Bază de date de supraveghere

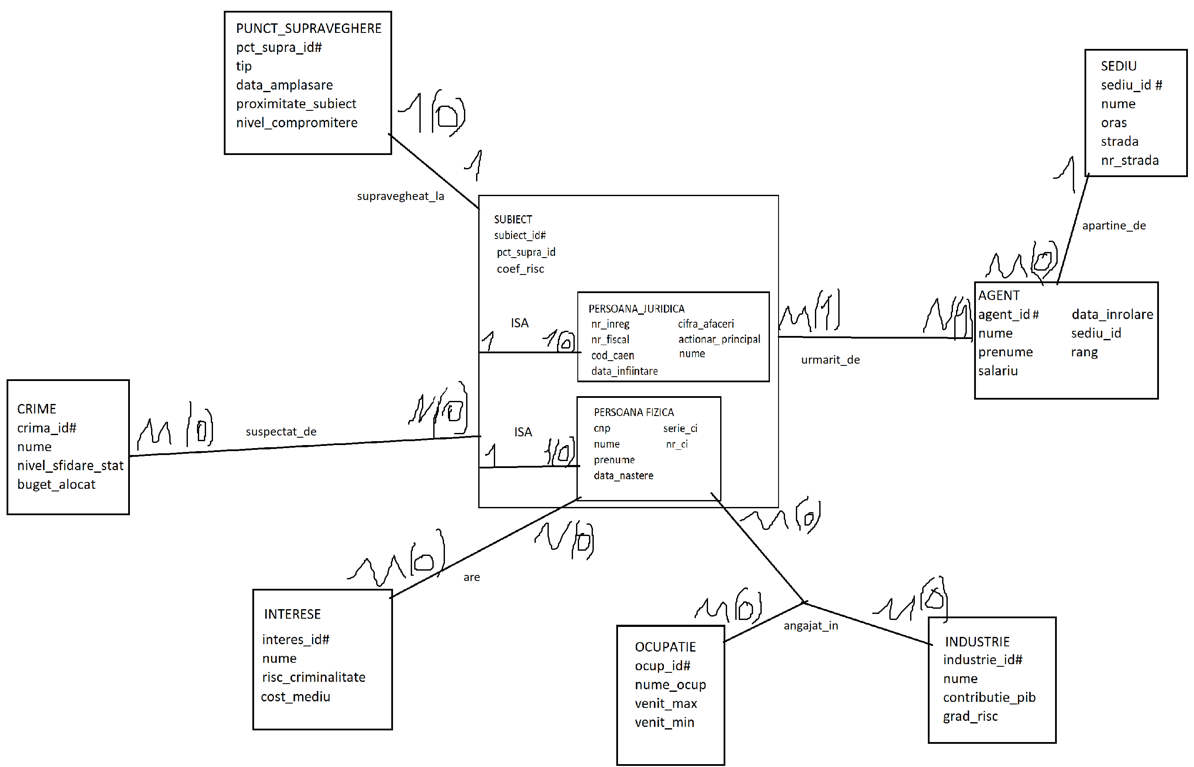
a Serviciului Român de Informații

Tudose Alexandru-Ștefan, 231

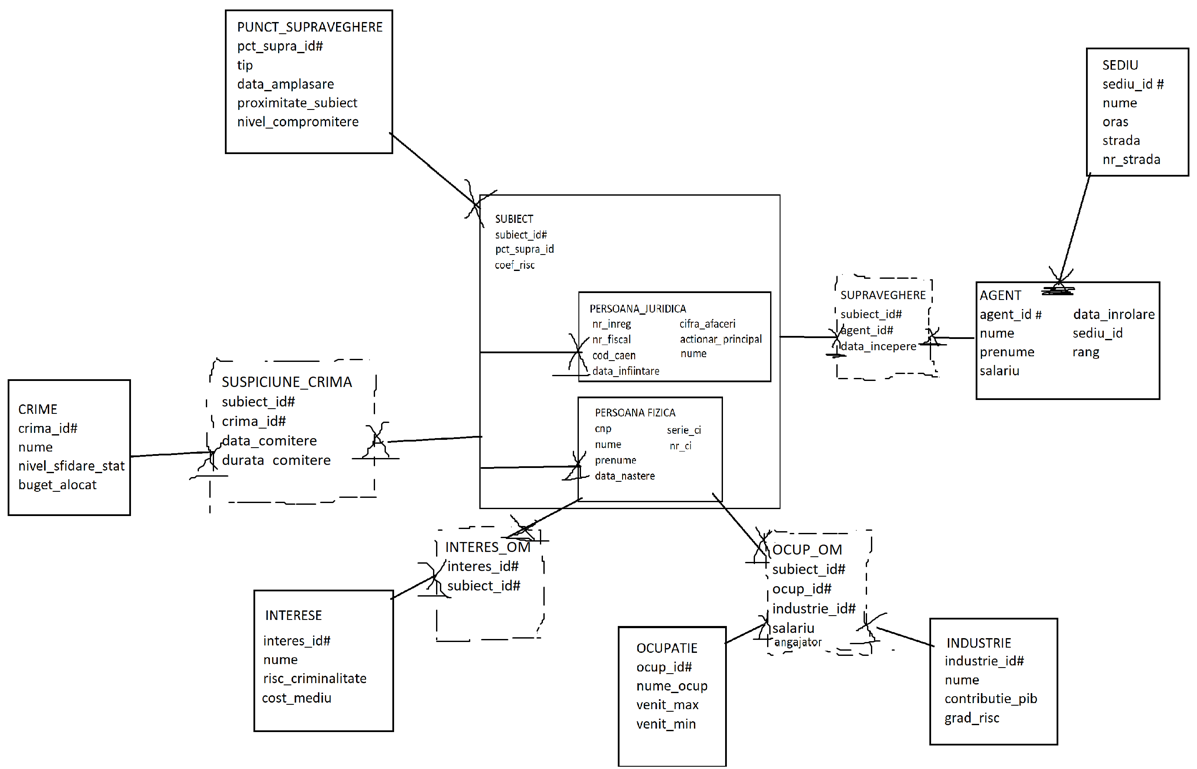
Exercițiul 1. Scurtă descriere a utilității bazei de date.

Baza de date alcatuită are ca scop principal centralizarea si facilitarea urmăririi unei serii de subiecți periculoși de către Serviciul Român de Informații. Baza de date conține informații personale despre cei pe care ii supravegheaza, (date despre ocupațiile, interesele persoanelor fizice și date din interior despre cele juridice), dar și cunoștințe logistice esențiale în urmărirea acțiunilor lor (cine ii supravegheaza, pentru ce, si de unde).

Exercițiul 2. Diagrama Entitate-Relație



Exercițiul 3. Diagrama Conceptuală



Exercițiul 4. Implementarea diagramei în Oracle

create table PUNCT\_SUPRAVEGHERE(

pct\_supra\_id number(5) constraint pk\_pct\_suprav primary key,

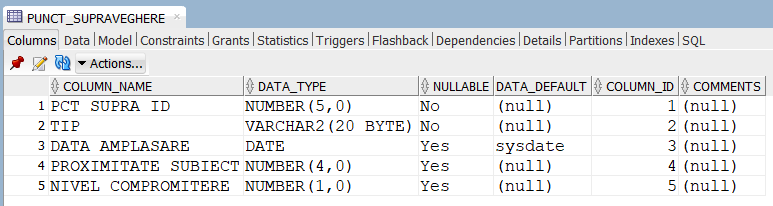
tip varchar(20) constraint nn\_tip\_pctsupra not null,

data\_amplasare date default sysdate,

proximitate\_subiect number(4),

nivel\_compromitere number(1)

);



create table SEDIU(

sediu\_id number(5) constraint pk\_sed primary key,

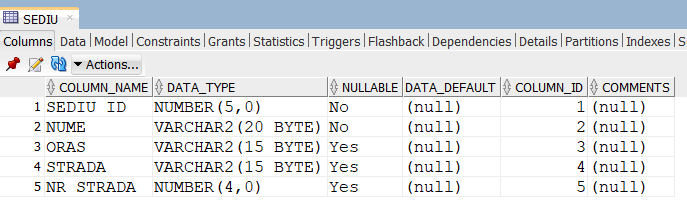
nume varchar(20) constraint nn\_nume\_sediu not null,

oras varchar(15),

strada varchar(15),

nr\_strada number(4)

);



create table AGENTT(

agent\_id number(5) constraint pk\_age primary key,

nume varchar(10) constraint nn\_nume\_age not null,

prenume varchar(10) constraint nn\_prenume\_age not null,

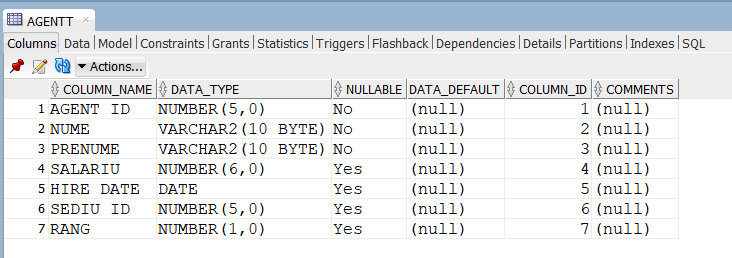
salariu number(6),

hire\_date date,

sediu\_id number(5) constraint fk\_sed\_age references SEDIU(sediu\_id),

rang number(1)

);



create table CRIME(

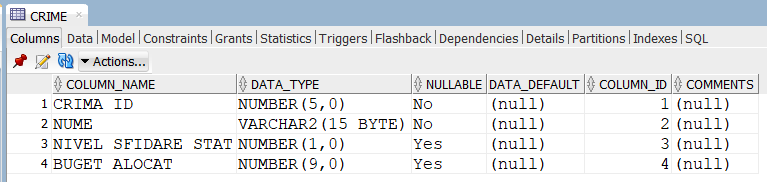
crima\_id number(5) constraint pk\_crime primary key,

nume varchar(15) constraint nn\_nume\_crime not null,

nivel\_sfidare\_stat number(1),

buget\_alocat number(9)

);



create table INDUSTRIE(

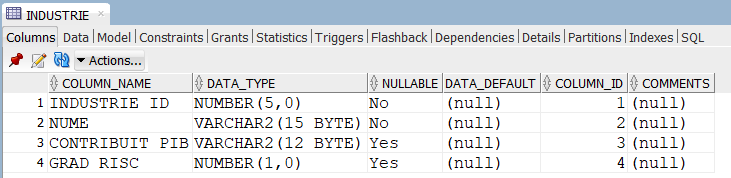
industrie\_id number(5) constraint pk\_ind primary key,

nume varchar(15) constraint nn\_nume\_ind not null,

contribuit\_pib varchar(12),

grad\_risc number(1)

);



create table INTERESE(

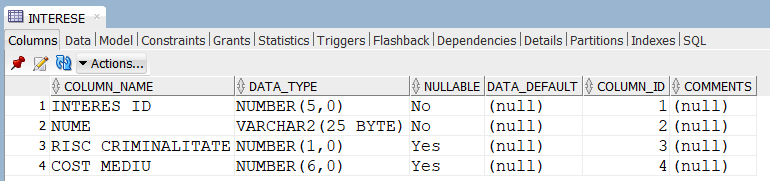
interes\_id number(5) constraint pk\_int primary key,

nume varchar(25) constraint nn\_nume\_int not null,

risc\_criminalitate number(1),

cost\_mediu number(6)

);



