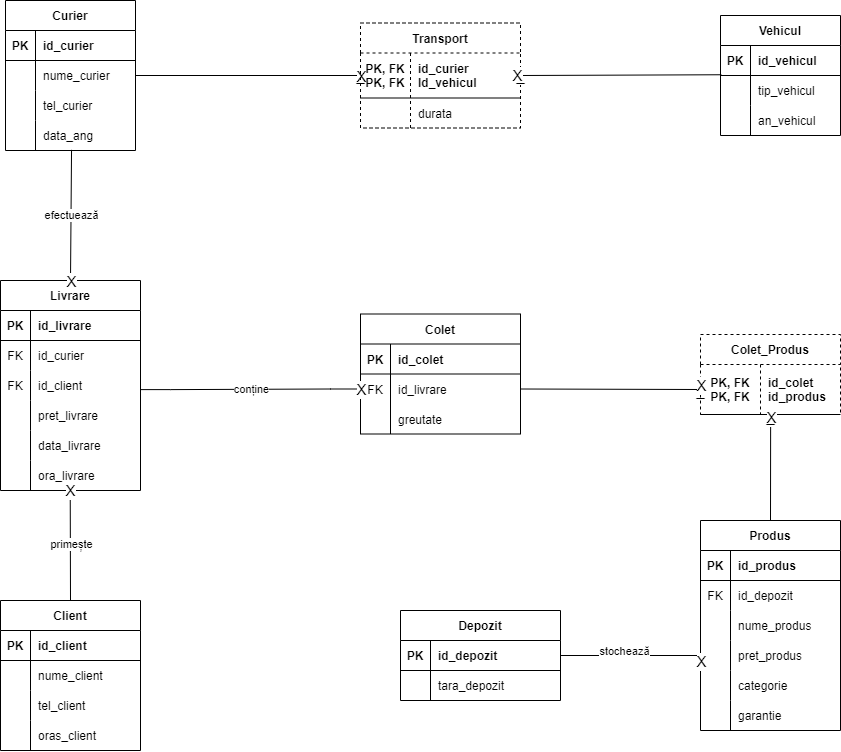
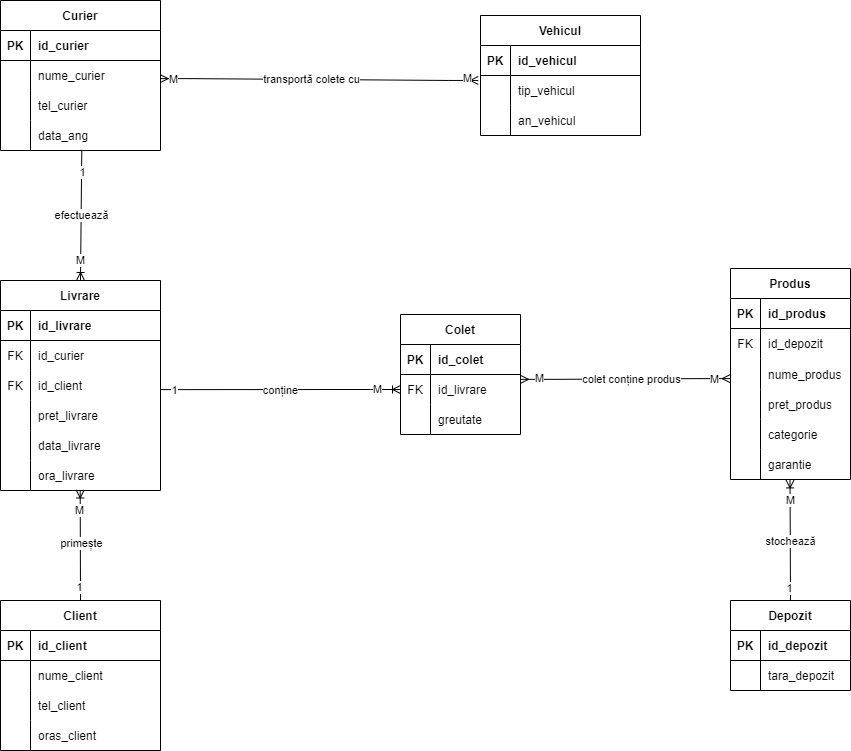
Livrarea coletelor de către o firmă de curierat

1. Tematica proiectului meu este proiectarea bazei de date a unei firme de curierat care se ocupă cu livrarea coletelor. În acest program voi prezenta traseul coletului de la curier la client (nu toată structura firmei de curierat). Baza de date conține informații cu privire la datele de contact ale curierului și clientului, particularitățile produselor și coletelor, detaliile livrării.

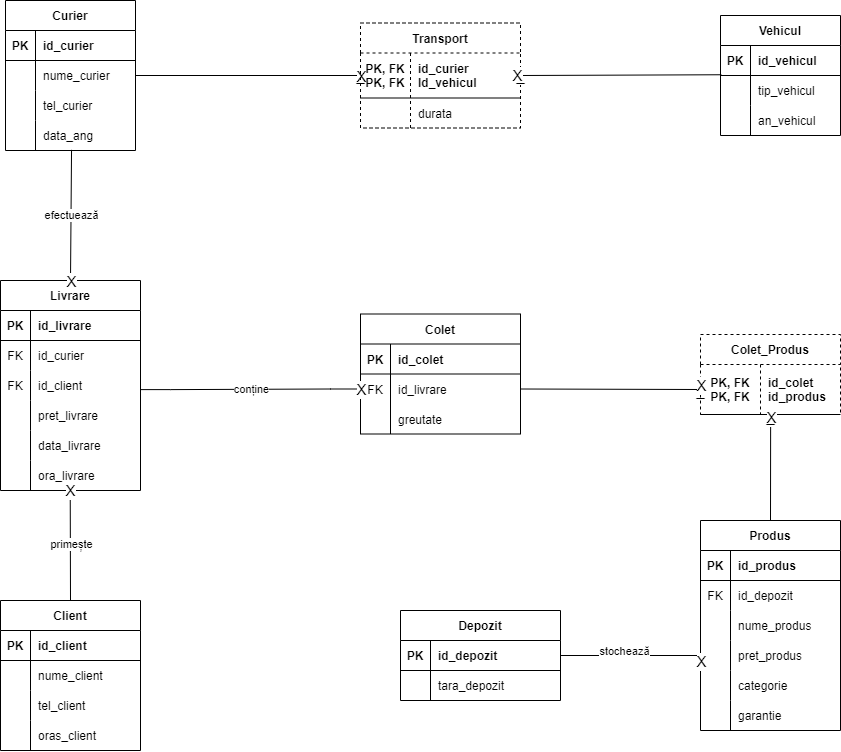
Câteva clarificări în legatură cu atributele unor entități: an\_vehicul – anul de fabricație al vehiculului; durata – durata de contract (exprimată în luni) pentru care curierul poate să foloseasca respectivul vehicul în scopul transportării coletelor; pret\_livrare – prețul transportului livrării, ci nu prețul coletelor; garantie – garanția produsului (exprimată în ani), poate fi NULL; oras\_client – orașul adresei de livrare a clientului; tel\_curier, tel\_client – pentru simplitate vor avea 4 cifre fiecare.



