Proiect- Sisteme de Gestiune a Bazelor de Date

Proiectarea si implementarea unei baze de date relationale

Creita Andreea-Georgiana

Grupa 232

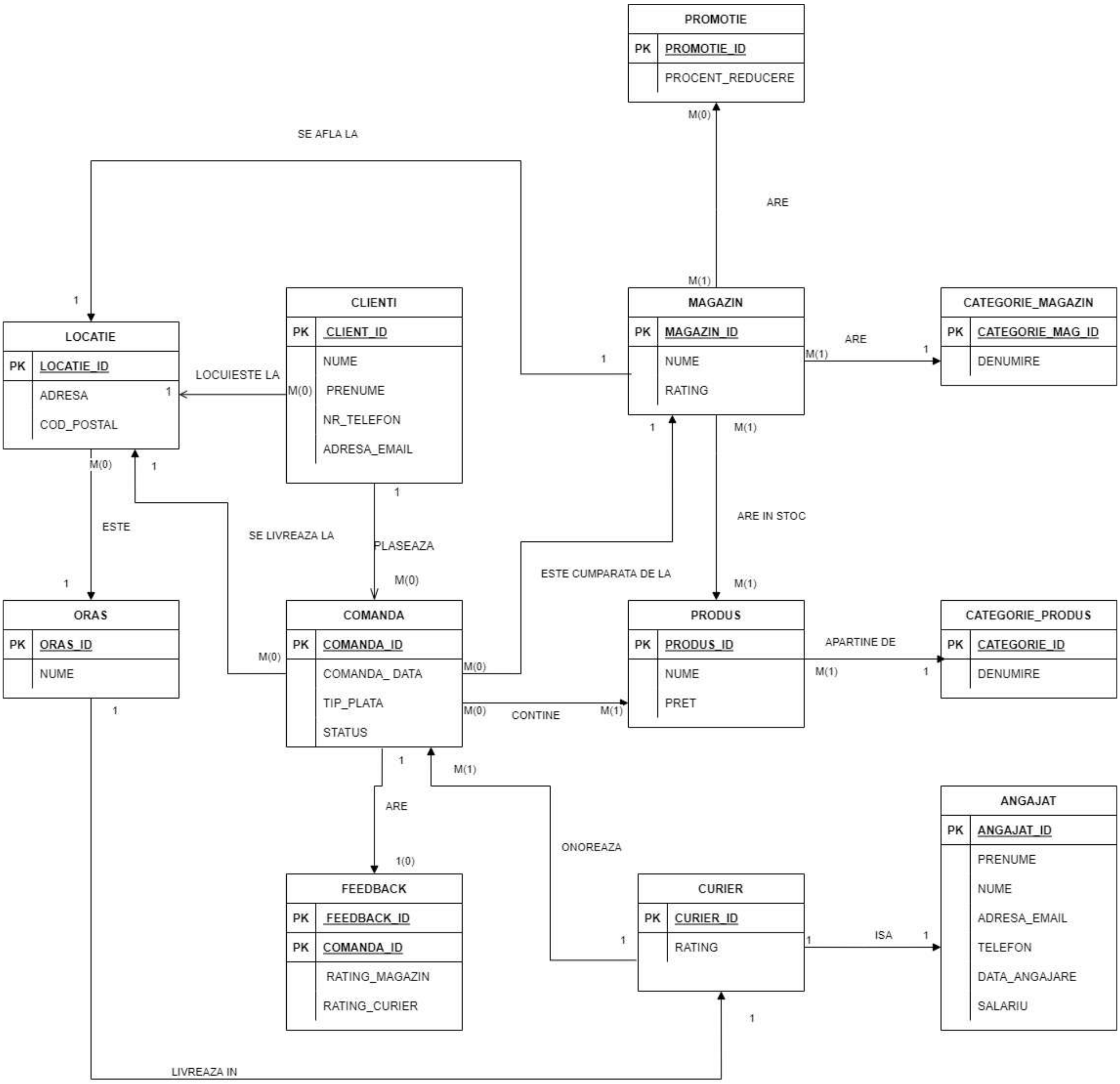
1.**Prezentați pe scurt baza de date (utilitatea ei)**.

Baza de date prezentată va fi cea a unei aplicații de livrare a produselor din magazine la domiciliu, birou, etc.

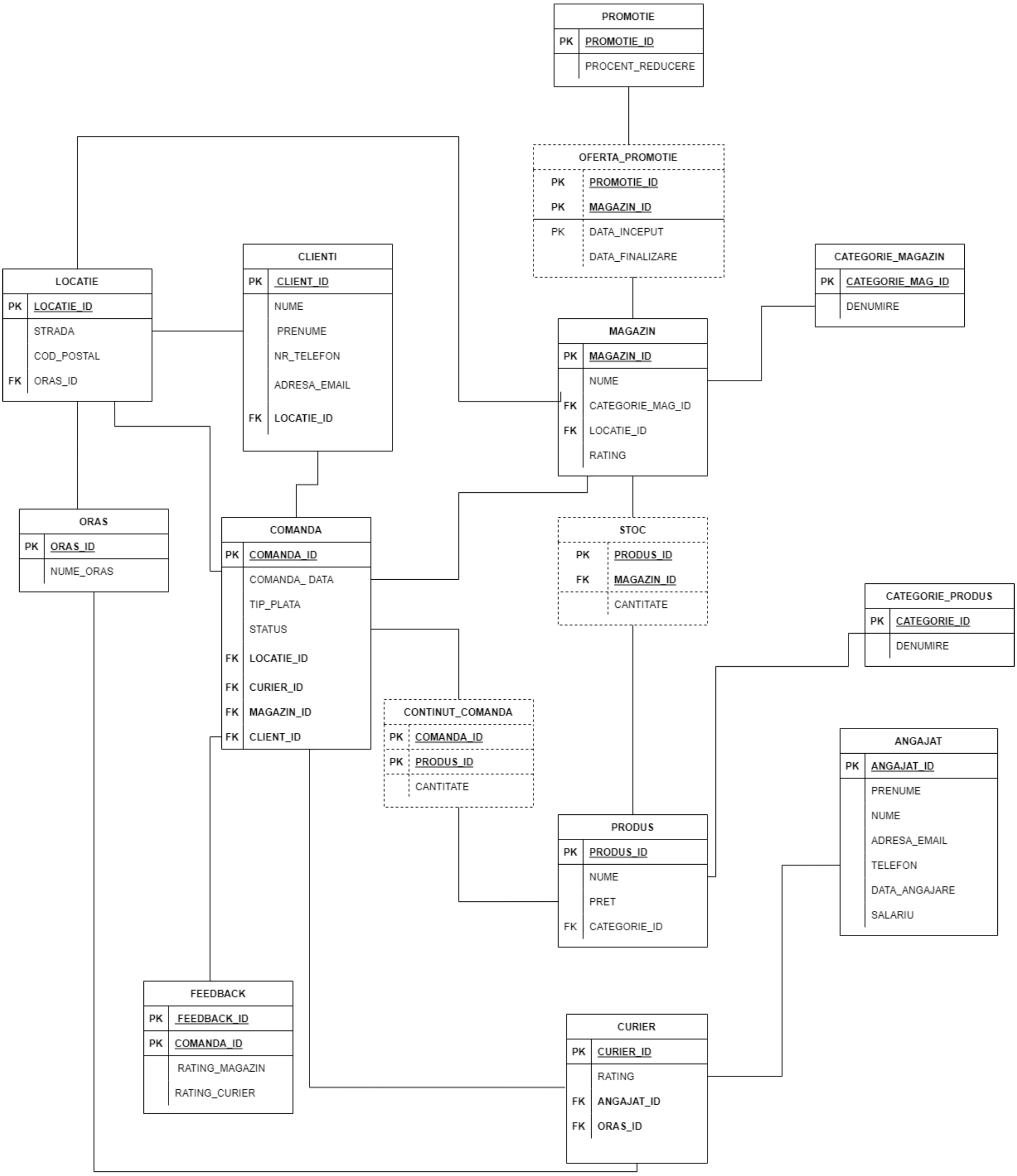
Precum aplicațiile folosite pentru comandarea de mâncare, o astfel de aplicație ce permite achiziționarea de produse ce vor fi livrate la o adresa anume, oferă o oportunitate pentru persoanele care nu au timp sa meargă la magazin sau nu au posibilitatea de a o face.

2. Realizați diagrama entitate-relație (ERD).

Am creat o baza de date ce contine 12 entitati independente.



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

--tabele

create table oras (

```
   oras_id NUMBER(5) not null,
   nume_oras Varchar2(25) not null,
   constraint PK_oras  primary key (oras_id)
);

create table locatie (
   locatie_id            number(5)                not null,
   strada                varchar2(60)             not null,
   cod_postal            number(10),
   oras_id               number(5)                not null,
   constraint PK_locatie  primary key (locatie_id),
   foreign key(oras_id) references oras(oras_id) on delete cascade
);

create table angajat (
   angajat_id                   number(5)        not null,
   prenume                    varchar2(35)       not null,
   nume                   varchar2(35)       not null,
   adresa_email                varchar2(35),
   telefon                    number(21)        not null,
   data_angajare                 date          not null,
   salariu                      number(21)        not null,
   constraint PK_angajat  primary key ( angajat_id)



);

create table curier (                                    --id, rating, ang id, oras id
   curier_id        number(5) not null,
   rating           number(3,2) default 5,
   angajat_id       number(5) not null,
   oras_id          number(5) ,
   constraint PK_curier_id  primary key (curier_id),
   foreign key(angajat_id) references angajat(angajat_id) on delete cascade,
   foreign key(oras_id) references oras(oras_id) on delete cascade
);


create table clienti(                                   --client id, prenume, nume, tel, email, locatie_id
   client_id          number(5)          not null,
   prenume            varchar2(20)       not null,
   nume               varchar2(25)       not null,
   nr_telefon         varchar2(20)       not null,
   adresa_email       varchar2(35),
   locatie_id         number(5),
   constraint PK_client_id  primary key (client_id),
   foreign key(locatie_id) references locatie(locatie_id) on delete cascade

);

create table categorie_magazin (                              --categorie id, denumire
   categorie_mag_id          number(5)       not null,
   denumire                  varchar2(25)    not null,
   constraint PK_categorie_mag_id  primary key (categorie_mag_id)
);

create table magazin (                                    --magazin id, nume, categorie_mag_id,locatie_id, rating
   magazin_id         number(5)            not null,
   nume               varchar2(25)          not null,
   categorie_mag_id    number(5)           not null,
   locatie_id         number(5)            not null,
   rating             number(3,2) default 5,


   constraint PK_magazin_id  primary key (magazin_id),

   foreign key(locatie_id) references locatie(locatie_id) on delete cascade,


   foreign key (categorie_mag_id) references categorie_magazin(categorie_mag_id) on delete
cascade

);






create table promotie (                                    --prom id, procent reducere

   promotie_id        number(5)      not null,

   procent_reducere    number(5)      not null,

   constraint promotie_id  primary key (promotie_id)

);


create table oferta_promotie (                             --prom id, mag id, data inceput, data
finalizare

   promotie_id        number(5)         not null references promotie(promotie_id),

   magazin_id         number(5)         not null references magazin(magazin_id),

   data_inceput       date            not null,

   data_finalizare     date            not null,

   constraint data_inceput  primary key (data_inceput)

);





create table categorie_produs (                    --categ id, den,

   categorie_id       number(5) not null,

   denumire          varchar(25) not null,
   constraint PK_categorie_id  primary key(categorie_id)
);



create table produs (                           --produs id, nume, pret, categorie_id
   produs_id      number(5)      not null,
   nume          varchar(25)    not null,
   pret          number(5)      ,
   categorie_id    number(5)      ,
   constraint PK_produs_id   primary key(produs_id ),
   foreign key (categorie_id) references categorie_produs(categorie_id) on delete cascade
);



create table stoc (                              --stoc id, mag id, cantitate
   produs_id      number(5)         not null references produs(produs_id),
   magazin_id     number(5)         not null references magazin(magazin_id),
   cantitate      number(5)
);

create table comanda (                                  --comanda id, comanda data, tip plata, status,
locatie id, curier id, mag id, client id
   comanda_id       number(5)              not null,
   comanda_data     date DEFAULT sysdate     not null,
   tip_plata        varchar(25)             ,
   status          varchar(25)            not null,
   locatie_id      number(5)               not null,
   curier_id       number(5),
   magazin_id      number(5)              not null,
   client_id       number(5)              not null,
   constraint PK_comanda_id    primary key(comanda_id  ),
   foreign key (locatie_id) references locatie(locatie_id) on delete cascade,
   foreign key (curier_id) references curier(curier_id) on delete cascade,
   foreign key (magazin_id) references magazin(magazin_id) on delete cascade,
   foreign key (client_id) references clienti(client_id) on delete cascade
);
create table continut_comanda (                    --comanda_id,produs_id, cantitate
   comanda_id     number(5)        not null references comanda(comanda_id),
   produs_id      number(5)        not null references produs(produs_id),
   cantitate      number(5) not null
);
create table  feedback_comanda(                    --feedback_id, comanda_id, rating_mag,
rating_curier
   feedback_id        number(5) not null,
   comanda_id         number(5) not null,
   rating_magazin     number(3,2) default 5,
   rating_curier      number(3,2) default 5,
   constraint PK_feedback_id    primary key(feedback_id  ),
   foreign key (comanda_id) references comanda(comanda_id) on delete cascade

);

select * from oras;
select * from locatie;
select * from angajat;
select * from curier;
select * from clienti;
select * from categorie_magazin;
select * from magazin;
select * from promotie;
select * from oferta_promotie;
select * from categorie_produs;
```

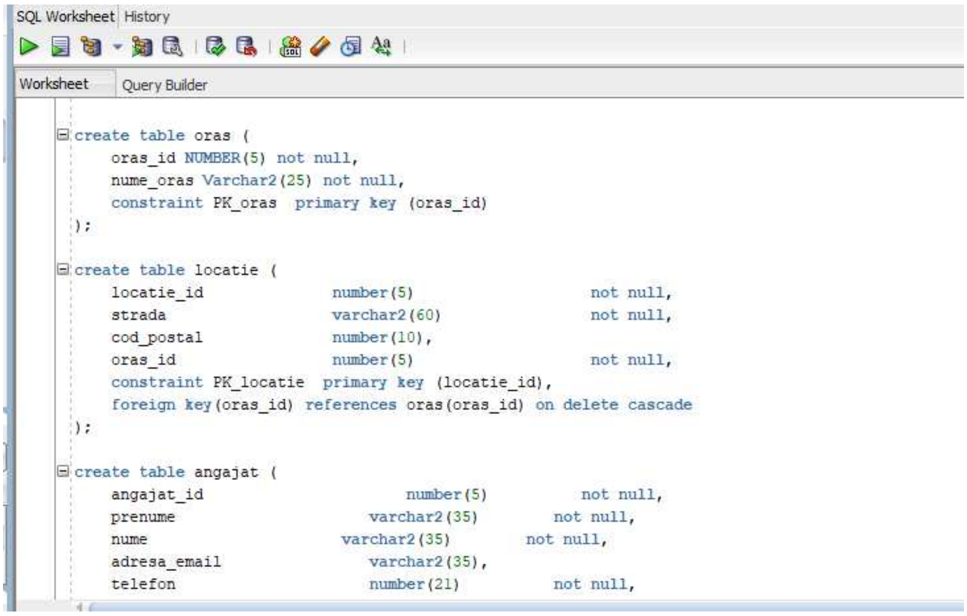
select * from produs;

select * from stoc;

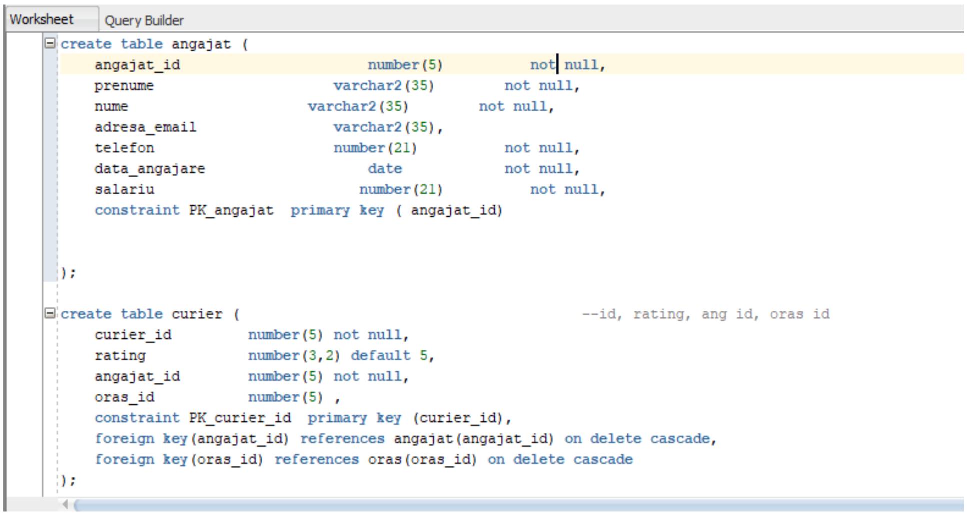
select * from comanda;

select * from continut_comanda;

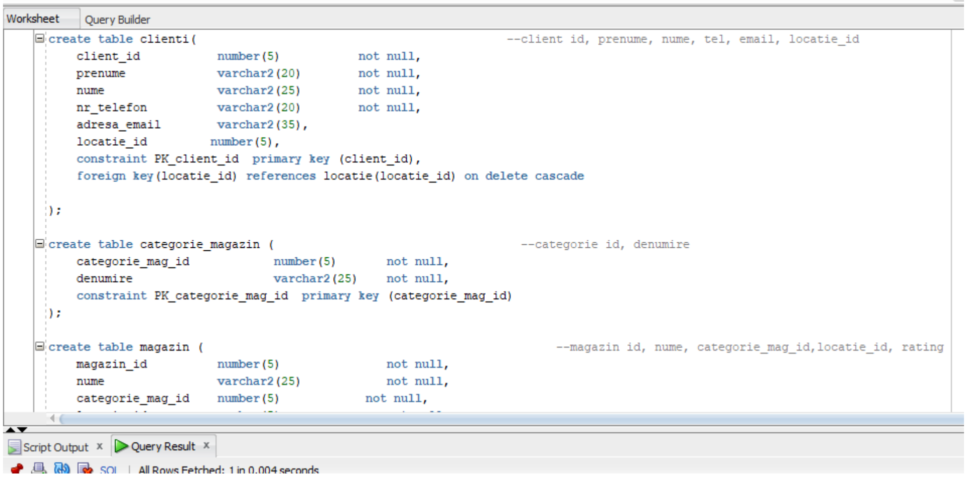
select * from feedback_comanda;



```
create table oras (
    oras_id NUMBER(5) not null,
    nume_oras Varchar2(25) not null,
    constraint PK_oras  primary key (oras_id)
);

create table locatie (
    locatie_id             number(5)                not null,
    strada                 varchar2(60)             not null,
    cod_postal             number(10),
    oras_id                number(5)                not null,
    constraint PK_locatie  primary key (locatie_id),
    foreign key(oras_id) references oras(oras_id) on delete cascade
);

create table angajat (
    angajat_id                     number(5)          not null,
    prenume                    varchar2(35)         not null,
    nume                   varchar2(35)         not null,
    adresa_email               varchar2(35),
    telefon                    number(21)           not null,
```

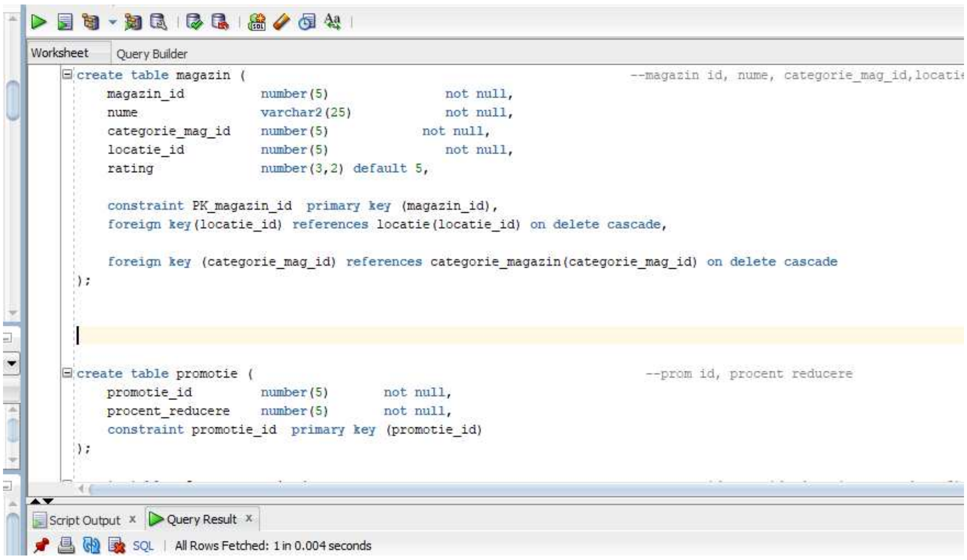


```
create table angajat (
    angajat_id                      number(5)          not null,
    prenume                     varchar2(35)         not null,
    nume                    varchar2(35)         not null,
    adresa_email                varchar2(35),
    telefon                     number(21)           not null,
    data_angajare                   date             not null,
    salariu                        number(21)           not null,
    constraint PK_angajat  primary key ( angajat_id)


);

create table curier (                                     --id, rating, ang id, oras id
    curier_id          number(5) not null,
    rating             number(3,2) default 5,
    angajat_id         number(5) not null,
    oras_id            number(5) ,
    constraint PK_curier_id  primary key (curier_id),
    foreign key(angajat_id) references angajat(angajat_id) on delete cascade,
    foreign key(oras_id) references oras(oras_id) on delete cascade
);
```



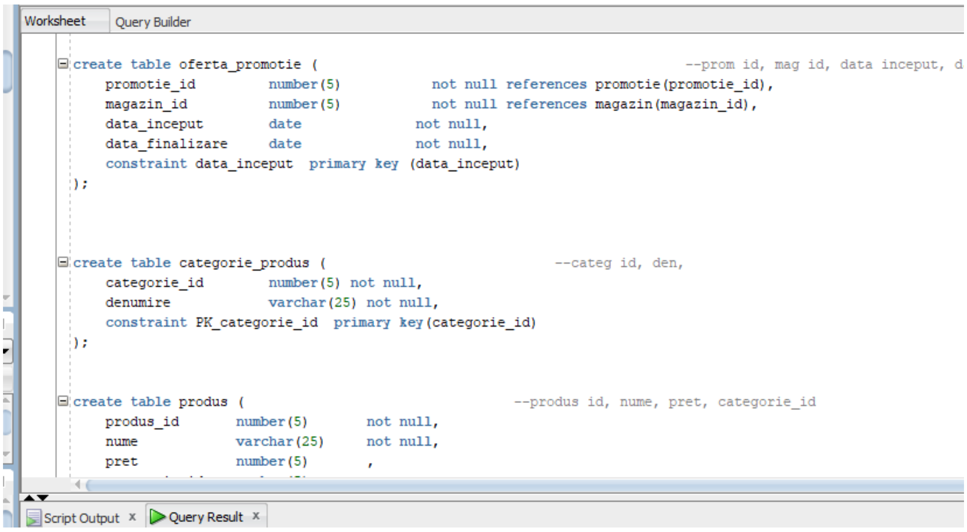
```
create table clienti(                                              --client id, prenume, nume, tel, email, locatie_id
    client_id           number(5)           not null,
    prenume             varchar2(20)        not null,
    nume                varchar2(25)        not null,
    nr_telefon          varchar2(20)        not null,
    adresa_email        varchar2(35),
    locatie_id         number(5),
    constraint PK_client_id  primary key (client_id),
    foreign key(locatie_id) references locatie(locatie_id) on delete cascade

);

create table categorie_magazin (                                     --categorie id, denumire
    categorie_mag_id            number(5)       not null,
    denumire                    varchar2(25)    not null,
    constraint PK_categorie_mag_id  primary key (categorie_mag_id)
);

create table magazin (                                                    --magazin id, nume, categorie_mag_id,locatie_id, rating
    magazin_id          number(5)               not null,
    nume                varchar2(25)            not null,
    categorie_mag_id    number(5)            not null,
```

All Rows Fetched: 1 in 0.004 seconds

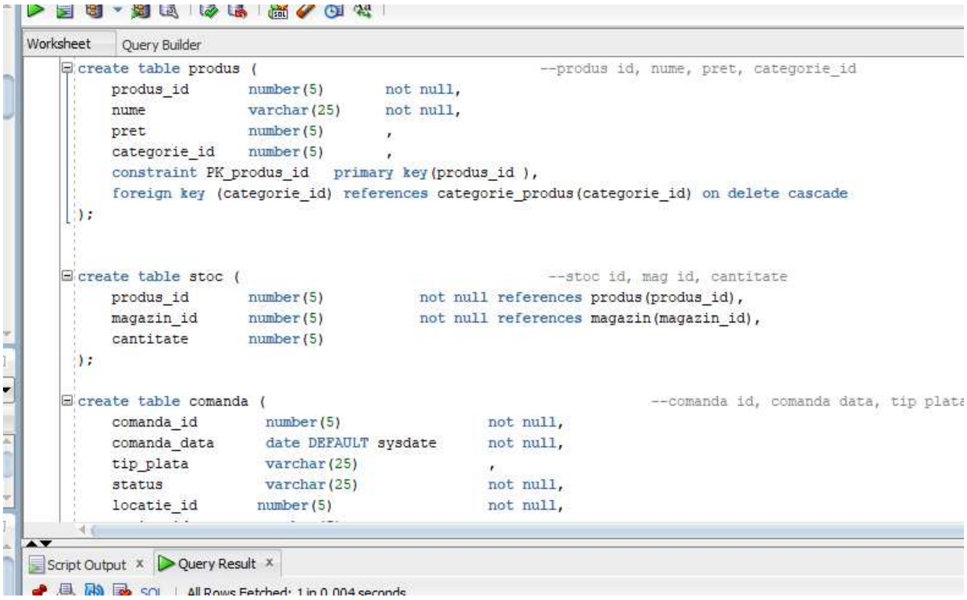


```
create table magazin (                                              --magazin id, nume, categorie_mag_id,locatie
    magazin_id          number(5)               not null,
    nume                varchar2(25)            not null,
    categorie_mag_id    number(5)            not null,
    locatie_id          number(5)               not null,
    rating              number(3,2) default 5,

    constraint PK_magazin_id  primary key (magazin_id),
    foreign key(locatie_id) references locatie(locatie_id) on delete cascade,

    foreign key (categorie_mag_id) references categorie_magazin(categorie_mag_id) on delete cascade
);



create table promotie (                                                --prom id, procent reducere
    promotie_id         number(5)       not null,
    procent_reducere    number(5)       not null,
    constraint promotie_id  primary key (promotie_id)
);
```