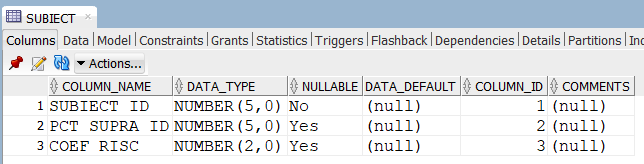
create table SUBIECT(

subiect\_id number(5) constraint pk\_subiect primary key,

pct\_supra\_id number(5) constraint fk\_pct\_suprav\_sub references PUNCT\_SUPRAVEGHERE(pct\_supra\_id) constraint un\_pct\_supra\_sub unique,

coef\_risc number(2)

);



create table INTERES\_OM(

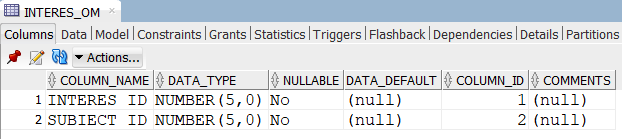
interes\_id number(5),

subiect\_id number(5) constraint fk\_sub\_int\_om references SUBIECT(subiect\_id) on delete cascade,

constraint pk\_int\_om primary key(interes\_id, subiect\_id),

constraint fk\_int\_int\_om foreign key(interes\_id) references INTERESE(interes\_id)

);



create table OCUPATIE(

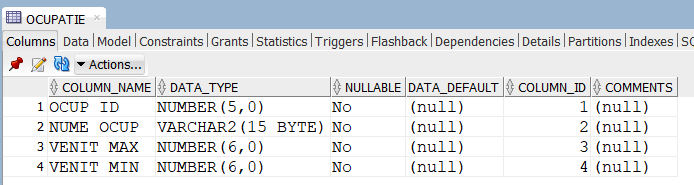
ocup\_id number (5) constraint pk\_ocup primary key,

nume\_ocup varchar(15) constraint nn\_nume\_ocup not null,

venit\_max number(6) constraint nn\_vmax\_ocup not null,

venit\_min number(6) constraint nn\_vmin\_ocup not null

);



create table PERSOANA\_FIZICA(

subiect\_id number(5) constraint pk\_pers\_fiz primary key,

cnp varchar(14) constraint nn\_cnp\_persfiz not null constraint un\_cnp\_persfiz unique,

nume varchar (15) constraint nn\_nume\_persfiz not null,

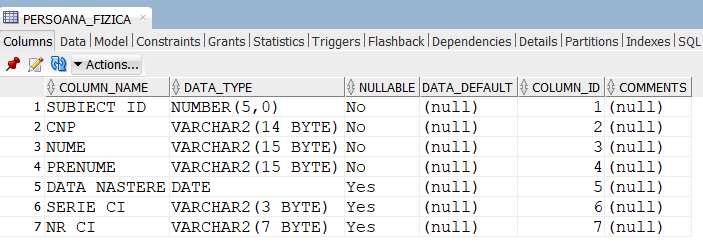
prenume varchar (15) constraint nn\_prenume\_persfiz not null,

data\_nastere date,

serie\_ci varchar(3),

nr\_ci varchar(7)

);



create table PERSOANA\_JURIDICA(

subiect\_id number(5) constraint pk\_pers\_jur primary key,

nr\_inreg varchar(15) constraint nn\_inreg\_pers\_jur not null constraint un\_inreg\_persjur unique,

nr\_fiscal varchar(15)constraint nn\_fisc\_pers\_jur not null constraint un\_fisc\_persjur unique,

cod\_caen varchar(5) constraint nn\_caen\_persjur not null,

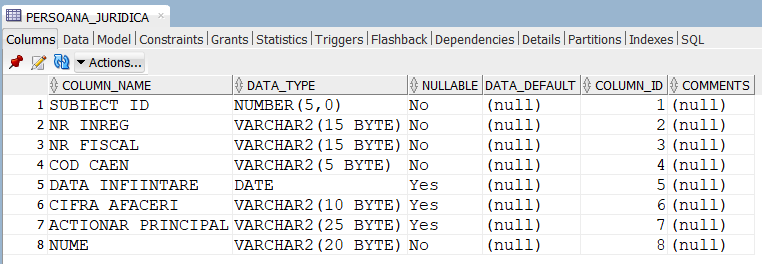
data\_infiintare date,

cifra\_afaceri varchar(10),

actionar\_principal varchar(25),

nume varchar(20) constraint nn\_nume\_persjur not null

);



create table SUSPICIUNE\_CRIMA(

subiect\_id number(5) constraint fk\_sub\_sus references SUBIECT(subiect\_id),

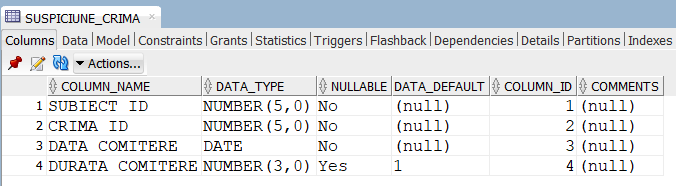
crima\_id number(5) constraint fk\_crima\_sus references CRIME(crima\_id),

data\_comitere date constraint nn\_data\_sus not null,

durata\_comitere number(3) default 1,

constraint pk\_sus primary key(subiect\_id, crima\_id)

);

create table SUPRAVEGHERE(

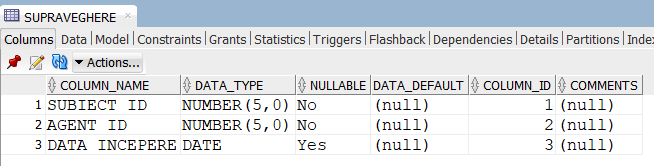
subiect\_id number(5) constraint fk\_sub\_suprav references SUBIECT(subiect\_id) on delete cascade,

agent\_id number(5) constraint fk\_age\_suprav references AGENTT(agent\_id) on delete cascade,

data\_incepere date,

constraint pk\_suprav primary key(subiect\_id, agent\_id)

);



create table OCUP\_OM(

subiect\_id number(5) constraint fk\_sub\_ocupom references SUBIECT(subiect\_id) on delete cascade,

ocup\_id number(5) constraint fk\_ocup\_ocupom references OCUPATIE(ocup\_id),

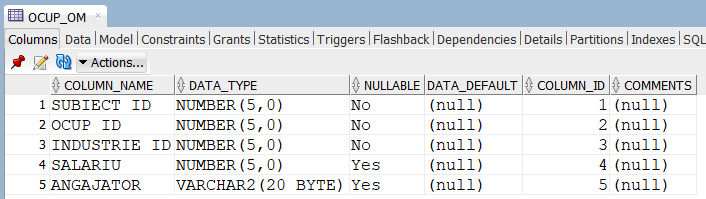
industrie\_id number(5) constraint fk\_int\_ocupom references INDUSTRIE(industrie\_id),

salariu number (5),

angajator varchar(20),

constraint pk\_ocupom primary key(subiect\_id, ocup\_id, industrie\_id)

);



Exercițiul 5. Adăugați înformații coerente în tabele

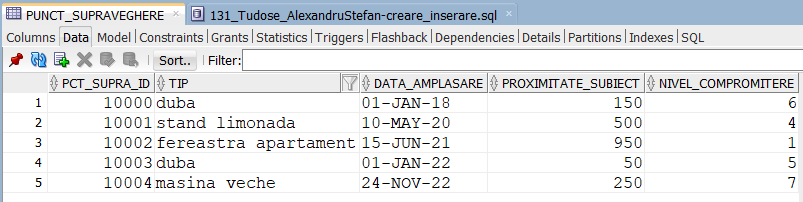
insert into PUNCT\_SUPRAVEGHERE values(secv\_id.nextval, 'duba', to\_date('01-01-2018', 'dd-mm-yyyy'), 150, 6);

insert into PUNCT\_SUPRAVEGHERE values(secv\_id.nextval, 'stand limonada', to\_date('10-05-2020', 'dd-mm-yyyy'), 500, 4);

insert into PUNCT\_SUPRAVEGHERE values(secv\_id.nextval, 'fereastra apartament', to\_date('15-06-2021', 'dd-mm-yyyy'), 950, 1);

insert into PUNCT\_SUPRAVEGHERE values(secv\_id.nextval, 'duba', to\_date('01-01-2022', 'dd-mm-yyyy'), 50, 5);

insert into PUNCT\_SUPRAVEGHERE values(secv\_id.nextval, 'masina veche', sysdate-50, 250, 7);



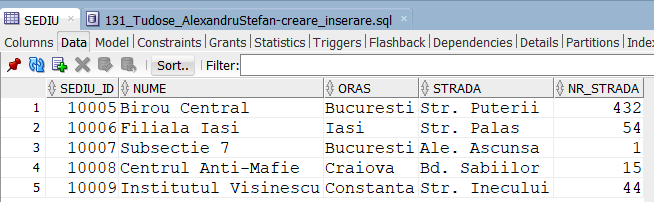
insert into SEDIU values(secv\_id.nextval, 'Birou Central', 'Bucuresti', 'Str. Puterii', 432);

insert into SEDIU values(secv\_id.nextval, 'Filiala Iasi', 'Iasi', 'Str. Palas', 54);

insert into SEDIU values(secv\_id.nextval, 'Subsectie 7', 'Bucuresti', 'Ale. Ascunsa', 1);

insert into SEDIU values(secv\_id.nextval, 'Centrul Anti-Mafie', 'Craiova', 'Bd. Sabiilor', 15);

insert into SEDIU values(secv\_id.nextval, 'Institutul Visinescu', 'Constanta', 'Str. Inecului', 44);



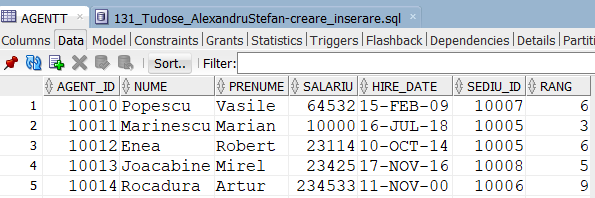
insert into AGENTT values (secv\_id.nextval ,'Popescu', 'Vasile', 64532, to\_date('15-02-2009', 'dd-mm-yyyy'), 10007, 6);

insert into AGENTT values (secv\_id.nextval ,'Marinescu', 'Marian', 10000, to\_date('16-07-2018', 'dd-mm-yyyy'), 10005, 3);

insert into AGENTT values (secv\_id.nextval ,'Enea', 'Robert', 23114, to\_date('10-10-2014', 'dd-mm-yyyy'), 10005, 6);

insert into AGENTT values (secv\_id.nextval ,'Joacabine', 'Mirel', 23425, to\_date('17-11-2016', 'dd-mm-yyyy'), 10008, 5);

insert into AGENTT values (secv\_id.nextval ,'Rocadura', 'Artur', 234533, to\_date('11-11-2000', 'dd-mm-yyyy'), 10006, 9);



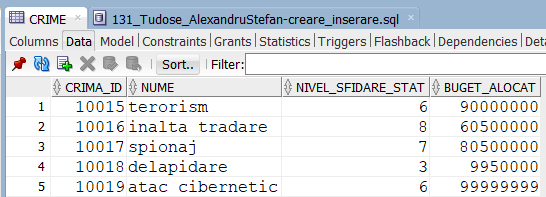
insert into CRIME values(secv\_id.nextval, 'terorism', 6, 90000000);

insert into CRIME values(secv\_id.nextval, 'inalta tradare', 8, 60500000);

insert into CRIME values(secv\_id.nextval, 'spionaj', 7, 80500000);

insert into CRIME values(secv\_id.nextval, 'delapidare', 3, 9950000);

insert into CRIME values(secv\_id.nextval, 'atac cibernetic', 6, 99999999);