2. Diagrama Entitate-Relație (ERD)



3. Diagrama conceptuală



4.

create table vehicul(

id\_vehicul number(3) primary key,

tip\_vehicul varchar2(100),

an\_vehicul number(4)

);

create table curier(

id\_curier number(3) primary key,

nume\_curier varchar2(100),

tel\_curier number(4),

data\_ang date not null

);

create table transport(

id\_curier number(3) not null,

foreign key (id\_curier) references curier(id\_curier),

id\_vehicul number(3) not null,

foreign key (id\_vehicul) references vehicul(id\_vehicul),

durata number(3)

);

create table client(

id\_client number(3) primary key,

nume\_client varchar2(100),

tel\_client number(4),

oras\_client varchar2(100) not null

);

create table livrare(

id\_livrare number(3) primary key,

id\_curier number(3) not null,

foreign key (id\_curier) references curier(id\_curier),

id\_client number(3) not null,

foreign key (id\_client) references client(id\_client),

pret\_livrare number(10),

data\_livrare date,

ora\_livrare number(2)

);

create table colet(

id\_colet number(3) primary key,

id\_livrare number(3) not null,

foreign key (id\_livrare) references livrare(id\_livrare),

greutate number(5) not null

);

create table depozit(

id\_depozit number(3) primary key,

tara\_depozit varchar2(100)

);

create table produs(

id\_produs number(3) primary key,

id\_depozit number(3) not null,

foreign key (id\_depozit) references depozit(id\_depozit),

nume\_produs varchar2(200),

pret\_produs number(10),

categorie varchar2(100),

garantie number(3)

);

create table colet\_produs(

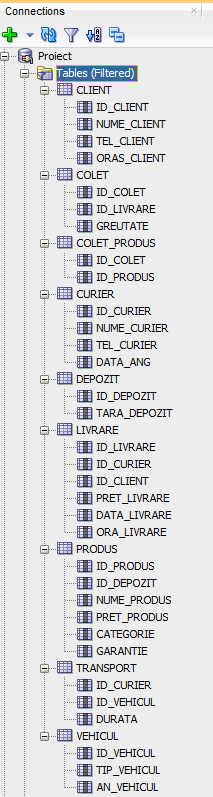
id\_colet number(3) not null,

foreign key (id\_colet) references colet(id\_colet),

id\_produs number(3) not null,

foreign key (id\_produs) references produs(id\_produs)

);



5.

insert into vehicul (id\_vehicul, tip\_vehicul, an\_vehicul)

values (700, 'tir', 2014);

insert into vehicul (id\_vehicul, tip\_vehicul, an\_vehicul)

values (701, 'duba', 2009);

insert into vehicul (id\_vehicul, tip\_vehicul, an\_vehicul)

values (702, 'avion', 2021);

insert into vehicul (id\_vehicul, tip\_vehicul, an\_vehicul)

values (703, 'duba', 2017);

insert into vehicul (id\_vehicul, tip\_vehicul, an\_vehicul)

values (704, 'duba', 2010);

insert into vehicul (id\_vehicul, tip\_vehicul, an\_vehicul)

values (705, 'tir', 2005);

insert into curier (id\_curier, nume\_curier, tel\_curier, data\_ang)

values (100, 'Vlad', 1567, '12-FEB-2009');

insert into curier (id\_curier, nume\_curier, tel\_curier, data\_ang)

values (101, 'John', 1891, '28-SEP-2014');

insert into curier (id\_curier, nume\_curier, tel\_curier, data\_ang)

values (102, 'Dan', 8505, '04-JUL-2022');

insert into curier (id\_curier, nume\_curier, tel\_curier, data\_ang)

values (103, 'Alex', 0912, '19-MAR-2009');

insert into curier (id\_curier, nume\_curier, tel\_curier, data\_ang)

values (104, 'David', 6480, '23-OCT-2015');

insert into curier (id\_curier, nume\_curier, tel\_curier, data\_ang)

values (105, 'Jack', 6157, '02-APR-2012');

insert into transport (id\_curier, id\_vehicul, durata)

values (100, 704, 12);

insert into transport (id\_curier, id\_vehicul, durata)

values (101, 700, 6);

insert into transport (id\_curier, id\_vehicul, durata)

values (102, 701, 24);

insert into transport (id\_curier, id\_vehicul, durata)

values (103, 705, 12);

insert into transport (id\_curier, id\_vehicul, durata)

values (104, 703, 36);

insert into transport (id\_curier, id\_vehicul, durata)

values (105, 702, 18);

insert into transport (id\_curier, id\_vehicul, durata)

values (100, 701, 18);

insert into transport (id\_curier, id\_vehicul, durata)

values (100, 703, 36);

insert into transport (id\_curier, id\_vehicul, durata)

values (101, 705, 24);

insert into transport (id\_curier, id\_vehicul, durata)

values (102, 703, 6);

insert into transport (id\_curier, id\_vehicul, durata)

values (102, 704, 12);

insert into transport (id\_curier, id\_vehicul, durata)

values (103, 700, 30);

insert into transport (id\_curier, id\_vehicul, durata)

values (104, 701, 12);

insert into transport (id\_curier, id\_vehicul, durata)

values (104, 704, 18);

insert into client (id\_client, nume\_client, tel\_client, oras\_client)

values (200, 'Raul', 1898, 'Tokyo');

insert into client (id\_client, nume\_client, tel\_client, oras\_client)

values (201, 'Laura', 7645, 'Roma');

insert into client (id\_client, nume\_client, tel\_client, oras\_client)

values (202, 'Chris', 3271, 'Londra');

insert into client (id\_client, nume\_client, tel\_client, oras\_client)

values (203, 'Eric', 6459, 'Barcelona');

insert into client (id\_client, nume\_client, tel\_client, oras\_client)

values (204, 'Maria', 1327, 'Miami');

insert into client (id\_client, nume\_client, tel\_client, oras\_client)

values (205, 'Gabi', 5573, 'Bucuresti');

