachete pentru tara data');

end if;

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

exemplu2 := cerinta91(tara);

if exemplu2> -1 then

dbms\_output.put\_line('Prenumele cumparatorului care a achizitionat un pachet de vanta in tara ' || tara || ' este ' || exemplu2);

end if;

end;

end cerinta14;

/

execute cerinta14.apelare(504, 'UK');