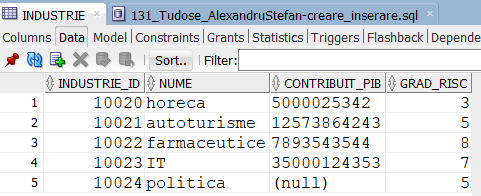
insert into INDUSTRIE values(secv\_id.nextval, 'horeca', '5000025342', 3);

insert into INDUSTRIE values(secv\_id.nextval, 'autoturisme', '12573864243', 5);

insert into INDUSTRIE values(secv\_id.nextval, 'farmaceutice', '7893543544', 8);

insert into INDUSTRIE values(secv\_id.nextval, 'IT', '35000124353', 7);

insert into INDUSTRIE values(secv\_id.nextval, 'politica', null, 5);



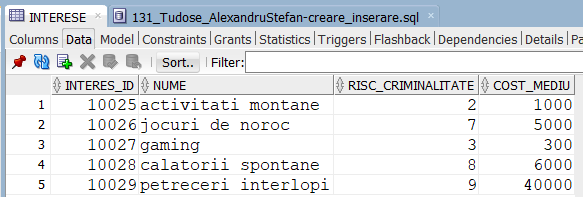
insert into INTERESE values(secv\_id.nextval, 'activitati montane', 2, 1000);

insert into INTERESE values(secv\_id.nextval, 'jocuri de noroc', 7, 5000);

insert into INTERESE values(secv\_id.nextval, 'gaming', 3, 300);

insert into INTERESE values(secv\_id.nextval, 'calatorii spontane', 8, 6000);

insert into INTERESE values(secv\_id.nextval, 'petreceri interlopi', 9, 40000);



insert into SUBIECT values(secv\_id.nextval, 10000, 3);

insert into SUBIECT values(secv\_id.nextval, 10001, 7);

insert into SUBIECT values(secv\_id.nextval, null, 5);

insert into SUBIECT values(secv\_id.nextval, 10002, 4);

insert into SUBIECT values(secv\_id.nextval, null, 3);

insert into SUBIECT values(secv\_id.nextval, null, 7);

insert into SUBIECT values(secv\_id.nextval, null, 8);

insert into SUBIECT values(secv\_id.nextval, null, 8);

insert into SUBIECT values(secv\_id.nextval, 10004, 4);

insert into SUBIECT values(secv\_id.nextval, null, 1);

insert into SUBIECT values(secv\_id.nextval, null, 3);

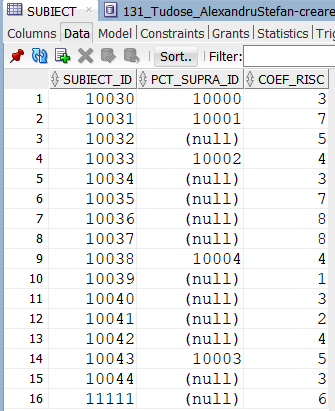
insert into SUBIECT values(secv\_id.nextval, null, 2);

insert into SUBIECT values(secv\_id.nextval, null, 4);

insert into SUBIECT values(secv\_id.nextval, 10003, 5);

insert into SUBIECT values(secv\_id.nextval, null, 3);

insert into SUBIECT values(11111, null, 6);



insert into INTERES\_OM values (10026, 10034);

insert into INTERES\_OM values (10027, 10036);

insert into INTERES\_OM values (10026, 10041);

insert into INTERES\_OM values (10029, 10041);

insert into INTERES\_OM values (10029, 10042);

insert into INTERES\_OM values (10025, 10035);

insert into INTERES\_OM values (10028, 10032);

insert into INTERES\_OM values (10026, 10036);

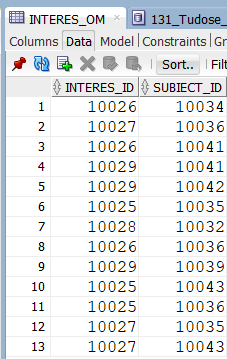
insert into INTERES\_OM values (10029, 10039);

insert into INTERES\_OM values (10025, 10043);

insert into INTERES\_OM values (10025, 10036);

insert into INTERES\_OM values (10027, 10035);

insert into INTERES\_OM values (10027, 10043);



insert into OCUPATIE values(secv\_id.nextval, 'motostivuitor', 4000, 1500);

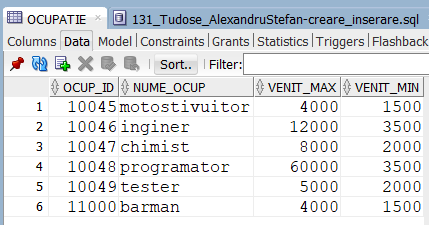
insert into OCUPATIE values(secv\_id.nextval, 'inginer', 12000, 3500);

insert into OCUPATIE values(secv\_id.nextval, 'chimist', 8000, 2000);

insert into OCUPATIE values(secv\_id.nextval, 'programator', 60000, 3500);

insert into OCUPATIE values(secv\_id.nextval, 'tester', 5000, 2000);

insert into OCUPATIE values(11000, 'barman', 4000, 1500);



insert into PERSOANA\_FIZICA values (10032, 6030116196416, 'Nemtanu', 'Violeta', to\_date('16-01-2003', 'dd-mm-yyyy'), 'MZ', '724312');

insert into PERSOANA\_FIZICA values (10034, '1940728155764', 'Vasilescu', 'Ion', to\_date('28-07-1994', 'dd-mm-yyyy'), 'MX', '483945');

insert into PERSOANA\_FIZICA values (10035, '1880109244504', 'Mormon', 'Marius', to\_date('09-01-1988', 'dd-mm-yyyy'), 'MZ', '457453');

insert into PERSOANA\_FIZICA values (10036, '1870218196401', 'Cormen', 'Vioren', to\_date('18-02-1987', 'dd-mm-yyyy'), 'IO', '275421');

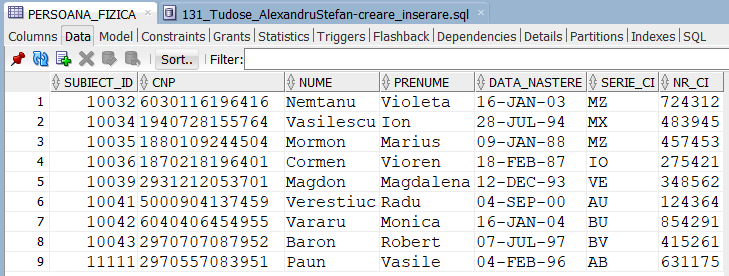
insert into PERSOANA\_FIZICA values (10039, '2931212053701', 'Magdon', 'Magdalena', to\_date('12-12-1993', 'dd-mm-yyyy'), 'VE', '348562');

insert into PERSOANA\_FIZICA values (10041, '5000904137459', 'Verestiuc', 'Radu', to\_date('04-09-2000', 'dd-mm-yyyy'), 'AU', '124364');

insert into PERSOANA\_FIZICA values (10042, '6040406454955', 'Vararu', 'Monica', to\_date('16-01-2004', 'dd-mm-yyyy'), 'BU', '854291');

insert into PERSOANA\_FIZICA values (10043, '2970707087952', 'Baron', 'Robert', to\_date('07-07-1997', 'dd-mm-yyyy'), 'BV', '415261');

insert into PERSOANA\_FIZICA values (11111, '2970557083951', 'Paun', 'Vasile', to\_date('04-02-1996', 'dd-mm-yyyy'), 'AB', '631175');



insert into PERSOANA\_JURIDICA values(10030, 'J22/973/2000', 'RO13611073C', '1243', to\_date('05-05-2000', 'dd-mm-yyyy'), '1325531', 'Ion Pavel', 'Trafic de pufarine');

insert into PERSOANA\_JURIDICA values(10031, 'J40/1973/2015', 'RO45361325C', '4241', to\_date('06-06-2016', 'dd-mm-yyyy'), '325634634', 'Victor Manea', 'Regatul Comertului');

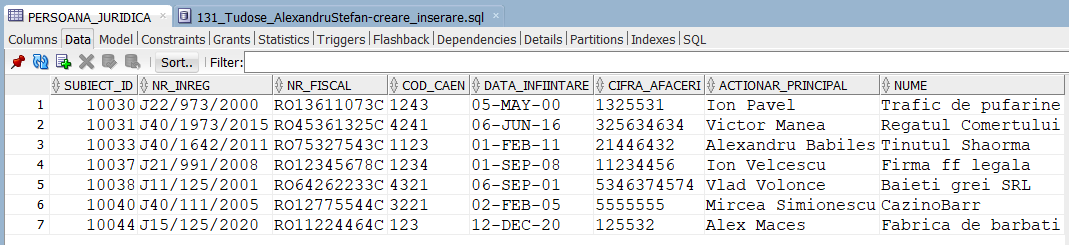
insert into PERSOANA\_JURIDICA values(10033, 'J40/1642/2011', 'RO75327543C', '1123', to\_date('01-02-2011', 'dd-mm-yyyy'), '21446432', 'Alexandru Babiles', 'Tinutul Shaorma');

insert into PERSOANA\_JURIDICA values(10037, 'J21/991/2008', 'RO12345678C', '1234', to\_date('01-09-2008', 'dd-mm-yyyy'), '11234456', 'Ion Velcescu', 'Firma ff legala');

insert into PERSOANA\_JURIDICA values(10038, 'J11/125/2001', 'RO64262233C', '4321', to\_date('06-09-2001', 'dd-mm-yyyy'), '5346374574', 'Vlad Volonce', 'Baieti grei SRL');

insert into PERSOANA\_JURIDICA values(10040, 'J40/111/2005', 'RO12775544C', '3221', to\_date('02-02-2005', 'dd-mm-yyyy'), '5555555', 'Mircea Simionescu', 'CazinoBarr');

insert into PERSOANA\_JURIDICA values(10044, 'J15/125/2020', 'RO11224464C', '123', to\_date('12-12-2020', 'dd-mm-yyyy'), '125532', 'Alex Maces', 'Fabrica de barbati');



insert into SUSPICIUNE\_CRIMA values(10037, 10015, to\_date('15-07-2018', 'dd-mm-yyyy'), 3);

insert into SUSPICIUNE\_CRIMA values(10041, 10018, to\_date('19-02-2020', 'dd-mm-yyyy'), 12);

insert into SUSPICIUNE\_CRIMA values(10043, 10019, to\_date('12-12-2013', 'dd-mm-yyyy'), 2);

insert into SUSPICIUNE\_CRIMA values(10035, 10016, to\_date('01-05-2016', 'dd-mm-yyyy'), 1);

insert into SUSPICIUNE\_CRIMA values(10044, 10018, to\_date('02-03-2019', 'dd-mm-yyyy'), 10);