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (300, 100, 205, 20, '02-JAN-2023', 17);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (301, 100, 200, 90, '17-DEC-2021', 12);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (302, 100, 201, 0, '30-JUN-2019', 9);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (303, 101, 205, 120, '09-JUL-2022', 20);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (304, 102, 202, 10, '15-OCT-2019', 8);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (305, 102, 203, 50, '23-MAR-2020', 13);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (306, 103, 205, 30, '27-SEP-2018', 10);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (307, 104, 200, 100, '11-APR-2019', 19);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (308, 104, 204, 60, '04-SEP-2017', 21);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (309, 105, 205, 25, '15-AUG-2022', 8);

insert into livrare (id\_livrare, id\_curier, id\_client, pret\_livrare, data\_livrare, ora\_livrare)

values (310, 105, 200, 40, '20-AUG-2020', 17);

insert into colet (id\_colet, id\_livrare, greutate)

values (400, 302, 4);

insert into colet (id\_colet, id\_livrare, greutate)

values (401, 307, 1);

insert into colet (id\_colet, id\_livrare, greutate)

values (402, 305, 12);

insert into colet (id\_colet, id\_livrare, greutate)

values (403, 308, 8);

insert into colet (id\_colet, id\_livrare, greutate)

values (404, 306, 3);

insert into colet (id\_colet, id\_livrare, greutate)

values (405, 300, 10);

insert into colet (id\_colet, id\_livrare, greutate)

values (406, 310, 5);

insert into colet (id\_colet, id\_livrare, greutate)

values (407, 304, 8);

insert into colet (id\_colet, id\_livrare, greutate)

values (408, 309, 4);

insert into colet (id\_colet, id\_livrare, greutate)

values (409, 301, 6);

insert into colet (id\_colet, id\_livrare, greutate)

values (410, 307, 20);

insert into colet (id\_colet, id\_livrare, greutate)

values (411, 303, 50);

insert into depozit (id\_depozit, tara\_depozit)

values (600, 'China');

insert into depozit (id\_depozit, tara\_depozit)

values (601, 'Belgia');

insert into depozit (id\_depozit, tara\_depozit)

values (602, 'Romania');

insert into depozit (id\_depozit, tara\_depozit)

values (603, 'Mexic');

insert into depozit (id\_depozit, tara\_depozit)

values (604, 'Franta');

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (500, 601, 'TV', 4000, 'electronice', 2);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (501, 603, 'creion', 2, 'rechizite', null);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (502, 604, 'boxe', 800, 'electronice', 3);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (503, 601, 'telefon', 3000, 'electronice', 1);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (504, 604, 'dulap', 1500, 'mobila', 1);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (505, 602, 'stilou', 70, 'rechizite', null);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (506, 600, 'masa', 600, 'mobila', 1);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (507, 603, 'canapea', 5000, 'mobila', 2);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (508, 601, 'catan', 300, 'jocuri', null);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (509, 602, 'scaun', 700, 'mobila', 1);

insert into produs (id\_produs, id\_depozit, nume\_produs, pret\_produs, categorie, garantie)

values (510, 601, 'activity', 200, 'jocuri', null);

insert into colet\_produs(id\_colet, id\_produs)

values (400, 508);

insert into colet\_produs(id\_colet, id\_produs)

values (400, 510);

insert into colet\_produs(id\_colet, id\_produs)

values (401, 501);

insert into colet\_produs(id\_colet, id\_produs)

values (401, 505);

insert into colet\_produs(id\_colet, id\_produs)

values (402, 500);

insert into colet\_produs(id\_colet, id\_produs)

values (403, 503);

insert into colet\_produs(id\_colet, id\_produs)

values (403, 506);

insert into colet\_produs(id\_colet, id\_produs)

values (404, 507);

insert into colet\_produs(id\_colet, id\_produs)

values (405, 504);

insert into colet\_produs(id\_colet, id\_produs)

values (405, 510);

insert into colet\_produs(id\_colet, id\_produs)

values (406, 500);

insert into colet\_produs(id\_colet, id\_produs)

values (406, 506);

insert into colet\_produs(id\_colet, id\_produs)

values (406, 508);

insert into colet\_produs(id\_colet, id\_produs)

values (406, 502);

insert into colet\_produs(id\_colet, id\_produs)

values (407, 503);

insert into colet\_produs(id\_colet, id\_produs)

values (408, 505);

insert into colet\_produs(id\_colet, id\_produs)

values (408, 509);

insert into colet\_produs(id\_colet, id\_produs)

values (409, 510);

insert into colet\_produs(id\_colet, id\_produs)

values (409, 501);

insert into colet\_produs(id\_colet, id\_produs)

values (410, 509);

insert into colet\_produs(id\_colet, id\_produs)

values (410, 506);

insert into colet\_produs(id\_colet, id\_produs)

values (411, 502);

6. Un client dorește să afle din ce țări provin produsele din colet. Astfel, să se afișeze toate țările cu depozite care stochează produse aflate în coletul cu codul dat ca parametru.

