

PROIECT SGBD

“Magazin alimentar”

Nume: Burdulea Diana-Ana-Maria

Facultate: CSIE

Specializare: Informatica Economica

Seria: C

Grupa: 1048

I. Descrierea problemei

Baza mea de date este compusa din elementele pe care le-am considerat esentiale in vederea gestionarii unui magazin alimentar. Tabelele pe care le-am dezvoltat sunt: functii, angajati, comenzi, client, depozit, rand_comenzi, produse, furnizor, factura. Pentru inceput, orice afacere nu poate functiona fara cele 3 elemente cheie: angajati, client si obiectele/serviciile comercializate. Astfel, aceste 3 tabele se regasesc si in baza mea de date.

Tabela “**Functii**” are ca si cheie primara “id_functie” si retine denumirile functiilor angajatilor dar si limitele salariului(minim si maxim). Tabela “**Functii**” este legata de urmatoarea tabela “**Angajati**” printr-o legatura “*one to many*” (un angajat poate avea o singura functie, in timp ce o functie poate fi detinuta de mai multi angajati). Tabela “**Angajati**” are ca si cheie primara “id_angajat” si prezinta cateva caracteristici ale angajatilor precum nume, prenume, adresa etc.

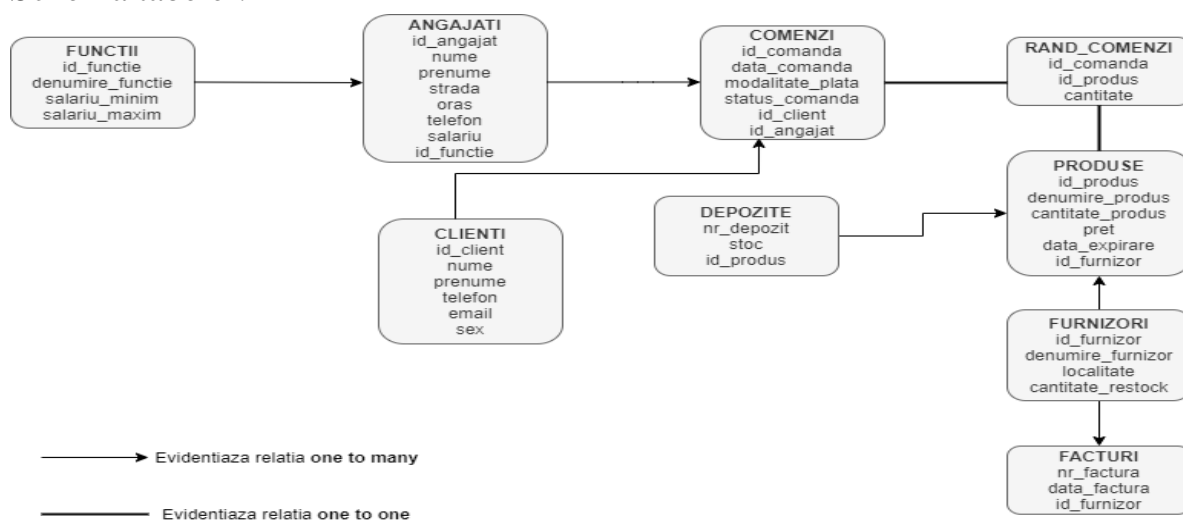
Tabela “**Clienti**” contine caracteristici specifice precum nume, prenume, telefon, email, id_client care are rol de cheie primara.

Cheile primare ale tabelelor “**Angajati**” si “**Clienti**” formeaza foreign keys pentru tabela “**Comenzi**”, iar cheile primare ale tabelelor “**Comenzi**” si “**Produse**” formeaza foreign keys pentru tabela “**Rand_comenzi**”, tabela care este legata de celelalte doua printr-o relatie “*one to one*”.

Un alt element pe care l-am considerat a fi important a fost existenta unui “**Furnizor**” care urmeaza sa mentina stocurile incarcate astfel determinand afacerea sa continue. La fiecare achizitie a produselor, “**Furnizorul**” poate emite una sau mai multe “**Facturi**”.

Toate alimentele vor fi stocate intr-un “**Depozit**” pentru gestionarea mai facila.