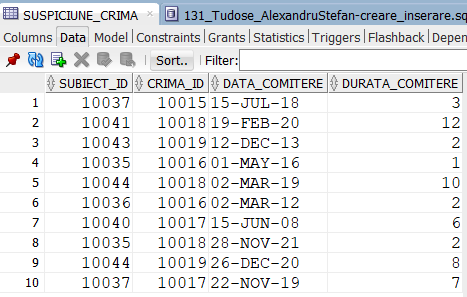
insert into SUSPICIUNE\_CRIMA values(10036, 10016, to\_date('02-03-2012', 'dd-mm-yyyy'), 2);

insert into SUSPICIUNE\_CRIMA values(10040, 10017, to\_date('15-06-2008', 'dd-mm-yyyy'), 6);

insert into SUSPICIUNE\_CRIMA values(10035, 10018, to\_date('28-11-2021', 'dd-mm-yyyy'), 2);

insert into SUSPICIUNE\_CRIMA values(10044, 10019, to\_date('26-12-2020', 'dd-mm-yyyy'), 8);

insert into SUSPICIUNE\_CRIMA values(10037, 10017, to\_date('22-11-2019', 'dd-mm-yyyy'), 7);



insert into SUPRAVEGHERE values(10032, 10013, to\_date('16-08-2021', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10033, 10010, to\_date('19-04-2020', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10044, 10014, to\_date('26-01-2021', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10034, 10012, to\_date('19-08-2021', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10042, 10010, to\_date('16-03-2019', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10044, 10013, to\_date('07-12-2019', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10034, 10011, to\_date('11-10-2019', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10042, 10014, to\_date('06-11-2021', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10031, 10012, to\_date('21-09-2021', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10032, 10011, to\_date('13-02-2022', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10037, 10013, to\_date('07-10-2016', 'dd-mm-yyyy'));

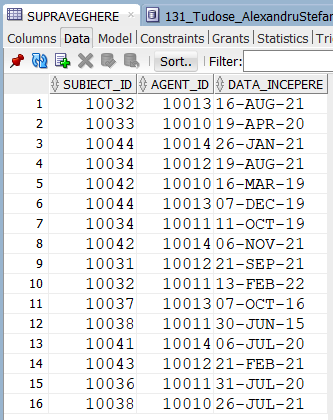
insert into SUPRAVEGHERE values(10038, 10011, to\_date('30-06-2015', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10041, 10014, to\_date('06-07-2020', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10043, 10012, to\_date('21-02-2021', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10036, 10011, to\_date('31-07-2020', 'dd-mm-yyyy'));

insert into SUPRAVEGHERE values(10038, 10010, to\_date('26-07-2021', 'dd-mm-yyyy'));



insert into OCUP\_OM values (10034, 10045, 10021, 7000, 'Tractoare si lopeti');

insert into OCUP\_OM values (10032, 10048, 10024, 15000, 'Vervontis');

insert into OCUP\_OM values (10042, 10046, 10020, 3500, 'Mercurian');

insert into OCUP\_OM values (10041, 10049, 10022, 6432, 'Babion');

insert into OCUP\_OM values (10043, 10047, 10021, 11003, 'Boninu');

insert into OCUP\_OM values (10039, 10045, 10020, 5322, 'McOron');

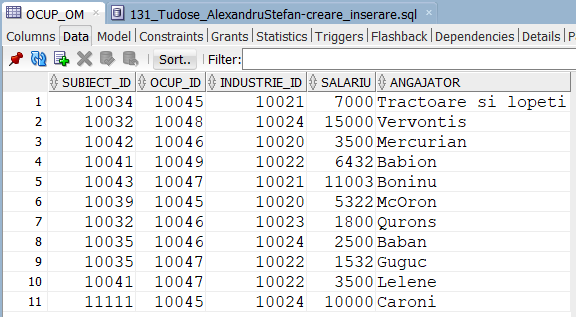
insert into OCUP\_OM values (10032, 10046, 10023, 1800, 'Qurons');

insert into OCUP\_OM values (10035, 10046, 10024, 2500, 'Baban');

insert into OCUP\_OM values (10035, 10047, 10022, 1532, 'Guguc');

insert into OCUP\_OM values (10041, 10047, 10022, 3500, 'Lelene');

insert into OCUP\_OM values (11111, 10045, 10024, 10000, 'Caroni');



6. Problema

-subprogram stocat independent

-2 tipuri diferite de colectii (eu am 3 pentru complexitate)

Cerință: Realizati un top al performantei agentilor: Datele despre agenti vor fi afisate in ordine (id, nume, prenume, rang), dupa suma coeficientilor de risc ai tuturor subiectilor pe care ii au sub supraveghere. In caz de egalitate intre doi agenti, departajarea va fi facuta dupa rangul agentului. Vor fi afisati si subiectii pe care ii au sub supraveghere (id, coef\_risc).

CREATE OR REPLACE PROCEDURE ex6 AS

type tipColSub is record (subId subiect.subiect\_id%type, coefRisc subiect.coef\_risc%type);

type colSub is table of tipColSub;

type tabSub is table of colSub index by pls\_integer;

subiecti tabSub; --colectie de colectii de subiecti supravegheati, indexata de id-ul agentului

type recAg is record(agId agentt.agent\_id%type, nume agentt.nume%type, prenume agentt.prenume%type, rang agentt.rang%type);

type tabAg is varray(20) of recAg;

agenti tabAg := tabAg();

agIdCurent agentt.agent\_id%type;

BEGIN

select a.agent\_id, a.nume, a.prenume, a.rang

bulk collect into agenti

from agentt a left join supraveghere sup

on a.agent\_id = sup.agent\_id

left join subiect s on s.subiect\_id = sup.subiect\_id

group by a.agent\_id, a.nume, a.prenume, a.rang

order by nvl(sum(s.coef\_risc), 0) desc, a.rang desc;

if (agenti.count = 0) then

raise\_application\_error(-20000, 'Nu exista niciun agent inregistrat');

end if;

for i in agenti.first..agenti.last loop

agIdCurent := agenti(i).agId;

select s.subiect\_id, s.coef\_risc

bulk collect into subiecti(agIdCurent)

from supraveghere sup join subiect s

on sup.subiect\_id = s.subiect\_id

where sup.agent\_id = agIdCurent;

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

dbms\_output.put\_line('Locul ' || i || ': agentul ' || agenti(i).nume || ' ' || agenti(i).prenume ||', cu rangul ' || agenti(i).rang || ' . Suspecti supravegheati: ');

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

if (subiecti(agIdCurent).count = 0) then

dbms\_output.put\_line('Niciun suspect supravegheat');

else

for j in subiecti(agIdCurent).first..subiecti(agIdCurent).last loop

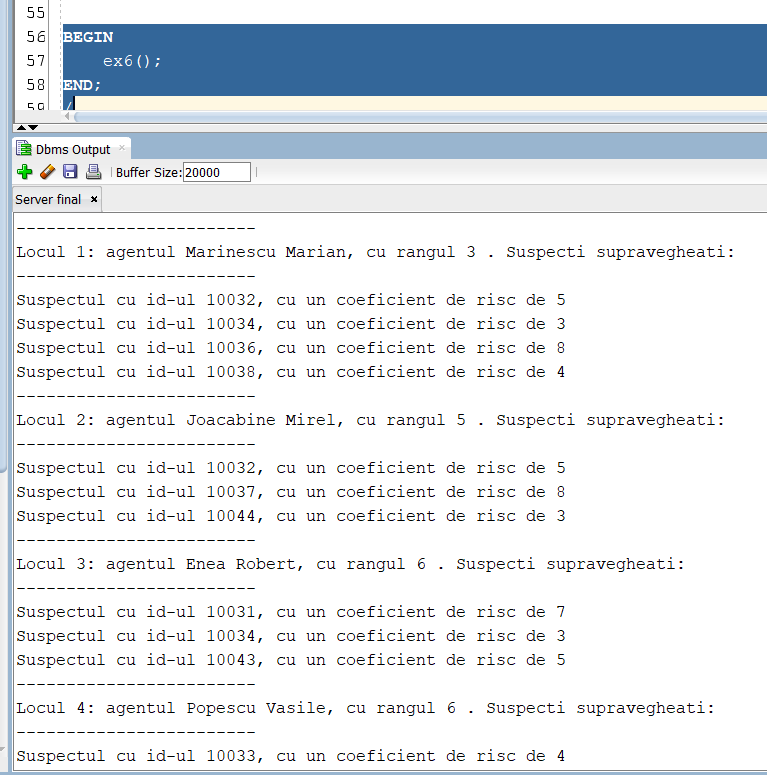
dbms\_output.put\_line('Suspectul cu id-ul ' || subiecti(agIdCurent)(j).subId || ', cu un coeficient de risc de ' || subiecti(agIdCurent)(j).coefRisc);

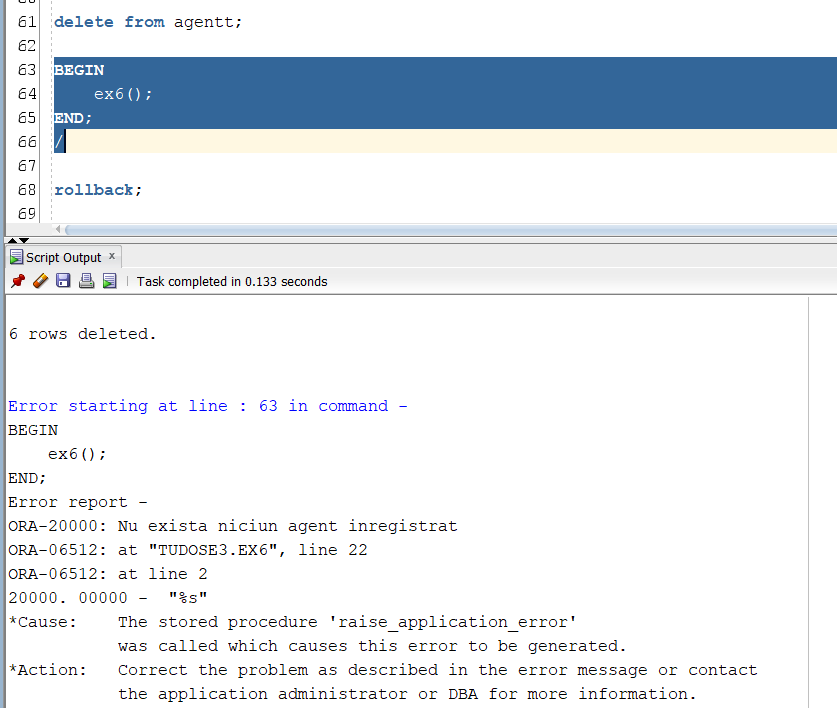
end loop;

end if;

end loop;

END ex6;





7. Problema

-subprogram stocat independent

-2 cursoare diferite (eu am 3 pentru complexitate)

-cel putin un cursor parametrizat

Cerinta: In functie de pachetul de interese ales de la tastatura dintre cele de mai jos, sa se afiseze toate persoanele fizice care au respectivul interes, iar pentru fiecare persoana fizica sa fie mentionate numele agentilor care o urmaresc

--pachet 1: interesele cu risc de criminalitate cuprins intre 1 si 3

--pachet 2: interesele cu risc de criminalitate cuprins intre 4 si 6

--pachet 3: interesele cu risc de criminalitate cuprins intre 7 si 9

create or replace procedure ex7 as

type refcursorInt is ref cursor return interese%rowtype;

type pers is record (subId persoana\_fizica.subiect\_id%type, nume persoana\_fizica.nume%type, prenume persoana\_fizica.prenume%type);

type agenttt is record (nume agentt.nume%type, prenume agentt.nume%type);

type refcursorAg is ref cursor return agenttt;

agenti refcursorAg;

cursInterese refcursorInt;

pachet number := &p;

interes interese%rowtype;

numeSub persoana\_fizica.nume%type;

prenumeSub persoana\_fizica.prenume%type;

numeAg agentt.nume%type;

prenumeAg agentt.prenume%type;

cursor persFiz(interes interese.interes\_id%type) is --expresie cursor

select distinct nume, prenume,

cursor (select a.nume, a.prenume

from agentt a join supraveghere sup

on a.agent\_id = sup.agent\_id

where p.subiect\_id = sup.subiect\_id)

from persoana\_fizica p join interes\_om i

on p.subiect\_id = i.subiect\_id

where i.interes\_id = interes;

begin

if (pachet = 1) then

open cursInterese for --cursor dinamic

select \*

from interese

where risc\_criminalitate < 4;

elsif (pachet = 2) then

open cursInterese for

select \*

from interese

where risc\_criminalitate > 3 and risc\_criminalitate < 7;

elsif (pachet = 3) then

open cursInterese for

select \*

from interese

where risc\_criminalitate > 6;

else

raise\_application\_error(-20001, 'Optiune incorecta');

end if;

loop

fetch cursInterese into interes;

exit when cursInterese%notfound;