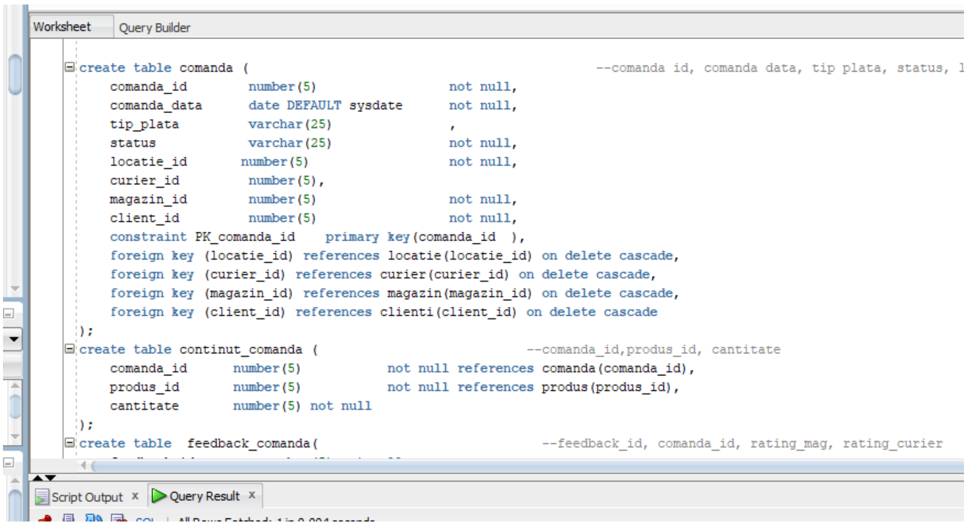
All Rows Fetched: 1 in 0.004 seconds



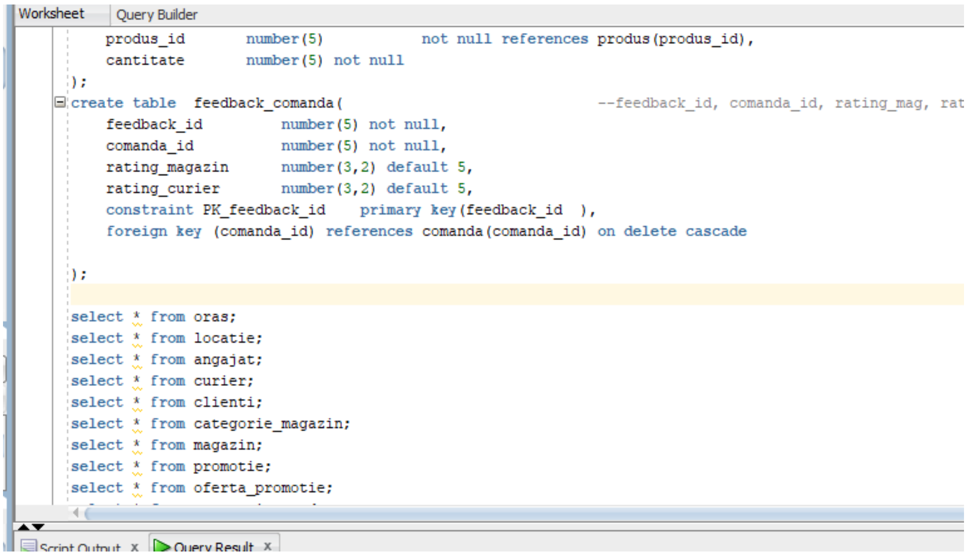
```
create table oferta_promotie (                                              --prom id, mag id, data inceput, d
    promotie_id         number(5)           not null references promotie(promotie_id),
    magazin_id          number(5)           not null references magazin(magazin_id),
    data_inceput        date              not null,
    data_finalizare     date              not null,
    constraint data_inceput  primary key (data_inceput)
);



create table categorie_produs (                               --categ id, den,
    categorie_id        number(5) not null,
    denumire            varchar(25) not null,
    constraint PK_categorie_id  primary key(categorie_id)
);


create table produs (                                   --produs id, nume, pret, categorie_id
    produs_id       number(5)       not null,
    nume            varchar(25)     not null,
    pret            number(5)       ,
```



```
create table produs (                                       --produs id, nume, pret, categorie_id
    produs_id       number(5)       not null,
    nume            varchar(25)     not null,
    pret            number(5)       ,
    categorie_id    number(5)       ,
    constraint PK_produs_id   primary key(produs_id ),
    foreign key (categorie_id) references categorie_produs(categorie_id) on delete cascade
);


create table stoc (                                          --stoc id, mag id, cantitate
    produs_id       number(5)           not null references produs(produs_id),
    magazin_id      number(5)           not null references magazin(magazin_id),
    cantitate       number(5)
);

create table comanda (                                                  --comanda id, comanda data, tip plata
    comanda_id          number(5)               not null,
    comanda_data        date DEFAULT sysdate    not null,
    tip_plata           varchar(25)             ,
    status              varchar(25)             not null,
    locatie_id         number(5)                not null,
```



```
create table comanda (                                          --comanda id, comanda data, tip plata, status, l
    comanda_id          number(5)               not null,
    comanda_data        date DEFAULT sysdate    not null,
    tip_plata           varchar(25)             ,
    status              varchar(25)             not null,
    locatie_id         number(5)                not null,
    curier_id           number(5),
    magazin_id          number(5)               not null,
    client_id           number(5)               not null,
    constraint PK_comanda_id    primary key(comanda_id  ),
    foreign key (locatie_id) references locatie(locatie_id) on delete cascade,
    foreign key (curier_id) references curier(curier_id) on delete cascade,
    foreign key (magazin_id) references magazin(magazin_id) on delete cascade,
    foreign key (client_id) references clienti(client_id) on delete cascade
);
create table continut_comanda (                         --comanda_id,produs_id, cantitate
    comanda_id      number(5)           not null references comanda(comanda_id),
    produs_id       number(5)           not null references produs(produs_id),
    cantitate       number(5) not null
);
create table  feedback_comanda(                           --feedback_id, comanda_id, rating_mag, rating_curier
```



```
    produs_id       number(5)           not null references produs(produs_id),
    cantitate       number(5) not null
);
create table  feedback_comanda(                           --feedback_id, comanda_id, rating_mag, rat
    feedback_id         number(5) not null,
    comanda_id          number(5) not null,
    rating_magazin      number(3,2) default 5,
    rating_curier       number(3,2) default 5,
    constraint PK_feedback_id    primary key(feedback_id  ),
    foreign key (comanda_id) references comanda(comanda_id) on delete cascade

);

select * from oras;
select * from locatie;
select * from angajat;
select * from curier;
select * from clienti;
select * from categorie_magazin;
select * from magazin;
select * from promotie;
select * from oferta_promotie;
```

5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

--inserare in tabele

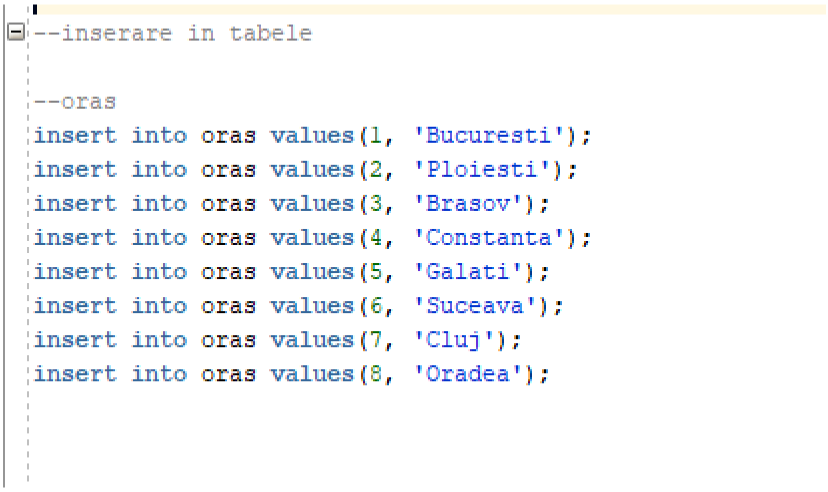
```
--oras
insert into oras values(1, 'Bucuresti');
insert into oras values(2, 'Ploiesti');
insert into oras values(3, 'Brasov');
insert into oras values(4, 'Constanta');
insert into oras values(5, 'Galati');
insert into oras values(6, 'Suceava');
insert into oras values(7, 'Cluj');
insert into oras values(8, 'Oradea');



--locatie
insert into locatie values(1,'Strada George Cosbuc',410001,8);
insert into locatie values(2,'Strada Avram Iancu', 500001,3);
insert into locatie values(3,'Strada Olteniei',900001,4);
insert into locatie values(4,'Strada Baraganului',900001,4);
insert into locatie values(5,'Strada Brasov', 400000,7);
insert into locatie values(6,'Strada Ropotului',010001,1);
insert into locatie values(7,'Strada Izvoare',100078,2);
insert into locatie values(8,'Strada General Manu',900001,4);




--angajat
```

```
--id,prenume,nume,mail,tel,data ang, salariu



insert into angajat values(1,'Ana',
'Grigorescu','ana.grigo@yahoo.com',2354365745,to_date('22/03/1999', 'DD/MM/YYYY'),230000);

insert into angajat values(2,'Roxana'
,'Amerinde','roxi.roxi@yahoo.com',123428950,to_date('22/03/1989', 'DD/MM/YYYY'),233444);

insert into angajat values(3,'Christian','
Garry','christian_garry@yahoo.com',5363245367,to_date('22/03/1998', 'DD/MM/YYYY'),456432);

insert into angajat values(4,'Robert','
Pattiosana','rob.pattiosana@gmail.com',3425349786,to_date('22/03/2000',
'DD/MM/YYYY'),342111);

insert into angajat values(5,'Emilia','
Venedi','emilia_venedi@gmail.com',4573645687,to_date('22/03/2011', 'DD/MM/YYYY'),100000);

insert into angajat values(6,'Venti','
Vernindo','Venti_Vent@yahoo.com',6445636667,to_date('22/03/2013', 'DD/MM/YYYY'),200000);

insert into angajat values(7,'Malumi','
Shay','Malumi.Ver@gmail.com',4357564568,to_date('22/03/2019', 'DD/MM/YYYY'),340560);

insert into angajat values(8,'Surimi','
Summer','Summer_surimi@gmail.com',989078896,to_date('22/03/2014', 'DD/MM/YYYY'),340000);

insert into angajat values(9,'Masamoto','
Yukata','Masa_moto@yahoo.com',0086575445,to_date('22/03/2021', 'DD/MM/YYYY'),100000);



--curier

--id, rating, ang id, oras id



insert into curier values(1,2,3,2);

insert into curier values(2,3.45,3,1);

insert into curier values(3,5,1,4);

insert into curier values(4,6,8,3);
insert into curier values(5,4.4,9,1);

insert into curier values(6,3,5,7);

insert into curier values(7,1,6,5);

insert into curier values(8,4.45,7,1);

insert into curier values(9,4.67,4,6);


select * from curier;

--categorie magazin

  --categorie id, denumire


insert into categorie_magazin values(1,'restaurant');

insert into categorie_magazin values(2,'electrocasnice');

insert into categorie_magazin values(3,'magazin alimente');

insert into categorie_magazin values(4,'macelarie');

insert into categorie_magazin values(5,'piata');

insert into categorie_magazin values(6,'florarie');

insert into categorie_magazin values(7,'pet shop');

select * from categorie_magazin;

--magazin

----magazin id, nume, categorie_mag_id,locatie_id, rating


insert into magazin values(1,'Carrefour', 3, 1,5);                  --mag alimente

insert into magazin values(2,'Altex',2,5,3.6);                      --electro

insert into magazin values(3,'Piata Obor',5,1,4.5);                    --piata

insert into magazin values(4,'Arabella',4,5,3.3);                   --macelarie

insert into magazin values(5,'Nirvana',6,7,4.2);                    --florarie

insert into magazin values(6,'Anavrin',7,8,3.4);                    --pet shop

insert into magazin values(7,'Siren',1,3,5);                     --restaurant

insert into magazin values(8,'Shanon',4, 2, 4.5);                   --macelarie

insert into magazin values(9,'La Nuci',1,7,4);                     --restaurant

insert into magazin values(10,'Chirila',3,2,2.3);                    --mag alim
insert into magazin values(11,'Simp',4,5, 5);                    --macelarie

insert into magazin values(12,'Afrodita', 7,8,3);                 --pet shop

insert into magazin values(13,'La Teo',1,4,4.4);                  --restaurant

insert into magazin values(14,'Chika',2,6,5);                    --electro


select * from magazin;


--prom id, procent reducere

insert into promotie values(1,10);

insert into promotie values(2,20);

insert into promotie values(3,20);

insert into promotie values(4,30);

insert into promotie values(5,40);

insert into promotie values(6,70);

insert into promotie values(7,50);


--oferta_promotie

--prom id, mag id, data inceput, data finalizare


insert into oferta_promotie values(1,1,to_date('22/03/2022', 'DD/MM/YYYY'),to_date('22/04/2022',
'DD/MM/YYYY'));

insert into oferta_promotie values(2,2,to_date('12/01/2022', 'DD/MM/YYYY'),to_date('20/03/2022',
'DD/MM/YYYY'));

insert into oferta_promotie values(3,3,to_date('05/07/2022', 'DD/MM/YYYY'),to_date('22/07/2022',
'DD/MM/YYYY'));

insert into oferta_promotie values(4,4,to_date('29/08/2012', 'DD/MM/YYYY'),to_date('05/09/2022',
'DD/MM/YYYY'));

insert into oferta_promotie values(5,5,to_date('13/12/2022', 'DD/MM/YYYY'),to_date('13/10/2022',
'DD/MM/YYYY'));

insert into oferta_promotie values(6,6,to_date('22/09/2012', 'DD/MM/YYYY'),to_date('22/11/2022',
'DD/MM/YYYY'));

insert into oferta_promotie values(7,7,to_date('26/03/2022', 'DD/MM/YYYY'),to_date('22/05/2022',
'DD/MM/YYYY'));
```

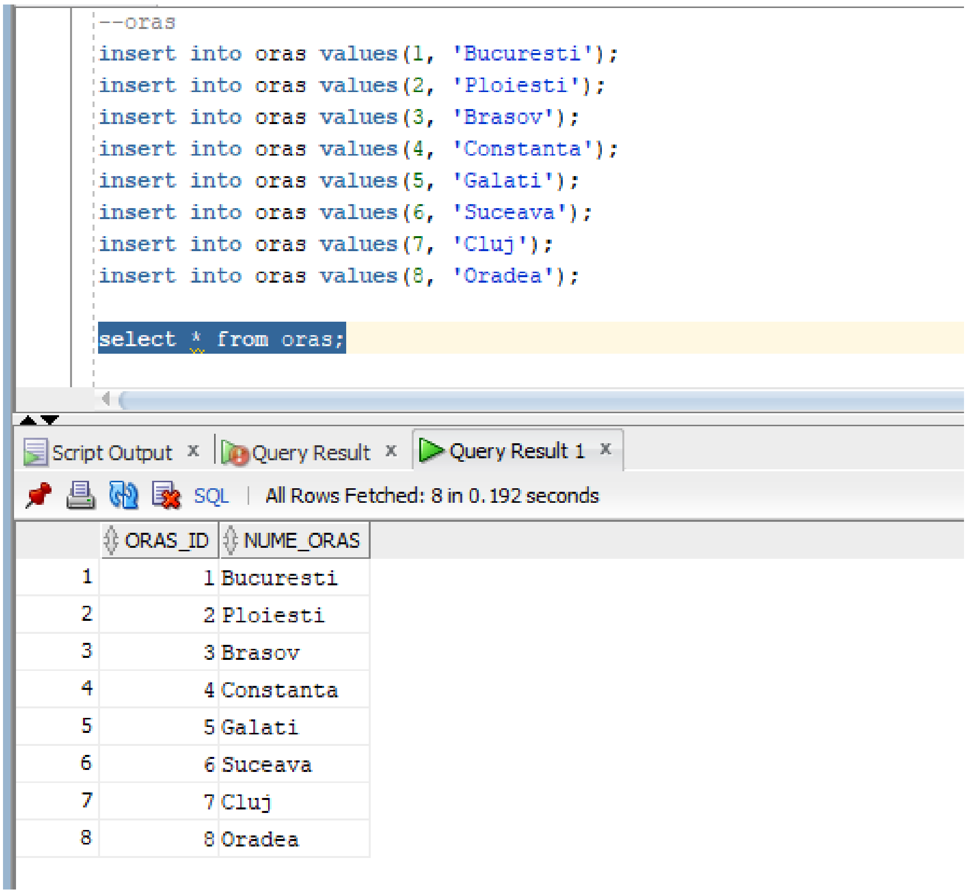
```
--insert into oferta_promotie values(8,10,to_date('16/04/2022',
'DD/MM/YYYY'),to_date('22/04/2022', 'DD/MM/YYYY'));
--insert into oferta_promotie values(8,2,to_date('08/03/2022',
'DD/MM/YYYY'),to_date('14/03/2022', 'DD/MM/YYYY'));
--insert into oferta_promotie values(9,12,to_date('03/05/2022',
'DD/MM/YYYY'),to_date('02/06/2022', 'DD/MM/YYYY'));
--insert into oferta_promotie values(10,8,to_date('22/10/2022',
'DD/MM/YYYY'),to_date('27/10/2022', 'DD/MM/YYYY'));


select * from oferta_promotie;
--categorie_produs
--categ id, den
insert into categorie_produs values(1,'electrocasnice');
insert into categorie_produs values(2,'alimente');
insert into categorie_produs values(3,'mezeluri');
insert into categorie_produs values(4, 'lactate');
insert into categorie_produs values(5,'hrana animale');
insert into categorie_produs values(6, 'flori');
insert into categorie_produs values(7, 'carne');
insert into categorie_produs values(8, 'panificatie');
insert into categorie_produs values(9, 'gadget');
insert into categorie_produs values(10, 'laptop');
insert into categorie_produs values(11, 'jucarii animale');
insert into categorie_produs values(12, 'fructe');



--produs
--produs id, nume, pret, categorie_id


insert into produs values(1,'trandafir',3, 6);
insert into produs values(2,'lalele',1,6);
insert into produs values(3,'narcisa',2,6);
insert into produs values(4,'ceas health',200,9);
insert into produs values(5,'ananas',6,12);
insert into produs values(6,'sunca',6,3);
insert into produs values(7,'salam',4,3);
insert into produs values(8,'parizer',3,3);
insert into produs values(9,'zgarda',12,12);
insert into produs values(10,'jucarie zgomot',5,12);
insert into produs values(11,'laptop asus',3000,10);
insert into produs values(12,'paine',1,8);
insert into produs values(13,'covrig',2,8);
insert into produs values(14,'pulpe pui',7,7);
insert into produs values(15,'spate porc',7,7);
insert into produs values(16,'pliculete Felix',15,5);
insert into produs values(17,'lapte Mill',7,4);
insert into produs values(18,'Danonino',5,4);
insert into produs values(19,'blender',100,1);
insert into produs values(19,'cereale lapte',10,2);
insert into produs values(20,'clatite cioco',15,2);
insert into produs values(21,'piure cu pui',25,2);
insert into produs values(22,'somon cu orez',30,2);
insert into produs values(23,'creveti',40,2);

select * from produs;

--stoc
--stoc id, mag id, cantitate
insert into stoc values(1,13, 2);
insert into stoc values(2,12, 4);
insert into stoc values(3,11,10);
insert into stoc values(4,1,4);
insert into stoc values(5,2,20);
insert into stoc values(6,3,4);
insert into stoc values(7,4,6);
insert into stoc values(8,9,1);
insert into stoc values(9,8,1);
insert into stoc values(10,7,2);
insert into stoc values(11,6,3);
insert into stoc values(12,5,40);
insert into stoc values(13,13,60);
insert into stoc values(14,12,3);
insert into stoc values(15,1,3);
insert into stoc values(16,7,7);
insert into stoc values(17,7,8);
insert into stoc values(18,2,4);
insert into stoc values(19,3,3);

select * from stoc;



--clienti

--client id, prenume, nume, tel, email, locatie_id

insert into clienti values(1,'Alexandru','Larel','0745654687','alex.la@gmail.com',1);
insert into clienti values(2,'Ionel','Serban','0445676545','ionel.serban@yahoo.com',2);
insert into clienti values(3,'Petru','Meabefir','0453453232','meamea@yahoo.com',3);
insert into clienti values(4,'Gigel','Constantinov','0987654356','gigel_gigel@yahoo.com',4);
insert into clienti values(5,'Alexandra','Paul','0987654657','alex_alex@yahoo.com',5);
insert into clienti values(6,'Carmen','Sasha','0876543455','carmen_sasha@yahoo.com',6);
insert into clienti values(7,'Iolanda','Alexandrescu','0687534665','iolanda_io@yahoo.com',7);
insert into clienti values(8,'Simbad','Mufasa','7584658465','simba_ad.gmail.com',8);
insert into clienti values(9,'Yumina','Shitori','4853456198','yumina_shitori@yahoo.com',1);

insert into clienti values(10,'Ilias','Ionie','2342345467','ilias.ionei@gmail.com',1);

insert into clienti values(12,'Chim','Chimchim','0987656765','chimchimie@gmail.com',2);

insert into clienti values(11,'Mihai','Gabriel','0987634565','mihaita@gmail.com',1);

select * from clienti;

--comanda

--comanda id, comanda data, tip plata, status, locatie id, curier id, mag id, client id


insert into comanda values(1,to_date('22/03/2022', 'DD/MM/YYYY'),'card','finalizat',1,1,1,5);  --restaurant             --

insert into comanda values(2,to_date('01/05/2022', 'DD/MM/YYYY'),'card','finalizat',2,1,8,2); --macelarie

insert into comanda values(3,to_date('30/03/2022', 'DD/MM/YYYY'),'cash','finalizat',3,4,7,3); --restaurent

insert into comanda values(4,to_date('13/12/2022', 'DD/MM/YYYY'),'card','finalizat',2,1,10,3); --mag alim

insert into comanda values(5,to_date('02/09/2022', 'DD/MM/YYYY'),'cash','in progres',4,3,13,4); --rest

insert into comanda values(6,to_date('12/11/2022', 'DD/MM/YYYY'),'card','finalizat',7,3,6,4);   --pet shop

insert into comanda values(7,to_date('22/11/2022', 'DD/MM/YYYY'),'card','in progres',8,3,6,4); --pet shop

insert into comanda values(8,to_date('04/09/2022', 'DD/MM/YYYY'),'card','finalizat',5,6,4,5); --macelarie

insert into comanda values(9,to_date('06/07/2022', 'DD/MM/YYYY'),'card','finalizat',1,null,3,10); --piata

insert into comanda values(10,to_date('29/04/2022', 'DD/MM/YYYY'),'cash','finalizat',3,3,7,3); --rest

insert into comanda values(11,to_date('10/08/2022', 'DD/MM/YYYY'),'cash','finalizat',4,3,13,4); --rest

insert into comanda values(12,to_date('10/10/2022', 'DD/MM/YYYY'),'cash','in progres',7,1,5,7);  --florarie

insert into comanda values(13,to_date('02/02/2022', 'DD/MM/YYYY'),'card','finalizat',6,8,14,6);  --electro


select * from comanda;
```

```
--continut_comanda
--comanda_id,produs_id, cantitate
insert into continut_comanda values(1,20,2);
insert into continut_comanda values(2,14,10);
insert into continut_comanda values(3,21,3);
insert into continut_comanda values(4,17,12);
insert into continut_comanda values(5,23,2);
insert into continut_comanda values(6,16,10);
insert into continut_comanda values(7,9,3);
insert into continut_comanda values(8,15,3);
insert into continut_comanda values(9,5,30);
insert into continut_comanda values(10,23,4);
insert into continut_comanda values(11,22,1);
insert into continut_comanda values(12,2,50);
insert into continut_comanda values(13,19,4);

select * from continut_comanda;
rollback;
--feedback
  --feedback_id, comanda_id, rating_mag, rating_curier


insert into feedback_comanda values(1,13,3,4);
insert into feedback_comanda values(2,12,4,5);
insert into feedback_comanda values(3,11,5,5);
insert into feedback_comanda values(4,10,3,4);
insert into feedback_comanda values(5,9,2,2);
insert into feedback_comanda values(6,8,5,5);
insert into feedback_comanda values(7,7,3,3);
insert into feedback_comanda values(8,6,2,3);

insert into feedback_comanda values(9,5,4,4);

insert into feedback_comanda values(10,4,3,4);

insert into feedback_comanda values(11,3,4,2);

insert into feedback_comanda values(12,2,5,1);

insert into feedback_comanda values(13,1,4,1);


select * from feedback_comanda;
```

oras



```
--inserare in tabele

--oras
insert into oras values(1, 'Bucuresti');
insert into oras values(2, 'Ploiesti');
insert into oras values(3, 'Brasov');
insert into oras values(4, 'Constanta');
insert into oras values(5, 'Galati');
insert into oras values(6, 'Suceava');
insert into oras values(7, 'Cluj');
insert into oras values(8, 'Oradea');
```

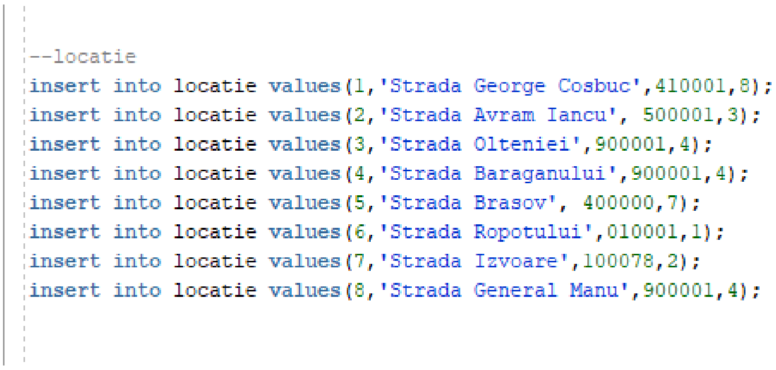


```
--oras
insert into oras values(1, 'Bucuresti');
insert into oras values(2, 'Ploiesti');
insert into oras values(3, 'Brasov');
insert into oras values(4, 'Constanta');
insert into oras values(5, 'Galati');
insert into oras values(6, 'Suceava');
insert into oras values(7, 'Cluj');
insert into oras values(8, 'Oradea');

select * from oras;
```

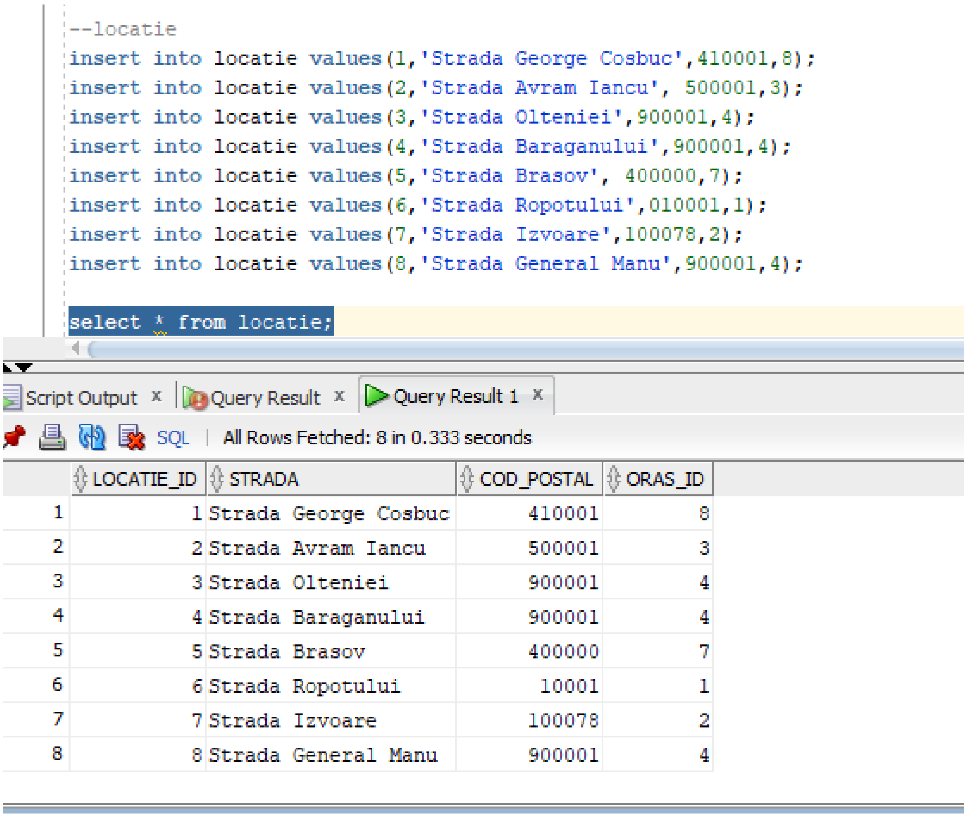
All Rows Fetched: 8 in 0.192 seconds

	ORAS_ID	NUME_ORAS
1	1	Bucuresti
2	2	Ploiesti
3	3	Brasov
4	4	Constanta
5	5	Galati
6	6	Suceava
7	7	Cluj
8	8	Oradea

locatie



```
--locatie
insert into locatie values(1,'Strada George Cosbuc',410001,8);
insert into locatie values(2,'Strada Avram Iancu', 500001,3);
insert into locatie values(3,'Strada Olteniei',900001,4);
insert into locatie values(4,'Strada Baraganului',900001,4);
insert into locatie values(5,'Strada Brasov', 400000,7);
insert into locatie values(6,'Strada Ropotului',010001,1);
insert into locatie values(7,'Strada Izvoare',100078,2);
insert into locatie values(8,'Strada General Manu',900001,4);
```

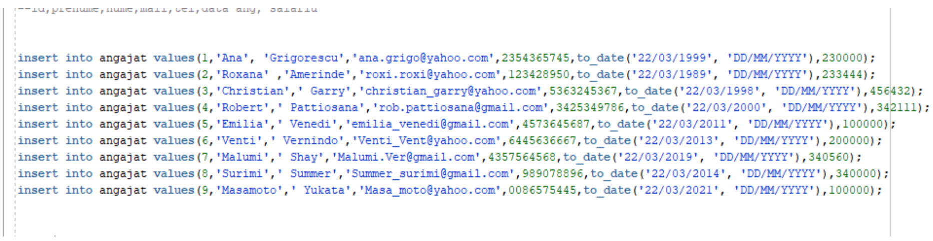


```
--locatie
insert into locatie values(1,'Strada George Cosbuc',410001,8);
insert into locatie values(2,'Strada Avram Iancu', 500001,3);
insert into locatie values(3,'Strada Olteniei',900001,4);
insert into locatie values(4,'Strada Baraganului',900001,4);
insert into locatie values(5,'Strada Brasov', 400000,7);
insert into locatie values(6,'Strada Ropotului',010001,1);
insert into locatie values(7,'Strada Izvoare',100078,2);
insert into locatie values(8,'Strada General Manu',900001,4);

select * from locatie;
```

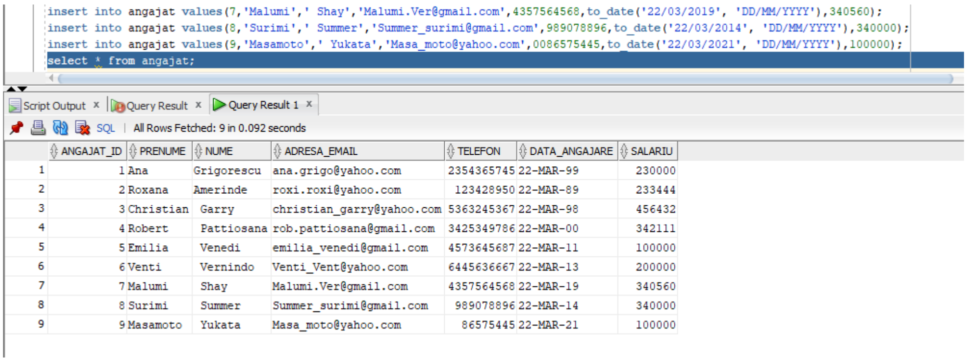
All Rows Fetched: 8 in 0.333 seconds

	LOCATIE_ID	STRADA	COD_POSTAL	ORAS_ID
1	1	Strada George Cosbuc	410001	8
2	2	Strada Avram Iancu	500001	3
3	3	Strada Olteniei	900001	4
4	4	Strada Baraganului	900001	4
5	5	Strada Brasov	400000	7
6	6	Strada Ropotului	10001	1
7	7	Strada Izvoare	100078	2
8	8	Strada General Manu	900001	4

angajat



```
--id,prenume,nume,mail,tel,data ang, salariu


insert into angajat values(1,'Ana', 'Grigorescu','ana.grigo@yahoo.com',2354365745,to_date('22/03/1999', 'DD/MM/YYYY'),230000);
insert into angajat values(2,'Roxana' ,'Amerinde','roxi.roxi@yahoo.com',123428950,to_date('22/03/1989', 'DD/MM/YYYY'),233444);
insert into angajat values(3,'Christian',' Garry','christian_garry@yahoo.com',5363245367,to_date('22/03/1998', 'DD/MM/YYYY'),456432);
insert into angajat values(4,'Robert',' Pattiosana','rob.pattiosana@gmail.com',3425349786,to_date('22/03/2000', 'DD/MM/YYYY'),342111);
insert into angajat values(5,'Emilia',' Venedi','emilia_venedi@gmail.com',4573645687,to_date('22/03/2011', 'DD/MM/YYYY'),100000);
insert into angajat values(6,'Venti',' Vernindo','Venti_Vent@yahoo.com',6445636667,to_date('22/03/2013', 'DD/MM/YYYY'),200000);
insert into angajat values(7,'Malumi',' Shay','Malumi.Ver@gmail.com',4357564568,to_date('22/03/2019', 'DD/MM/YYYY'),340560);
insert into angajat values(8,'Surimi',' Summer','Summer_surimi@gmail.com',989078896,to_date('22/03/2014', 'DD/MM/YYYY'),340000);
insert into angajat values(9,'Masamoto',' Yukata','Masa_moto@yahoo.com',0086575445,to_date('22/03/2021', 'DD/MM/YYYY'),100000);
```