create or replace procedure e6

(cod\_colet colet\_produs.id\_colet%type)

as

type t\_id\_produs is table of colet\_produs.id\_produs%type;

t t\_id\_produs;

type t\_id\_depozit is table of produs.id\_depozit%type index by PLS\_INTEGER;

z t\_id\_depozit;

type dif is varray(6) of number(1);

d dif := dif(0, 0, 0, 0, 0);

cnt number(3);

a number(3);

i number(3);

verif number(3) := 0;

cod\_depozit produs.id\_depozit%type;

aux depozit.tara\_depozit%type;

begin

select unique id\_colet into verif

from colet\_produs

where id\_colet = cod\_colet;

if verif = 0 then

DBMS\_OUTPUT.PUT\_LINE('Coletul ' || cod\_colet || ' nu exista.');

else

select id\_produs bulk collect into t

from colet\_produs

where id\_colet = cod\_colet;

select count(\*) into cnt

from colet\_produs

where id\_colet = cod\_colet;

if cnt = 0 then

DBMS\_OUTPUT.PUT\_LINE('Coletul ' || cod\_colet || ' nu contine niciun produs.');

else

DBMS\_OUTPUT.PUT\_LINE('Coletul ' || cod\_colet || ' contine produse din urmatoarele tari:');

for i in 1..cnt loop

select id\_depozit into z(i)

from produs

where t(i) = id\_produs;

select tara\_depozit into aux

from depozit

where id\_depozit = z(i);

a := z(i) - 599;

if d(a) = 0 then

d(a) := 1;

DBMS\_OUTPUT.PUT\_LINE(aux);

end if;

end loop;

end if;

end if;

end;

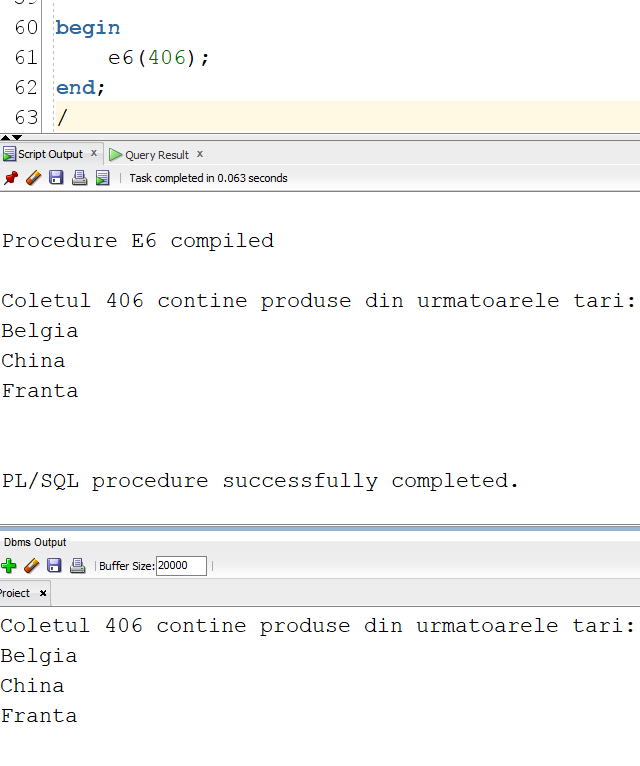
/

begin

e6(406);

end;

/



7. Să se afișeze prețul total al livrărilor efectuate de curierul cu numele dat ca prim parametru, respectiv numărul de colete care au maxim greutatea dată ca al doilea parametru și câte produse conțin în total aceste colete.

create or replace procedure e7

(nume curier.nume\_curier%type,

gr colet.greutate%type)

as

cursor cp (x colet.greutate%type) is

select id\_colet

from colet

where greutate <= x;

cursor cn is

select pret\_livrare pl

from livrare

where id\_curier = (select id\_curier

from curier

where nume\_curier = nume);

cod\_colet colet.id\_colet%type;

suma number(10) := 0;

nr\_colete number(3) := 0;

nr\_produse number(4) := 0;

aux number(4);

begin

for i in cn loop

suma := suma + i.pl;

end loop;

DBMS\_OUTPUT.PUT\_LINE('Pretul total al livrarilor efectuate de curierul '|| nume || ' este ' || suma || '.');

open cp(gr);

loop

fetch cp into cod\_colet;

exit when cp%notfound;

select count(\*) into aux

from colet\_produs

where cod\_colet = id\_colet;

nr\_colete := nr\_colete + 1;

nr\_produse := nr\_produse + aux;

end loop;

close cp;

DBMS\_OUTPUT.PUT\_LINE('Sunt ' || nr\_colete || ' colete care au maxim ' || gr || ' kg si contin in total ' || nr\_produse || ' produse.');

end;

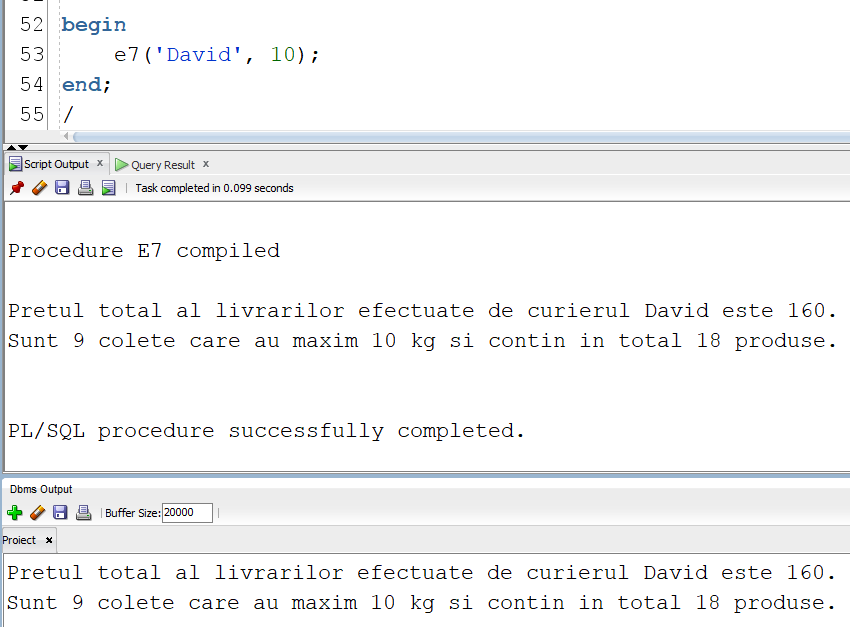
/

begin

e7('David', 10);

end;

/



8. Să se afișeze câte vehicule poate conduce un curier cu numele dat ca prim parametru. Vehiculele trebuie să fie produse după anul dat ca al doilea parametru.

create or replace function e8

(nume curier.nume\_curier%type,

an vehicul.an\_vehicul%type)

return number

is

rez number(3);

v number(3);

curryear number(4);

begin

select count(\*) into v

from curier

where nume\_curier = nume;

if v = 0 then

--DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun curier cu numele ' || nume || '.');