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

dbms\_output.put\_line('Pentru interesul ' || interes.nume|| ', cu un risc de criminalitate de ' || interes.risc\_criminalitate);

dbms\_output.put\_line('au fost identificati urmatorii:');

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

--for i in persFiz(interes.interes\_id) loop

open persFiz(interes.interes\_id); --cursor clasic parametrizat

loop

fetch persFiz into numeSub, prenumeSub, agenti;

exit when persFiz%notfound;

dbms\_output.put(numeSub || ' ' || prenumeSub || '; supravegheat de: ');

loop

fetch agenti into numeAg, prenumeAg;

exit when agenti%notfound;

dbms\_output.put(numeAg || ' ' || prenumeAg || ', ');

end loop;

if (agenti%rowcount = 0) then

dbms\_output.put(' niciun agent');

end if;

dbms\_output.new\_line();

end loop;

if (persFiz%rowcount = 0) then

dbms\_output.put\_line('Nu a fost identificat niciun subiect cu acest interes');

end if;

close persFiz;

end loop;

if cursInterese%rowcount = 0 then

dbms\_output.put\_line('Nu exista niciun interes in intervalul selectat al riscului de criminalitate');

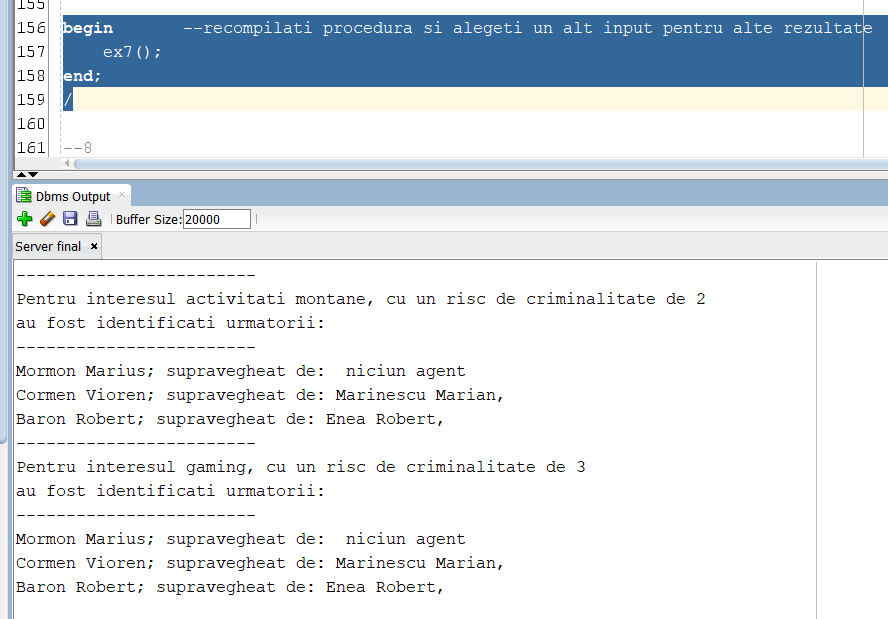
end if;

close cursInterese;

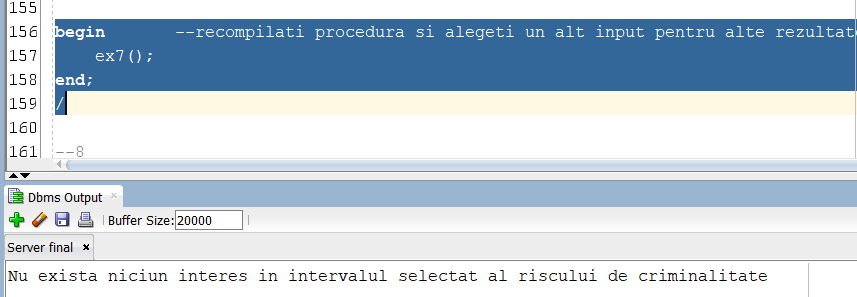
end ex7;

/

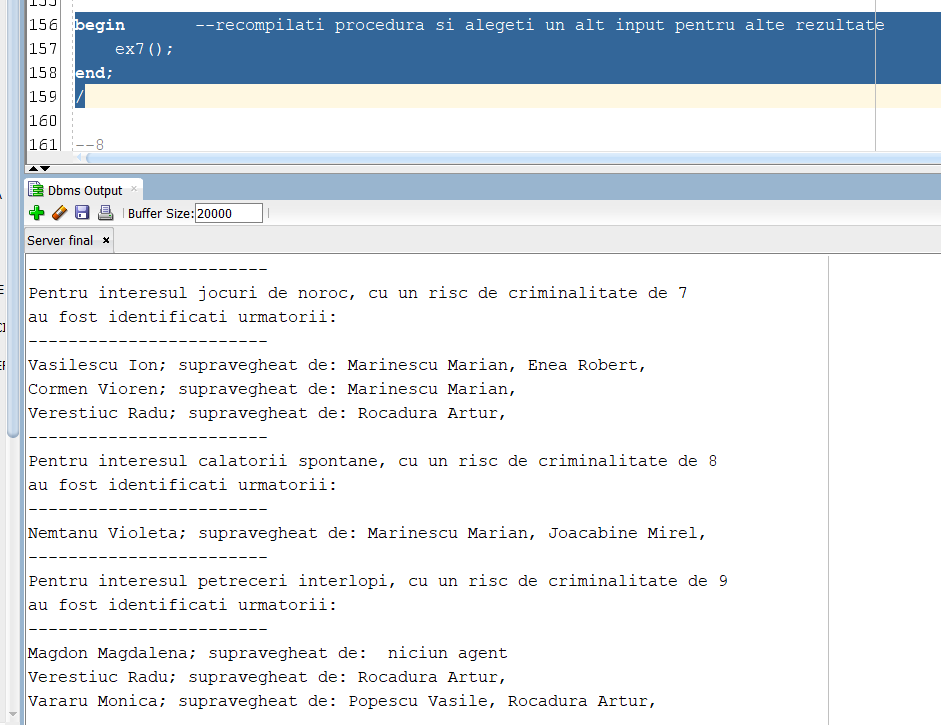
Pachet 1



Pachet 2



Pachet 3



8. Problema

--subprogram stocat independent de tip functie

--intr-o singura comanda sql utilizarea a 3 tabele

--minim 2 exceptii

Cerinta: Sa se obtina durata dezavarsirii crimei (in luni) de care este suspectat un subiect, unde id-ul acestuia si numele crimei sunt cunoscute

create or replace function ex8(numeCr crime.nume%type, idSub subiect.subiect\_id%type) return number

is

nonExistSubject exception;

pragma exception\_init(nonExistSubject, -20005);

nonExistCrime exception;

pragma exception\_init(nonExistCrime, -20006);

durata number;

testt1 number;

begin

select count(subiect\_id) into testt1

from subiect

where subiect\_id = idSub;

if (testt1 = 0) then

raise nonExistSubject;

end if;

select count(crima\_id) into testt1

from crime

where upper(nume) = upper(numeCr);

if (testt1 = 0) then

raise nonExistCrime;

end if;

select durata\_comitere into durata

from suspiciune\_crima sc join crime c

on sc.crima\_id = c.crima\_id

join subiect s

on s.subiect\_id = sc.subiect\_id

where upper(c.nume) = upper(numeCr) and idSub = s.subiect\_id;

return durata;

EXCEPTION

when no\_data\_found then --cand exista si crima si subiectul, dar sunt asociati

dbms\_output.put\_line('Subiectul nu este suspect de aceasta crima');

when nonExistCrime then

dbms\_output.put\_line('Serviciul roman de informatii nu se ocupa cu astfel de crime');

when nonExistSubject then

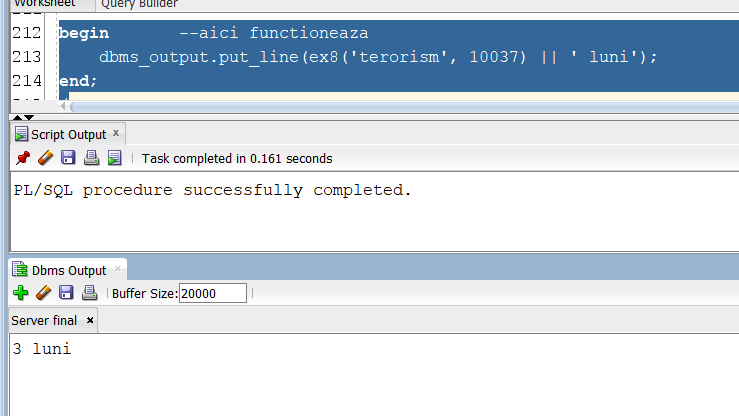
dbms\_output.put\_line('Id-ul transmis nu apartine niciunui subiect inregistrat.');

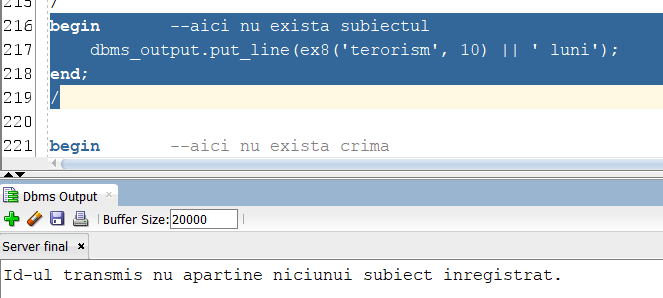
when others then

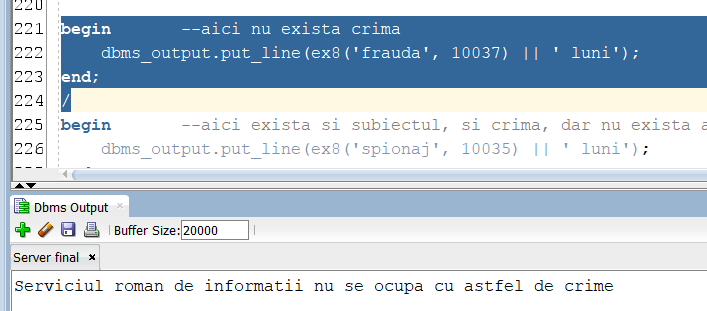
dbms\_output.put\_line('Exista o eroare neprevazuta. Ceea ce nu e posibil, pentru ca noi suntem SRI si prevedem totul. Cel mai probabil

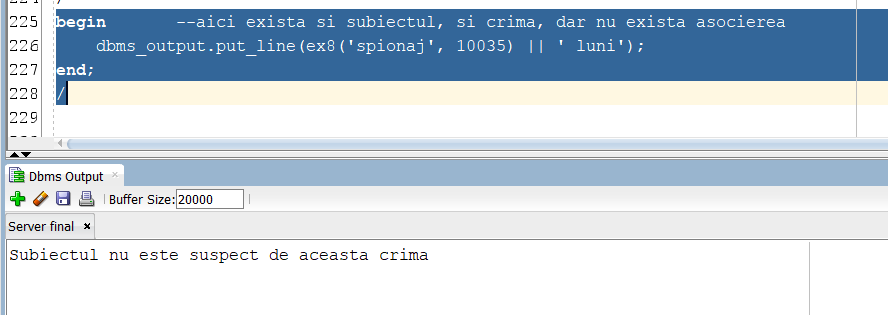
ati ajuns aici pentru ca a gresit procesorul la calcule');

end ex8;