II. Schema tabelei:



SCHEMA FINALA A BAZEI DE DATE "MAGAZIN ALIMENTAR"

III. Crearea tabelelor

- Tabela „**Functii**”

```
CREATE TABLE bd_functii
(
    id_functie NUMBER(6),
    CONSTRAINT functie_pk PRIMARY KEY(id_functie),
    denumire_functie VARCHAR2(40) NOT NULL,
    salariul_minim NUMBER(4),
    salariul_maxim NUMBER(5)
);
```

- Tabela „**Angajati**”

```
CREATE TABLE bd_angajati
(
    id_angajat NUMBER(4),
    CONSTRAINT angajat_pk PRIMARY KEY(id_angajat),
    nume VARCHAR2(30) NOT NULL,
    prenume VARCHAR2(30) NOT NULL,
    strada VARCHAR2(30),
    oras VARCHAR2(50),
    telefon VARCHAR2(10) UNIQUE,
    salariu NUMBER(5),
    id_functie NUMBER(6),
    CONSTRAINT angajat_fk FOREIGN KEY(id_functie) REFERENCES bd_functii(id_functie)
);
```

- Tabela „**Clienti**”

```
CREATE TABLE bd_clienti
(
    id_client NUMBER(6),
    CONSTRAINT client_pk PRIMARY KEY(id_client),
    nume VARCHAR2(50) NOT NULL,
    prenume VARCHAR2(50) NOT NULL,
    telefon VARCHAR2(10) UNIQUE,
    email VARCHAR2(50) NOT NULL,
    sex VARCHAR2(1)
);
ALTER TABLE bd_clienti
    ADD CONSTRAINT email_uq UNIQUE(email);
ALTER TABLE bd_clienti
    ADD CONSTRAINT ck_email CHECK (email LIKE '%@%.%');
```

- Tabela „**Produse**”

```
CREATE TABLE bd_produse
(
    id_produs NUMBER(6),
    CONSTRAINT produs_pk PRIMARY KEY(id_produs),
    denumire_produs VARCHAR2(50) NOT NULL,
    categorii_produs VARCHAR2(50),
    pret NUMBER(8,2),
```

```

        data_expirare DATE,
        id_furnizor NUMBER(5)
    );
ALTER TABLE bd_produce
ADD CONSTRAINT pret_ck CHECK(pret>=0);

```

○ Tabela „Comenzi”

```

CREATE TABLE bd_comenzi
(
    id_comanda NUMBER(6),
    CONSTRAINT comanda_pk PRIMARY KEY(id_comanda),
    data_comanda DATE,
    modalitate_plata VARCHAR2(20),
    status_comanda VARCHAR2(50),
    id_angajat NUMBER(4),
    CONSTRAINT c_angajat_fk FOREIGN KEY(id_angajat) REFERENCES bd_angajati(id_
angajat),
    id_client NUMBER(6),
    CONSTRAINT c_client_fk FOREIGN KEY(id_client) REFERENCES bd_clienti(id_cli
ent)
);

```

○ Tabela „Rand_comenzi”

```

CREATE TABLE bd_rand_comenzi
(
    id_comanda NUMBER(6),
    id_produș NUMBER(6),
    cantitate_comandata NUMBER(8)
);
ALTER TABLE bd_rand_comenzi
ADD CONSTRAINT rand_pk PRIMARY KEY(id_comanda, id_produș);
ALTER TABLE bd_rand_comenzi
ADD CONSTRAINT rand_com_com_fk FOREIGN KEY(id_comanda) REFERENCES bd_comenzi(i
d_comanda);
ALTER TABLE bd_rand_comenzi
ADD CONSTRAINT rand_com_prod_fk FOREIGN KEY(id_produș) REFERENCES bd_produce(i
d_produș);

```

○ Tabela „Furnizori”

```

CREATE TABLE bd_furnizori
(
    id_furnizor NUMBER(5),
    CONSTRAINT furnizor_pk PRIMARY KEY(id_furnizor),
    denumire_furnizor VARCHAR2(50),
    localitate VARCHAR2(50),
    cantitate_restock NUMBER(5)
);
ALTER TABLE bd_produce
ADD CONSTRAINT furn_fk FOREIGN KEY(id_furnizor) REFERENCES bd_furnizori(id_fur
nizor);