RAISE\_APPLICATION\_ERROR(-20001, 'Nu exista niciun curier cu numele ' || nume || '.');

return -1;

end if;

select to\_char(sysdate, 'YYYY') into curryear

from dual;

if an < 0 or an > curryear then

--DBMS\_OUTPUT.PUT\_LINE('Anul introdus este invalid.');

RAISE\_APPLICATION\_ERROR(-20002, 'Anul introdus este invalid.');

return -1;

end if;

select count(\*) into rez

from vehicul v, transport t, curier c

where an < v.an\_vehicul and

v.id\_vehicul = t.id\_vehicul and

t.id\_curier = c.id\_curier and

c.nume\_curier = nume;

DBMS\_OUTPUT.PUT\_LINE('Numarul de vehicule produse dupa anul ' || an || ' pe care le poate conduce curierul ' || nume || ' este: ');

return rez;

end;

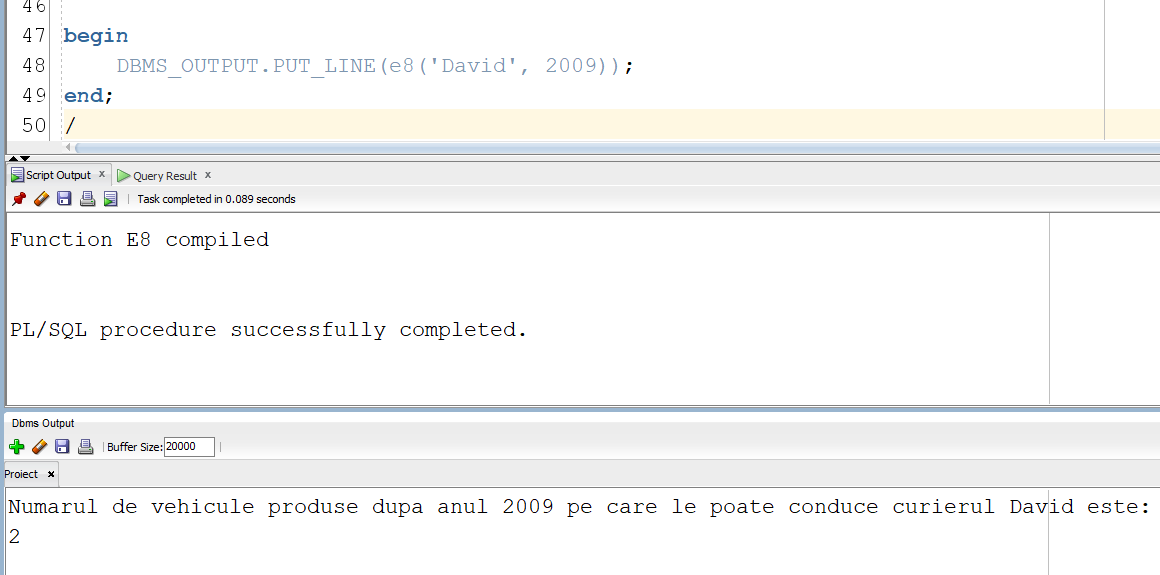
/

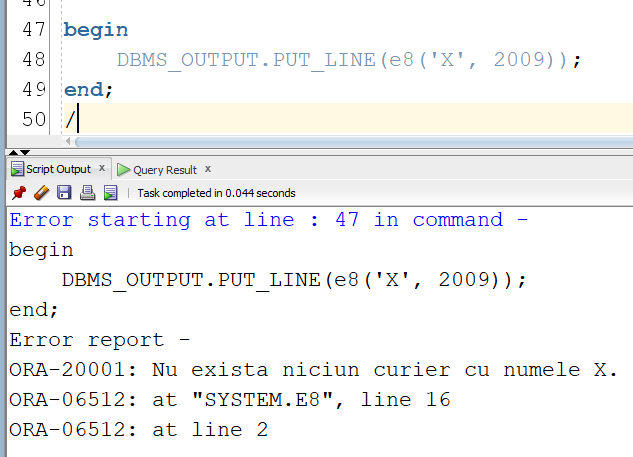
begin

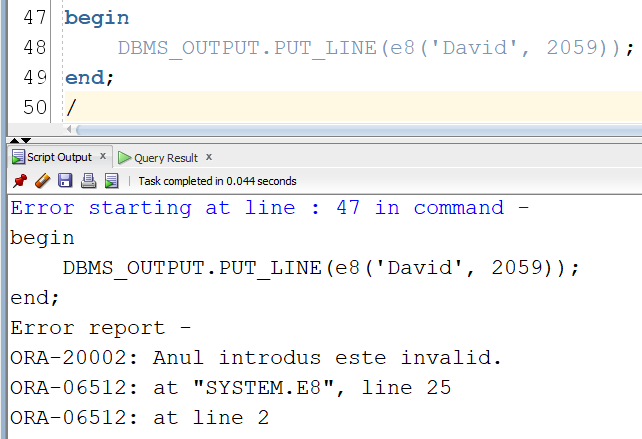
DBMS\_OUTPUT.PUT\_LINE(e8('David', 2008));

end;

/







9. Să se afișeze câți lei a cheltuit în total pe produse clientul cu numele dat ca parametru.

create or replace procedure ex9

(nume client.nume\_client%type)

as

suma number(11);

vf client.id\_client%type;

begin

-- folosesc acest query pentru a vedea

-- daca numele clientului este unul valid

-- astfel, daca numele nu exista in baza

-- de date, va interveni eroarea

select id\_client into vf

from client

where nume\_client = nume;

select sum(p.pret\_produs) into suma

from client c

join livrare l on c.id\_client = l.id\_client

join colet co on l.id\_livrare = co.id\_livrare

join colet\_produs cp on co.id\_colet = cp.id\_colet

join produs p on cp.id\_produs = p.id\_produs

where c.nume\_client = nume;

DBMS\_OUTPUT.PUT\_LINE(nume || ' a cumparat produse in valoare de ' || suma || ' lei.');

exception

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Clientul ' || nume || ' nu exista.');

--DBMS\_OUTPUT.PUT\_LINE('Clientul ' || nume || ' nu exista.');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Too many rows.');

end;