9. Problema

-subprogram stocat independent de tip procedura

-intr-o singura comanda sql utilizare a 5 tabele

-tratarea tuturor exceptiilor

Cerinta: Sa se afiseze interesul (atentie mare, interesul si nu interesele) unui subiect, despre care sunt cunoscute ocupatia si industria in care activeaza

create or replace procedure ex9(nameInd industrie.nume%type, nameOcup ocupatie.nume\_ocup%type) is

nonExistOcup exception;

nonExistInd exception;

tooManySubj exception;

noSubj exception;

pragma Exception\_Init(tooManySubj, -20010);

pragma Exception\_Init(noSubj, -2009);

pragma Exception\_Init(nonExistOcup, -20007);

pragma Exception\_init(nonExistInd, -20008);

testt number;

interesRez interese.nume%type;

BEGIN

select count(ocup\_id) into testt

from ocupatie

where upper(nume\_ocup) = upper(nameOcup);

if (testt = 0) then

raise nonExistOcup;

end if;

select count(industrie\_id) into testt

from industrie

where upper(nume) = upper(nameInd);

if(testt = 0) then

raise nonExistInd;

end if;

select count (subiect\_id) into testt

from ocup\_om oo join ocupatie ocup on oo.ocup\_id = ocup.ocup\_id

join industrie i on i.industrie\_id = oo.industrie\_id

where upper(i.nume) = upper(nameInd) and upper(ocup.NUME\_OCUP) = upper(nameOcup);

if (testt = 0) then

raise noSubj;

elsif (testt > 1) then

raise tooManySubj;

end if;

select i.nume into interesRez

from interese i join interes\_om io

on i.interes\_id = io.interes\_id

join persoana\_fizica pf

on pf.subiect\_id = io.subiect\_id

join ocup\_om oo

on pf.subiect\_id = oo.subiect\_id

join industrie ind

on ind.industrie\_id = oo.industrie\_id

join ocupatie ocup

on ocup.ocup\_id = oo.ocup\_id

where upper(ind.nume) = upper(nameInd) and upper(ocup.NUME\_OCUP) = upper(nameOcup);

dbms\_output.put\_line('Interesul unui subiect ce respecta acesti parametrii este ' || interesRez);

exception

when nonExistInd then

dbms\_output.put\_line('Industria introdusa nu este analizata de SRI. Incercati la STS.');

when nonExistOcup then

dbms\_output.put\_line('Ocupatia introdusa nu este suficient de importanta pentru a fi urmarita de noi.');

when tooManySubj then

dbms\_output.put\_line('Exista mai mult de un subiect care se incadreaza in parametrii oferiti');

when noSubj then

dbms\_output.put\_line('Nu se incadreaza nimeni in parametrii oferiti');

when too\_many\_rows then

dbms\_output.put\_line('Subiectul identificat are mai mult de un singur interes');

when no\_data\_found then

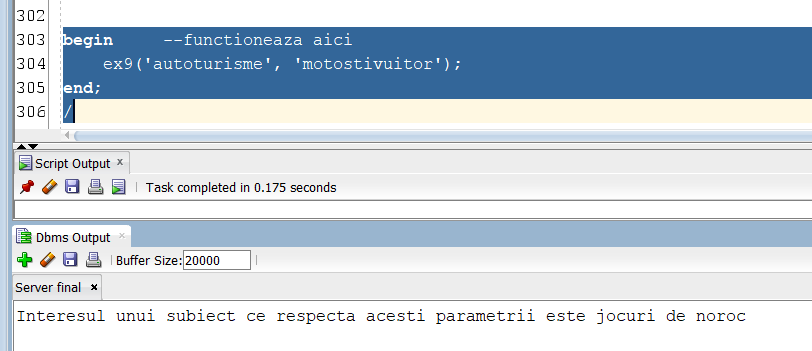
dbms\_output.put\_line('Subiectul nu are niciun interes');

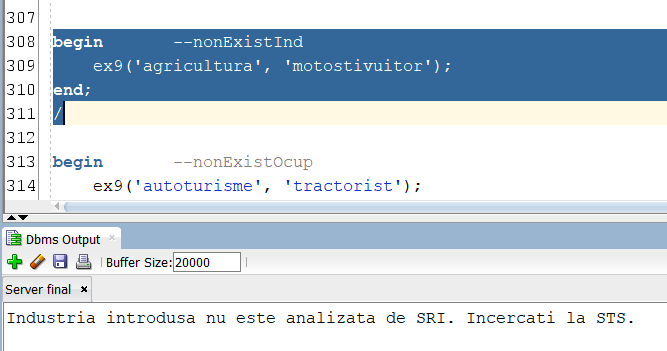
when others then

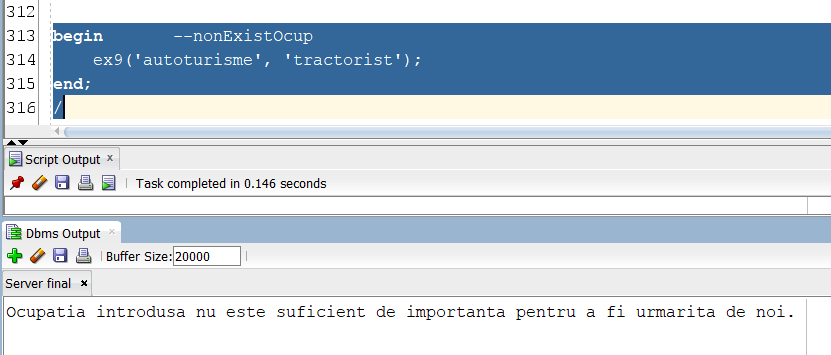
dbms\_output.put\_line('Exista o eroare neprevazuta. Ceea ce nu e posibil, pentru ca noi suntem SRI si prevedem totul. Cel mai probabil

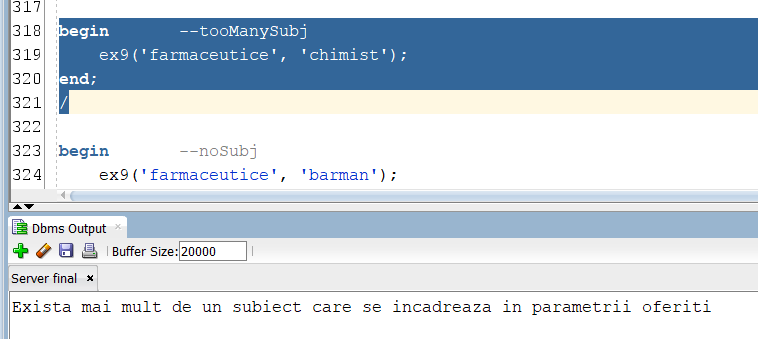
ati ajuns aici pentru ca a gresit procesorul la calcule');

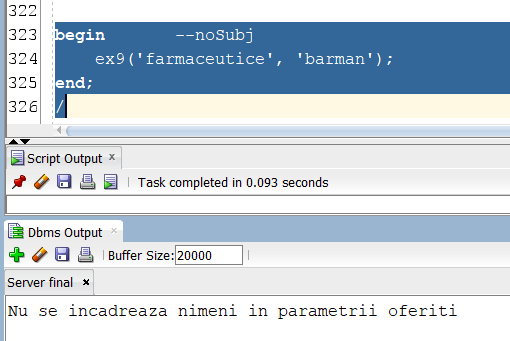
END ex9;

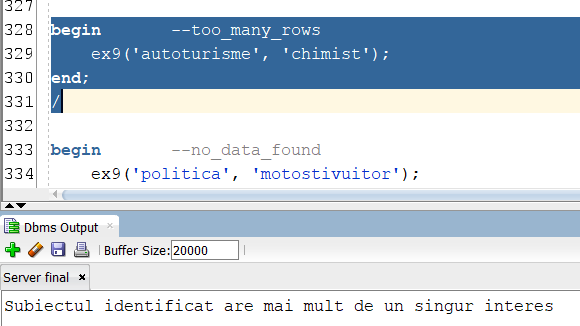


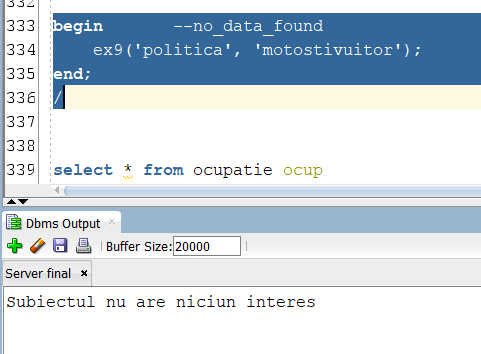












Ex 10 (trigger LMD comanda) si 11 (trigger LMD linie) sunt corelate

Cerinta 10: Sa nu se poata insera un subiect direct in lista agregata a subiectilor si nici sa nu se poata sterge direct de aici

Cerinta 11: Sa se automatizeze procesul de adaugare al subiectilor in baza de date. La momentul adaugarii unei persoane fizice sau juridice, sa se adauge o intrare in lista agregata a subiectilor, si sa se adauge o instanta de supraveghere a acestuia efectuata de cel mai putin solicitat agent. La momentul stergerii unei persoane fizice sau juridice, sa se stearga intrarea echivalenta din tabelul "subiect"

--10

create or replace package ex10aux is

canInsert number := 0;

idAgNesolicit agentt.agent\_id%type;

end ex10aux;

create or replace trigger ex10

before insert or delete on subiect

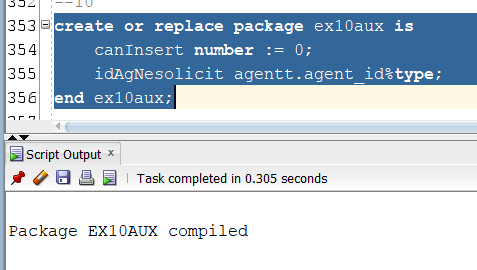
begin

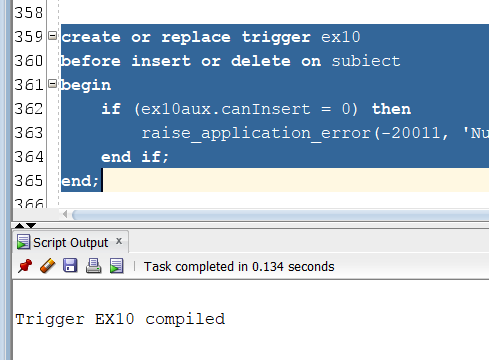
if (ex10aux.canInsert = 0) then

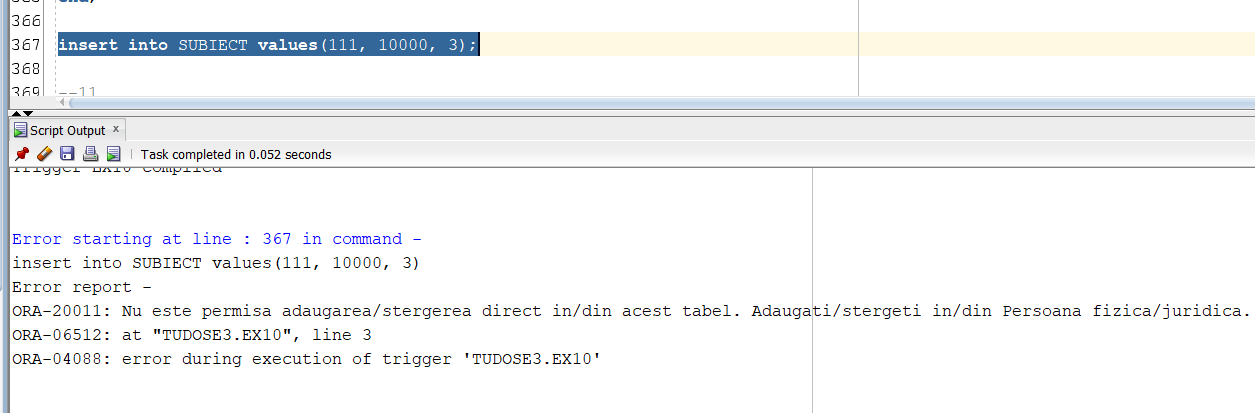
raise\_application\_error(-20011, 'Nu este permisa adaugarea/stergerea direct in/din acest tabel. Adaugati/stergeti in/din Persoana fizica/juridica.');

end if;

end;