```

○ Tabela „Depozite”

```

CREATE TABLE bd_depozite
(
    nr_depozit NUMBER(4),

```

```

        CONSTRAINT depozit_pk PRIMARY KEY(nr_depozit),
        stoc NUMBER(5),
        id_produș NUMBER(6),
        CONSTRAINT produs_dep_fk FOREIGN KEY(id_produș) REFERENCES bd_produșe(id_p
roduș)
    );

```

○ Tabela „Facturi”

```

CREATE TABLE bd_facturi
(
    nr_factura NUMBER(6),
    CONSTRAINT factura_pk PRIMARY KEY(nr_factura),
    data_factura DATE,
    id_furnizor NUMBER(5),
    CONSTRAINT factura_fk FOREIGN KEY(id_furnizor) REFERENCES bd_furnizori(id_
furnizor)
);

```

IV. Adaugarea datelor in tabele

--functii

```

INSERT INTO functii (id_functie, denumire_functie, salariul_minim, salariul_maxim)
VALUES('1', 'Vanzator', '1000', '4000');
INSERT INTO bd_functii (id_functie, denumire_functie, salariul_minim, salariul_maxim)
VALUES('2', 'Manager magazin', '2000', '7000');
INSERT INTO bd_functii (id_functie, denumire_functie, salariul_minim, salariul_maxim)
VALUES('3', 'Ingrijitor', '1000', '2500');
INSERT INTO bd_functii (id_functie, denumire_functie, salariul_minim, salariul_maxim)
VALUES('4', 'Asistent manager', '2000', '3500');
INSERT INTO bd_functii (id_functie, denumire_functie, salariul_minim, salariul_maxim)
VALUES('5', 'Contabil', '2000', '4000');
INSERT INTO bd_functii (id_functie, denumire_functie, salariul_minim, salariul_maxim)
VALUES('6', 'Manager marketing', '1000', '4000');

```

--angajati

```

INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)
VALUES('1', 'Trandafir', 'Robert', 'Strada Ciresului', 'Mangalia', '0712345678', '2500', '1
');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)
VALUES('2', 'Dobre', 'Alina', 'Strada Dumbravei', 'Bucuresti', '0722345678', '2700', '1');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)
VALUES('3', 'Eftimie', 'Mircea', 'Strada Grivitei', 'Bucuresti', '0732345678', '3500', '6')
;
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)
VALUES('4', 'Teodorescu', 'Andreea', 'Strada Zorilor', 'Bucuresti', '0742345678', '3400', '
4');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)
VALUES('5', 'Cristea', 'Cornel', 'Strada Oituz', 'Slobozia', '0752345678', '2500', '3');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)
VALUES('6', 'Paraschiv', 'Monica', 'Strada Dorobantilor', 'Bucuresti', '0762345678', '2300
', '3');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_funct
ie)

```

```
VALUES('7', 'Paduraru', 'Victor', 'Strada Garii', 'Calarasi', '0772345678', '6000', '2');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_functie)
VALUES('8', 'Rusu', 'Flavia', 'Strada Traian', 'Bucuresti', '0782345678', '4000', '5');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_functie)
VALUES('9', 'Stroe', 'Ioana', 'Strada Teilor', 'Pitesti', '0792345678', '3250', '4');
INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id_functie)
VALUES('10', 'Moraru', 'Cristina', 'Strada Fagaras', 'Bucuresti', '0731345678', '2750', '1');
);
```

--clienti

```
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('10', 'Chirila', 'Ileana', '0731234567', 'chirila@gmail.com', 'F');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('11', 'Nedelcu', 'Tudor', '0732234567', 'nedelcu@gmail.com', 'M');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('12', 'Lazar', 'Marian', '0733234567', 'lazar@gmail.com', 'M');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('13', 'Serban', 'Daniel', '0734234567', 'serban@gmail.com', 'F');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('14', 'Sava', 'Corina', '0735234567', 'sava@gmail.com', 'F');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('15', 'Sava', 'Gabriel', '0736234567', 'gsava@gmail.com', 'M');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('16', 'Ivascu', 'Rodica', '0737234567', 'ivascu@gmail.com', 'F');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('17', 'Iosif', 'Adelin', '0738234567', 'iosif@gmail.com', 'M');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('18', 'Militaru', 'George', '0739234567', 'militaru@gmail.com', 'M');
INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)
VALUES('19', 'Badea', 'Silvia', '0749234567', 'badea@gmail.com', 'F');
```