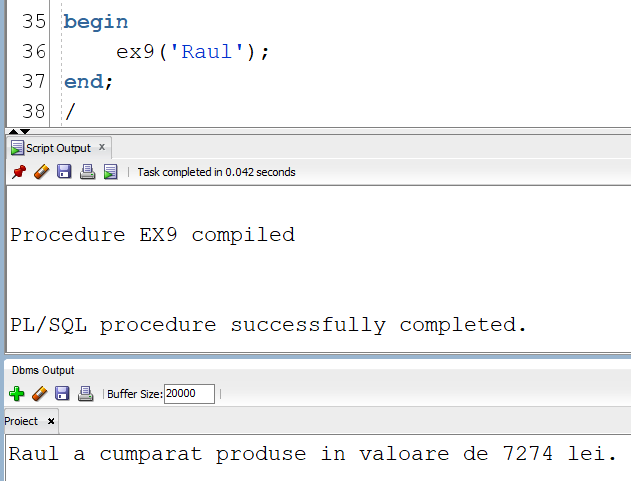
/

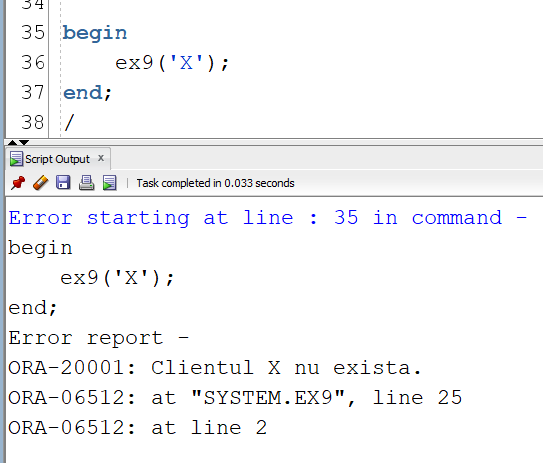
begin

ex9('Raul');

end;

/





10. În urma unui studiu, am aflat că este profitabil ca firma noastra de curierat să aibă depozite în cât mai puține țări diferite. Creați un trigger care să nu permită să se insereze (înființeze) depozite în mai mult de 8 țări diferite.

create or replace trigger limitare\_depozite

before insert or update on depozit

declare

cnt number(3);

begin

select count(distinct tara\_depozit) into cnt

from depozit;

if cnt > 8 then

RAISE\_APPLICATION\_ERROR(-20100, 'S-a atins limita de tari diferite inca care putem avea depozite.');

end if;

end;

/

begin

insert into depozit (id\_depozit, tara\_depozit)

values (605, 'Anglia');

insert into depozit (id\_depozit, tara\_depozit)

values (606, 'Grecia');

insert into depozit (id\_depozit, tara\_depozit)

values (607, 'Egypt');

insert into depozit (id\_depozit, tara\_depozit)

values (608, 'Peru');

insert into depozit (id\_depozit, tara\_depozit)

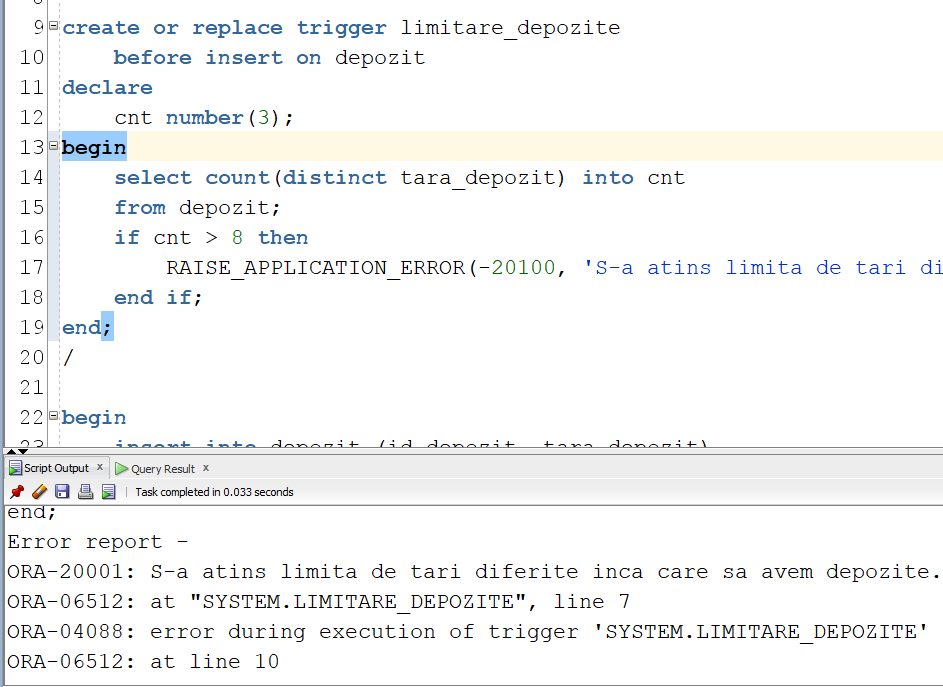
values (609, 'India');

-- delete from depozit where id\_depozit > 604;

end;

/

drop trigger limitare\_depozite;



11. Fiind sezonul reducerilor, majoritatea produselor vor fi reduse și nu se poate mări prețul acestora. Creați un trigger prin care să nu se permită creșterea prețului produselor.

create or replace trigger crestere\_pret

before update of pret\_produs on produs

for each row

when (new.pret\_produs > old.pret\_produs)

begin

RAISE\_APPLICATION\_ERROR(-20101, 'Pretul produselor nu poate fi majorat.');

end;