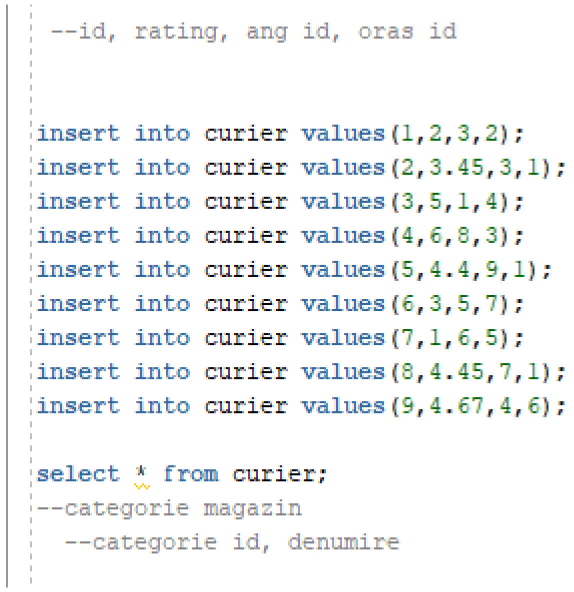


```
insert into angajat values(7,'Malumi',' Shay','Malumi.Ver@gmail.com',4357564568,to_date('22/03/2019', 'DD/MM/YYYY'),340560);
insert into angajat values(8,'Surimi',' Summer','Summer_surimi@gmail.com',989078896,to_date('22/03/2014', 'DD/MM/YYYY'),340000);
insert into angajat values(9,'Masamoto',' Yukata','Masa_moto@yahoo.com',0086575445,to_date('22/03/2021', 'DD/MM/YYYY'),100000);
select * from angajat;
```

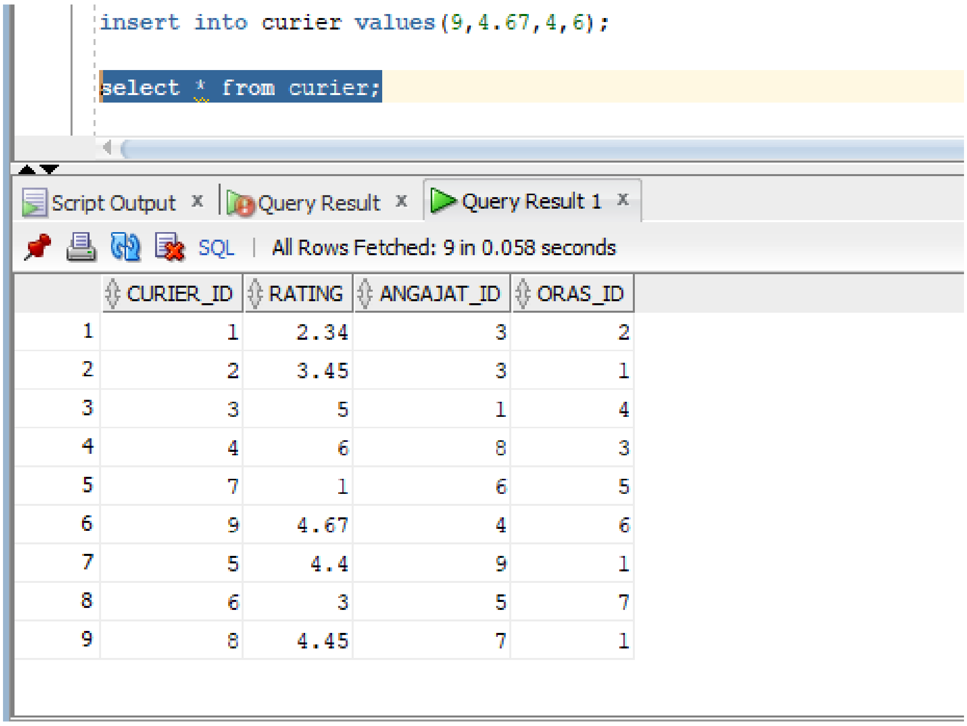
All Rows Fetched: 9 in 0.092 seconds

	ANGAJAT_ID	PRENUME	NUME	ADRESA_EMAIL	TELEFON	DATA_ANGAJARE	SALARIU
1	1	Ana	Grigorescu	ana.grigo@yahoo.com	2354365745	22-MAR-99	230000
2	2	Roxana	Amerinde	roxi.roxi@yahoo.com	123428950	22-MAR-89	233444
3	3	Christian	Garry	christian_garry@yahoo.com	5363245367	22-MAR-98	456432
4	4	Robert	Pattiosana	rob.pattiosana@gmail.com	3425349786	22-MAR-00	342111
5	5	Emilia	Venedi	emilia_venedi@gmail.com	4573645687	22-MAR-11	100000
6	6	Venti	Vernindo	Venti_Vent@yahoo.com	6445636667	22-MAR-13	200000
7	7	Malumi	Shay	Malumi.Ver@gmail.com	4357564568	22-MAR-19	340560
8	8	Surimi	Summer	Summer_surimi@gmail.com	989078896	22-MAR-14	340000
9	9	Masamoto	Yukata	Masa_moto@yahoo.com	86575445	22-MAR-21	100000

curier



```
 --id, rating, ang id, oras id


insert into curier values(1,2,3,2);
insert into curier values(2,3.45,3,1);
insert into curier values(3,5,1,4);
insert into curier values(4,6,8,3);
insert into curier values(5,4.4,9,1);
insert into curier values(6,3,5,7);
insert into curier values(7,1,6,5);
insert into curier values(8,4.45,7,1);
insert into curier values(9,4.67,4,6);

select * from curier;
--categorie magazin
  --categorie id, denumire
```

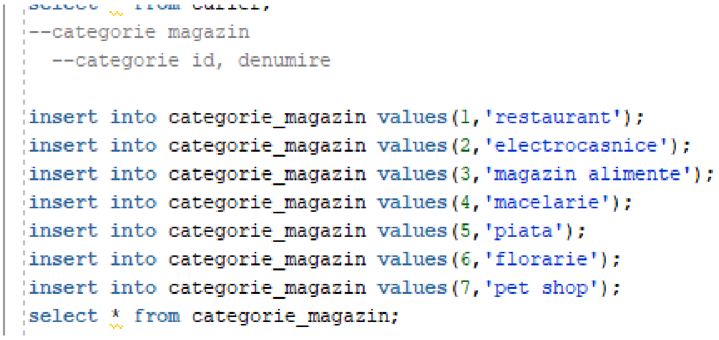


```
insert into curier values(9,4.67,4,6);

select * from curier;
```

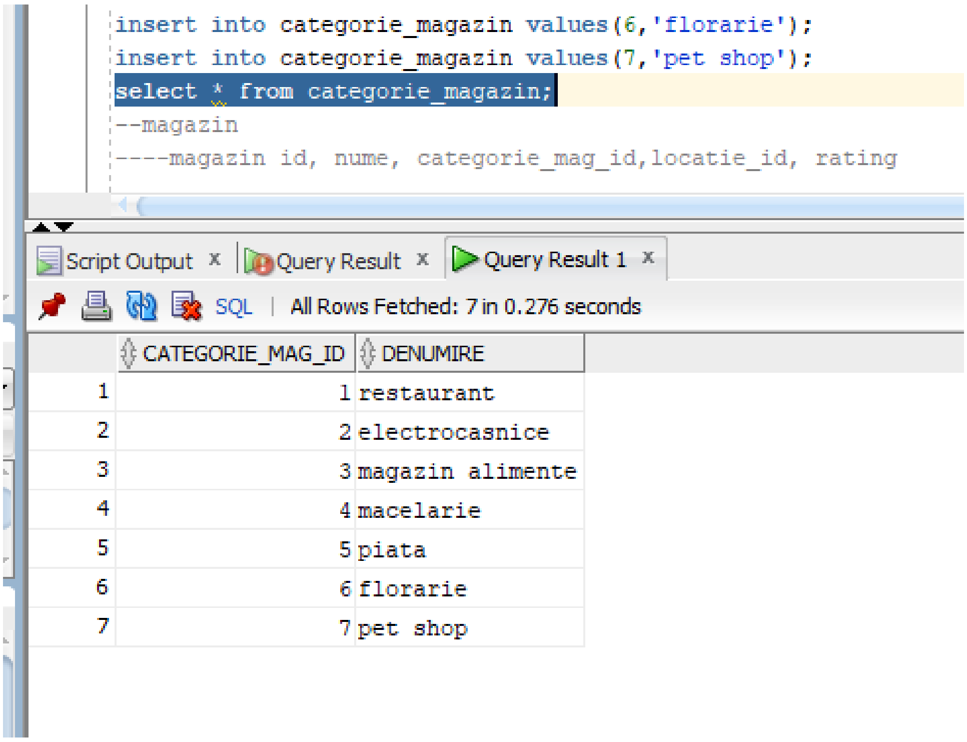
All Rows Fetched: 9 in 0.058 seconds

	CURIER_ID	RATING	ANGAJAT_ID	ORAS_ID
1	1	2.34	3	2
2	2	3.45	3	1
3	3	5	1	4
4	4	6	8	3
5	7	1	6	5
6	9	4.67	4	6
7	5	4.4	9	1
8	6	3	5	7
9	8	4.45	7	1

Categorie_magazin



```
select * from curier;
--categorie magazin
  --categorie id, denumire

insert into categorie_magazin values(1,'restaurant');
insert into categorie_magazin values(2,'electrocasnice');
insert into categorie_magazin values(3,'magazin alimente');
insert into categorie_magazin values(4,'macelarie');
insert into categorie_magazin values(5,'piata');
insert into categorie_magazin values(6,'florarie');
insert into categorie_magazin values(7,'pet shop');
select * from categorie_magazin;
```

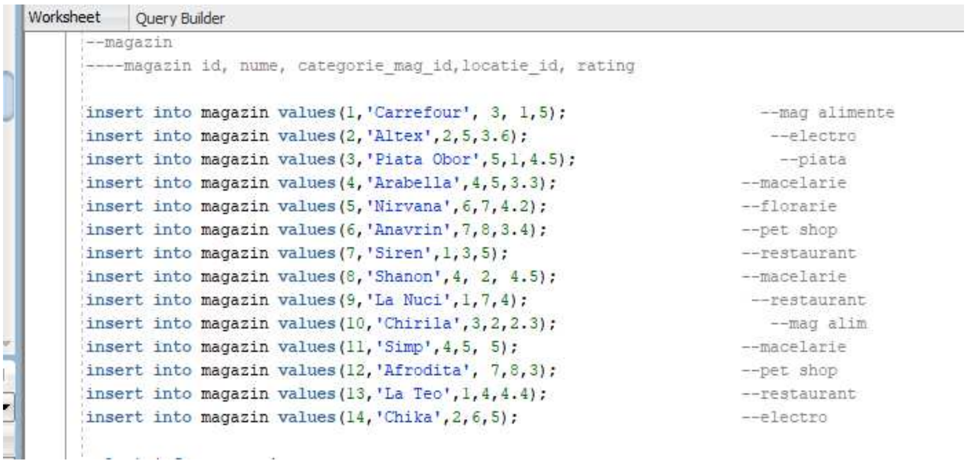


```
insert into categorie_magazin values(6,'florarie');
insert into categorie_magazin values(7,'pet shop');
select * from categorie_magazin;
--magazin
----magazin id, nume, categorie_mag_id,locatie_id, rating
```

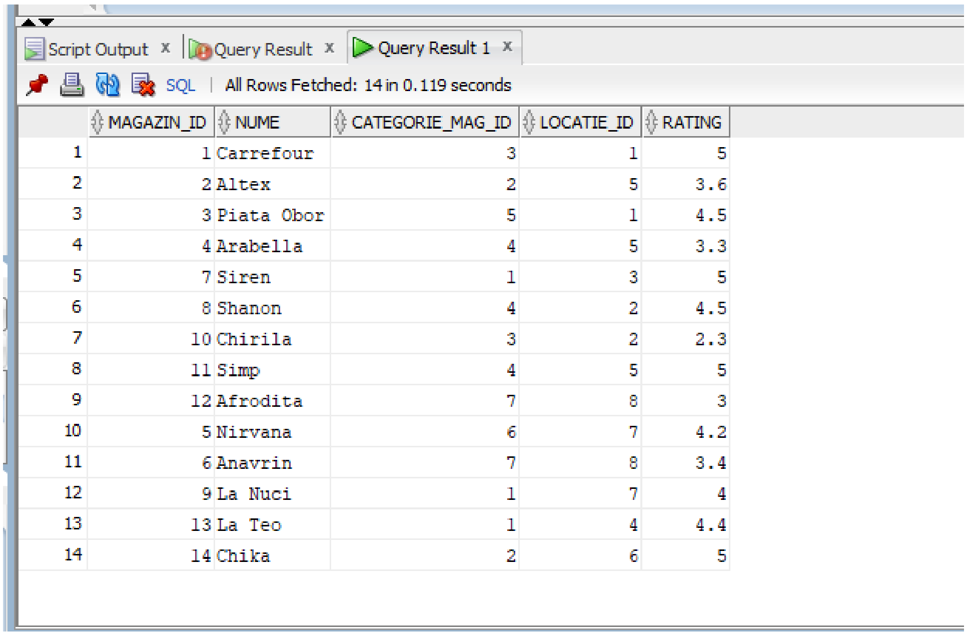
All Rows Fetched: 7 in 0.276 seconds

	CATEGORIE_MAG_ID	DENUMIRE
1	1	restaurant
2	2	electrocasnice
3	3	magazin alimente
4	4	macelarie
5	5	piata
6	6	florarie
7	7	pet shop

magazin



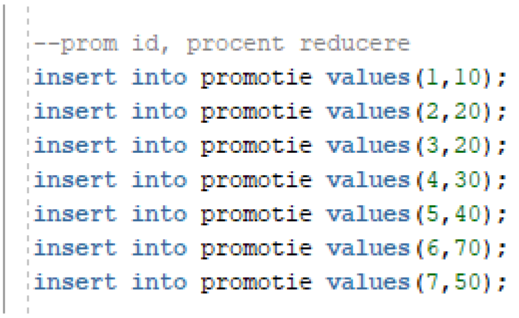
```
--magazin
----magazin id, nume, categorie_mag_id,locatie_id, rating

insert into magazin values(1,'Carrefour', 3, 1,5);                 --mag alimente
insert into magazin values(2,'Altex',2,5,3.6);                      --electro
insert into magazin values(3,'Piata Obor',5,1,4.5);                  --piata
insert into magazin values(4,'Arabella',4,5,3.3);                 --macelarie
insert into magazin values(5,'Nirvana',6,7,4.2);                  --florarie
insert into magazin values(6,'Anavrin',7,8,3.4);                  --pet shop
insert into magazin values(7,'Siren',1,3,5);                      --restaurant
insert into magazin values(8,'Shanon',4, 2, 4.5);                 --macelarie
insert into magazin values(9,'La Nuci',1,7,4);                     --restaurant
insert into magazin values(10,'Chirila',3,2,2.3);                   --mag alim
insert into magazin values(11,'Simp',4,5, 5);                     --macelarie
insert into magazin values(12,'Afrodita', 7,8,3);                 --pet shop
insert into magazin values(13,'La Teo',1,4,4.4);                  --restaurant
insert into magazin values(14,'Chika',2,6,5);                     --electro
```



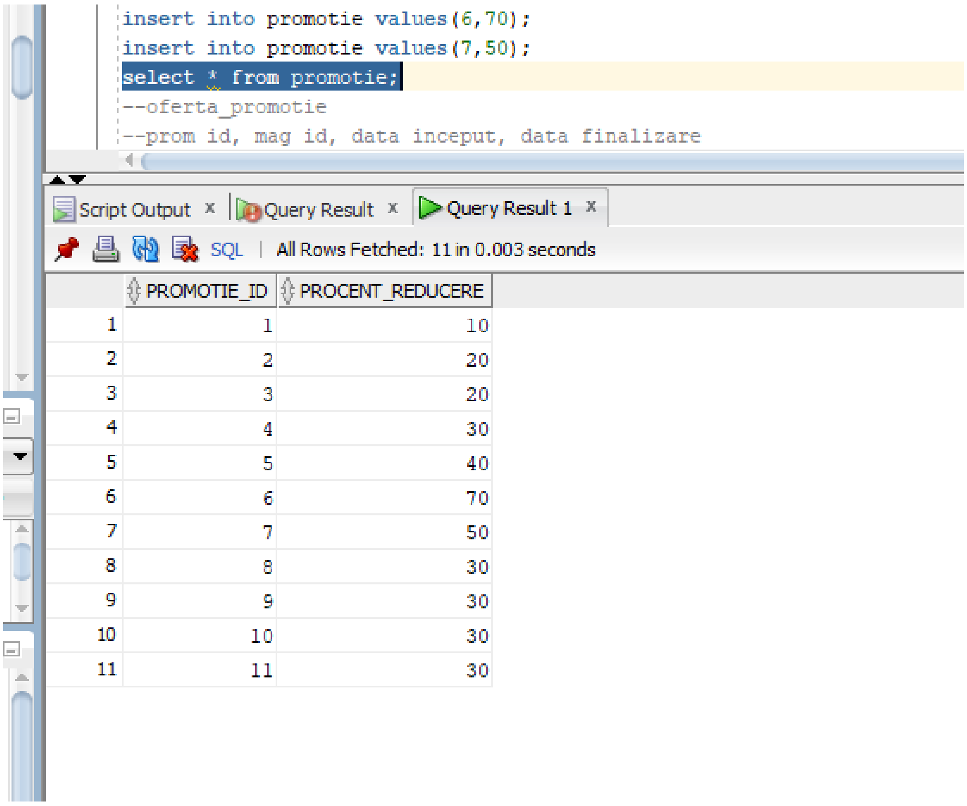
All Rows Fetched: 14 in 0.119 seconds

	MAGAZIN_ID	NUME	CATEGORIE_MAG_ID	LOCATIE_ID	RATING
1	1	Carrefour	3	1	5
2	2	Altex	2	5	3.6
3	3	Piata Obor	5	1	4.5
4	4	Arabella	4	5	3.3
5	7	Siren	1	3	5
6	8	Shanon	4	2	4.5
7	10	Chirila	3	2	2.3
8	11	Simp	4	5	5
9	12	Afrodita	7	8	3
10	5	Nirvana	6	7	4.2
11	6	Anavrin	7	8	3.4
12	9	La Nuci	1	7	4
13	13	La Teo	1	4	4.4
14	14	Chika	2	6	5

promotie



```
--prom id, procent reducere
insert into promotie values(1,10);
insert into promotie values(2,20);
insert into promotie values(3,20);
insert into promotie values(4,30);
insert into promotie values(5,40);
insert into promotie values(6,70);
insert into promotie values(7,50);
```

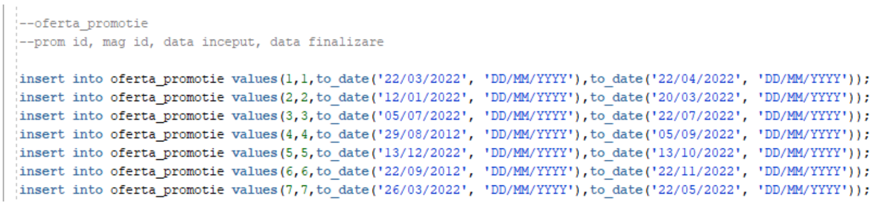


```
insert into promotie values(6,70);
insert into promotie values(7,50);
select * from promotie;
--oferta_promotie
--prom id, mag id, data inceput, data finalizare
```

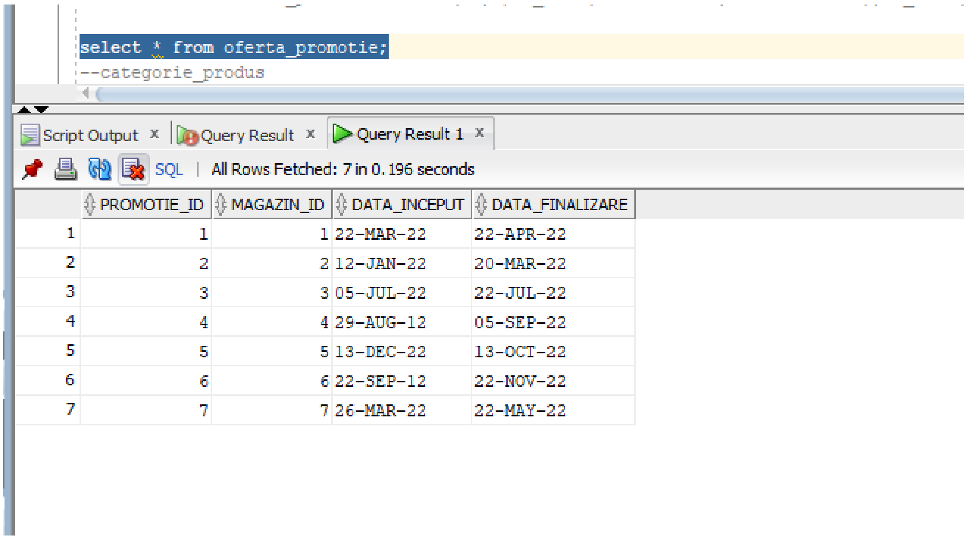
All Rows Fetched: 11 in 0.003 seconds

	PROMOTIE_ID	PROCENT_REDUCERE
1	1	10
2	2	20
3	3	20
4	4	30
5	5	40
6	6	70
7	7	50
8	8	30
9	9	30
10	10	30
11	11	30

Oferta_promotie



```
--oferta_promotie
--prom id, mag id, data inceput, data finalizare

insert into oferta_promotie values(1,1,to_date('22/03/2022', 'DD/MM/YYYY'),to_date('22/04/2022', 'DD/MM/YYYY'));
insert into oferta_promotie values(2,2,to_date('12/01/2022', 'DD/MM/YYYY'),to_date('20/03/2022', 'DD/MM/YYYY'));
insert into oferta_promotie values(3,3,to_date('05/07/2022', 'DD/MM/YYYY'),to_date('22/07/2022', 'DD/MM/YYYY'));
insert into oferta_promotie values(4,4,to_date('29/08/2012', 'DD/MM/YYYY'),to_date('05/09/2022', 'DD/MM/YYYY'));
insert into oferta_promotie values(5,5,to_date('13/12/2022', 'DD/MM/YYYY'),to_date('13/10/2022', 'DD/MM/YYYY'));
insert into oferta_promotie values(6,6,to_date('22/09/2012', 'DD/MM/YYYY'),to_date('22/11/2022', 'DD/MM/YYYY'));
insert into oferta_promotie values(7,7,to_date('26/03/2022', 'DD/MM/YYYY'),to_date('22/05/2022', 'DD/MM/YYYY'));
```

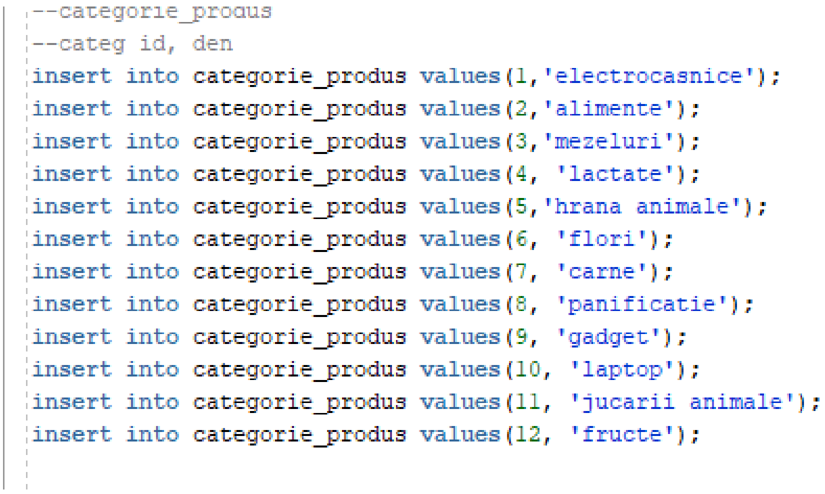


```
select * from oferta_promotie;
--categorie_produs
```

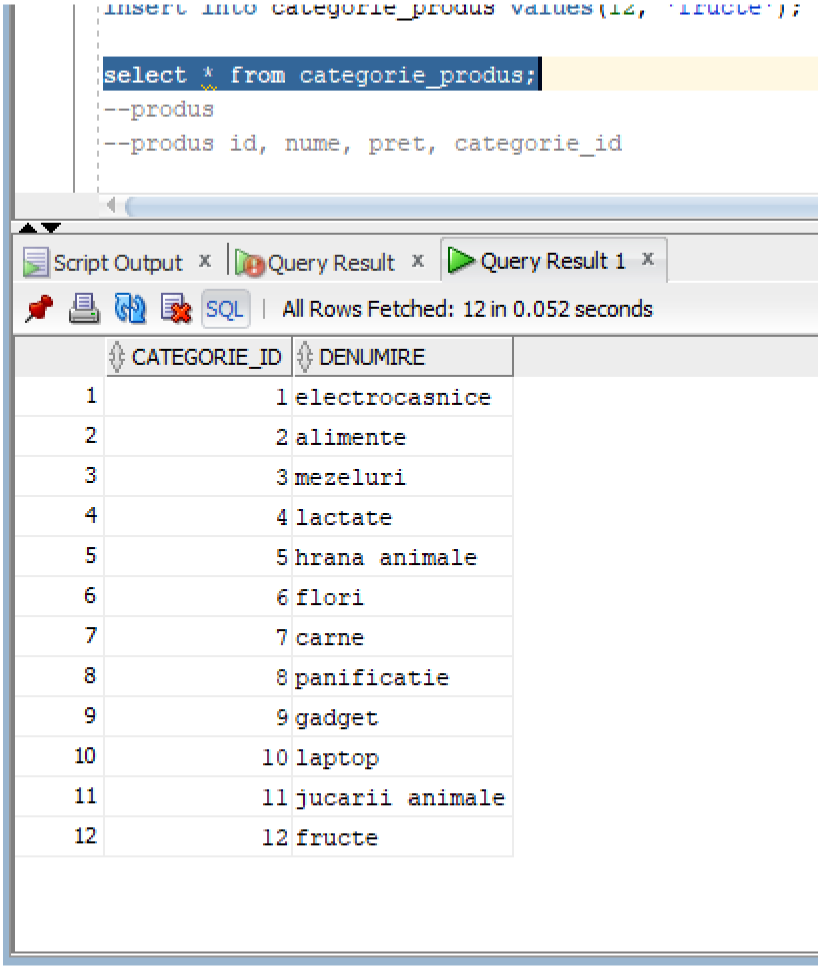
All Rows Fetched: 7 in 0.196 seconds

	PROMOTIE_ID	MAGAZIN_ID	DATA_INCEPUT	DATA_FINALIZARE
1	1	1	22-MAR-22	22-APR-22
2	2	2	12-JAN-22	20-MAR-22
3	3	3	05-JUL-22	22-JUL-22
4	4	4	29-AUG-12	05-SEP-22
5	5	5	13-DEC-22	13-OCT-22
6	6	6	22-SEP-12	22-NOV-22
7	7	7	26-MAR-22	22-MAY-22

Categorie_produs



```
--categorie_produs
--categ id, den
insert into categorie_produs values(1,'electrocasnice');
insert into categorie_produs values(2,'alimente');
insert into categorie_produs values(3,'mezeluri');
insert into categorie_produs values(4, 'lactate');
insert into categorie_produs values(5,'hrana animale');
insert into categorie_produs values(6, 'flori');
insert into categorie_produs values(7, 'carne');
insert into categorie_produs values(8, 'panificatie');
insert into categorie_produs values(9, 'gadget');
insert into categorie_produs values(10, 'laptop');
insert into categorie_produs values(11, 'jucarii animale');
insert into categorie_produs values(12, 'fructe');
```