--produse

```
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('100', 'Chipsuri Lays sare', 'Chipsuri', '5', to_date('20.06.2021', 'dd.mm.yyyy'), '1234');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('101', 'Ciocolata Milka', 'Ciocolata', '7', to_date('30.07.2021', 'dd.mm.yyyy'), '1235');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('102', 'Chipsuri Pringles sare', 'Chipsuri', '8', to_date('10.02.2022', 'dd.mm.yyyy'), '1236');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('103', 'Cereale Cheerios', 'Cereale', '10', to_date('15.06.2021', 'dd.mm.yyyy'), '1237');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('104', 'Musli Vitalis ciocolata', 'Cereale', '12', to_date('20.02.2021', 'dd.mm.yyyy'), '1245');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('105', 'Paine Boromir', 'Panificatie', '4', to_date('20.12.2020', 'dd.mm.yyyy'), '1238');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
VALUES('106', 'Lapte ZUZU', 'Lactate', '6', to_date('31.12.2020', 'dd.mm.yyyy'), '1239');
INSERT INTO bd_produse(id_produs, denumire_produs, categorie_produs, pret, data_expirare, id_furnizor)
```

```

VALUES('107', 'Telemea Napolact', 'Lactate', '13', to_date('10.01.2021', 'dd.mm.yyyy'), '1240');
INSERT INTO bd_produce(id_produș, denumire_produș, categoriē_produș, pret, data_expirare, id_furnizor)
VALUES('108', 'Coca-Cola Zero', 'Bauturi', '6', to_date('20.06.2023', 'dd.mm.yyyy'), '1241');
INSERT INTO bd_produce(id_produș, denumire_produș, categoriē_produș, pret, data_expirare, id_furnizor)
VALUES('109', 'Paste Barilla', 'Paste', '12', to_date('20.06.2021', 'dd.mm.yyyy'), '1242');
;
INSERT INTO bd_produce(id_produș, denumire_produș, categoriē_produș, pret, data_expirare, id_furnizor)
VALUES('110', 'Amestec legume Edenia', 'Congelate', '15', to_date('20.01.2021', 'dd.mm.yyyy'), '1243');
INSERT INTO bd_produce(id_produș, denumire_produș, categoriē_produș, pret, data_expirare, id_furnizor)
VALUES('111', 'Mix smoothie Edenia', 'Congelate', '18', to_date('20.06.2021', 'dd.mm.yyyy'), '1243');
INSERT INTO bd_produce(id_produș, denumire_produș, categoriē_produș, pret, data_expirare, id_furnizor)
VALUES('112', 'Pizza Ristorante', 'Congelate', '12', to_date('20.06.2021', 'dd.mm.yyyy'), '1245');
INSERT INTO bd_produce(id_produș, denumire_produș, categoriē_produș, pret, data_expirare, id_furnizor)
VALUES('113', 'Piept pui Fragedo', 'Carne', '30', to_date('20.06.2021', 'dd.mm.yyyy'), '1244');

```

--furnizori

```

INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1234', 'Lays', 'București', '50');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1235', 'Milka', 'Brasov', '35');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1236', 'Pringles', 'București', '20');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1237', 'Nestle', 'Constanta', '20');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1238', 'Boromir', 'Ramnicu Valcea', '60');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1239', 'Zuzu', 'Focsani', '20');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1240', 'Napolact', 'Cluj Napoca', '15');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1241', 'Pepsico', 'București', '45');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1242', 'Barilla', 'București', '50');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1243', 'Edenia', 'București', '12');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1244', 'Fragedo', 'București', '20');
INSERT INTO bd_furnizori(id_furnizor, denumire_furnizor, localitate, cantitate_restock)
VALUES('1245', 'Dr Oetker', 'București', '30');

```

--depozite

```

INSERT INTO bd_depozite(nr_depozit, stoc, id_produș)
VALUES('1', '50', '100');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produș)
VALUES('2', '23', '101');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produș)
VALUES('3', '5', '102');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produș)
VALUES('4', '2', '103');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produș)
VALUES('5', '17', '104');