11.

create or replace trigger ex11a1 --pentru pers\_fizica

before insert or delete on persoana\_fizica

for each row

begin

ex10aux.canInsert := 1;

if inserting then

insert into subiect values (:new.subiect\_id, null, null);

select distinct a.agent\_id into ex10aux.idAgNesolicit

from supraveghere s join agentt a

on a.agent\_id = s.agent\_id (+)

group by a.agent\_id

having count(subiect\_id) = (select min(count(subiect\_id)) --luam in considerare si agentii care nu supravegheaza pe nimeni

from supraveghere ss join agentt aa

on ss.agent\_id (+) = aa.agent\_id

group by aa.agent\_id)

fetch first 1 rows only;

insert into supraveghere

values (:new.subiect\_id, ex10aux.idAgNesolicit, sysdate);

else

delete from subiect where :old.subiect\_id = subiect\_id;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista niciun agent in baza de date care il poate urmari pe acest subiect');

end ex11a1;

create or replace trigger ex11a2

after insert or delete on persoana\_fizica

for each row

begin

ex10aux.canInsert := 0;

end ex11a2;

create or replace trigger ex11b1 --pentru pers\_juridica

before insert or delete on persoana\_juridica

for each row

begin

ex10aux.canInsert := 1;

if inserting then

insert into subiect values (:new.subiect\_id, null, null);

select distinct a.agent\_id into ex10aux.idAgNesolicit

from supraveghere s join agentt a

on a.agent\_id = s.agent\_id (+)

group by a.agent\_id

having count(subiect\_id) = (select min(count(subiect\_id)) --luam in considerare si agentii care nu supravegheaza pe nimeni

from supraveghere ss join agentt aa

on ss.agent\_id (+) = aa.agent\_id

group by aa.agent\_id)

fetch first 1 rows only;

insert into supraveghere

values (:new.subiect\_id, ex10aux.idAgNesolicit, sysdate);

else

delete from subiect where :old.subiect\_id = subiect\_id;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista niciun agent in baza de date care il poate urmari pe acest subiect');

end ex11b1;

create or replace trigger ex11b2

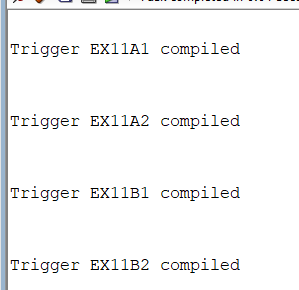
after insert or delete on persoana\_juridica

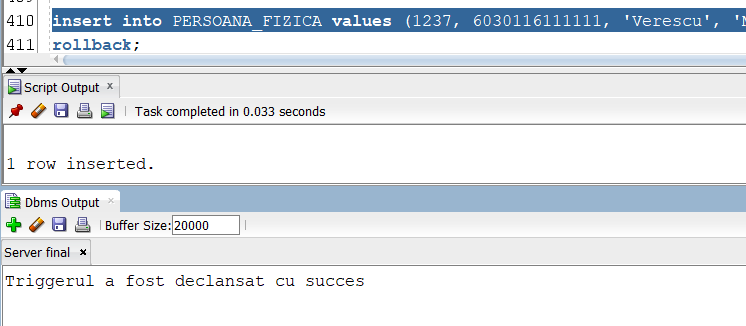
for each row

begin

ex10aux.canInsert := 0;

end ex11b2;





12. Trigger LDD.

Cerinta: A nu se permite stergerea de tabele (din cauza naturii de importanta si securitate a domeniului) si creearea de tabele cu un nume mai lung de 30 de caractere

create or replace trigger ex12a

before create on database

begin

if (length(ora\_dict\_obj\_name) > 30) then

raise\_application\_error(-20012, 'Lungimea numelui tabelului creat nu poate depasi 30 de caractere!');

end if;

end ex12a;

create or replace trigger ex12b --triggerul permite stergerea altor obiecte in afara de tabele

before drop on database

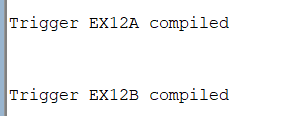
begin

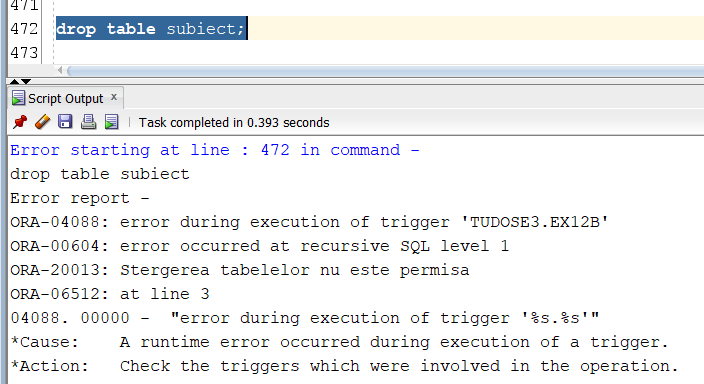
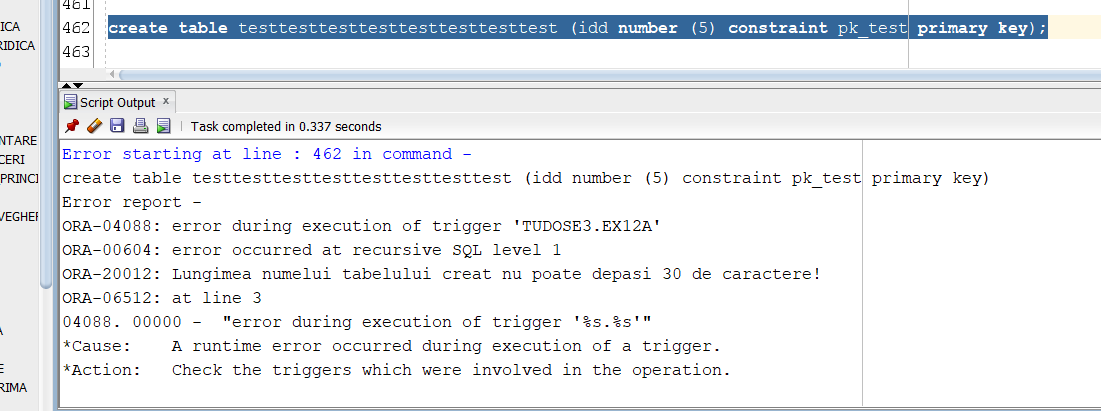
if (ora\_dict\_obj\_type = 'TABLE') then

raise\_application\_error(-20013, 'Stergerea tabelelor nu este permisa');

end if;

end ex12b;





Ex 13. Pachet cu obiecte deja predefinite

create or replace package ex13 as

PROCEDURE ex6;

procedure ex7;

function ex8(numeCr crime.nume%type, idSub subiect.subiect\_id%type) return number;

procedure ex9(nameInd industrie.nume%type, nameOcup ocupatie.nume\_ocup%type);

end ex13;

create or replace package body ex13 as

PROCEDURE ex6 AS

type tipColSub is record (subId subiect.subiect\_id%type, coefRisc subiect.coef\_risc%type);

type colSub is table of tipColSub;

type tabSub is table of colSub index by pls\_integer;

subiecti tabSub; --colectie de colectii de subiecti supravegheati, indexata de id-ul agentului

type recAg is record(agId agentt.agent\_id%type, nume agentt.nume%type, prenume agentt.prenume%type, rang agentt.rang%type);

type tabAg is varray(20) of recAg;

agenti tabAg := tabAg();

agIdCurent agentt.agent\_id%type;

BEGIN

select a.agent\_id, a.nume, a.prenume, a.rang

bulk collect into agenti

from agentt a left join supraveghere sup

on a.agent\_id = sup.agent\_id

left join subiect s on s.subiect\_id = sup.subiect\_id

group by a.agent\_id, a.nume, a.prenume, a.rang

order by nvl(sum(s.coef\_risc), 0) desc, a.rang desc;

if (agenti.count = 0) then

raise\_application\_error(-20000, 'Nu exista niciun agent inregistrat');

end if;

for i in agenti.first..agenti.last loop

agIdCurent := agenti(i).agId;

select s.subiect\_id, s.coef\_risc

bulk collect into subiecti(agIdCurent)

from supraveghere sup join subiect s

on sup.subiect\_id = s.subiect\_id

where sup.agent\_id = agIdCurent;

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

dbms\_output.put\_line('Locul ' || i || ': agentul ' || agenti(i).nume || ' ' || agenti(i).prenume ||', cu rangul ' || agenti(i).rang || ' . Suspecti supravegheati: ');

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

if (subiecti(agIdCurent).count = 0) then

dbms\_output.put\_line('Niciun suspect supravegheat');

else

for j in subiecti(agIdCurent).first..subiecti(agIdCurent).last loop

dbms\_output.put\_line('Suspectul cu id-ul ' || subiecti(agIdCurent)(j).subId || ', cu un coeficient de risc de ' || subiecti(agIdCurent)(j).coefRisc);

end loop;

end if;

end loop;

END ex6;

procedure ex7 as

type refcursorInt is ref cursor return interese%rowtype;

type pers is record (subId persoana\_fizica.subiect\_id%type, nume persoana\_fizica.nume%type, prenume persoana\_fizica.prenume%type);

type agenttt is record (nume agentt.nume%type, prenume agentt.nume%type);

type refcursorAg is ref cursor return agenttt;

agenti refcursorAg;

cursInterese refcursorInt;

pachet number := &p;

interes interese%rowtype;

numeSub persoana\_fizica.nume%type;

prenumeSub persoana\_fizica.prenume%type;

numeAg agentt.nume%type;

prenumeAg agentt.prenume%type;

cursor persFiz(interes interese.interes\_id%type) is --expresie cursor

select distinct nume, prenume,

cursor (select a.nume, a.prenume

from agentt a join supraveghere sup

on a.agent\_id = sup.agent\_id

where p.subiect\_id = sup.subiect\_id)

from persoana\_fizica p join interes\_om i

on p.subiect\_id = i.subiect\_id

where i.interes\_id = interes;

begin

if (pachet = 1) then

open cursInterese for --cursor dinamic

select \*

from interese

where risc\_criminalitate < 4;

elsif (pachet = 2) then

open cursInterese for

select \*

from interese

where risc\_criminalitate > 3 and risc\_criminalitate < 7;

elsif (pachet = 3) then

open cursInterese for

select \*

from interese

where risc\_criminalitate > 6;

else

raise\_application\_error(-20001, 'Optiune incorecta');

end if;

loop

fetch cursInterese into interes;

exit when cursInterese%notfound;