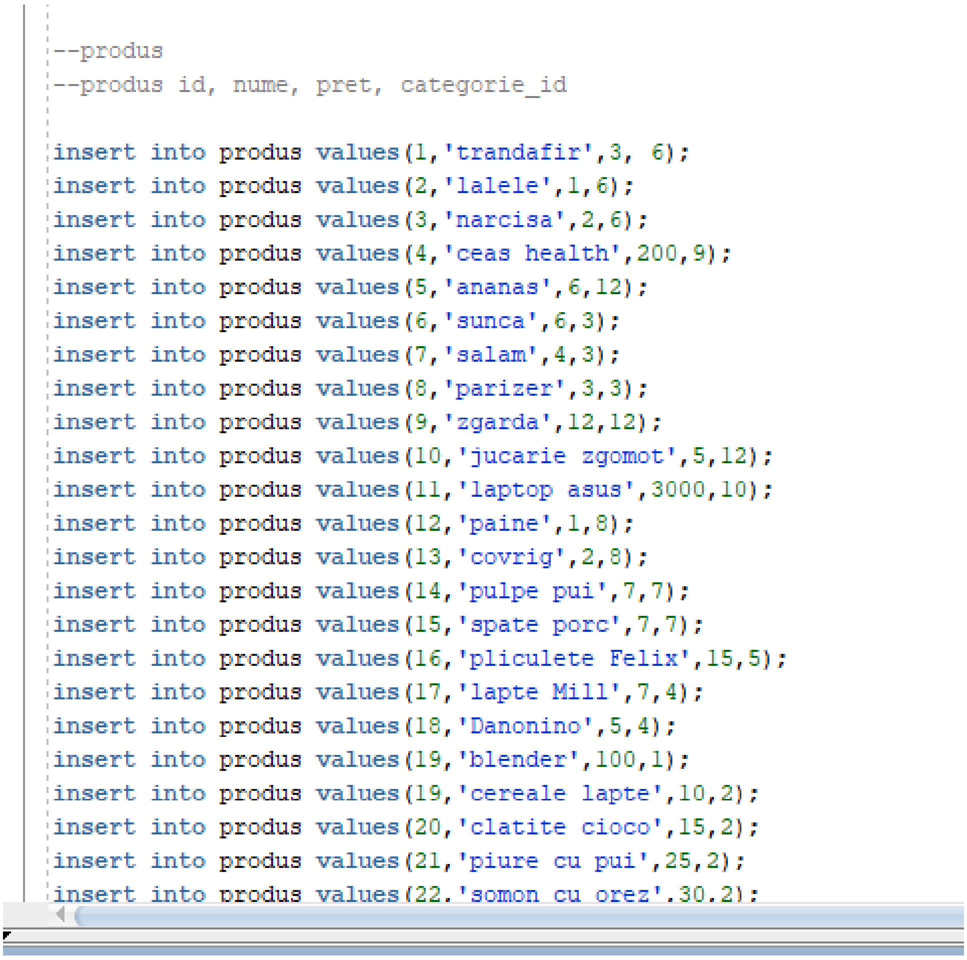


```
insert into categorie_produs values(12, 'fructe');

select * from categorie_produs;
--produs
--produs id, nume, pret, categorie_id
```

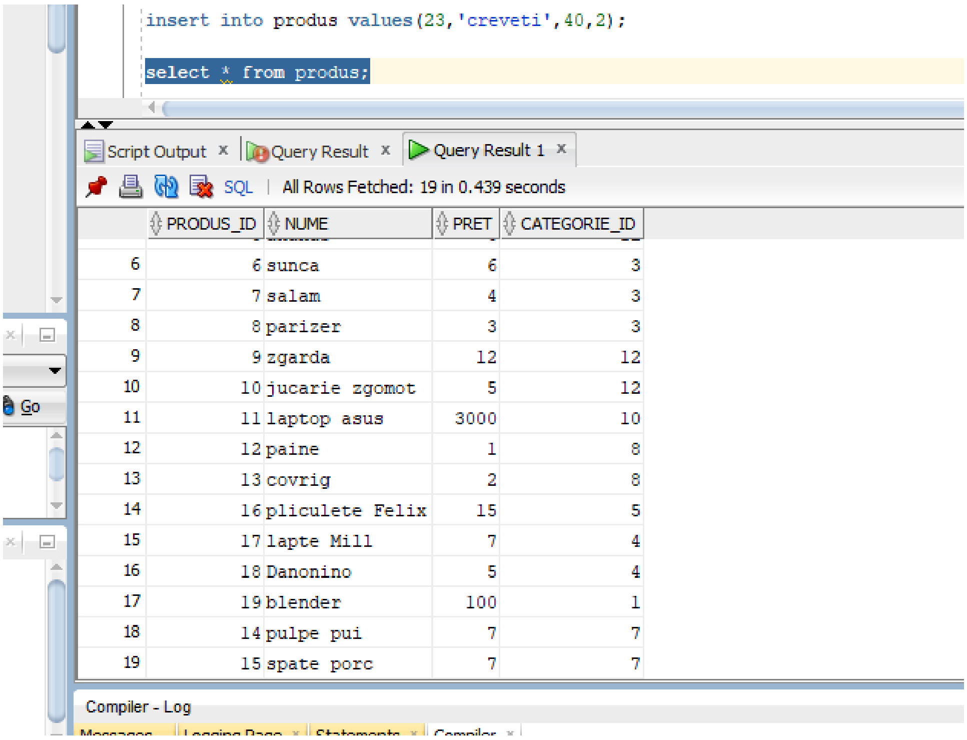
All Rows Fetched: 12 in 0.052 seconds

	CATEGORIE_ID	DENUMIRE
1	1	electrocasnice
2	2	alimente
3	3	mezeluri
4	4	lactate
5	5	hrana animale
6	6	flori
7	7	carne
8	8	panificatie
9	9	gadget
10	10	laptop
11	11	jucarii animale
12	12	fructe

produs



```
--produs
--produs id, nume, pret, categorie_id

insert into produs values(1,'trandafir',3, 6);
insert into produs values(2,'lalele',1,6);
insert into produs values(3,'narcisa',2,6);
insert into produs values(4,'ceas health',200,9);
insert into produs values(5,'ananas',6,12);
insert into produs values(6,'sunca',6,3);
insert into produs values(7,'salam',4,3);
insert into produs values(8,'parizer',3,3);
insert into produs values(9,'zgarda',12,12);
insert into produs values(10,'jucarie zgomot',5,12);
insert into produs values(11,'laptop asus',3000,10);
insert into produs values(12,'paine',1,8);
insert into produs values(13,'covrig',2,8);
insert into produs values(14,'pulpe pui',7,7);
insert into produs values(15,'spate porc',7,7);
insert into produs values(16,'pliculete Felix',15,5);
insert into produs values(17,'lapte Mill',7,4);
insert into produs values(18,'Danonino',5,4);
insert into produs values(19,'blender',100,1);
insert into produs values(19,'cereale lapte',10,2);
insert into produs values(20,'clatite cioco',15,2);
insert into produs values(21,'piure cu pui',25,2);
insert into produs values(22,'somon cu orez',30,2);
```

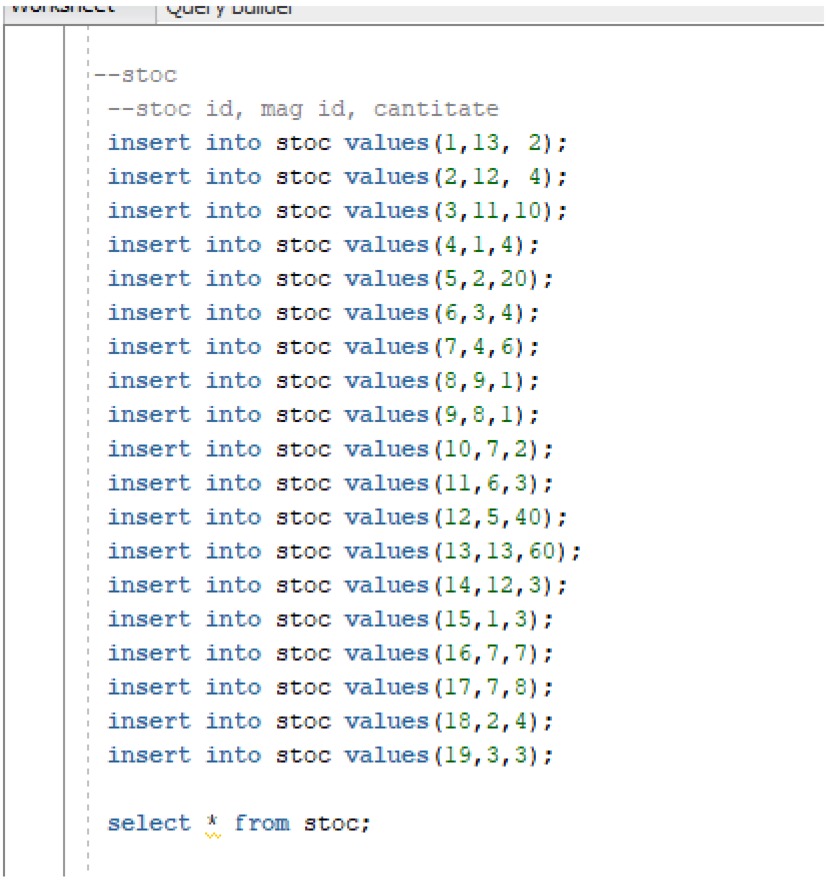


```
insert into produs values(23,'creveti',40,2);

select * from produs;
```

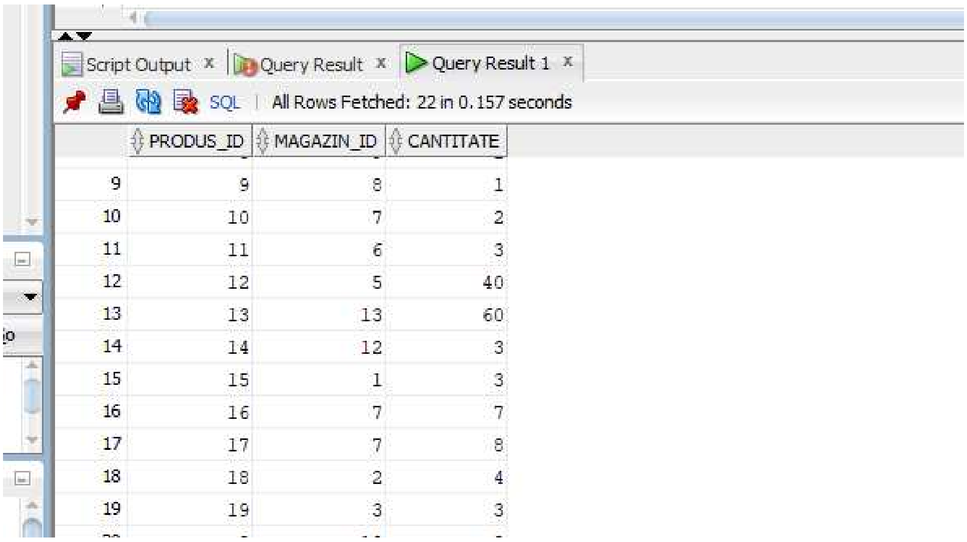
All Rows Fetched: 19 in 0.439 seconds

	PRODUS_ID	NUME	PRET	CATEGORIE_ID
6	6	sunca	6	3
7	7	salam	4	3
8	8	parizer	3	3
9	9	zgarda	12	12
10	10	jucarie zgomot	5	12
11	11	laptop asus	3000	10
12	12	paine	1	8
13	13	covrig	2	8
14	16	pliculete Felix	15	5
15	17	lapte Mill	7	4
16	18	Danonino	5	4
17	19	blender	100	1
18	14	pulpe pui	7	7
19	15	spate porc	7	7

Stoc



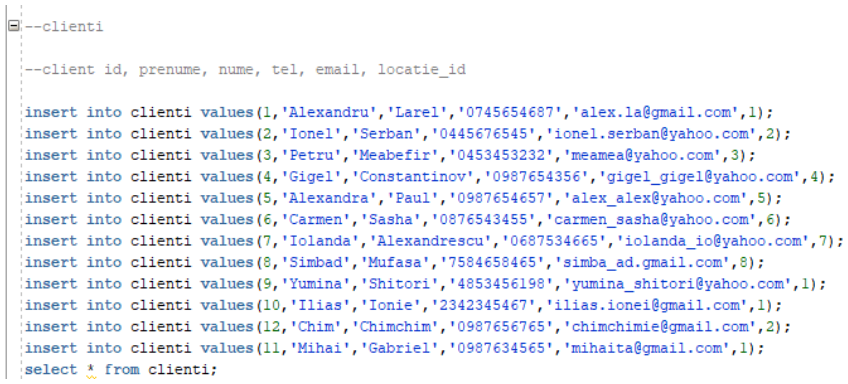
```
--stoc
 --stoc id, mag id, cantitate
 insert into stoc values(1,13, 2);
 insert into stoc values(2,12, 4);
 insert into stoc values(3,11,10);
 insert into stoc values(4,1,4);
 insert into stoc values(5,2,20);
 insert into stoc values(6,3,4);
 insert into stoc values(7,4,6);
 insert into stoc values(8,9,1);
 insert into stoc values(9,8,1);
 insert into stoc values(10,7,2);
 insert into stoc values(11,6,3);
 insert into stoc values(12,5,40);
 insert into stoc values(13,13,60);
 insert into stoc values(14,12,3);
 insert into stoc values(15,1,3);
 insert into stoc values(16,7,7);
 insert into stoc values(17,7,8);
 insert into stoc values(18,2,4);
 insert into stoc values(19,3,3);

 select * from stoc;
```



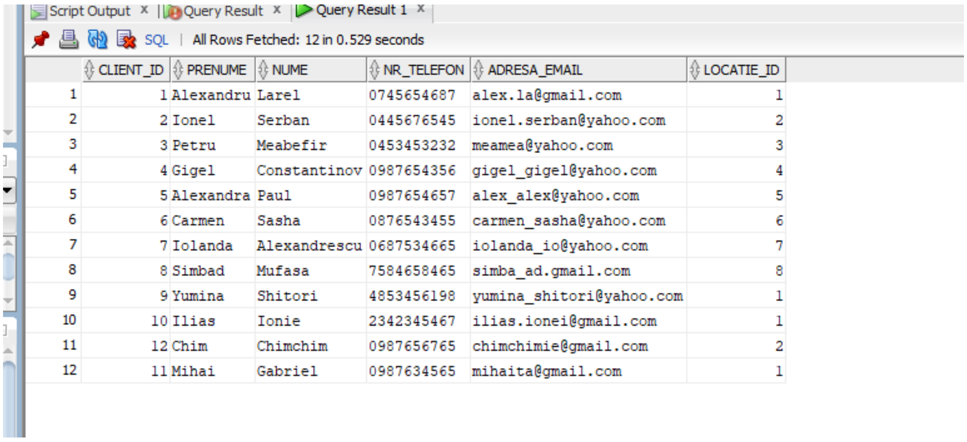
All Rows Fetched: 22 in 0.157 seconds

	PRODUS_ID	MAGAZIN_ID	CANTITATE
9	9	8	1
10	10	7	2
11	11	6	3
12	12	5	40
13	13	13	60
14	14	12	3
15	15	1	3
16	16	7	7
17	17	7	8
18	18	2	4
19	19	3	3

clienti



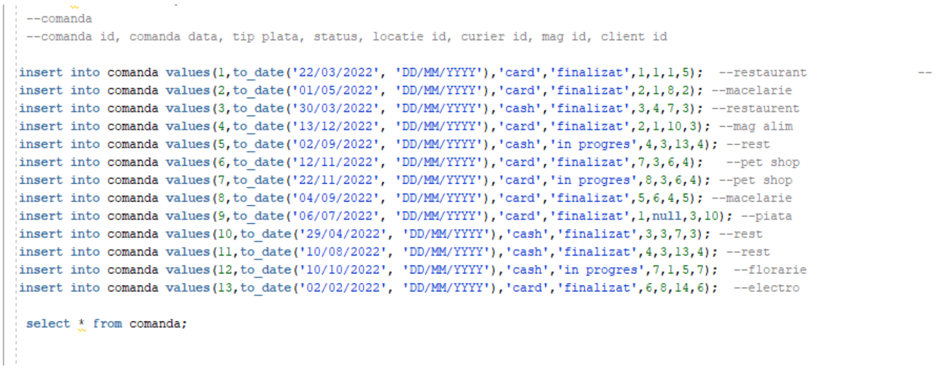
```
--clienti

--client id, prenume, nume, tel, email, locatie_id

insert into clienti values(1,'Alexandru','Larel','0745654687','alex.la@gmail.com',1);
insert into clienti values(2,'Ionel','Serban','0445676545','ionel.serban@yahoo.com',2);
insert into clienti values(3,'Petru','Meabefir','0453453232','meamea@yahoo.com',3);
insert into clienti values(4,'Gigel','Constantinov','0987654356','gigel_gigel@yahoo.com',4);
insert into clienti values(5,'Alexandra','Paul','0987654657','alex_alex@yahoo.com',5);
insert into clienti values(6,'Carmen','Sasha','0876543455','carmen_sasha@yahoo.com',6);
insert into clienti values(7,'Iolanda','Alexandrescu','0687534665','iolanda_io@yahoo.com',7);
insert into clienti values(8,'Simbad','Mufasa','7584658465','simba_ad.gmail.com',8);
insert into clienti values(9,'Yumina','Shitori','4853456198','yumina_shitori@yahoo.com',1);
insert into clienti values(10,'Ilias','Ionie','2342345467','ilias.ionei@gmail.com',1);
insert into clienti values(12,'Chim','Chimchim','0987656765','chimchimie@gmail.com',2);
insert into clienti values(11,'Mihai','Gabriel','0987634565','mihaita@gmail.com',1);
select * from clienti;
```



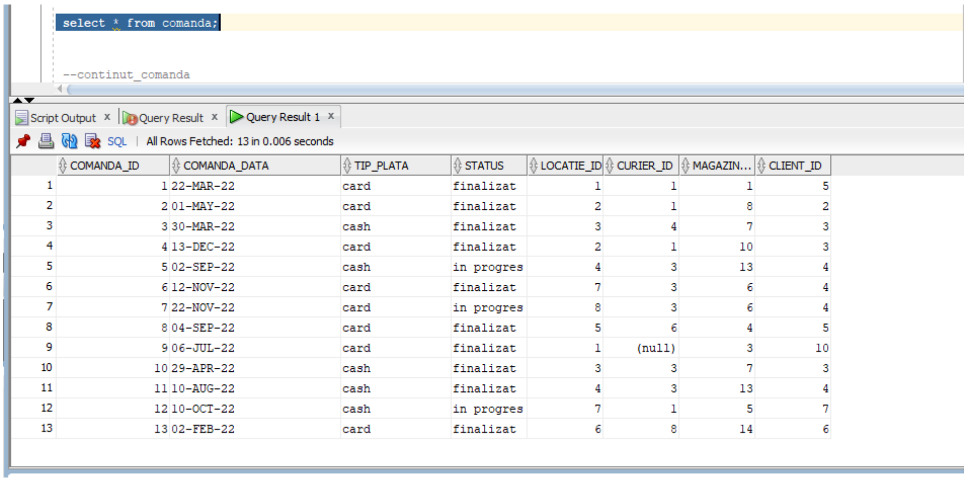
All Rows Fetched: 12 in 0.529 seconds

	CLIENT_ID	PRENUME	NUME	NR_TELEFON	ADRESA_EMAIL	LOCATIE_ID
1	1	Alexandru	Larel	0745654687	alex.la@gmail.com	1
2	2	Ionel	Serban	0445676545	ionel.serban@yahoo.com	2
3	3	Petru	Meabefir	0453453232	meamea@yahoo.com	3
4	4	Gigel	Constantinov	0987654356	gigel_gigel@yahoo.com	4
5	5	Alexandra	Paul	0987654657	alex_alex@yahoo.com	5
6	6	Carmen	Sasha	0876543455	carmen_sasha@yahoo.com	6
7	7	Iolanda	Alexandrescu	0687534665	iolanda_io@yahoo.com	7
8	8	Simbad	Mufasa	7584658465	simba_ad.gmail.com	8
9	9	Yumina	Shitori	4853456198	yumina_shitori@yahoo.com	1
10	10	Ilias	Ionie	2342345467	ilias.ionei@gmail.com	1
11	12	Chim	Chimchim	0987656765	chimchimie@gmail.com	2
12	11	Mihai	Gabriel	0987634565	mihaita@gmail.com	1

comanda



```
 --comanda
 --comanda id, comanda data, tip plata, status, locatie id, curier id, mag id, client id

insert into comanda values(1,to_date('22/03/2022', 'DD/MM/YYYY'),'card','finalizat',1,1,1,5);  --restaurant            --
insert into comanda values(2,to_date('01/05/2022', 'DD/MM/YYYY'),'card','finalizat',2,1,8,2); --macelarie
insert into comanda values(3,to_date('30/03/2022', 'DD/MM/YYYY'),'cash','finalizat',3,4,7,3); --restaurent
insert into comanda values(4,to_date('13/12/2022', 'DD/MM/YYYY'),'card','finalizat',2,1,10,3); --mag alim
insert into comanda values(5,to_date('02/09/2022', 'DD/MM/YYYY'),'cash','in progres',4,3,13,4); --rest
insert into comanda values(6,to_date('12/11/2022', 'DD/MM/YYYY'),'card','finalizat',7,3,6,4);   --pet shop
insert into comanda values(7,to_date('22/11/2022', 'DD/MM/YYYY'),'card','in progres',8,3,6,4); --pet shop
insert into comanda values(8,to_date('04/09/2022', 'DD/MM/YYYY'),'card','finalizat',5,6,4,5); --macelarie
insert into comanda values(9,to_date('06/07/2022', 'DD/MM/YYYY'),'card','finalizat',1,null,3,10); --piata
insert into comanda values(10,to_date('29/04/2022', 'DD/MM/YYYY'),'cash','finalizat',3,3,7,3); --rest
insert into comanda values(11,to_date('10/08/2022', 'DD/MM/YYYY'),'cash','finalizat',4,3,13,4); --rest
insert into comanda values(12,to_date('10/10/2022', 'DD/MM/YYYY'),'cash','in progres',7,1,5,7);  --florarie
insert into comanda values(13,to_date('02/02/2022', 'DD/MM/YYYY'),'card','finalizat',6,8,14,6);  --electro

 select * from comanda;
```

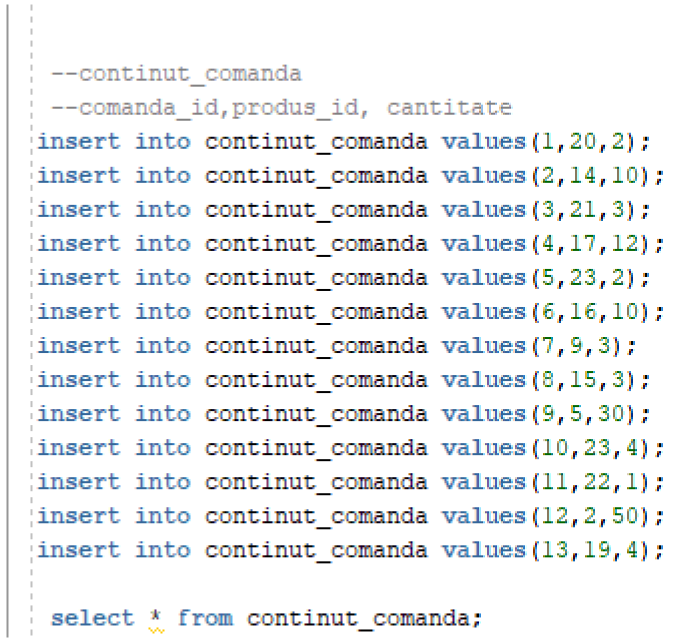


```
select * from comanda;


--continut_comanda
```

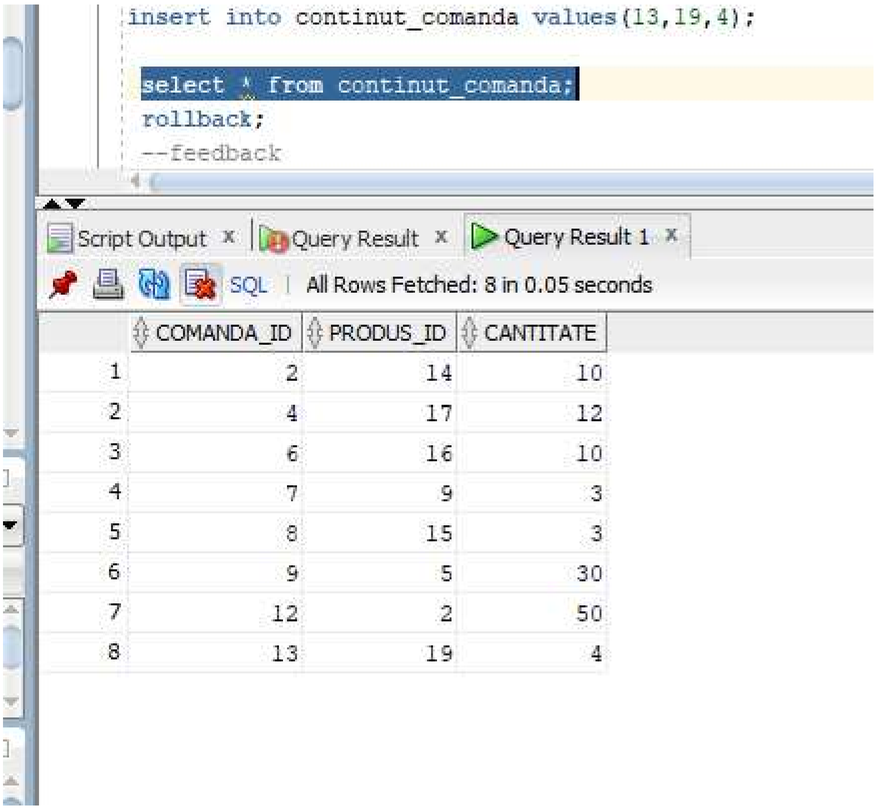
All Rows Fetched: 13 in 0.006 seconds

	COMANDA_ID	COMANDA_DATA	TIP_PLATA	STATUS	LOCATIE_ID	CURIER_ID	MAGAZIN...	CLIENT_ID
1	1	22-MAR-22	card	finalizat	1	1	1	5
2	2	01-MAY-22	card	finalizat	2	1	8	2
3	3	30-MAR-22	cash	finalizat	3	4	7	3
4	4	13-DEC-22	card	finalizat	2	1	10	3
5	5	02-SEP-22	cash	in progres	4	3	13	4
6	6	12-NOV-22	card	finalizat	7	3	6	4
7	7	22-NOV-22	card	in progres	8	3	6	4
8	8	04-SEP-22	card	finalizat	5	6	4	5
9	9	06-JUL-22	card	finalizat	1	(null)	3	10
10	10	29-APR-22	cash	finalizat	3	3	7	3
11	11	10-AUG-22	cash	finalizat	4	3	13	4
12	12	10-OCT-22	cash	in progres	7	1	5	7
13	13	02-FEB-22	card	finalizat	6	8	14	6

Continut_comanda



```
 --continut_comanda
 --comanda_id,produs_id, cantitate
insert into continut_comanda values(1,20,2);
insert into continut_comanda values(2,14,10);
insert into continut_comanda values(3,21,3);
insert into continut_comanda values(4,17,12);
insert into continut_comanda values(5,23,2);
insert into continut_comanda values(6,16,10);
insert into continut_comanda values(7,9,3);
insert into continut_comanda values(8,15,3);
insert into continut_comanda values(9,5,30);
insert into continut_comanda values(10,23,4);
insert into continut_comanda values(11,22,1);
insert into continut_comanda values(12,2,50);
insert into continut_comanda values(13,19,4);

 select * from continut_comanda;
```

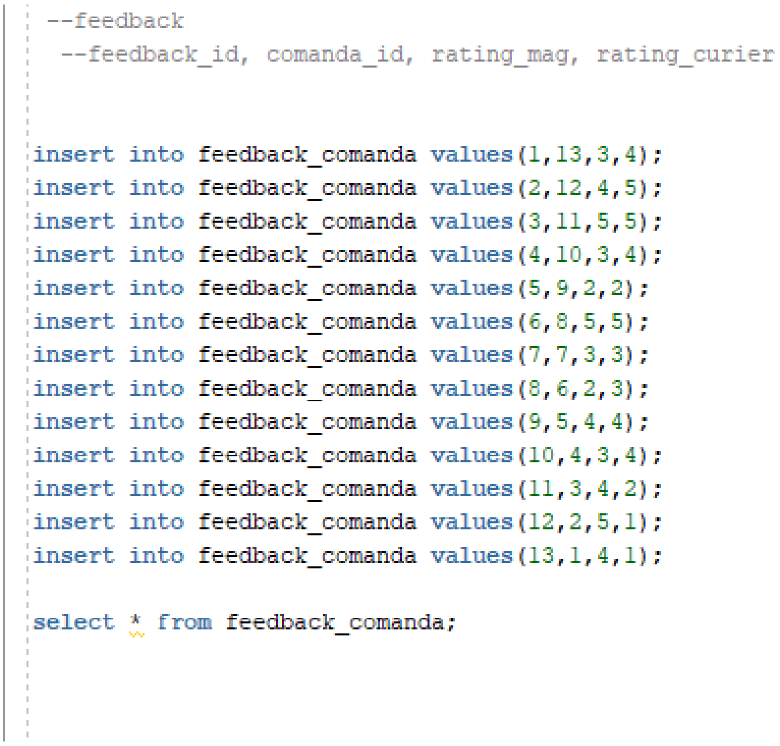


```
insert into continut_comanda values(13,19,4);

 select * from continut_comanda;
 rollback;
 --feedback
```

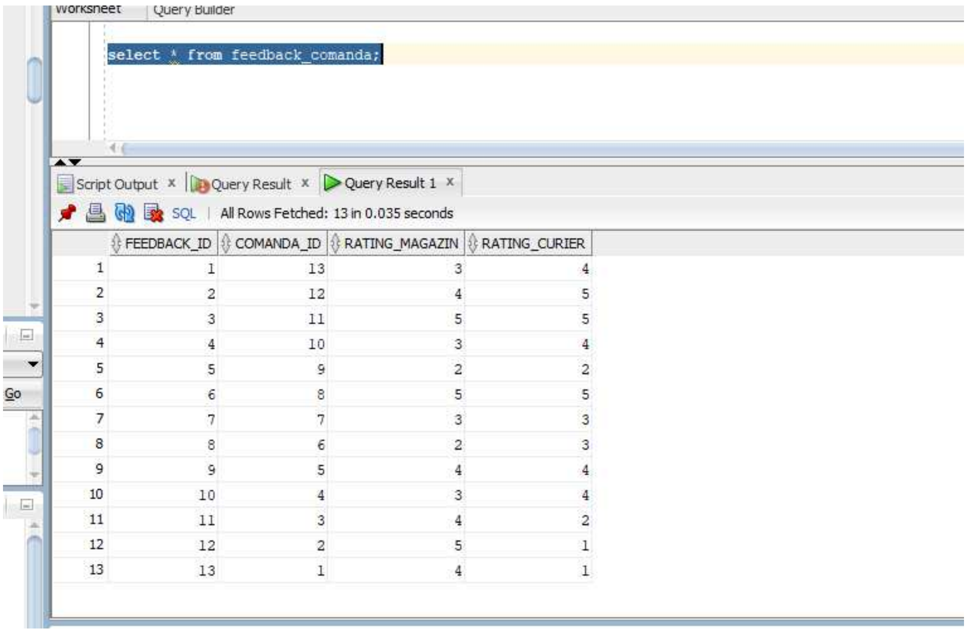
All Rows Fetched: 8 in 0.05 seconds

	COMANDA_ID	PRODUS_ID	CANTITATE
1	2	14	10
2	4	17	12
3	6	16	10
4	7	9	3
5	8	15	3
6	9	5	30
7	12	2	50
8	13	19	4

Feedback_comanda



```
 --feedback
  --feedback_id, comanda_id, rating_mag, rating_curier


insert into feedback_comanda values(1,13,3,4);
insert into feedback_comanda values(2,12,4,5);
insert into feedback_comanda values(3,11,5,5);
insert into feedback_comanda values(4,10,3,4);
insert into feedback_comanda values(5,9,2,2);
insert into feedback_comanda values(6,8,5,5);
insert into feedback_comanda values(7,7,3,3);
insert into feedback_comanda values(8,6,2,3);
insert into feedback_comanda values(9,5,4,4);
insert into feedback_comanda values(10,4,3,4);
insert into feedback_comanda values(11,3,4,2);
insert into feedback_comanda values(12,2,5,1);
insert into feedback_comanda values(13,1,4,1);

select * from feedback_comanda;
```



```
select * from feedback_comanda;
```

All Rows Fetched: 13 in 0.035 seconds

	FEEDBACK_ID	COMANDA_ID	RATING_MAGAZIN	RATING_CURIER
1	1	13	3	4
2	2	12	4	5
3	3	11	5	5
4	4	10	3	4
5	5	9	2	2
6	6	8	5	5
7	7	7	3	3
8	8	6	2	3
9	9	5	4	4
10	10	4	3	4
11	11	3	4	2
12	12	2	5	1
13	13	1	4	1

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

Am folosit doua tablouri, unul indexat pentru detalii_promotie si unul imbricat pentru detalii_comanda. Am folosit PLS_INTEGER pentru ca aduce ca avantaje spatiului folosit, avand nevoie de mult mai putin, si viteza, mult mai rapid decat number.