```

```

INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('6', '9', '105');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('7', '36', '106');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('8', '23', '107');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('9', '18', '108');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('10', '2', '109');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('11', '8', '110');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('12', '10', '111');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('13', '4', '112');
INSERT INTO bd_depozite(nr_depozit, stoc, id_produs)
VALUES('14', '24', '113');

```

--facturi

```

INSERT INTO bd_facturi(nr_factura, data_factura, id_furnizor)
VALUES('1111', to_date('16.06.2020', 'dd.mm.yyyy'), '1');
INSERT INTO bd_facturi(nr_factura, data_factura, id_furnizor)
VALUES('1112', to_date('16.06.2020', 'dd.mm.yyyy'), '2');
INSERT INTO bd_facturi(nr_factura, data_factura, id_furnizor)
VALUES('1113', to_date('16.06.2020', 'dd.mm.yyyy'), '5');
INSERT INTO bd_facturi(nr_factura, data_factura, id_furnizor)
VALUES('1114', to_date('16.06.2020', 'dd.mm.yyyy'), '7');

```

--comenzi

```

INSERT INTO bd_comenzi(id_comanda, data_comanda, modalitate_plata, status_comanda, id_angaj
at, id_client )
VALUES('1', to_date('01.12.2020', 'dd.mm.yyyy'), 'cash', 'procesare', '1', '10');
INSERT INTO bd_comenzi(id_comanda, data_comanda, modalitate_plata, status_comanda, id_angaj
at, id_client )
VALUES('2', to_date('12.06.2020', 'dd.mm.yyyy'), 'card', 'terminat', '1', '12');
INSERT INTO bd_comenzi(id_comanda, data_comanda, modalitate_plata, status_comanda, id_angaj
at, id_client )
VALUES('3', to_date('23.10.2020', 'dd.mm.yyyy'), 'card', 'terminat', '5', '14');
INSERT INTO bd_comenzi(id_comanda, data_comanda, modalitate_plata, status_comanda, id_angaj
at, id_client )
VALUES('4', to_date('19.02.2020', 'dd.mm.yyyy'), 'cash', 'terminat', '2', '13');
INSERT INTO bd_comenzi(id_comanda, data_comanda, modalitate_plata, status_comanda, id_angaj
at, id_client )
VALUES('5', to_date('02.12.2020', 'dd.mm.yyyy'), 'cash', 'terminat', '4', '12');
INSERT INTO bd_comenzi(id_comanda, data_comanda, modalitate_plata, status_comanda, id_angaj
at, id_client )
VALUES('6', to_date('18.08.2020', 'dd.mm.yyyy'), 'cash', 'terminat', '1', '18');

```

--rand comenzi

```

INSERT INTO bd_rand_comenzi(id_comanda, id_produs, cantitate_comandata)
VALUES('1', '100', '3');
INSERT INTO bd_rand_comenzi(id_comanda, id_produs, cantitate_comandata)
VALUES('1', '101', '1');
INSERT INTO bd_rand_comenzi(id_comanda, id_produs, cantitate_comandata)
VALUES('2', '105', '4');
INSERT INTO bd_rand_comenzi(id_comanda, id_produs, cantitate_comandata)
VALUES('3', '100', '5');
INSERT INTO bd_rand_comenzi(id_comanda, id_produs, cantitate_comandata)
VALUES('1', '110', '1');