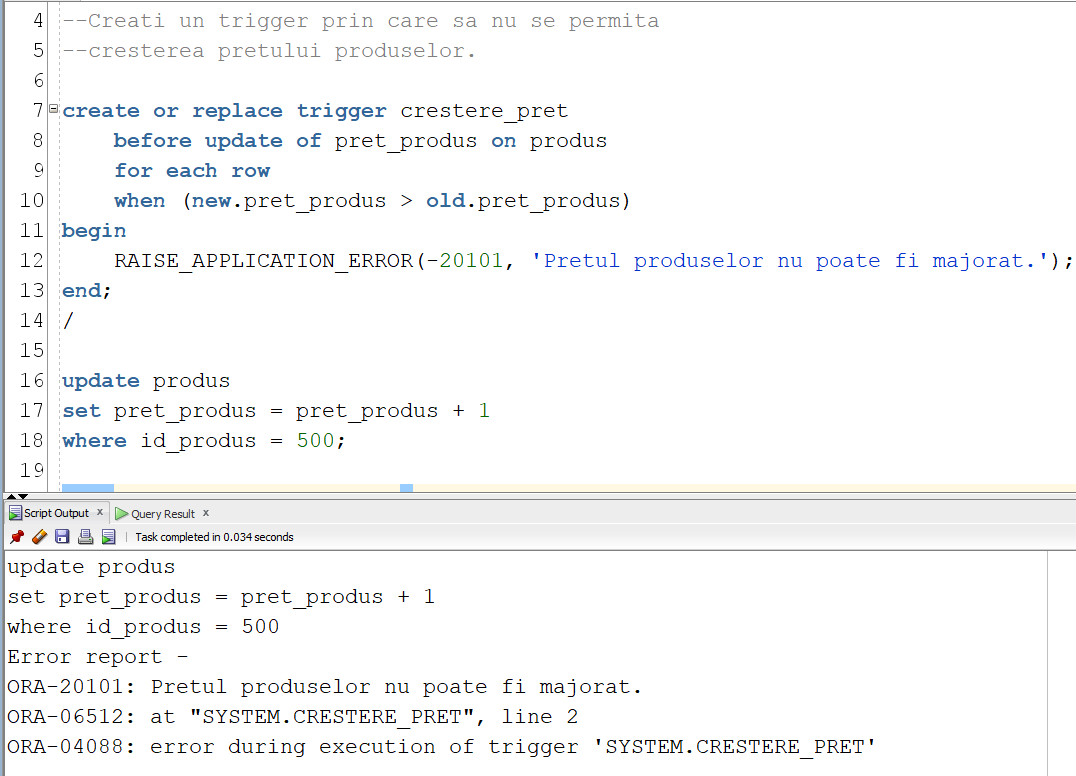
/

update produs

set pret\_produs = pret\_produs + 1

where id\_produs = 500;

drop trigger crestere\_pret;



12. Creați un trigger care să îi dea permisiuni de modificare a schemei doar userului 'SYSTEM'. Salvați toate modificările în tabelul audit\_user.

create table audit\_user(

utilizator varchar2(30),

nume\_bd varchar2(50),

eveniment varchar2(20),

nume\_obiect varchar2(30),

data date

);

create or replace trigger audit\_schema

before create or alter or drop on schema

begin

if user != 'SYSTEM' then

RAISE\_APPLICATION\_ERROR(-20102, 'Doar admin-ul are dreptul sa modifice schema.');

end if;

insert into audit\_user

values (sys.login\_user, sys.database\_name, sys.sysevent, sys.dictionary\_obj\_name, sysdate);

end;

/

alter table depozit

add supraf number(4);

alter table depozit

add nr\_zone number(4);

rollback;

create table ambalaj (

id\_ambalaj number(3) primary key,

id\_produs number(3) not null,

foreign key (id\_produs) references produs(id\_produs),

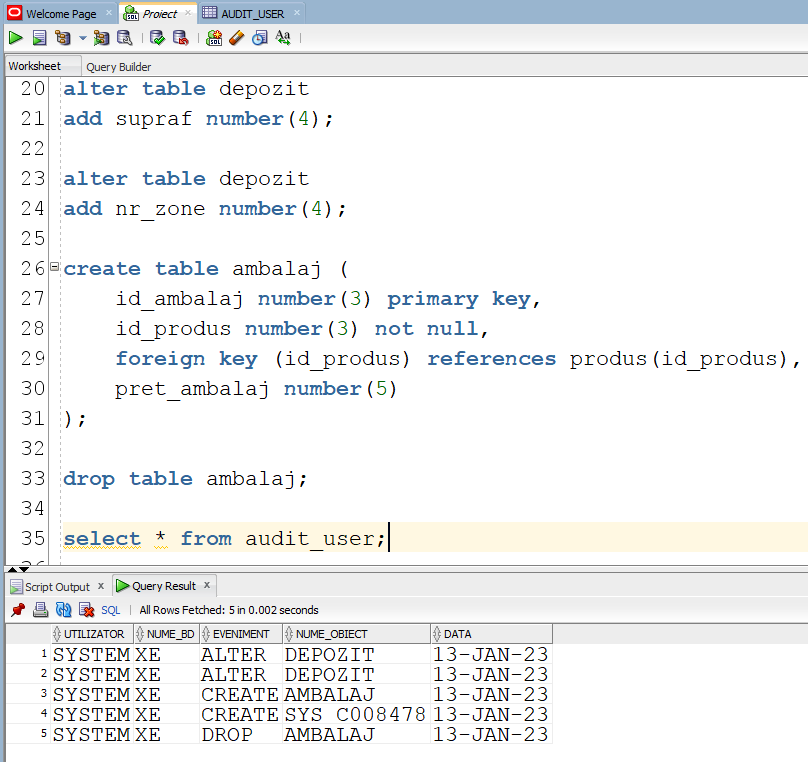
pret\_ambalaj number(5)

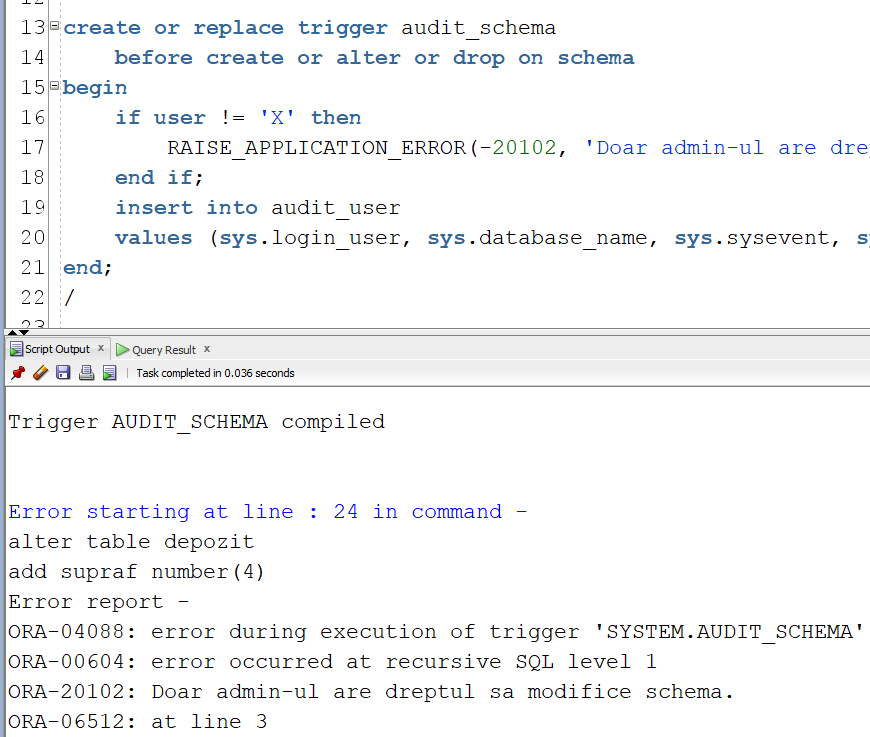
);