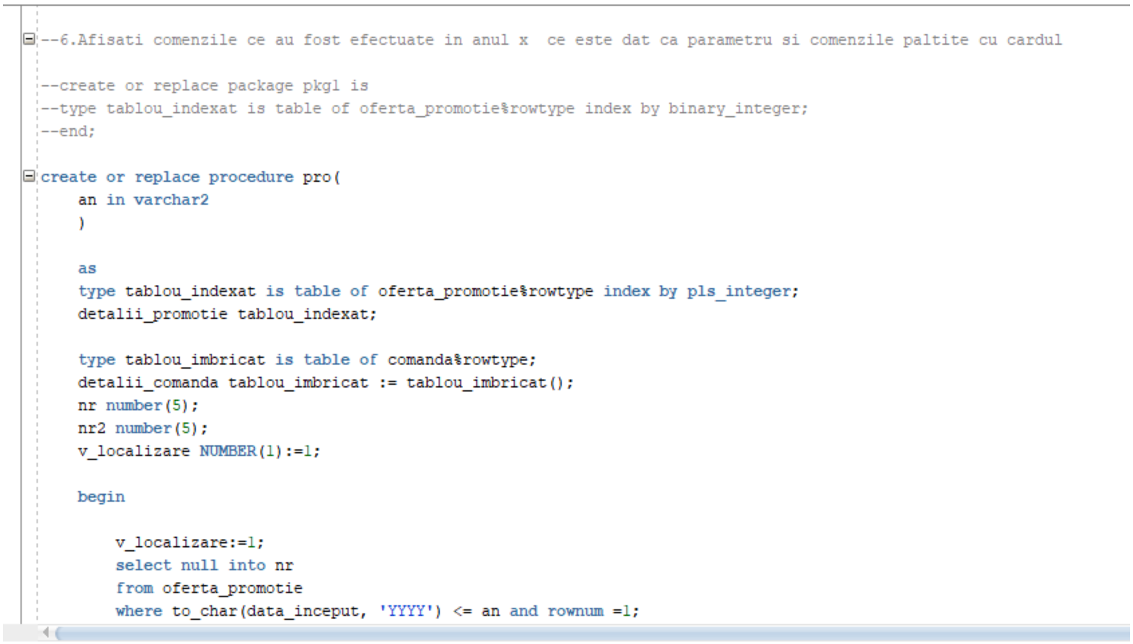
De asemenea, in cazul in care apar erori sau intra pe exceptii si vrem sa aflam care dintre comenzile select nu este buna, am folosit o variabila v_localizare, pe care am incrementat-o la fiecare select. O *eventuala problema ar fi fost nefolosirea “rownum =1” care ar fi produs afisarea exceptiei TOO_MANY_ROWS.*

--6.Afisati comenzile ce au fost efectuate in anul x ce este dat ca parametru si comenzile paltite cu cardul

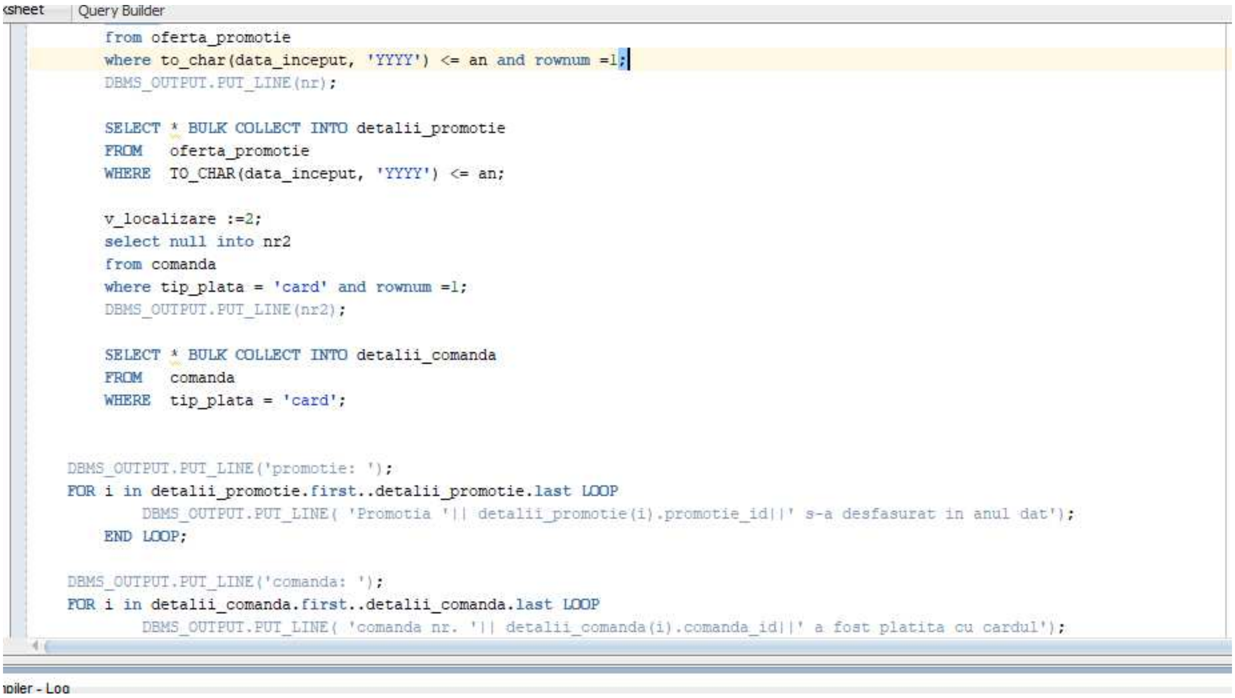
```
create or replace procedure pro(
  an in varchar2
  )

  as
  type tablou_indexat is table of oferta_promotie%rowtype index by pls_integer;
  detalii_promotie tablou_indexat;

  type tablou_imbricat is table of comanda%rowtype;
  detalii_comanda tablou_imbricat := tablou_imbricat();
  nr number(5);
  nr2 number(5);
  v_localizare NUMBER(1):=1;

  begin

    v_localizare:=1;
    select null into nr
    from oferta_promotie
    where to_char(data_inceput, 'YYYY') <= an and rownum =1;
    DBMS_OUTPUT.PUT_LINE(nr);

    SELECT * BULK COLLECT INTO detalii_promotie
    FROM   oferta_promotie
    WHERE  TO_CHAR(data_inceput, 'YYYY') <= an;
```

```
        v_localizare :=2;
        select null into nr2
        from comanda
        where tip_plata = 'card' and rownum =1;
        DBMS_OUTPUT.PUT_LINE(nr2);

        SELECT * BULK COLLECT INTO detalii_comanda
        FROM   comanda
        WHERE  tip_plata = 'card';



    DBMS_OUTPUT.PUT_LINE('promotie: ');
    FOR i in detalii_promotie.first..detalii_promotie.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'Promotia '|| detalii_promotie(i).promotie_id||' s-a desfasurat in anul dat');
        END LOOP;

    DBMS_OUTPUT.PUT_LINE('comanda: ');
    FOR i in detalii_comanda.first..detalii_comanda.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'comanda nr. '|| detalii_comanda(i).comanda_id||' a fost platita cu cardul');
        END LOOP;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
       RAISE_APPLICATION_ERROR(-20000, 'Nu exista comenzi');
--    WHEN OTHERS THEN
--       RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
when TOO_MANY_ROWS then
           RAISE_APPLICATION_ERROR(-20001, 'comanda select '||v_localizare || ' nu returneaza nimic ' || ' Exista mai multe reduceri cu acelasi cod');
  END pro;
```

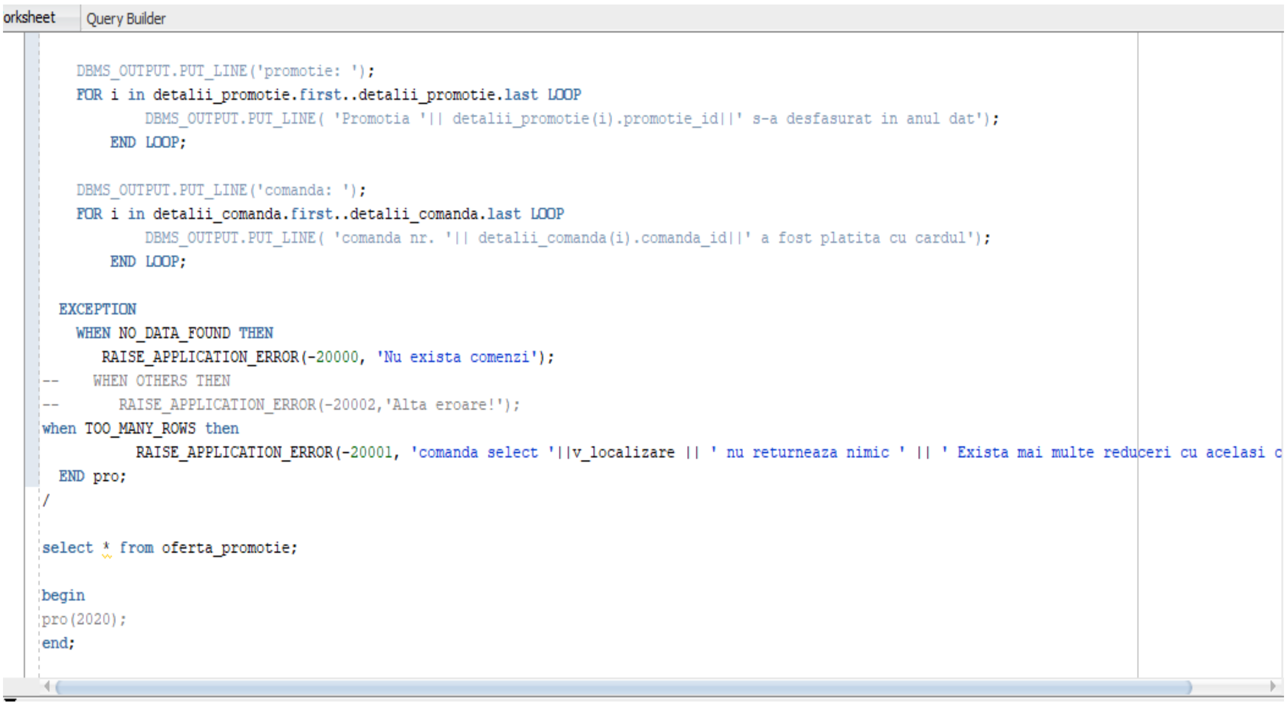
/



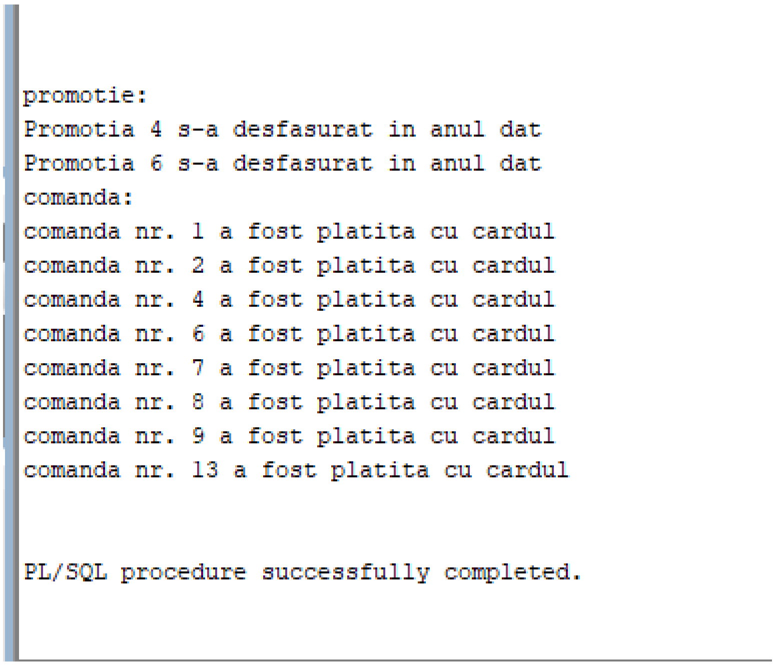
```
--6.Afisati comenzile ce au fost efectuate in anul x  ce este dat ca parametru si comenzile paltite cu cardul

--create or replace package pkg1 is
--type tablou_indexat is table of oferta_promotie%rowtype index by binary_integer;
--end;

create or replace procedure pro(
    an in varchar2
    )

    as
    type tablou_indexat is table of oferta_promotie%rowtype index by pls_integer;
    detalii_promotie tablou_indexat;

    type tablou_imbricat is table of comanda%rowtype;
    detalii_comanda tablou_imbricat := tablou_imbricat();
    nr number(5);
    nr2 number(5);
    v_localizare NUMBER(1):=1;

    begin

        v_localizare:=1;
        select null into nr
        from oferta_promotie
        where to_char(data_inceput, 'YYYY') <= an and rownum =1;
```



```
        from oferta_promotie
        where to_char(data_inceput, 'YYYY') <= an and rownum =1;
        DBMS_OUTPUT.PUT_LINE(nr);

        SELECT * BULK COLLECT INTO detalii_promotie
        FROM   oferta_promotie
        WHERE  TO_CHAR(data_inceput, 'YYYY') <= an;

        v_localizare :=2;
        select null into nr2
        from comanda
        where tip_plata = 'card' and rownum =1;
        DBMS_OUTPUT.PUT_LINE(nr2);

        SELECT * BULK COLLECT INTO detalii_comanda
        FROM   comanda
        WHERE  tip_plata = 'card';


    DBMS_OUTPUT.PUT_LINE('promotie: ');
    FOR i in detalii_promotie.first..detalii_promotie.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'Promotia '|| detalii_promotie(i).promotie_id||' s-a desfasurat in anul dat');
        END LOOP;

    DBMS_OUTPUT.PUT_LINE('comanda: ');
    FOR i in detalii_comanda.first..detalii_comanda.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'comanda nr. '|| detalii_comanda(i).comanda_id||' a fost platita cu cardul');
```



```
    DBMS_OUTPUT.PUT_LINE('promotie: ');
    FOR i in detalii_promotie.first..detalii_promotie.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'Promotia '|| detalii_promotie(i).promotie_id||' s-a desfasurat in anul dat');
        END LOOP;

    DBMS_OUTPUT.PUT_LINE('comanda: ');
    FOR i in detalii_comanda.first..detalii_comanda.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'comanda nr. '|| detalii_comanda(i).comanda_id||' a fost platita cu cardul');
        END LOOP;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
       RAISE_APPLICATION_ERROR(-20000, 'Nu exista comenzi');
--     WHEN OTHERS THEN
--        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
when TOO_MANY_ROWS then
           RAISE_APPLICATION_ERROR(-20001, 'comanda select '||v_localizare || ' nu returneaza nimic ' || ' Exista mai multe reduceri cu acelasi c
  END pro;
/

select * from oferta_promotie;

begin
pro(2020);
end;
```



```
promotie:
Promotia 4 s-a desfasurat in anul dat
Promotia 6 s-a desfasurat in anul dat
comanda:
comanda nr. 1 a fost platita cu cardul
comanda nr. 2 a fost platita cu cardul
comanda nr. 4 a fost platita cu cardul
comanda nr. 6 a fost platita cu cardul
comanda nr. 7 a fost platita cu cardul
comanda nr. 8 a fost platita cu cardul
comanda nr. 9 a fost platita cu cardul
comanda nr. 13 a fost platita cu cardul


PL/SQL procedure successfully completed.
```

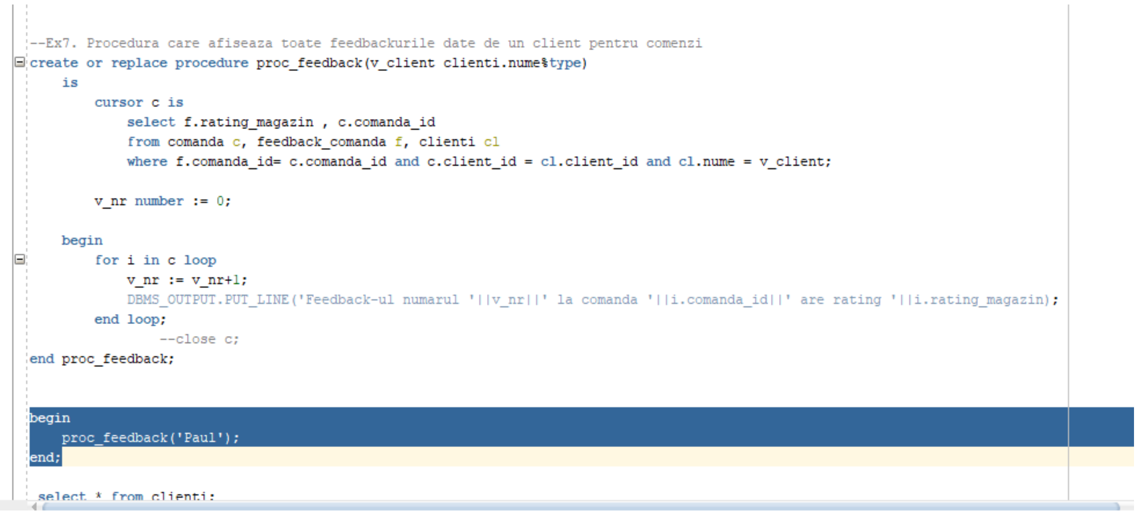
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

Pentru acest exercitiu am utilizat un cursor cu parametru dat numele unui client.

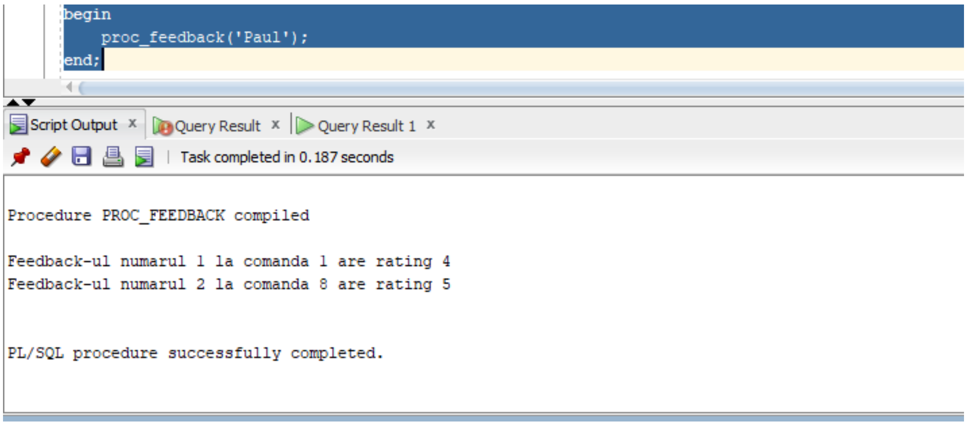
--Ex7. Procedura care afiseaza toate feedbackurile date de un client pentru comenzi

create or replace procedure proc_feedback(v_client clienti.nume%type)

```
   is
      cursor c is
         select f.rating_magazin , c.comanda_id
         from comanda c, feedback_comanda f, clienti cl
         where f.comanda_id= c.comanda_id and c.client_id = cl.client_id and cl.nume = v_client;

      v_nr number := 0;

   begin
      for i in c loop
         v_nr := v_nr+1;
         DBMS_OUTPUT.PUT_LINE('Feedback-ul numarul '||v_nr||' la comanda '||i.comanda_id||'
are rating '||i.rating_magazin);
      end loop;
            --close c;
end proc_feedback;


begin
   proc_feedback('Paul');
end;

select * from clienti;
   select * from comanda;
   select * from feedback_comanda;
```



```
--Ex7. Procedura care afiseaza toate feedbackurile date de un client pentru comenzi
create or replace procedure proc_feedback(v_client clienti.nume%type)
    is
        cursor c is
            select f.rating_magazin , c.comanda_id
            from comanda c, feedback_comanda f, clienti cl
            where f.comanda_id= c.comanda_id and c.client_id = cl.client_id and cl.nume = v_client;

        v_nr number := 0;

    begin
        for i in c loop
            v_nr := v_nr+1;
            DBMS_OUTPUT.PUT_LINE('Feedback-ul numarul '||v_nr||' la comanda '||i.comanda_id||' are rating '||i.rating_magazin);
        end loop;
                --close c;
end proc_feedback;


begin
    proc_feedback('Paul');
end;

 select * from clienti;
```



```
begin
    proc_feedback('Paul');
end;
```

```
Procedure PROC_FEEDBACK compiled

Feedback-ul numarul 1 la comanda 1 are rating 4
Feedback-ul numarul 2 la comanda 8 are rating 5


PL/SQL procedure successfully completed.
```

8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Cele trei tabelele folosite sunt magazin, stoc si produs. Functia are ca parametru id-ul magazinului.

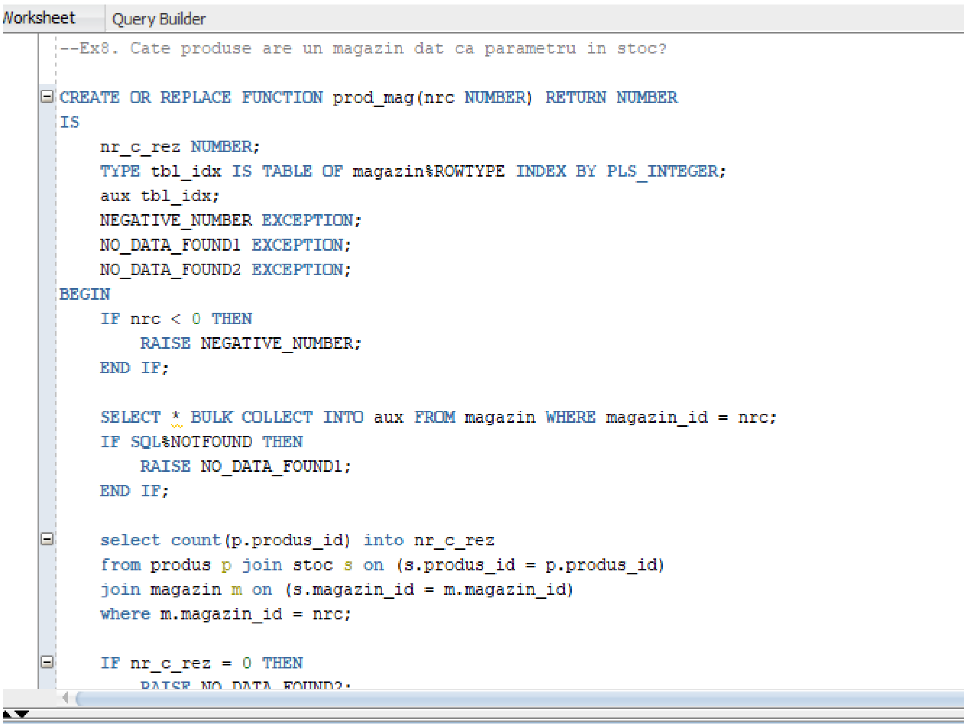
Din nou am folosit PLS_INTEGER pentru avantajele sale. Pentru magazine si pentru produse am utilizat doua exceptii NO_DATA_FOUND in caz ca nu exista magazin sau nu exista produse. In cazul in care apar valori negative va aparea mesajul corespunzator, iar in cazul altor erori se va afisa codul erorii si mesajul acesteia prin SQLCODE si SQLERRM.

--Ex8. Cate produse are un magazin dat ca parametru in stoc?