```


-----OPERATII PL/SQL-----

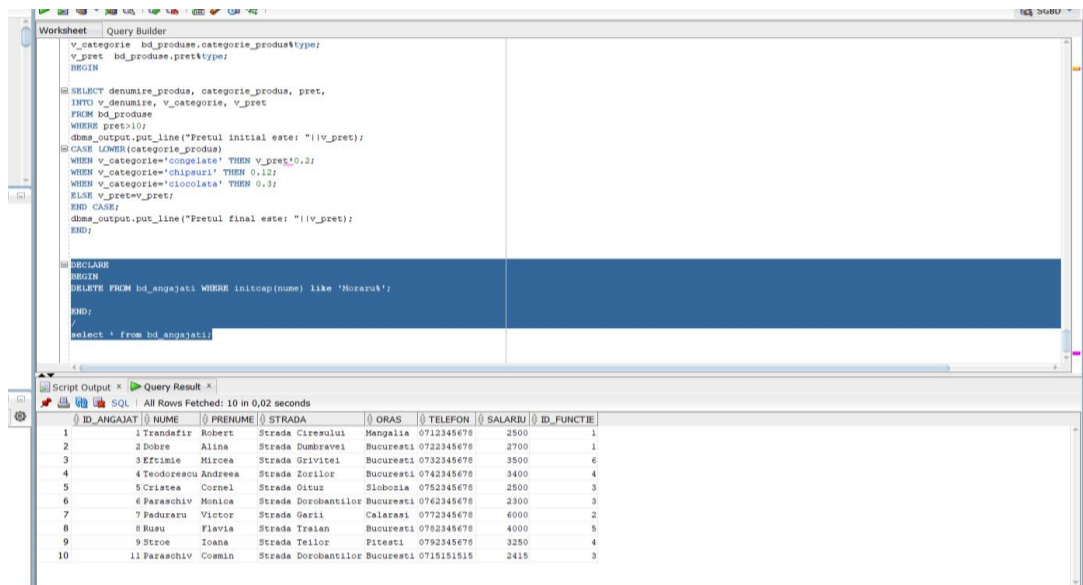
Partea 1. Interactiunea cu serverul Oracle

1. Sa se stearga angajatul cu numele "Moraru".

```
DECLARE
BEGIN
DELETE FROM bd_angajati WHERE initcap(nume) like 'Moraru%';

END;
/

select * from bd_angajati;
```



2. Sa se adauge o noua inregistrare in tabela client prin introducerea valorilor cu ajutorul variabilelor de substitutie.

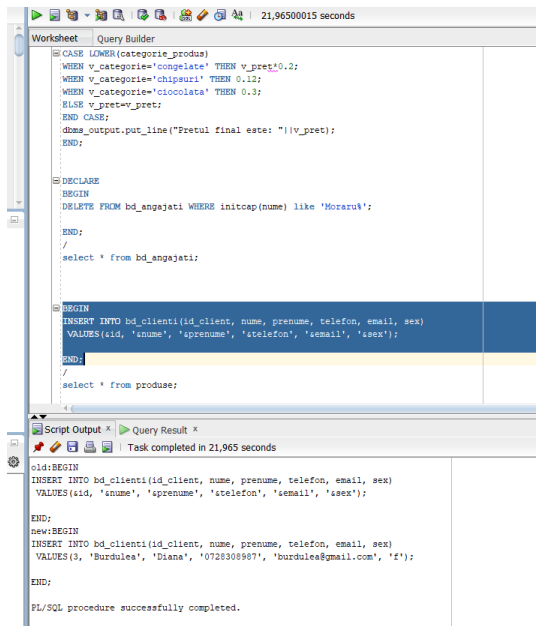
```
BEGIN

INSERT INTO bd_clienti(id_client, nume, prenume, telefon, email, sex)

VALUES(&id, '&nume', '&prenume', '&telefon', '&email', '&sex');

END;
/

select * from produse;
```



Partea 2. Structuri de control. Cursori impliciti. Cursori expliciti.

1. Comparati pretul produselor avand id-ul citit de la tastatura, cu o variabila egala cu 10 si afisati un mesaj(daca pretul este mai mare decat 10, se afiseaza mesajul „Produsul este scump”, daca pretul este egal cu 10, se afiseaza mesajul „Produsul este obisnuit” si daca pretul este mai mic decat 10 se afiseaza mesajul „Produsul este ieftin”).

declare

variabila number(5):=10;

v_pret bd_produce.pret%type;

begin

select pret into v_pret from bd_produce where id_produce=&p;

if v_pret<variabila

then dbms_output.put_line('Produsul este ieftin');