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

dbms\_output.put\_line('Pentru interesul ' || interes.nume|| ', cu un risc de criminalitate de ' || interes.risc\_criminalitate);

dbms\_output.put\_line('au fost identificati urmatorii:');

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

--for i in persFiz(interes.interes\_id) loop

open persFiz(interes.interes\_id); --cursor clasic parametrizat

loop

fetch persFiz into numeSub, prenumeSub, agenti;

exit when persFiz%notfound;

dbms\_output.put(numeSub || ' ' || prenumeSub || '; supravegheat de: ');

loop

fetch agenti into numeAg, prenumeAg;

exit when agenti%notfound;

dbms\_output.put(numeAg || ' ' || prenumeAg || ', ');

end loop;

if (agenti%rowcount = 0) then

dbms\_output.put(' niciun agent');

end if;

dbms\_output.new\_line();

end loop;

if (persFiz%rowcount = 0) then

dbms\_output.put\_line('Nu a fost identificat niciun subiect cu acest interes');

end if;

close persFiz;

end loop;

if cursInterese%rowcount = 0 then

dbms\_output.put\_line('Nu exista niciun interes in intervalul selectat al riscului de criminalitate');

end if;

close cursInterese;

end ex7;

function ex8(numeCr crime.nume%type, idSub subiect.subiect\_id%type) return number

is

nonExistSubject exception;

pragma exception\_init(nonExistSubject, -20005);

nonExistCrime exception;

pragma exception\_init(nonExistCrime, -20006);

durata number;

testt1 number;

begin

select count(subiect\_id) into testt1

from subiect

where subiect\_id = idSub;

if (testt1 = 0) then

raise nonExistSubject;

end if;

select count(crima\_id) into testt1

from crime

where upper(nume) = upper(numeCr);

if (testt1 = 0) then

raise nonExistCrime;

end if;

select durata\_comitere into durata

from suspiciune\_crima sc join crime c

on sc.crima\_id = c.crima\_id

join subiect s

on s.subiect\_id = sc.subiect\_id

where upper(c.nume) = upper(numeCr) and idSub = s.subiect\_id;

return durata;

EXCEPTION

when no\_data\_found then --cand exista si crima si subiectul, dar sunt asociati

dbms\_output.put\_line('Subiectul nu este suspect de aceasta crima');

when nonExistCrime then

dbms\_output.put\_line('Serviciul roman de informatii nu se ocupa cu astfel de crime');

when nonExistSubject then

dbms\_output.put\_line('Id-ul transmis nu apartine niciunui subiect inregistrat.');

when others then

dbms\_output.put\_line('Exista o eroare neprevazuta, ceea ce nu e posibil, pentru ca noi suntem SRI si prevedem totul. Cel mai probabil

ati ajuns aici pentru ca a gresit procesorul la calcule');

end ex8;

procedure ex9(nameInd industrie.nume%type, nameOcup ocupatie.nume\_ocup%type) is

nonExistOcup exception;

nonExistInd exception;

tooManySubj exception;

noSubj exception;

pragma Exception\_Init(tooManySubj, -20010);

pragma Exception\_Init(noSubj, -2009);

pragma Exception\_Init(nonExistOcup, -20007);

pragma Exception\_init(nonExistInd, -20008);

testt number;

interesRez interese.nume%type;

BEGIN

select count(ocup\_id) into testt

from ocupatie

where upper(nume\_ocup) = upper(nameOcup);

if (testt = 0) then

raise nonExistOcup;

end if;

select count(industrie\_id) into testt

from industrie

where upper(nume) = upper(nameInd);

if(testt = 0) then

raise nonExistInd;

end if;

select count (subiect\_id) into testt

from ocup\_om oo join ocupatie ocup on oo.ocup\_id = ocup.ocup\_id

join industrie i on i.industrie\_id = oo.industrie\_id

where upper(i.nume) = upper(nameInd) and upper(ocup.NUME\_OCUP) = upper(nameOcup);

if (testt = 0) then

raise noSubj;

elsif (testt > 1) then

raise tooManySubj;

end if;

select i.nume into interesRez

from interese i join interes\_om io

on i.interes\_id = io.interes\_id

join persoana\_fizica pf

on pf.subiect\_id = io.subiect\_id

join ocup\_om oo

on pf.subiect\_id = oo.subiect\_id

join industrie ind

on ind.industrie\_id = oo.industrie\_id

join ocupatie ocup

on ocup.ocup\_id = oo.ocup\_id

where upper(ind.nume) = upper(nameInd) and upper(ocup.NUME\_OCUP) = upper(nameOcup);

dbms\_output.put\_line('Interesul unui subiect ce respecta acesti parametrii este ' || interesRez);

exception

when nonExistInd then

dbms\_output.put\_line('Industria introdusa nu este analizata de SRI. Incercati la STS.');

when nonExistOcup then

dbms\_output.put\_line('Ocupatia introdusa nu este suficient de importanta pentru a fi urmarita de noi.');

when tooManySubj then

dbms\_output.put\_line('Exista mai mult de un subiect care se incadreaza in parametrii oferiti');

when noSubj then

dbms\_output.put\_line('Nu se incadreaza nimeni in parametrii oferiti');

when too\_many\_rows then

dbms\_output.put\_line('Subiectul identificat are mai mult de un singur interes');

when no\_data\_found then

dbms\_output.put\_line('Subiectul nu are niciun interes');

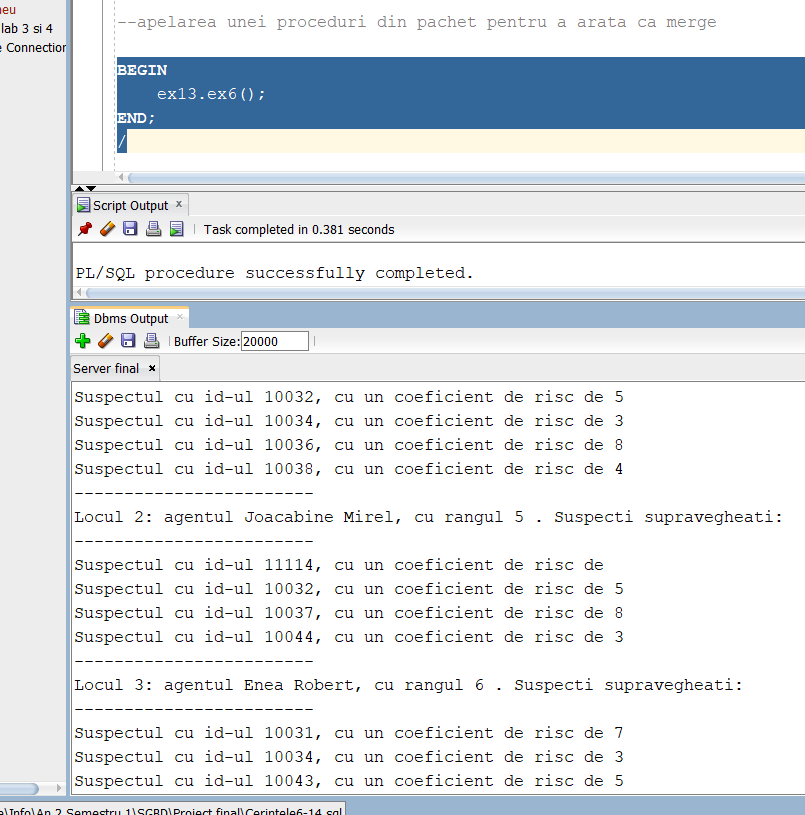
when others then

dbms\_output.put\_line('Exista o eroare neprevazuta, ceea ce nu e posibil, pentru ca noi suntem SRI si prevedem totul. Cel mai probabil

ati ajuns aici pentru ca a gresit procesorul la calcule');

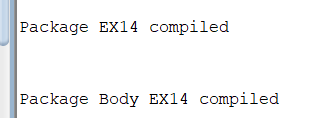
END ex9;

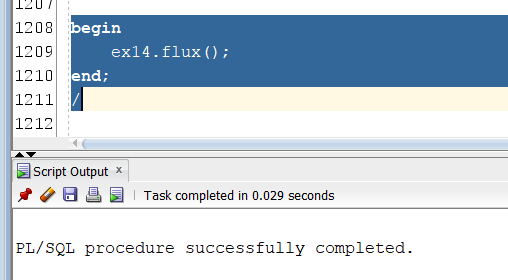
end ex13;



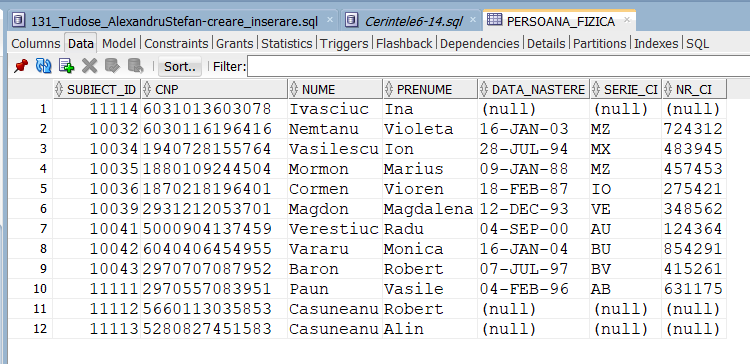
Ex 14.

Cerinta: sa se creeze un job care adauga in fiecare zi o persoana fizica. In lipsa unui serviciu de colectare a informatiei despre acesta, numele, prenumele, si cnp-ul, interesele si suspiciunile sale de crima vor fi automatizate

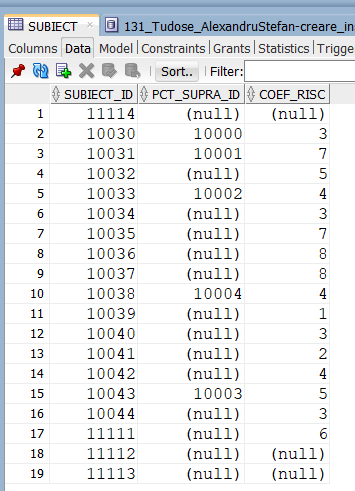




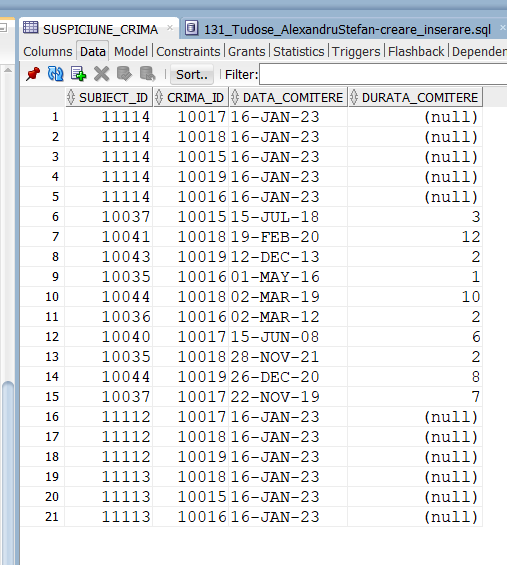
Am dat run de doua ori si s-au inserat 2 persoane fizice



s-au propagat in tabela “subiect” prin trigger-ul de la ex 11.



Li s-au asociat suspiciuni de crima, conform metodei insertDataCr din flux.



Li s-au asociat interese, conform metodei insertDataInt din flux.