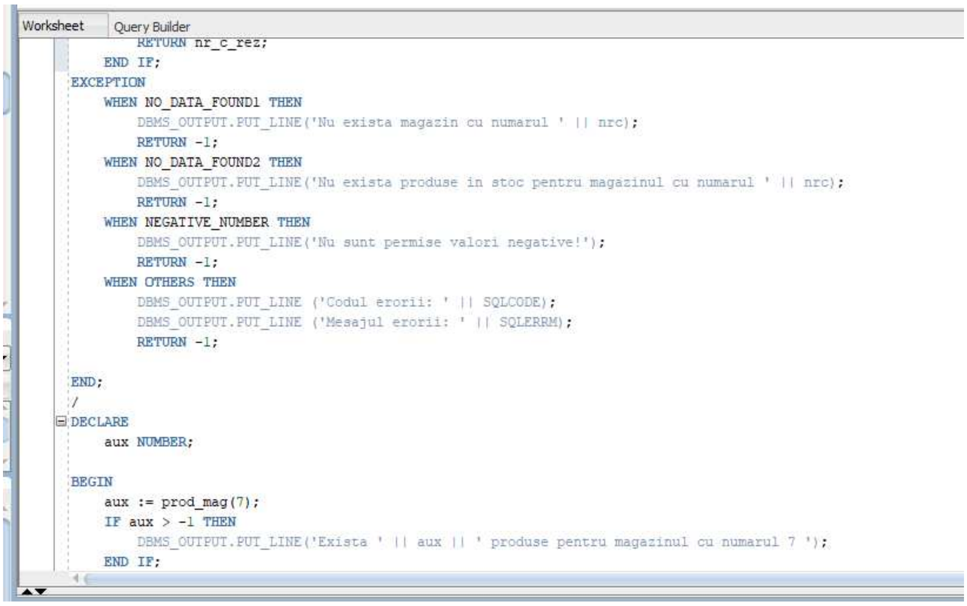
```
CREATE OR REPLACE FUNCTION prod_mag(nrc NUMBER) RETURN NUMBER

IS

    nr_c_rez NUMBER;

    TYPE tbl_idx IS TABLE OF magazin%ROWTYPE INDEX BY PLS_INTEGER;

    aux tbl_idx;
```

```
   NEGATIVE_NUMBER EXCEPTION;
   NO_DATA_FOUND1 EXCEPTION;
   NO_DATA_FOUND2 EXCEPTION;
BEGIN
   IF nrc < 0 THEN
      RAISE NEGATIVE_NUMBER;
   END IF;

   SELECT * BULK COLLECT INTO aux FROM magazin WHERE magazin_id = nrc;
   IF SQL%NOTFOUND THEN
      RAISE NO_DATA_FOUND1;
   END IF;

   select count(p.produs_id) into nr_c_rez
   from produs p join stoc s on (s.produs_id = p.produs_id)
   join magazin m on (s.magazin_id = m.magazin_id)
   where m.magazin_id = nrc;

   IF nr_c_rez = 0 THEN
      RAISE NO_DATA_FOUND2;
   ELSE
      RETURN nr_c_rez;
   END IF;
EXCEPTION
   WHEN NO_DATA_FOUND1 THEN
      DBMS_OUTPUT.PUT_LINE('Nu exista magazin cu numarul ' || nrc);
      RETURN -1;
   WHEN NO_DATA_FOUND2 THEN
      DBMS_OUTPUT.PUT_LINE('Nu exista produse in stoc pentru magazinul cu numarul ' || nrc);
      RETURN -1;
   WHEN NEGATIVE_NUMBER THEN
```

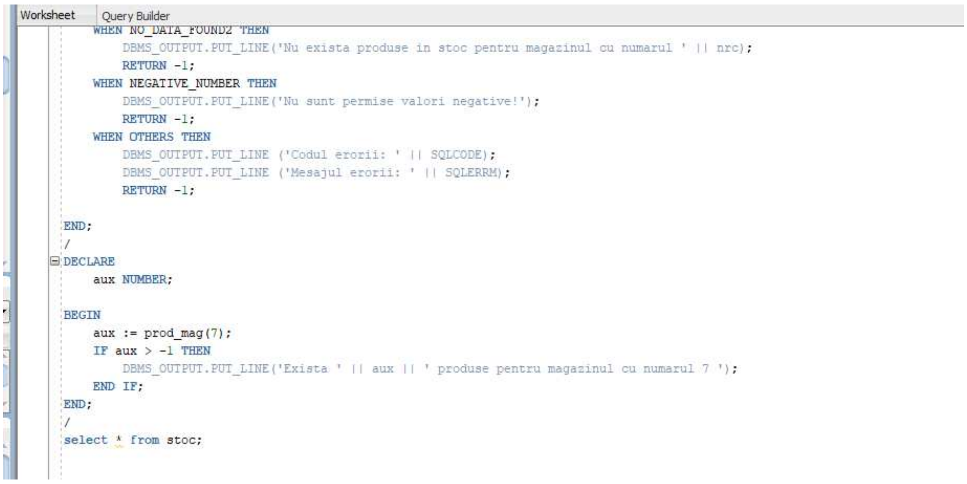
```
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;

END;
/
DECLARE
    aux NUMBER;

BEGIN
    aux := prod_mag(7);
    IF aux > -1 THEN
        DBMS_OUTPUT.PUT_LINE('Exista ' || aux || ' produse pentru magazinul cu numarul 7 ');
    END IF;
END;
/
```



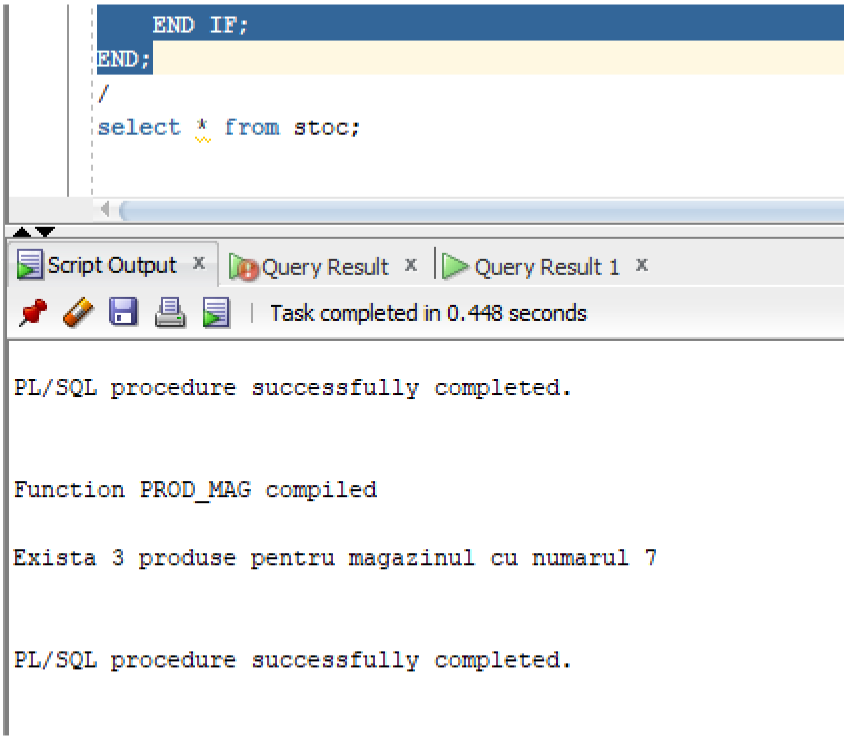
```
--Ex8. Cate produse are un magazin dat ca parametru in stoc?

CREATE OR REPLACE FUNCTION prod_mag(nrc NUMBER) RETURN NUMBER
IS
    nr_c_rez NUMBER;
    TYPE tbl_idx IS TABLE OF magazin%ROWTYPE INDEX BY PLS_INTEGER;
    aux tbl_idx;
    NEGATIVE_NUMBER EXCEPTION;
    NO_DATA_FOUND1 EXCEPTION;
    NO_DATA_FOUND2 EXCEPTION;
BEGIN
    IF nrc < 0 THEN
        RAISE NEGATIVE_NUMBER;
    END IF;

    SELECT * BULK COLLECT INTO aux FROM magazin WHERE magazin_id = nrc;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    select count(p.produs_id) into nr_c_rez
    from produs p join stoc s on (s.produs_id = p.produs_id)
    join magazin m on (s.magazin_id = m.magazin_id)
    where m.magazin_id = nrc;

    IF nr_c_rez = 0 THEN
        RAISE NO_DATA_FOUND2;
```



```
        RETURN nr_c_rez;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista magazin cu numarul ' || nrc);
        RETURN -1;
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista produse in stoc pentru magazinul cu numarul ' || nrc);
        RETURN -1;
    WHEN NEGATIVE_NUMBER THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;

END;
/
DECLARE
    aux NUMBER;

BEGIN
    aux := prod_mag(7);
    IF aux > -1 THEN
        DBMS_OUTPUT.PUT_LINE('Exista ' || aux || ' produse pentru magazinul cu numarul 7 ');
    END IF;
```



```
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista produse in stoc pentru magazinul cu numarul ' || nrc);
        RETURN -1;
    WHEN NEGATIVE_NUMBER THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;

END;
/
DECLARE
    aux NUMBER;

BEGIN
    aux := prod_mag(7);
    IF aux > -1 THEN
        DBMS_OUTPUT.PUT_LINE('Exista ' || aux || ' produse pentru magazinul cu numarul 7 ');
    END IF;
END;
/
select * from stoc;
```



```
    END IF;
END;
/
select * from stoc;
```

Task completed in 0.448 seconds

```
PL/SQL procedure successfully completed.


Function PROD_MAG compiled

Exista 3 produse pentru magazinul cu numarul 7


PL/SQL procedure successfully completed.
```

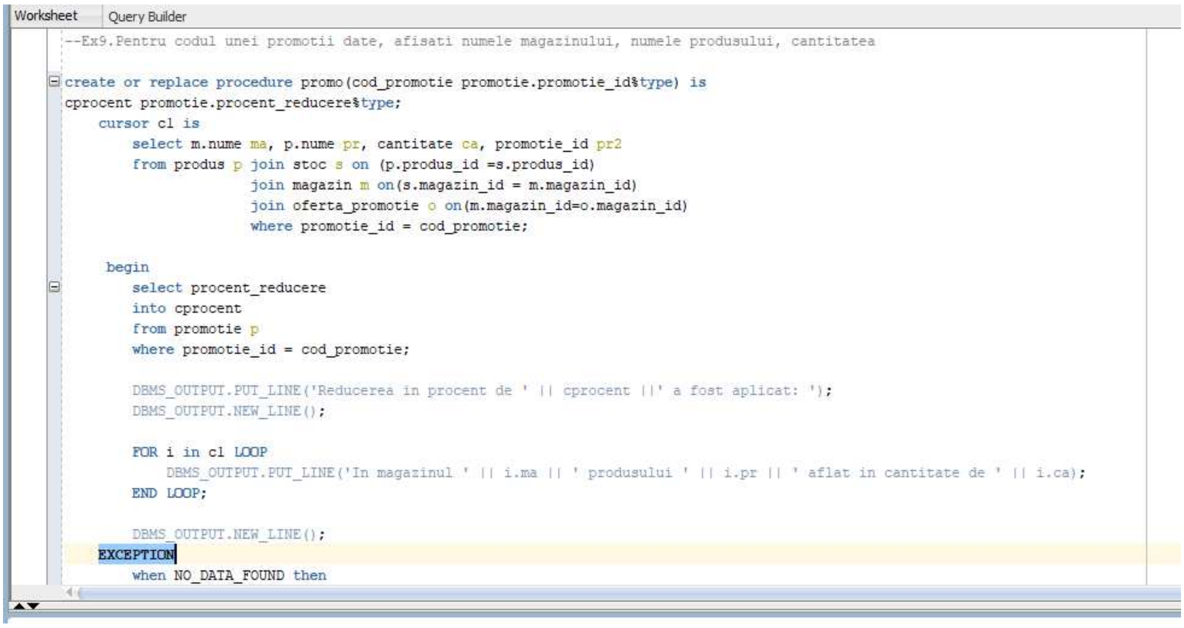
9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Tabelele utlizate sunt produs, stoc,magazin,oferta_promotie si promotie .

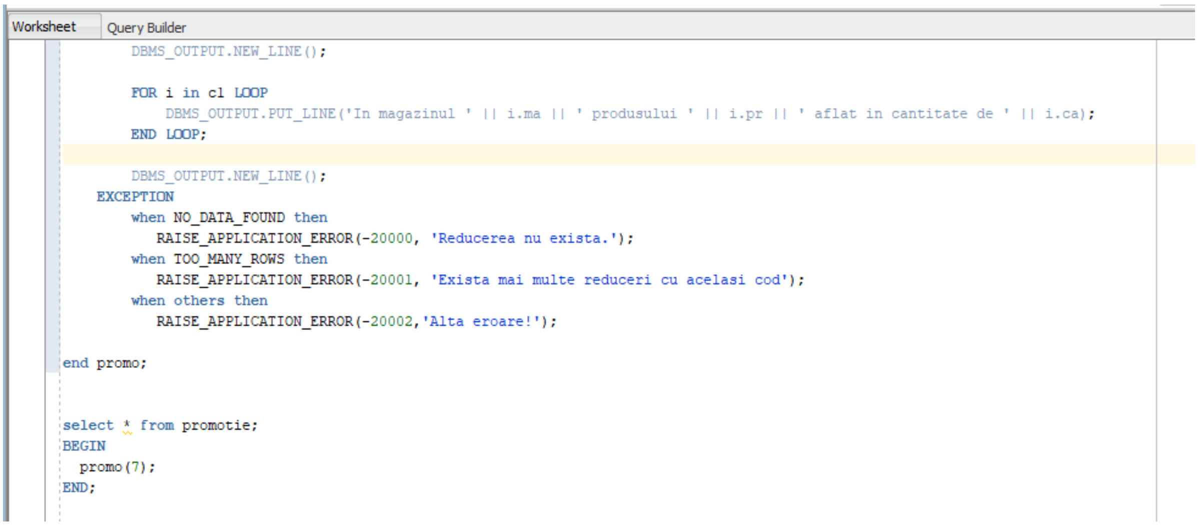
Exceptiile tratate au fost NO_DATA_FOUND, TOO_MANY_ROWS , iar pentru alte erori va fi afisat mesajul "alta eroare"

```
--Ex9.Pentru codul unei promotii date, afisati numele magazinului, numele produsului si cantitatea


create or replace procedure promo(cod_promotie promotie.promotie_id%type) is

cprocent promotie.procent_reducere%type;

  cursor c1 is

    select m.nume ma, p.nume pr, cantitate ca, promotie_id pr2

    from produs p join stoc s on (p.produs_id =s.produs_id)

           join magazin m on(s.magazin_id = m.magazin_id)

           join oferta_promotie o on(m.magazin_id=o.magazin_id)

           where promotie_id = cod_promotie;


   begin

    select procent_reducere

    into cprocent

    from promotie p

    where promotie_id = cod_promotie;


    DBMS_OUTPUT.PUT_LINE('Reducerea in procent de ' || cprocent ||' a fost aplicat: ');

    DBMS_OUTPUT.NEW_LINE();

    FOR i in c1 LOOP

      DBMS_OUTPUT.PUT_LINE('In magazinul ' || i.ma || ' produsului ' || i.pr || ' aflat in cantitate
de ' || i.ca);

    END LOOP;

    DBMS_OUTPUT.NEW_LINE();

  EXCEPTION
```

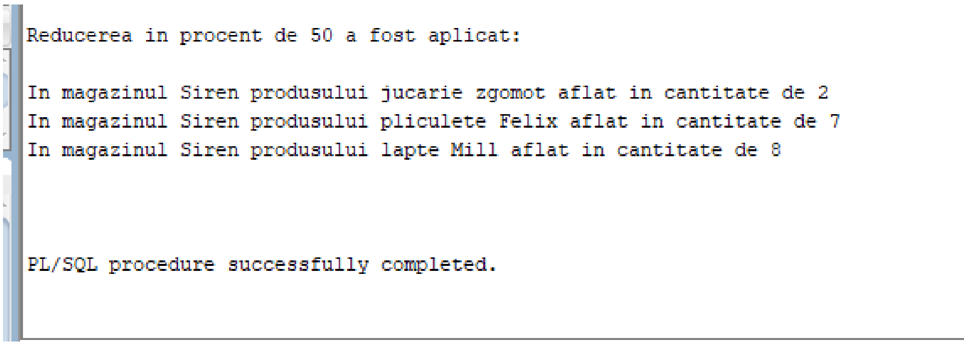
```
    when NO_DATA_FOUND then
      RAISE_APPLICATION_ERROR(-20000, 'Reducerea nu exista.');
    when TOO_MANY_ROWS then
      RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe reduceri cu acelasi cod');
    when others then
      RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

end promo;



select * from promotie;
BEGIN
  promo(7);
END;
```



```
--Ex9.Pentru codul unei promotii date, afisati numele magazinului, numele produsului, cantitatea

create or replace procedure promo(cod_promotie promotie.promotie_id%type) is
cprocent promotie.procent_reducere%type;
    cursor c1 is
        select m.nume ma, p.nume pr, cantitate ca, promotie_id pr2
        from produs p join stoc s on (p.produs_id =s.produs_id)
                      join magazin m on(s.magazin_id = m.magazin_id)
                      join oferta_promotie o on(m.magazin_id=o.magazin_id)
                      where promotie_id = cod_promotie;

     begin
        select procent_reducere
        into cprocent
        from promotie p
        where promotie_id = cod_promotie;

        DBMS_OUTPUT.PUT_LINE('Reducerea in procent de ' || cprocent ||' a fost aplicat: ');
        DBMS_OUTPUT.NEW_LINE();

        FOR i in c1 LOOP
            DBMS_OUTPUT.PUT_LINE('In magazinul ' || i.ma || ' produsului ' || i.pr || ' aflat in cantitate de ' || i.ca);
        END LOOP;

        DBMS_OUTPUT.NEW_LINE();
    EXCEPTION
        when NO_DATA_FOUND then
```



```
        DBMS_OUTPUT.NEW_LINE();

        FOR i in c1 LOOP
            DBMS_OUTPUT.PUT_LINE('In magazinul ' || i.ma || ' produsului ' || i.pr || ' aflat in cantitate de ' || i.ca);
        END LOOP;

        DBMS_OUTPUT.NEW_LINE();
    EXCEPTION
        when NO_DATA_FOUND then
           RAISE_APPLICATION_ERROR(-20000, 'Reducerea nu exista.');
        when TOO_MANY_ROWS then
           RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe reduceri cu acelasi cod');
        when others then
           RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

end promo;


select * from promotie;
BEGIN
  promo(7);
END;
```



```
Reducerea in procent de 50 a fost aplicat:

In magazinul Siren produsului jucarie zgomot aflat in cantitate de 2
In magazinul Siren produsului pliculete Felix aflat in cantitate de 7
In magazinul Siren produsului lapte Mill aflat in cantitate de 8



PL/SQL procedure successfully completed.
```

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

Am creat un trigger ce nu permite inserarea mai multor promotii fata de un numar dat, iar in cazul in care inserarea il depaseste, va afisa mesajul "Limita a fost depasita".

--Ex 10. -> trigger LMD la nivel de comanda

--10.Creati un trigger de tip LMD a nivel de comanda care sa nu permita inserarea

--mai multor promotii decat 10.

select * from promotie;

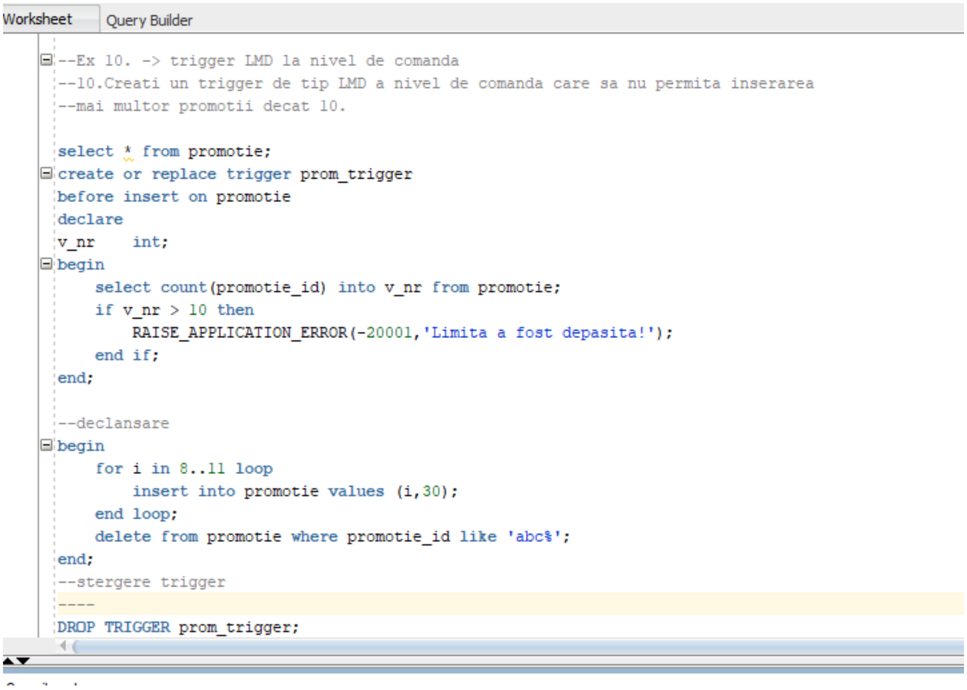
create or replace trigger prom_trigger

before insert on promotie

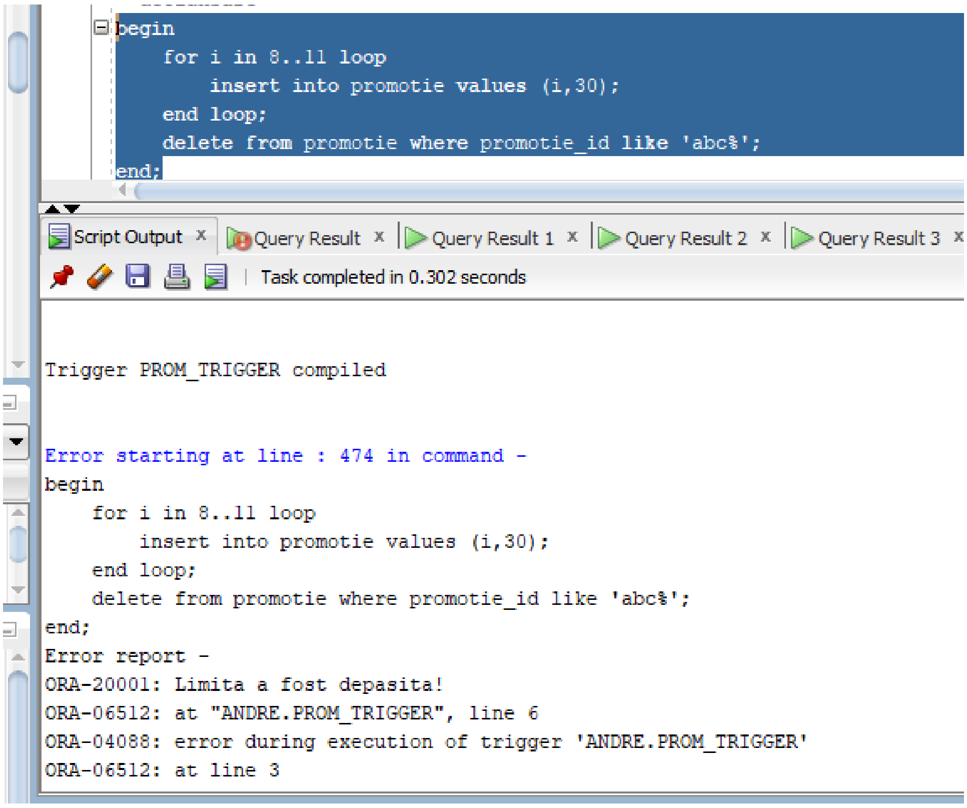
declare

v_nr   int;

```
begin
   select count(promotie_id) into v_nr from promotie;
   if v_nr > 10 then
      RAISE_APPLICATION_ERROR(-20001,'Limita a fost depasita!');
   end if;
end;

--declansare
begin
   for i in 8..11 loop
      insert into promotie values (i,30);
   end loop;
   delete from promotie where promotie_id like 'abc%';
end;
--stergere trigger


----

DROP TRIGGER prom_trigger;
```



```
--Ex 10. -> trigger LMD la nivel de comanda
--10.Creati un trigger de tip LMD a nivel de comanda care sa nu permita inserarea
--mai multor promotii decat 10.

select * from promotie;
create or replace trigger prom_trigger
before insert on promotie
declare
v_nr    int;
begin
    select count(promotie_id) into v_nr from promotie;
    if v_nr > 10 then
        RAISE_APPLICATION_ERROR(-20001,'Limita a fost depasita!');
    end if;
end;

--declansare
begin
    for i in 8..11 loop
        insert into promotie values (i,30);
    end loop;
    delete from promotie where promotie_id like 'abc%';
end;
--stergere trigger
----
DROP TRIGGER prom_trigger;
```



```
begin
    for i in 8..11 loop
        insert into promotie values (i,30);
    end loop;
    delete from promotie where promotie_id like 'abc%';
end;
```

Task completed in 0.302 seconds

```
Trigger PROM_TRIGGER compiled


Error starting at line : 474 in command -
begin
    for i in 8..11 loop
        insert into promotie values (i,30);
    end loop;
    delete from promotie where promotie_id like 'abc%';
end;
Error report -
ORA-20001: Limita a fost depasita!
ORA-06512: at "ANDRE.PROM_TRIGGER", line 6
ORA-04088: error during execution of trigger 'ANDRE.PROM_TRIGGER'
ORA-06512: at line 3
```

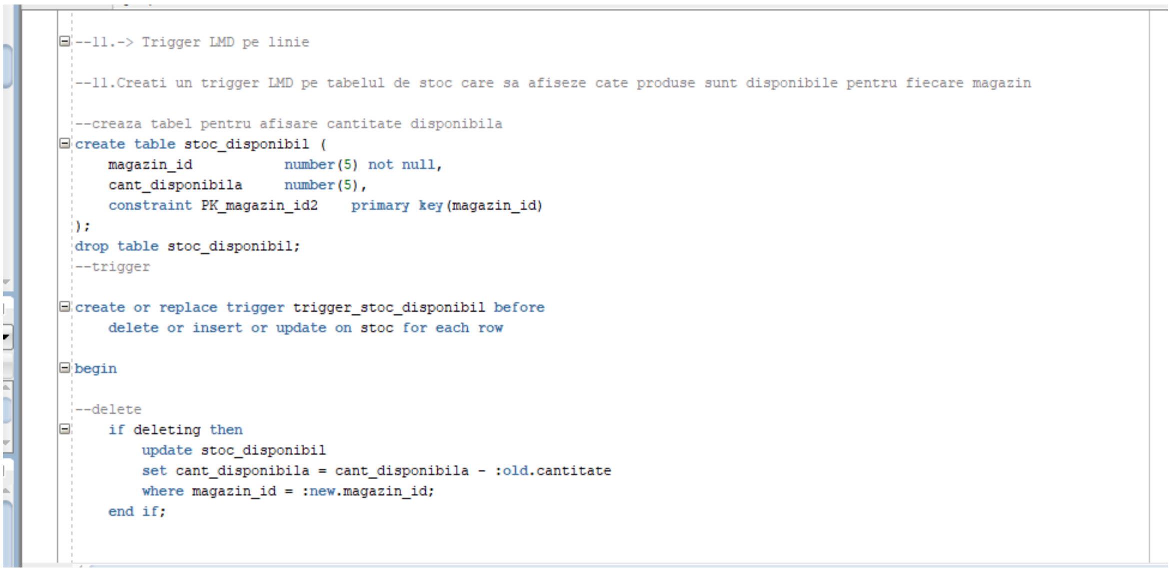
11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

Am creat un tabel nou stoc_disponibil in care am introdus cadate id ul magazinului si cantitatea de produse din el. De fiecare data cand se va face update, insert sau delete asupra tabelului stoc, tabelul nou creat va fi modificat si el in functie de comanda aleasa.

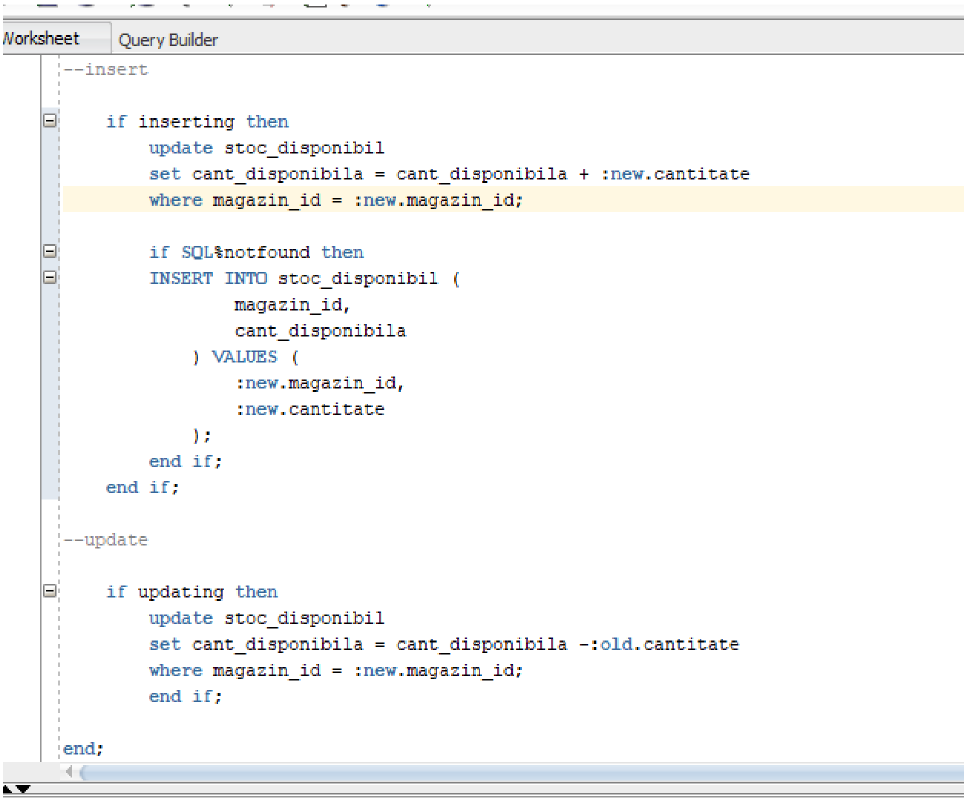
```
--11.-> Trigger LMD pe linie


--11.Creati un trigger LMD pe tabelul de stoc care sa afiseze cate produse sunt disponibile pentru fiecare magazin


--creaza tabel pentru afisare cantitate disponibila
create table stoc_disponibil (
  magazin_id          number(5) not null,
  cant_disponibila    number(5),
  constraint PK_magazin_id2    primary key(magazin_id)
);
--trigger


create or replace trigger trigger_stoc_disponibil before
  delete or insert or update on stoc for each row


begin


--delete
  if deleting then
    update stoc_disponibil
    set cant_disponibila = cant_disponibila - :old.cantitate
    where magazin_id = :new.magazin_id;
  end if;



--insert


  if inserting then
    update stoc_disponibil
```

```
        set cant_disponibila = cant_disponibila + :new.cantitate
        where magazin_id = :new.magazin_id;

        if SQL%notfound then
        INSERT INTO stoc_disponibil (
              magazin_id,
              cant_disponibila
          ) VALUES (
              :new.magazin_id,
              :new.cantitate
          );
        end if;
    end if;

--update

    if updating then
        update stoc_disponibil
        set cant_disponibila = cant_disponibila -:old.cantitate
        where magazin_id = :new.magazin_id;
        end if;

end;

select * from stoc_disponibil;
select * from stoc;
insert into stoc values(5,13, 2);
insert into stoc values (19, 3, 33);



select
```

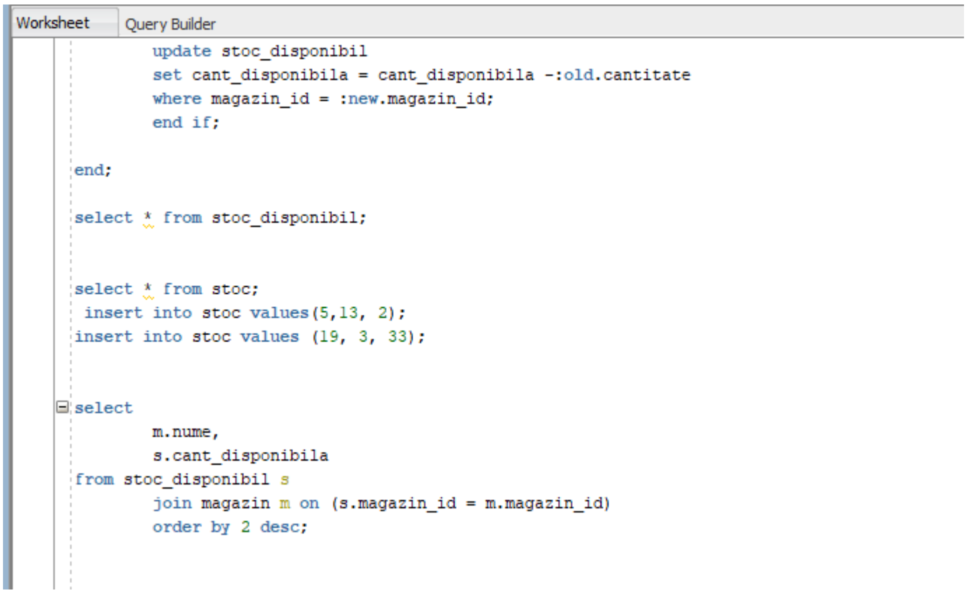
```
        m.nume,
        s.cant_disponibila
from stoc_disponibil s
        join magazin m on (s.magazin_id = m.magazin_id)
        order by 2 desc;
```



```
--11.-> Trigger LMD pe linie

--11.Creati un trigger LMD pe tabelul de stoc care sa afiseze cate produse sunt disponibile pentru fiecare magazin

--creaza tabel pentru afisare cantitate disponibila
create table stoc_disponibil (
    magazin_id          number(5) not null,
    cant_disponibila    number(5),
    constraint PK_magazin_id2    primary key(magazin_id)
);
drop table stoc_disponibil;
--trigger

create or replace trigger trigger_stoc_disponibil before
    delete or insert or update on stoc for each row

begin

--delete
    if deleting then
        update stoc_disponibil
        set cant_disponibila = cant_disponibila - :old.cantitate
        where magazin_id = :new.magazin_id;
    end if;
```