drop table ambalaj;

--select \* from audit\_user;

drop trigger audit\_schema;





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

create or replace package e13 as

procedure e6 (cod\_colet colet\_produs.id\_colet%type);

procedure e7 (nume curier.nume\_curier%type,

gr colet.greutate%type);

function e8 (nume curier.nume\_curier%type,

an vehicul.an\_vehicul%type)

return number;

procedure ex9 (nume client.nume\_client%type);

end e13;

/

create or replace package body e13 as

procedure e6 (cod\_colet colet\_produs.id\_colet%type)

as

type t\_id\_produs is table of colet\_produs.id\_produs%type;

t t\_id\_produs;

type t\_id\_depozit is table of produs.id\_depozit%type index by PLS\_INTEGER;

z t\_id\_depozit;

type dif is varray(6) of number(1);

d dif := dif(0, 0, 0, 0, 0);

cnt number(3);

a number(3);

i number(3);

verif number(3) := 0;

cod\_depozit produs.id\_depozit%type;

aux depozit.tara\_depozit%type;

begin

select unique id\_colet into verif

from colet\_produs

where id\_colet = cod\_colet;

if verif = 0 then

DBMS\_OUTPUT.PUT\_LINE('Coletul ' || cod\_colet || ' nu exista.');

else

select id\_produs bulk collect into t

from colet\_produs

where id\_colet = cod\_colet;

select count(\*) into cnt

from colet\_produs

where id\_colet = cod\_colet;

if cnt = 0 then

DBMS\_OUTPUT.PUT\_LINE('Coletul ' || cod\_colet || ' nu contine niciun produs.');

else

DBMS\_OUTPUT.PUT\_LINE('Coletul ' || cod\_colet || ' contine produse din urmatoarele tari:');

for i in 1..cnt loop

select id\_depozit into z(i)

from produs

where t(i) = id\_produs;

select tara\_depozit into aux

from depozit

where id\_depozit = z(i);

a := z(i) - 599;

if d(a) = 0 then

d(a) := 1;

DBMS\_OUTPUT.PUT\_LINE(aux);

end if;

end loop;

end if;

end if;

end e6;

procedure e7 (nume curier.nume\_curier%type,

gr colet.greutate%type)

as

cursor cp (x colet.greutate%type) is

select id\_colet

from colet

where greutate <= x;

cursor cn is

select pret\_livrare pl

from livrare

where id\_curier = (select id\_curier

from curier

where nume\_curier = nume);

cod\_colet colet.id\_colet%type;

suma number(10) := 0;

nr\_colete number(3) := 0;

nr\_produse number(4) := 0;

aux number(4);

begin

for i in cn loop

suma := suma + i.pl;

end loop;

DBMS\_OUTPUT.PUT\_LINE('Pretul total al livrarilor efectuate de curierul '|| nume || ' este ' || suma || '.');

open cp(gr);

loop

fetch cp into cod\_colet;

exit when cp%notfound;

select count(\*) into aux

from colet\_produs

where cod\_colet = id\_colet;

nr\_colete := nr\_colete + 1;

nr\_produse := nr\_produse + aux;

end loop;

close cp;

DBMS\_OUTPUT.PUT\_LINE('Sunt ' || nr\_colete || ' colete care au maxim ' || gr || ' kg si contin in total ' || nr\_produse || ' produse.');

end e7;

function e8 (nume curier.nume\_curier%type,

an vehicul.an\_vehicul%type)

return number

is

rez number(3);

v number(3);

curryear number(4);

begin

select count(\*) into v

from curier

where nume\_curier = nume;

if v = 0 then

--DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun curier cu numele ' || nume || '.');

RAISE\_APPLICATION\_ERROR(-20001, 'Nu exista niciun curier cu numele ' || nume || '.');

return -1;

end if;

select to\_char(sysdate, 'YYYY') into curryear

from dual;

if an < 0 or an > curryear then

--DBMS\_OUTPUT.PUT\_LINE('Anul introdus este invalid.');

RAISE\_APPLICATION\_ERROR(-20002, 'Anul introdus este invalid.');

return -1;

end if;

select count(\*) into rez

from vehicul v, transport t, curier c

where an < v.an\_vehicul and

v.id\_vehicul = t.id\_vehicul and

t.id\_curier = c.id\_curier and

c.nume\_curier = nume;

DBMS\_OUTPUT.PUT\_LINE('Numarul de vehicule produse dupa anul ' || an || ' pe care le poate conduce curierul ' || nume || ' este: ');

return rez;

end e8;

procedure ex9 (nume client.nume\_client%type)

as

suma number(11);

vf client.id\_client%type;

begin

select id\_client into vf

from client

where nume\_client = nume;

select sum(p.pret\_produs) into suma

from client c

join livrare l on c.id\_client = l.id\_client

join colet co on l.id\_livrare = co.id\_livrare

join colet\_produs cp on co.id\_colet = cp.id\_colet

join produs p on cp.id\_produs = p.id\_produs

where c.nume\_client = nume;

DBMS\_OUTPUT.PUT\_LINE(nume || ' a cumparat produse in valoare de ' || suma || ' lei.');

exception

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Clientul ' || nume || ' nu exista.');

--DBMS\_OUTPUT.PUT\_LINE('Clientul ' || nume || ' nu exista.');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Too many rows.');

end ex9;

end e13;

/