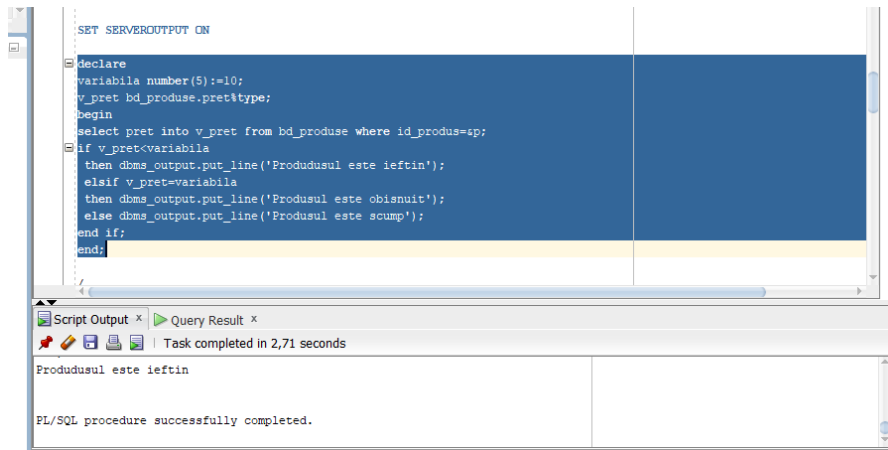
elsif v_pret=variabila

then dbms_output.put_line('Produsul este obisnuit');

else dbms_output.put_line('Produsul este scump');

end if;

end;

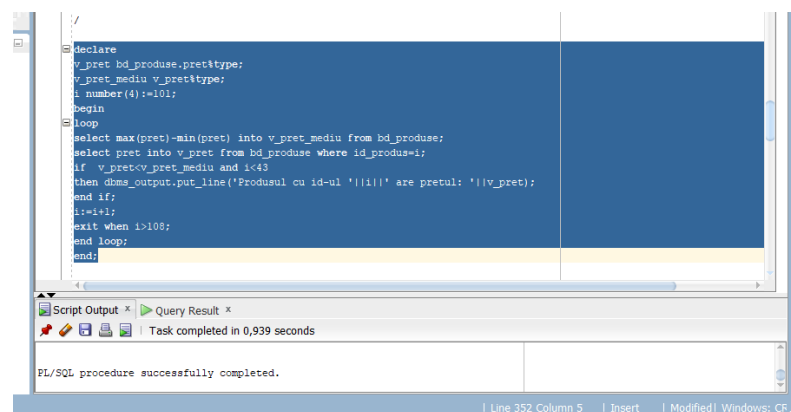


2. Sa se afiseze in ordine produsele cu id-ul intre 101 si 108 daca pretul acestora este mai mic decat diferenta dintre cel mai mare si cel mai mic pret dintre produsele existente.

```

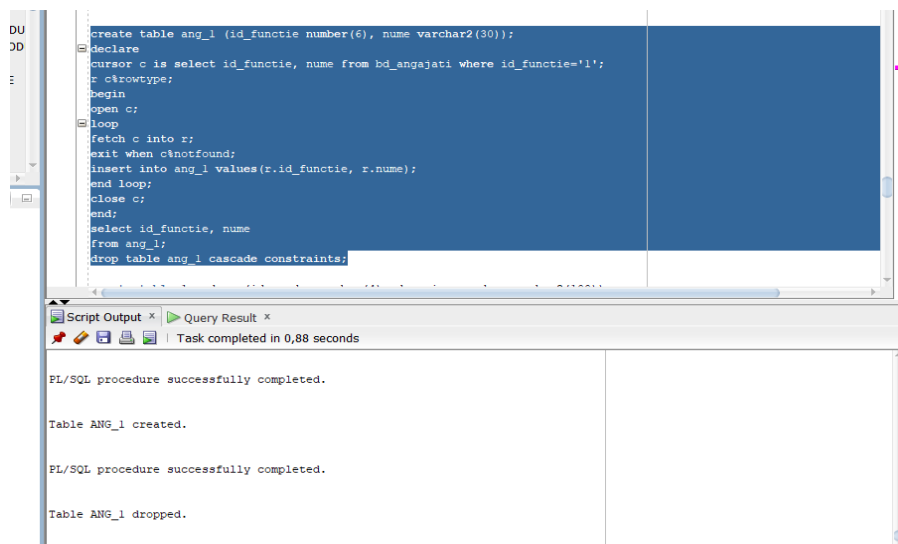
declare
v_pret bd_produce.pret%type;
v_pret_mediu v_pret%type;
i number(4):=101;
begin
loop
select max(pret)-min(pret) into v_pret_mediu from bd_produce;
select pret into v_pret from bd_produce where id_produce=i;
if v_pret<v_pret_mediu and i<43
then dbms_output.put_line('Produsul cu id-ul '||i||' are pretul: '||v_pret);
end if;
i:=i+1;
exit when i>108;
end loop;
end;

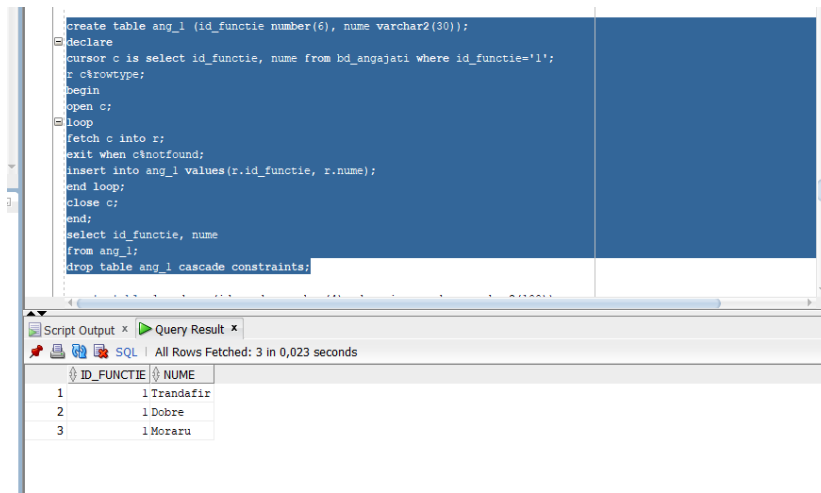
```