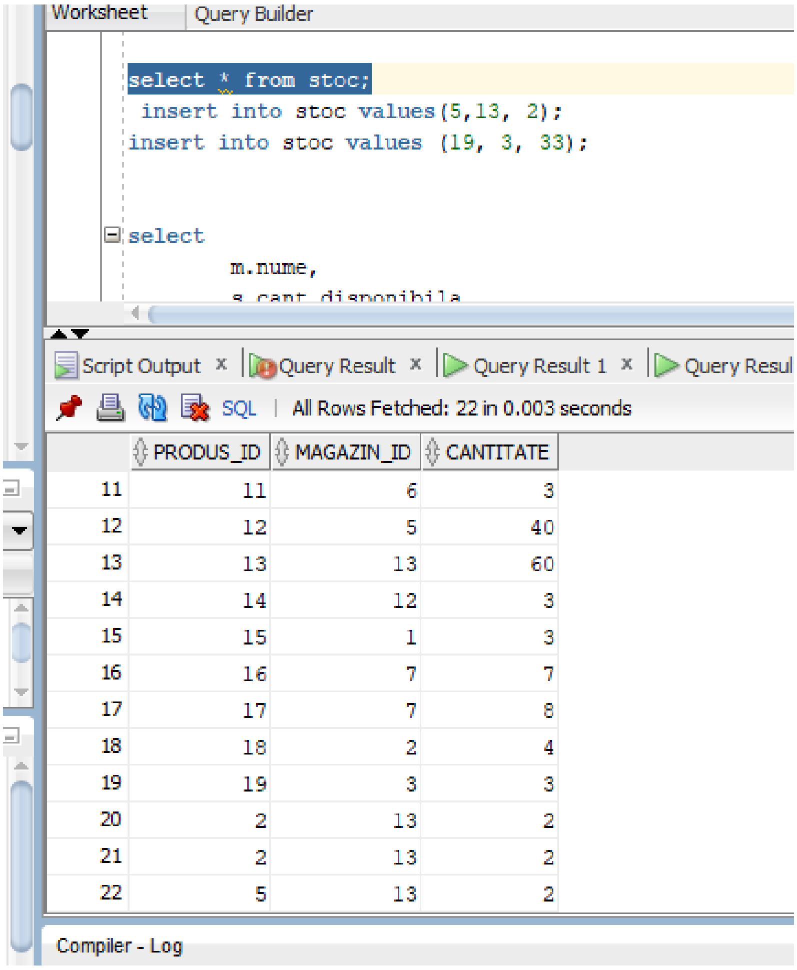


```
--insert

    if inserting then
        update stoc_disponibil
        set cant_disponibila = cant_disponibila + :new.cantitate
        where magazin_id = :new.magazin_id;

        if SQL%notfound then
        INSERT INTO stoc_disponibil (
                magazin_id,
                cant_disponibila
            ) VALUES (
                :new.magazin_id,
                :new.cantitate
            );
        end if;
    end if;

--update

    if updating then
        update stoc_disponibil
        set cant_disponibila = cant_disponibila -:old.cantitate
        where magazin_id = :new.magazin_id;
        end if;

end;
```



```
        update stoc_disponibil
        set cant_disponibila = cant_disponibila -:old.cantitate
        where magazin_id = :new.magazin_id;
        end if;

end;

select * from stoc_disponibil;


select * from stoc;
 insert into stoc values(5,13, 2);
insert into stoc values (19, 3, 33);


select
        m.nume,
        s.cant_disponibila
from stoc_disponibil s
        join magazin m on (s.magazin_id = m.magazin_id)
        order by 2 desc;
```

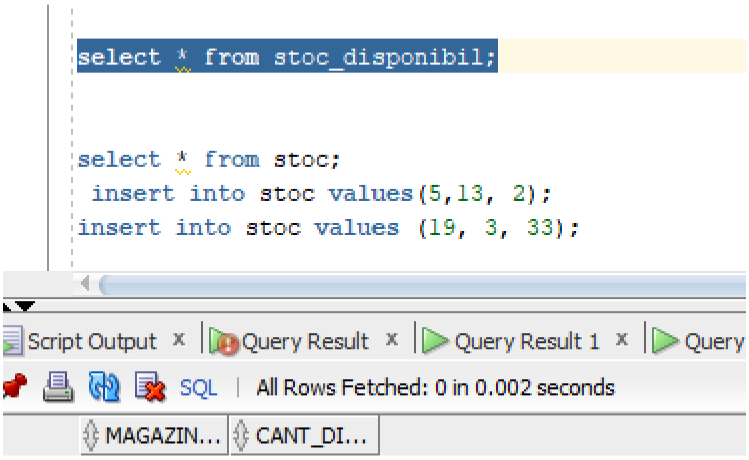
Inainte de a insera in tabelul stoc erau 22 de produse, iar in tabelul stoc_disponibil erau 0. Dupa inserare au tabelul stoc contine 24 de produse, iar in tabelul stoc_disponibil sunt 2 magazine



```
select * from stoc;
 insert into stoc values(5,13, 2);
insert into stoc values (19, 3, 33);


select
          m.nume,
          s.cant_disponibila
```

All Rows Fetched: 22 in 0.003 seconds

	PRODUS_ID	MAGAZIN_ID	CANTITATE
11	11	6	3
12	12	5	40
13	13	13	60
14	14	12	3
15	15	1	3
16	16	7	7
17	17	7	8
18	18	2	4
19	19	3	3
20	2	13	2
21	2	13	2
22	5	13	2

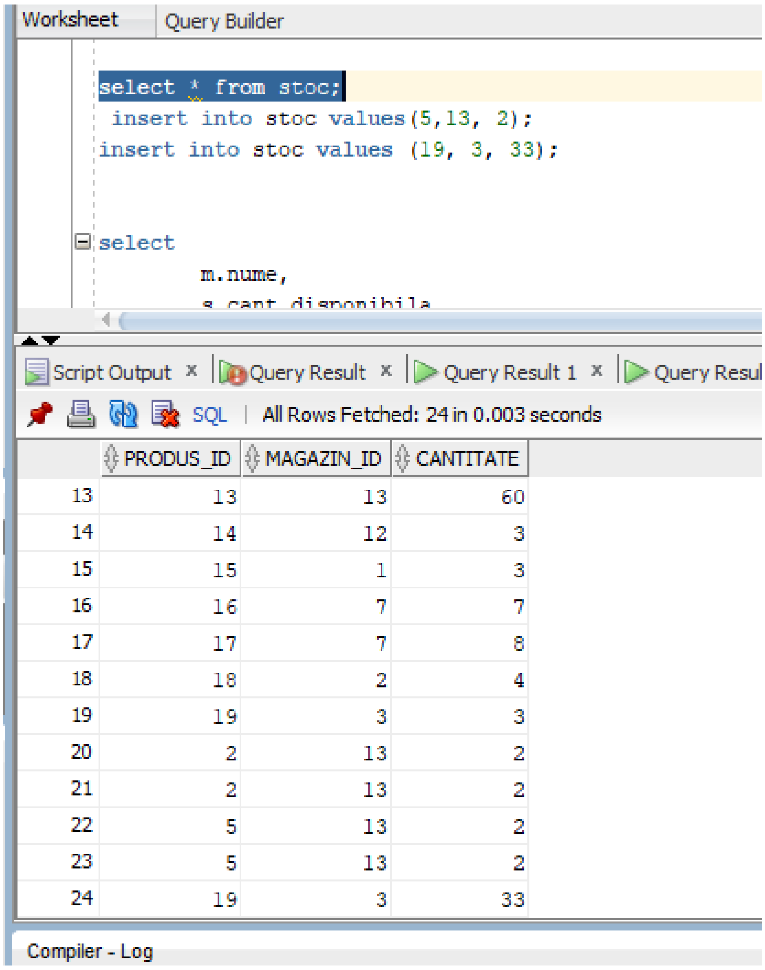


```
select * from stoc_disponibil;


select * from stoc;
 insert into stoc values(5,13, 2);
insert into stoc values (19, 3, 33);
```

All Rows Fetched: 0 in 0.002 seconds

MAGAZIN...	CANT_DI...

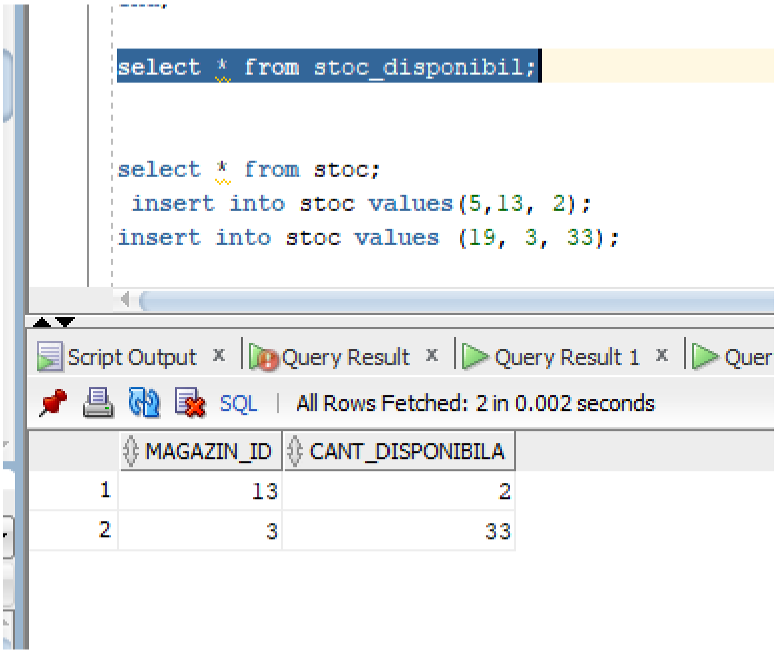
Dupa ce am inserat inca doua produse



```
select * from stoc;
 insert into stoc values(5,13, 2);
insert into stoc values (19, 3, 33);


select
        m.nume,
        s.cant_disponibila
```

All Rows Fetched: 24 in 0.003 seconds

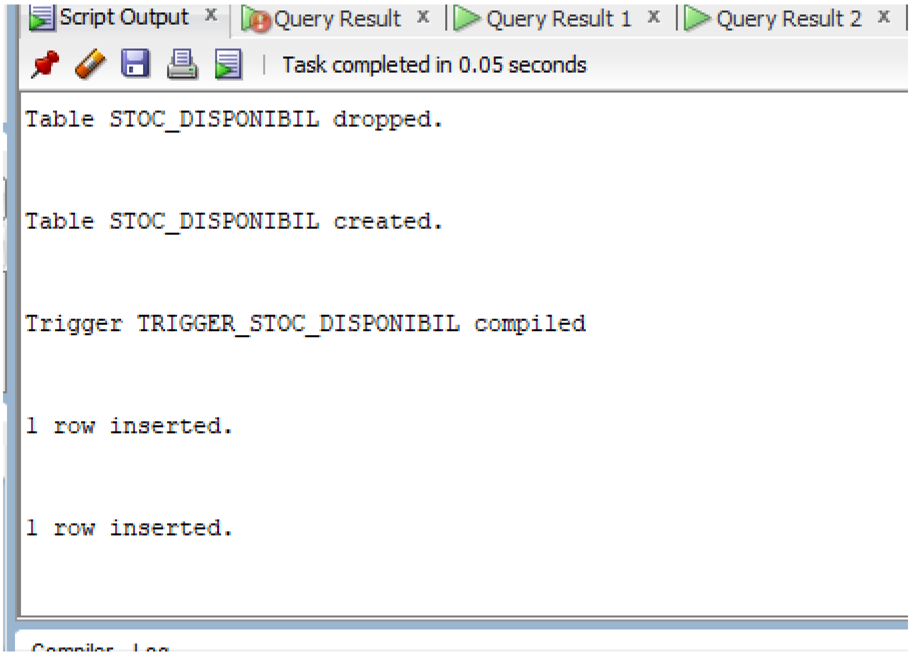
	PRODUS_ID	MAGAZIN_ID	CANTITATE
13	13	13	60
14	14	12	3
15	15	1	3
16	16	7	7
17	17	7	8
18	18	2	4
19	19	3	3
20	2	13	2
21	2	13	2
22	5	13	2
23	5	13	2
24	19	3	33



```
select * from stoc_disponibil;


select * from stoc;
 insert into stoc values(5,13, 2);
insert into stoc values (19, 3, 33);
```

All Rows Fetched: 2 in 0.002 seconds

	MAGAZIN_ID	CANT_DISPONIBILA
1	13	2
2	3	33



Task completed in 0.05 seconds

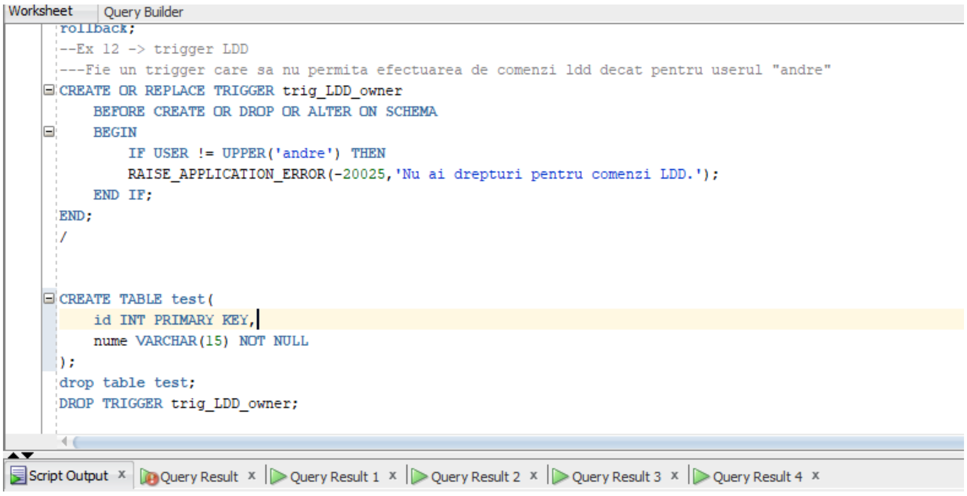
```
Table STOC_DISPONIBIL dropped.


Table STOC_DISPONIBIL created.


Trigger TRIGGER_STOC_DISPONIBIL compiled


1 row inserted.


1 row inserted.
```

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

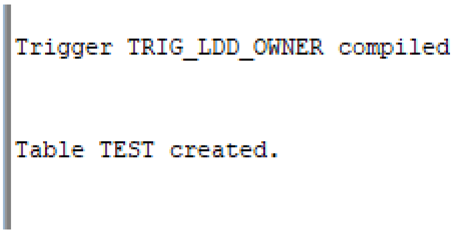
Am creat un trigger care nu permite efectuarea de comenzi altor utilizatori in afara de userul dat. In cazul de fata am pus ca utilizatorul sa fie 'andre' iar comenzile sunt permise. Daca am schimba numele utilizatorului din 'andre' in 'ana', atunci nu vor mai fi permise schimbari / comenzi.

```
--Ex 12 -> trigger LDD

---Fie un trigger care sa nu permita efectuarea de comenzi ldd decat pentru userul "andre"

CREATE OR REPLACE TRIGGER trig_LDD_owner

   BEFORE CREATE OR DROP OR ALTER ON SCHEMA

   BEGIN
```

```
        IF USER != UPPER('andre') THEN
        RAISE_APPLICATION_ERROR(-20025,'Nu ai drepturi pentru comenzi LDD.');
    END IF;
END;
/



CREATE TABLE test(
    id INT PRIMARY KEY,
    nume VARCHAR(15) NOT NULL
);
drop table test;
DROP TRIGGER trig_LDD_owner;
```

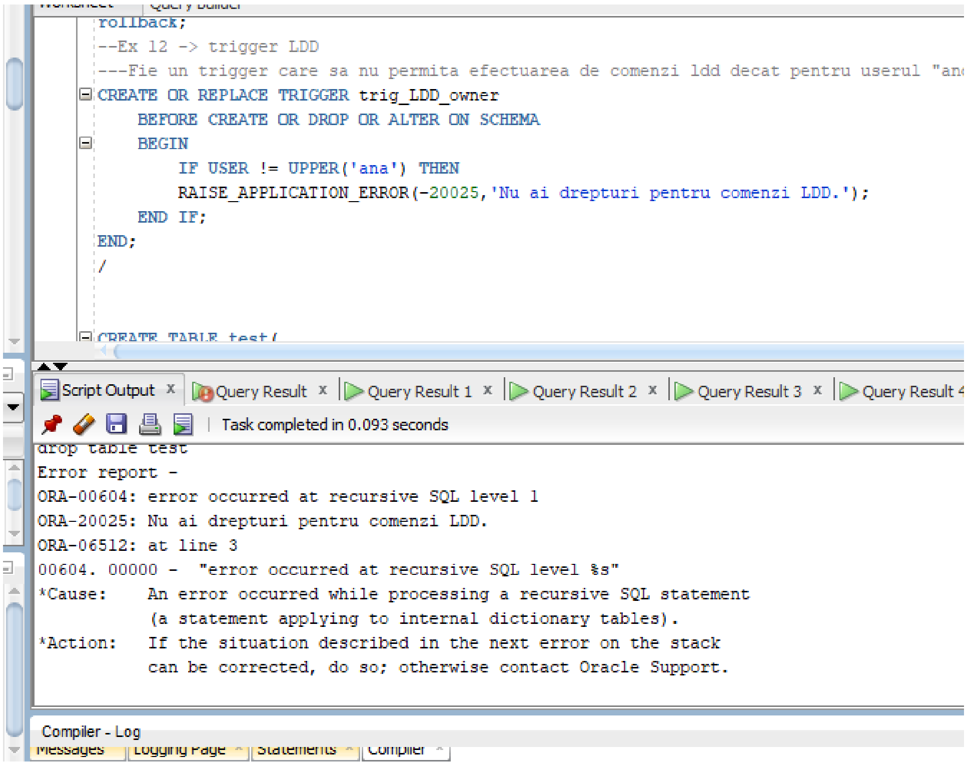


```
rollback;
--Ex 12 -> trigger LDD
---Fie un trigger care sa nu permita efectuarea de comenzi ldd decat pentru userul "andre"
CREATE OR REPLACE TRIGGER trig_LDD_owner
    BEFORE CREATE OR DROP OR ALTER ON SCHEMA
    BEGIN
        IF USER != UPPER('andre') THEN
        RAISE_APPLICATION_ERROR(-20025,'Nu ai drepturi pentru comenzi LDD.');
    END IF;
END;
/


CREATE TABLE test(
    id INT PRIMARY KEY,
    nume VARCHAR(15) NOT NULL
);
drop table test;
DROP TRIGGER trig_LDD_owner;
```



```
Trigger TRIG_LDD_OWNER compiled


Table TEST created.
```

Daca am modificat userul din ‘andre’ in ‘ana’, triggerul va provoca o eroare atunci cand incerc sa creez un tabel nou, sa ii dau drop etc.



```
rollback;
--Ex 12 -> trigger LDD
---Fie un trigger care sa nu permita efectuarea de comenzi ldd decat pentru userul "an
CREATE OR REPLACE TRIGGER trig_LDD_owner
    BEFORE CREATE OR DROP OR ALTER ON SCHEMA
    BEGIN
        IF USER != UPPER('ana') THEN
        RAISE_APPLICATION_ERROR(-20025,'Nu ai drepturi pentru comenzi LDD.');
    END IF;
END;
/


CREATE TABLE test(
```

Task completed in 0.093 seconds

```
drop table test
Error report -
ORA-00604: error occurred at recursive SQL level 1
ORA-20025: Nu ai drepturi pentru comenzi LDD.
ORA-06512: at line 3
00604. 00000 -  "error occurred at recursive SQL level %s"
*Cause:    An error occurred while processing a recursive SQL statement
           (a statement applying to internal dictionary tables).
*Action:   If the situation described in the next error on the stack
           can be corrected, do so; otherwise contact Oracle Support.
```

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

```
create or replace package proiect_cag as

   procedure pro(an in varchar2);

   procedure proc_feedback(v_client clienti.nume%type);

   function prod_mag(nrc NUMBER) RETURN NUMBER;

   procedure promo(cod_promotie promotie.promotie_id%type);

end proiect_cag;

/


create or replace package body proiect_cag as
```

--6.Afisati comenzile ce au fost efectuate in anul x ce este dat ca parametru si comenzile paltite cu cardul

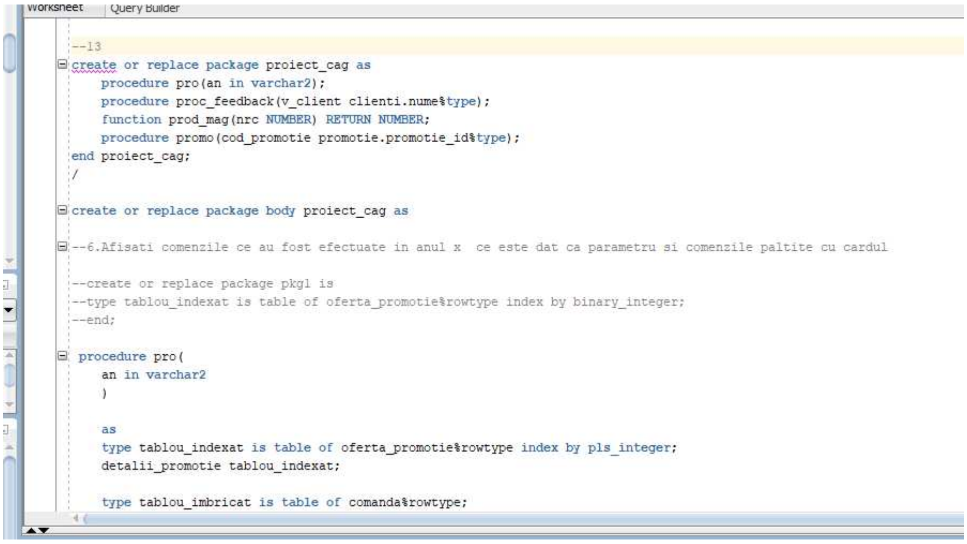
```
--create or replace package pkg1 is
--type tablou_indexat is table of oferta_promotie%rowtype index by binary_integer;
--end;

procedure pro(
   an in varchar2
   )

   as
   type tablou_indexat is table of oferta_promotie%rowtype index by pls_integer;
   detalii_promotie tablou_indexat;

   type tablou_imbricat is table of comanda%rowtype;
   detalii_comanda tablou_imbricat := tablou_imbricat();
   nr number(5);
   nr2 number(5);
   v_localizare NUMBER(1):=1;

   begin

      v_localizare:=1;
      select null into nr
      from oferta_promotie
      where to_char(data_inceput, 'YYYY') <= an and rownum =1;
      DBMS_OUTPUT.PUT_LINE(nr);

      SELECT * BULK COLLECT INTO detalii_promotie
      FROM   oferta_promotie
      WHERE  TO_CHAR(data_inceput, 'YYYY') <= an;

      v_localizare :=2;
```

```
    select null into nr2
    from comanda
    where tip_plata = 'card' and rownum =1;
    DBMS_OUTPUT.PUT_LINE(nr2);

    SELECT * BULK COLLECT INTO detalii_comanda
    FROM   comanda
    WHERE  tip_plata = 'card';



  DBMS_OUTPUT.PUT_LINE('promotie: ');
  FOR i in detalii_promotie.first..detalii_promotie.last LOOP
      DBMS_OUTPUT.PUT_LINE( 'Promotia '|| detalii_promotie(i).promotie_id||' s-a desfasurat in anul dat');
    END LOOP;

  DBMS_OUTPUT.PUT_LINE('comanda: ');
  FOR i in detalii_comanda.first..detalii_comanda.last LOOP
      DBMS_OUTPUT.PUT_LINE( 'comanda nr. '|| detalii_comanda(i).comanda_id||' a fost platita cu cardul');
    END LOOP;

 EXCEPTION
  WHEN NO_DATA_FOUND THEN
   RAISE_APPLICATION_ERROR(-20000, 'Nu exista comenzi');

  when TOO_MANY_ROWS then
     RAISE_APPLICATION_ERROR(-20001, 'comanda select '||v_localizare || ' nu returneaza nimic ' || ' Exista mai multe reduceri cu acelasi cod');
 END pro;
```

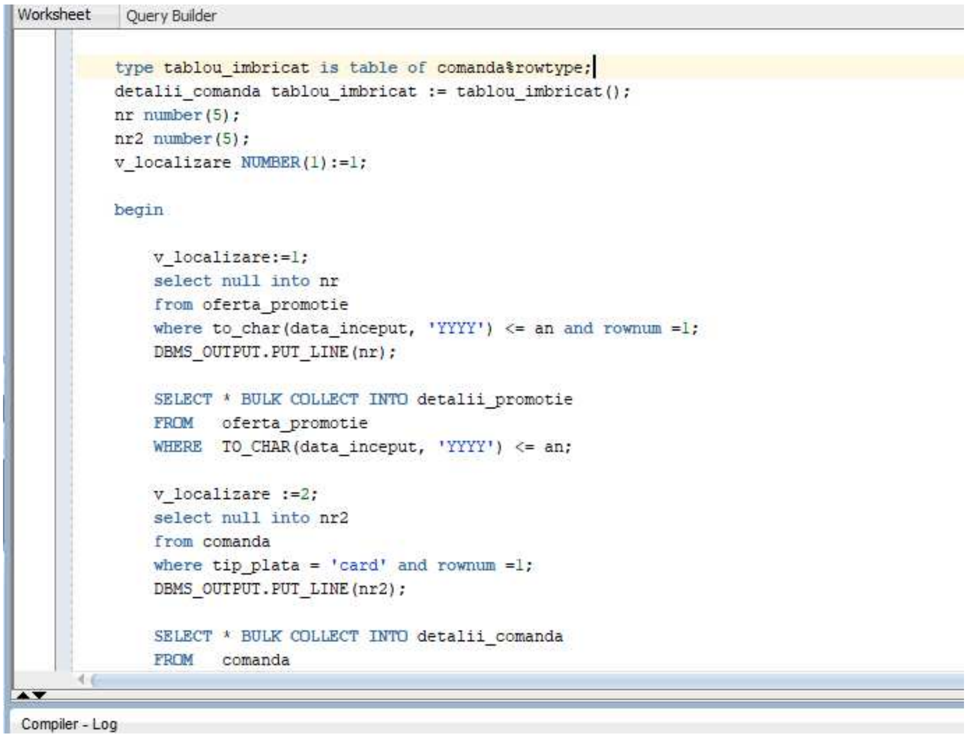
```
--Ex7. Procedura care afiseaza toate feedbackurile date de un client pentru comenzi
procedure proc_feedback(v_client clienti.nume%type)
   is
      cursor c is
         select f.rating_magazin , c.comanda_id
         from comanda c, feedback_comanda f, clienti cl
         where f.comanda_id= c.comanda_id and c.client_id = cl.client_id and cl.nume = v_client;

      v_nr number := 0;

   begin
      for i in c loop
         v_nr := v_nr+1;
         DBMS_OUTPUT.PUT_LINE('Feedback-ul numarul '||v_nr||' la comanda '||i.comanda_id||' are rating '||i.rating_magazin);
      end loop;
            --close c;
end proc_feedback;

--Ex8. Cate produse are un magazin dat ca parametru in stoc?

FUNCTION prod_mag(nrc NUMBER) RETURN NUMBER
IS
   nr_c_rez NUMBER;
   TYPE tbl_idx IS TABLE OF magazin%ROWTYPE INDEX BY PLS_INTEGER;
   aux tbl_idx;
   NEGATIVE_NUMBER EXCEPTION;
   NO_DATA_FOUND1 EXCEPTION;
   NO_DATA_FOUND2 EXCEPTION;
BEGIN
   IF nrc < 0 THEN
```

```
        RAISE NEGATIVE_NUMBER;
    END IF;

    SELECT * BULK COLLECT INTO aux FROM magazin WHERE magazin_id = nrc;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    select count(p.produs_id) into nr_c_rez
    from produs p join stoc s on (s.produs_id = p.produs_id)
    join magazin m on (s.magazin_id = m.magazin_id)
    where m.magazin_id = nrc;

    IF nr_c_rez = 0 THEN
        RAISE NO_DATA_FOUND2;
    ELSE
        RETURN nr_c_rez;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista magazin cu numarul ' || nrc);
        RETURN -1;
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista produse in stoc pentru magazinul cu numarul ' || nrc);
        RETURN -1;
    WHEN NEGATIVE_NUMBER THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;

END;



--Ex9.Pentru codul unei promotii date, afisati numele magazinului, numele produsului, cantitatea si mai vedem


procedure promo(cod_promotie promotie.promotie_id%type) is

cprocent promotie.procent_reducere%type;

    cursor c1 is

        select m.nume ma, p.nume pr, cantitate ca, promotie_id pr2

        from produs p join stoc s on (p.produs_id =s.produs_id)

                     join magazin m on(s.magazin_id = m.magazin_id)

                     join oferta_promotie o on(m.magazin_id=o.magazin_id)

                     where promotie_id = cod_promotie;


     begin

        select procent_reducere

        into cprocent

        from promotie p

        where promotie_id = cod_promotie;


        DBMS_OUTPUT.PUT_LINE('Reducerea in procent de ' || cprocent ||' a fost aplicat: ');

        DBMS_OUTPUT.NEW_LINE();


        FOR i in c1 LOOP

            DBMS_OUTPUT.PUT_LINE('In magazinul ' || i.ma || ' produsului ' || i.pr || ' aflat in cantitate de ' || i.ca);

        END LOOP;


        DBMS_OUTPUT.NEW_LINE();
```

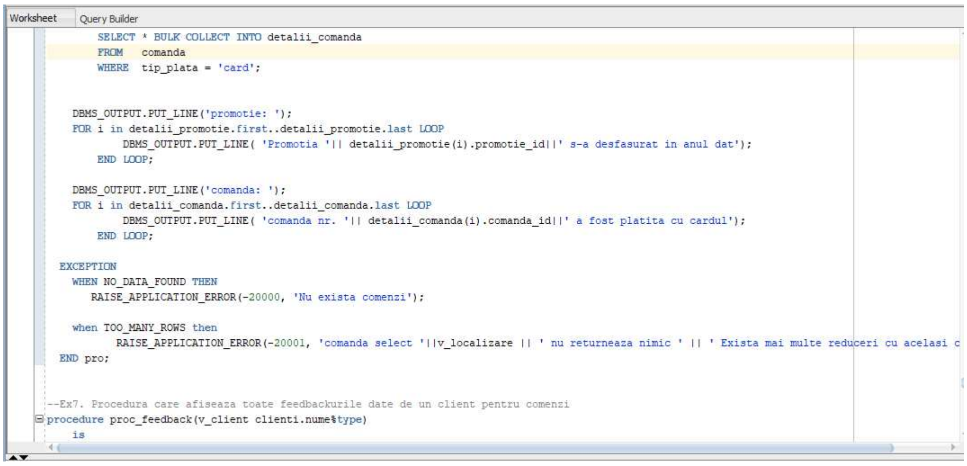
```
  EXCEPTION
     when NO_DATA_FOUND then
        RAISE_APPLICATION_ERROR(-20000, 'Reducerea nu exista.');
     when TOO_MANY_ROWS then
        RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe reduceri cu acelasi cod');
     when others then
        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

end promo;

end proiect_cag;
```



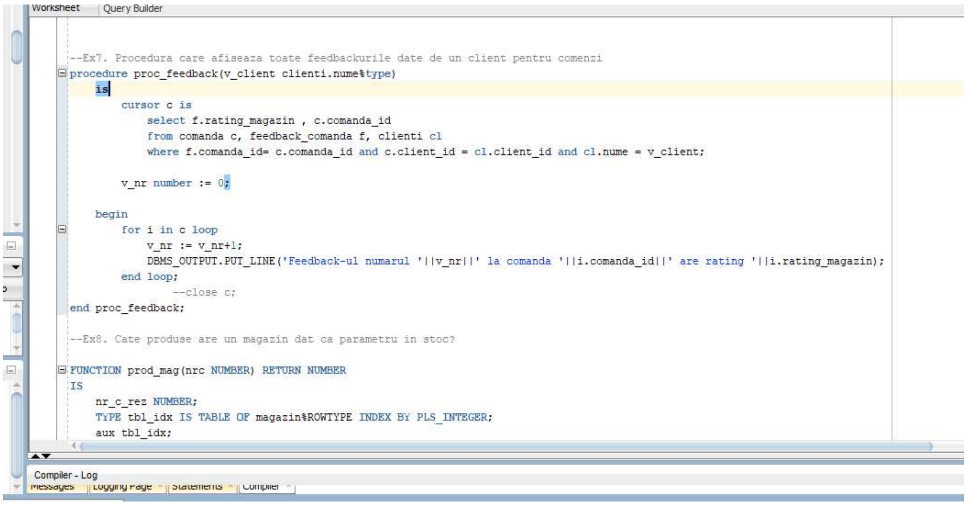
```
--13
create or replace package proiect_cag as
    procedure pro(an in varchar2);
    procedure proc_feedback(v_client clienti.nume%type);
    function prod_mag(nrc NUMBER) RETURN NUMBER;
    procedure promo(cod_promotie promotie.promotie_id%type);
end proiect_cag;
/

create or replace package body proiect_cag as

--6.Afisati comenzile ce au fost efectuate in anul x  ce este dat ca parametru si comenzile paltite cu cardul

--create or replace package pkg1 is
--type tablou_indexat is table of oferta_promotie%rowtype index by binary_integer;
--end;

 procedure pro(
    an in varchar2
    )

    as
    type tablou_indexat is table of oferta_promotie%rowtype index by pls_integer;
    detalii_promotie tablou_indexat;

    type tablou_imbricat is table of comanda%rowtype;
```



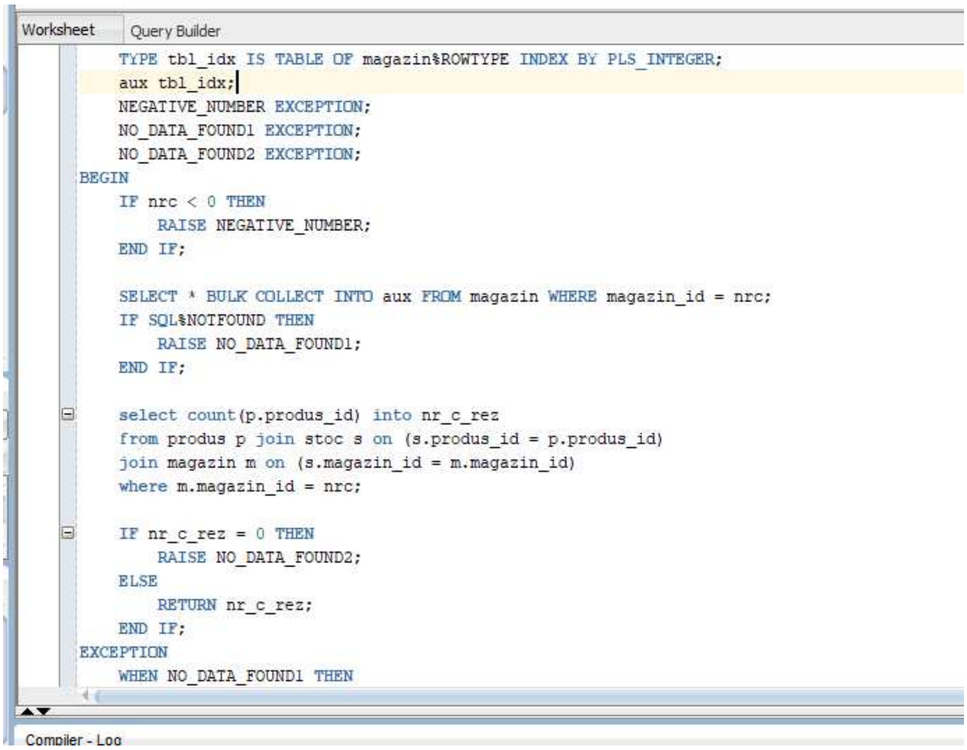
```
    type tablou_imbricat is table of comanda%rowtype;
    detalii_comanda tablou_imbricat := tablou_imbricat();
    nr number(5);
    nr2 number(5);
    v_localizare NUMBER(1):=1;

    begin

        v_localizare:=1;
        select null into nr
        from oferta_promotie
        where to_char(data_inceput, 'YYYY') <= an and rownum =1;
        DBMS_OUTPUT.PUT_LINE(nr);

        SELECT * BULK COLLECT INTO detalii_promotie
        FROM   oferta_promotie
        WHERE  TO_CHAR(data_inceput, 'YYYY') <= an;

        v_localizare :=2;
        select null into nr2
        from comanda
        where tip_plata = 'card' and rownum =1;
        DBMS_OUTPUT.PUT_LINE(nr2);

        SELECT * BULK COLLECT INTO detalii_comanda
        FROM   comanda
```



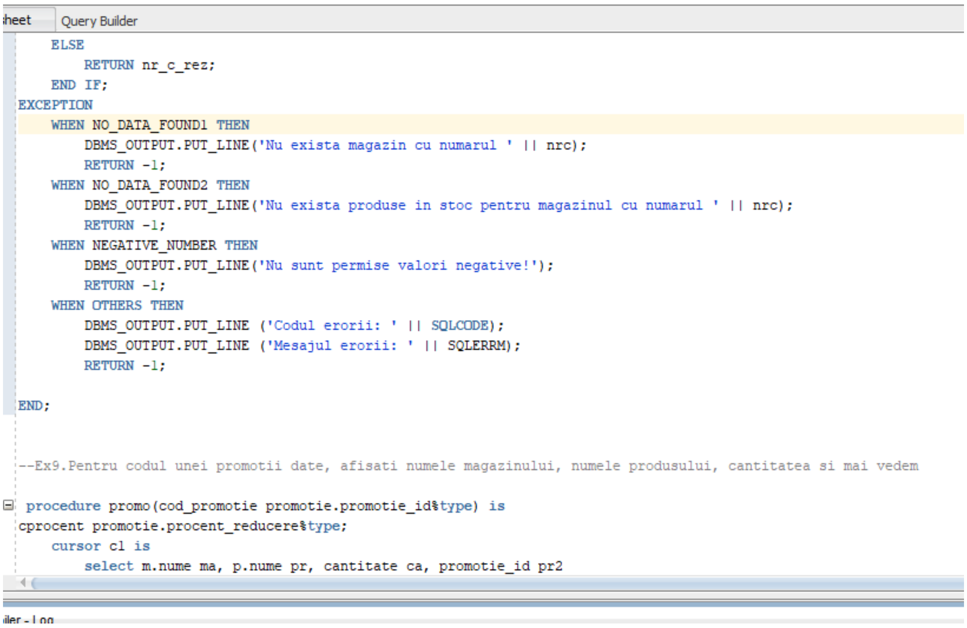
```
        SELECT * BULK COLLECT INTO detalii_comanda
        FROM   comanda
        WHERE  tip_plata = 'card';


    DBMS_OUTPUT.PUT_LINE('promotie: ');
    FOR i in detalii_promotie.first..detalii_promotie.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'Promotia '|| detalii_promotie(i).promotie_id||' s-a desfasurat in anul dat');
        END LOOP;

    DBMS_OUTPUT.PUT_LINE('comanda: ');
    FOR i in detalii_comanda.first..detalii_comanda.last LOOP
            DBMS_OUTPUT.PUT_LINE( 'comanda nr. '|| detalii_comanda(i).comanda_id||' a fost platita cu cardul');
        END LOOP;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
       RAISE_APPLICATION_ERROR(-20000, 'Nu exista comenzi');

    when TOO_MANY_ROWS then
           RAISE_APPLICATION_ERROR(-20001, 'comanda select '||v_localizare || ' nu returneaza nimic ' || ' Exista mai multe reduceri cu acelasi c
  END pro;


--Ex7. Procedura care afiseaza toate feedbackurile date de un client pentru comenzi
procedure proc_feedback(v_client clienti.nume%type)
    is
```



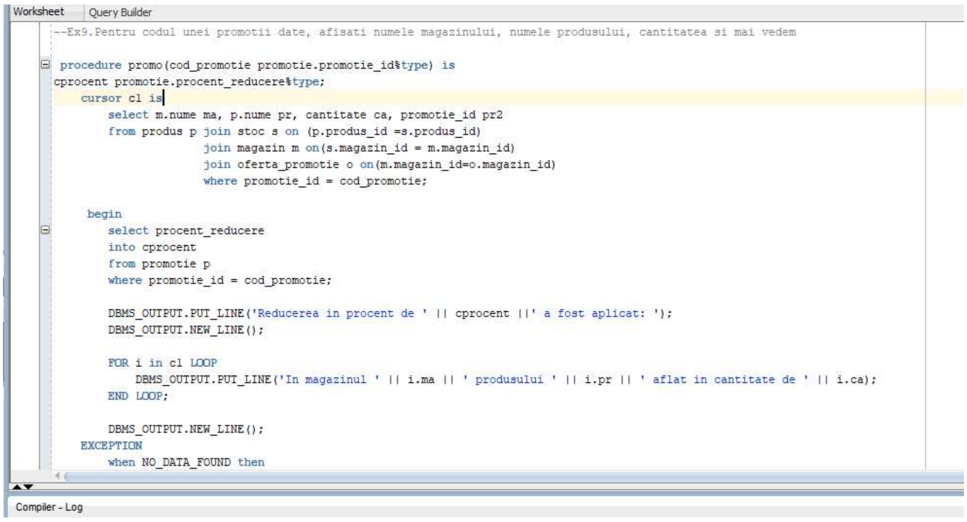
```
--Ex7. Procedura care afiseaza toate feedbackurile date de un client pentru comenzi
procedure proc_feedback(v_client clienti.nume%type)
    is
        cursor c is
            select f.rating_magazin , c.comanda_id
            from comanda c, feedback_comanda f, clienti cl
            where f.comanda_id= c.comanda_id and c.client_id = cl.client_id and cl.nume = v_client;

        v_nr number := 0;

    begin
        for i in c loop
            v_nr := v_nr+1;
            DBMS_OUTPUT.PUT_LINE('Feedback-ul numarul '||v_nr||' la comanda '||i.comanda_id||' are rating '||i.rating_magazin);
        end loop;
                --close c;
end proc_feedback;

--Ex8. Cate produse are un magazin dat ca parametru in stoc?

FUNCTION prod_mag(nrc NUMBER) RETURN NUMBER
IS
    nr_c_rez NUMBER;
    TYPE tbl_idx IS TABLE OF magazin%ROWTYPE INDEX BY PLS_INTEGER;
    aux tbl_idx;
```



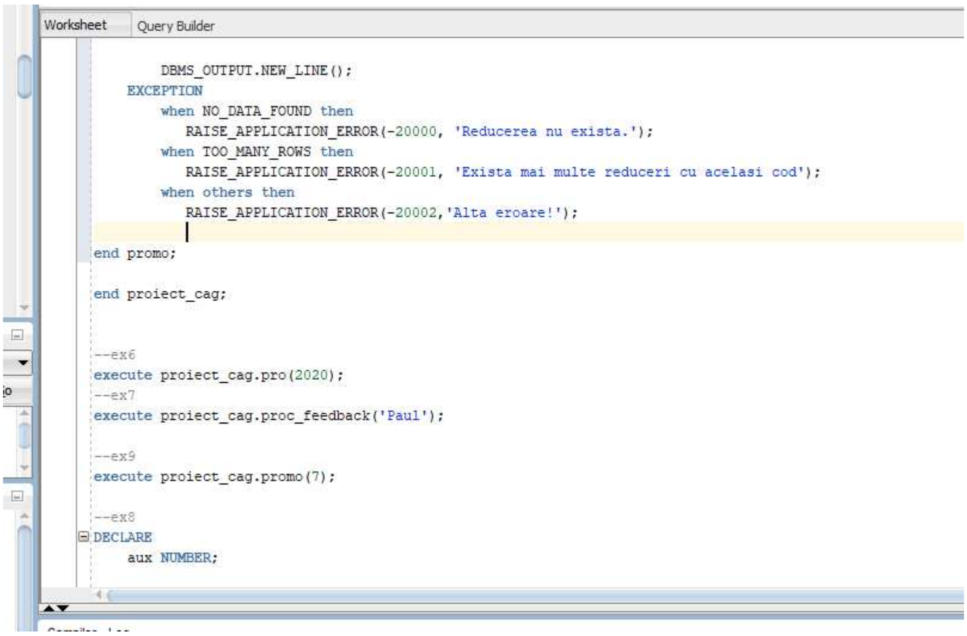
```
    TYPE tbl_idx IS TABLE OF magazin%ROWTYPE INDEX BY PLS_INTEGER;
    aux tbl_idx;
    NEGATIVE_NUMBER EXCEPTION;
    NO_DATA_FOUND1 EXCEPTION;
    NO_DATA_FOUND2 EXCEPTION;
BEGIN
    IF nrc < 0 THEN
        RAISE NEGATIVE_NUMBER;
    END IF;

    SELECT * BULK COLLECT INTO aux FROM magazin WHERE magazin_id = nrc;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    select count(p.produs_id) into nr_c_rez
    from produs p join stoc s on (s.produs_id = p.produs_id)
    join magazin m on (s.magazin_id = m.magazin_id)
    where m.magazin_id = nrc;

    IF nr_c_rez = 0 THEN
        RAISE NO_DATA_FOUND2;
    ELSE
        RETURN nr_c_rez;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
```



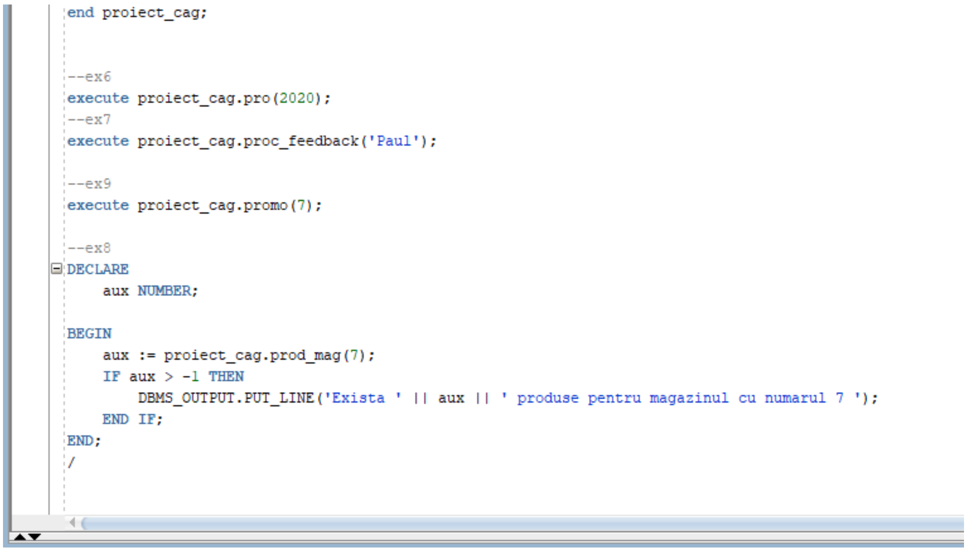
```
    ELSE
        RETURN nr_c_rez;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista magazin cu numarul ' || nrc);
        RETURN -1;
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista produse in stoc pentru magazinul cu numarul ' || nrc);
        RETURN -1;
    WHEN NEGATIVE_NUMBER THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;

END;


--Ex9.Pentru codul unei promotii date, afisati numele magazinului, numele produsului, cantitatea si mai vedem

 procedure promo(cod_promotie promotie.promotie_id%type) is
cprocent promotie.procent_reducere%type;
    cursor c1 is
        select m.nume ma, p.nume pr, cantitate ca, promotie_id pr2
```



```
--Ex9.Pentru codul unei promotii date, afisati numele magazinului, numele produsului, cantitatea si mai vedem

 procedure promo(cod_promotie promotie.promotie_id%type) is
cprocent promotie.procent_reducere%type;
    cursor c1 is
        select m.nume ma, p.nume pr, cantitate ca, promotie_id pr2
        from produs p join stoc s on (p.produs_id =s.produs_id)
                      join magazin m on(s.magazin_id = m.magazin_id)
                      join oferta_promotie o on(m.magazin_id=o.magazin_id)
                      where promotie_id = cod_promotie;

     begin
        select procent_reducere
        into cprocent
        from promotie p
        where promotie_id = cod_promotie;

        DBMS_OUTPUT.PUT_LINE('Reducerea in procent de ' || cprocent ||' a fost aplicat: ');
        DBMS_OUTPUT.NEW_LINE();

        FOR i in c1 LOOP
            DBMS_OUTPUT.PUT_LINE('In magazinul ' || i.ma || ' produsului ' || i.pr || ' aflat in cantitate de ' || i.ca);
        END LOOP;

        DBMS_OUTPUT.NEW_LINE();
    EXCEPTION
        when NO_DATA_FOUND then
```

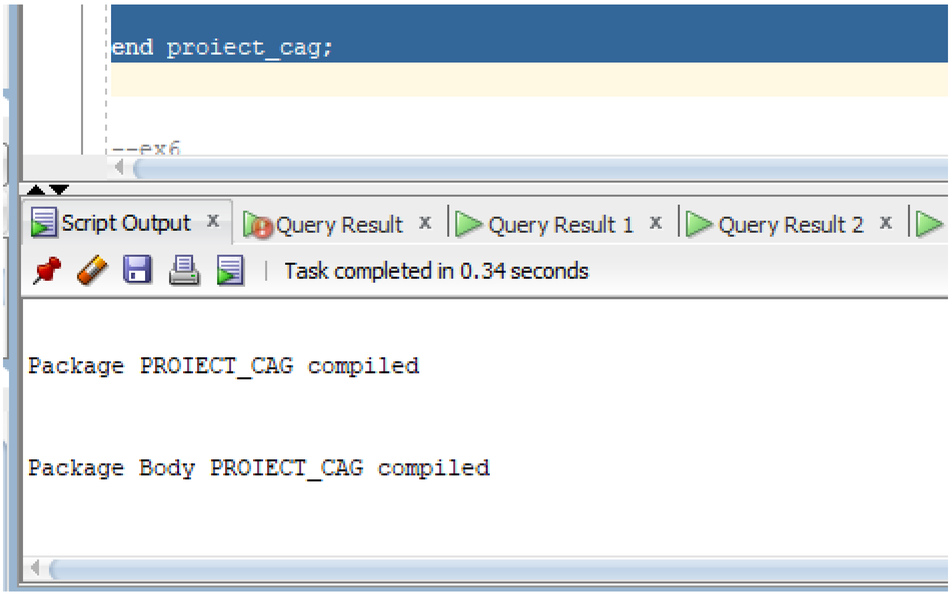


```
        DBMS_OUTPUT.NEW_LINE();
    EXCEPTION
        when NO_DATA_FOUND then
           RAISE_APPLICATION_ERROR(-20000, 'Reducerea nu exista.');
        when TOO_MANY_ROWS then
           RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe reduceri cu acelasi cod');
        when others then
           RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

end promo;

end proiect_cag;


--ex6
execute proiect_cag.pro(2020);
--ex7
execute proiect_cag.proc_feedback('Paul');

--ex9
execute proiect_cag.promo(7);

--ex8
DECLARE
    aux NUMBER;
```



```
end proiect_cag;


--ex6
execute proiect_cag.pro(2020);
--ex7
execute proiect_cag.proc_feedback('Paul');

--ex9
execute proiect_cag.promo(7);

--ex8
DECLARE
     aux NUMBER;

BEGIN
     aux := proiect_cag.prod_mag(7);
     IF aux > -1 THEN
          DBMS_OUTPUT.PUT_LINE('Exista ' || aux || ' produse pentru magazinul cu numarul 7 ');
     END IF;
END;
/
```

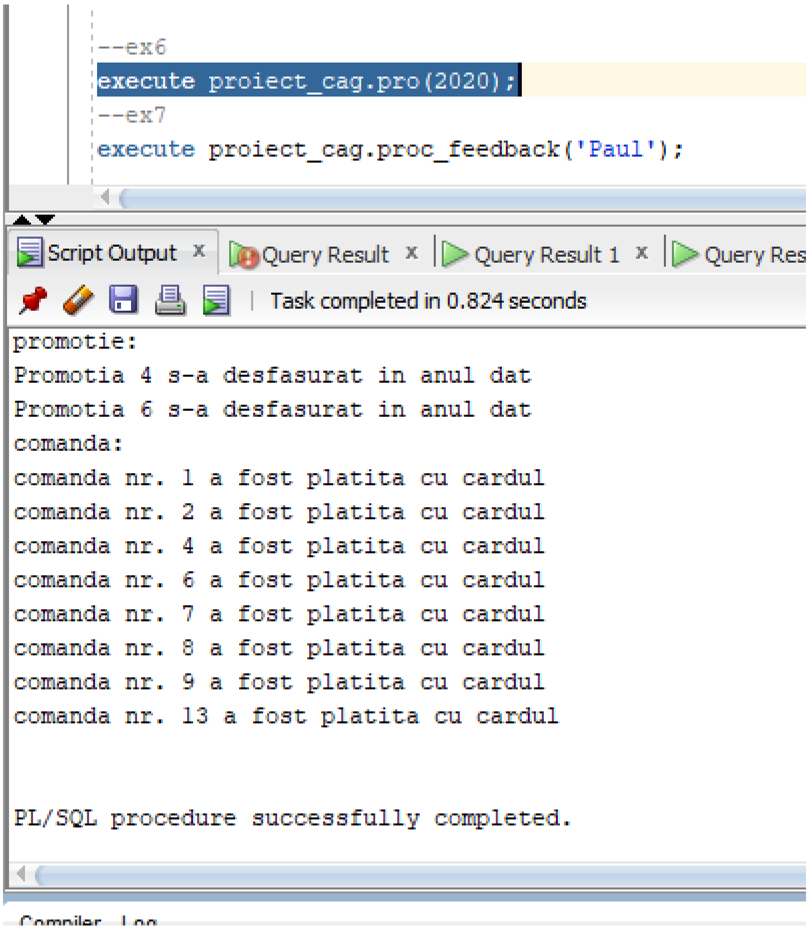
Compilare:



```
end proiect_cag;

--ex6
```

```
Package PROIECT_CAG compiled


Package Body PROIECT_CAG compiled
```

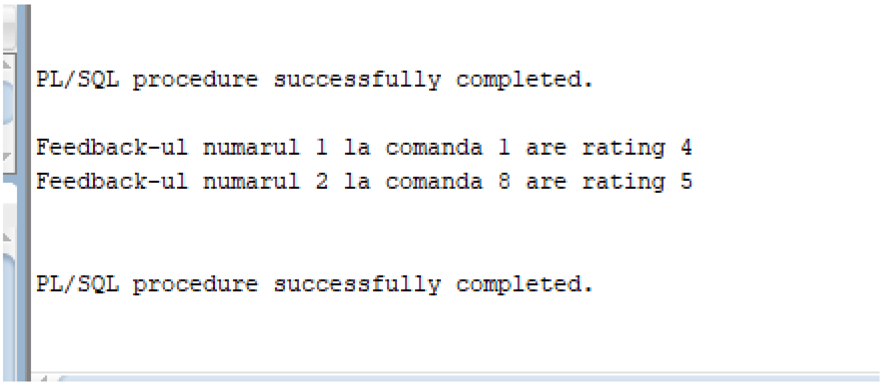
Ex6:



```
--ex6
execute proiect_cag.pro(2020);
--ex7
execute proiect_cag.proc_feedback('Paul');
```

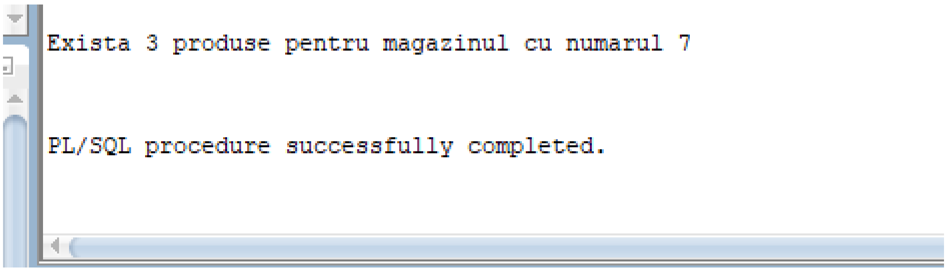
```
promotie:
Promotia 4 s-a desfasurat in anul dat
Promotia 6 s-a desfasurat in anul dat
comanda:
comanda nr. 1 a fost platita cu cardul
comanda nr. 2 a fost platita cu cardul
comanda nr. 4 a fost platita cu cardul
comanda nr. 6 a fost platita cu cardul
comanda nr. 7 a fost platita cu cardul
comanda nr. 8 a fost platita cu cardul
comanda nr. 9 a fost platita cu cardul
comanda nr. 13 a fost platita cu cardul


PL/SQL procedure successfully completed.
```

Ex7:



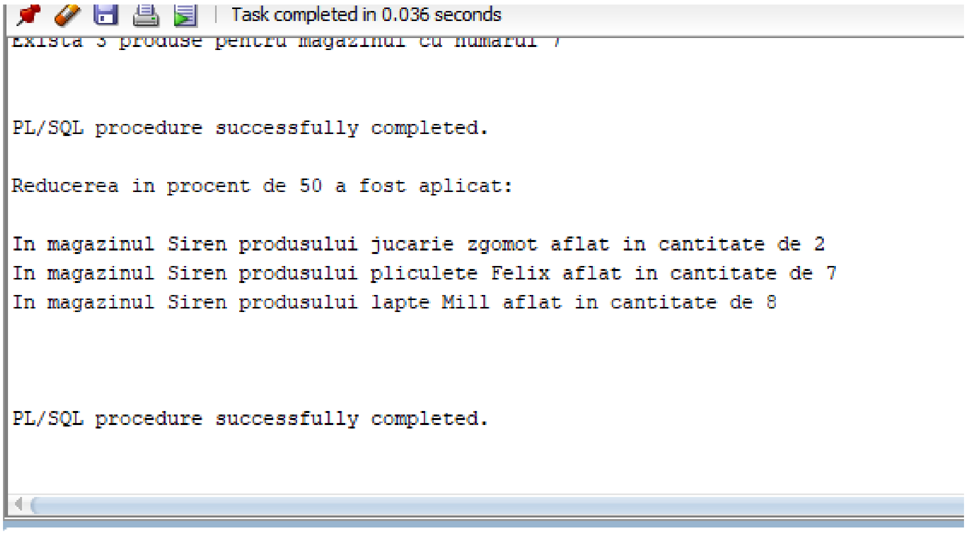
```
PL/SQL procedure successfully completed.

Feedback-ul numarul 1 la comanda 1 are rating 4
Feedback-ul numarul 2 la comanda 8 are rating 5


PL/SQL procedure successfully completed.
```

Ex8:



```
Exista 3 produse pentru magazinul cu numarul 7


PL/SQL procedure successfully completed.
```

Ex9:



```
Exista 3 produse pentru magazinul cu numarul 7


PL/SQL procedure successfully completed.

Reducerea in procent de 50 a fost aplicat:

In magazinul Siren produsului jucarie zgomot aflat in cantitate de 2
In magazinul Siren produsului pliculete Felix aflat in cantitate de 7
In magazinul Siren produsului lapte Mill aflat in cantitate de 8



PL/SQL procedure successfully completed.
```