3. Sa se creeze o tabela noua in care sa se insereze toti angajatii din tabela bd_angajati care au id_functie=1.

```
create table ang_1 (id_functie number(6), nume varchar2(30));  
  
declare  
  
cursor c is select id_functie, nume from bd_angajati where id_functie='1';  
r c%rowtype;  
  
begin  
open c;  
  
loop  
fetch c into r;  
exit when c%notfound;  
insert into ang_1 values(r.id_functie, r.nume);  
end loop;  
  
close c;  
  
end;  
  
select id_functie, nume  
from ang_1;  
  
drop table ang_1 cascade constraints;
```





4. Sa se scada pretul tuturor chipsurilor cu 10%. Afisati numarul modificarilor daca exista.

```
declare
```

```
begin
```

```
update bd_produce set pret=pret-0.1*pret where id_produus in(select id_produus from bd_produce where
categorie_produus='Chipsuri');
```

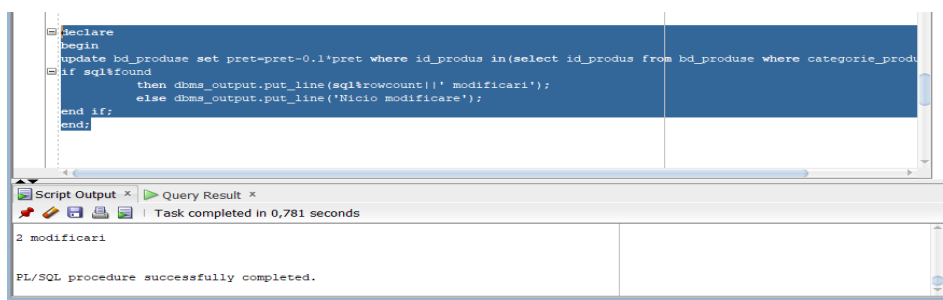
```
if sql%found
```

```
then dbms_output.put_line(sql%rowcount||' modificari');
```

```
else dbms_output.put_line('Nicio modificare');
```

```
end if;
```

```
end;
```



5. Sa se afiseze denumirea produsului si data expirarii pentru produsele care au fost valabile pana la finalul anului 2020.

```
DECLARE
```

```
CURSOR c IS SELECT denumire_produus, data_expirare FROM bd_produce WHERE
data_expirare<=TO_DATE('31.12.2020', 'DD.MM.YYYY');
```

```
v_denumire bd_produce.denumire_produus%type;
```

```
v_data bd_produce.data_expirare%type;
```

```
BEGIN
```

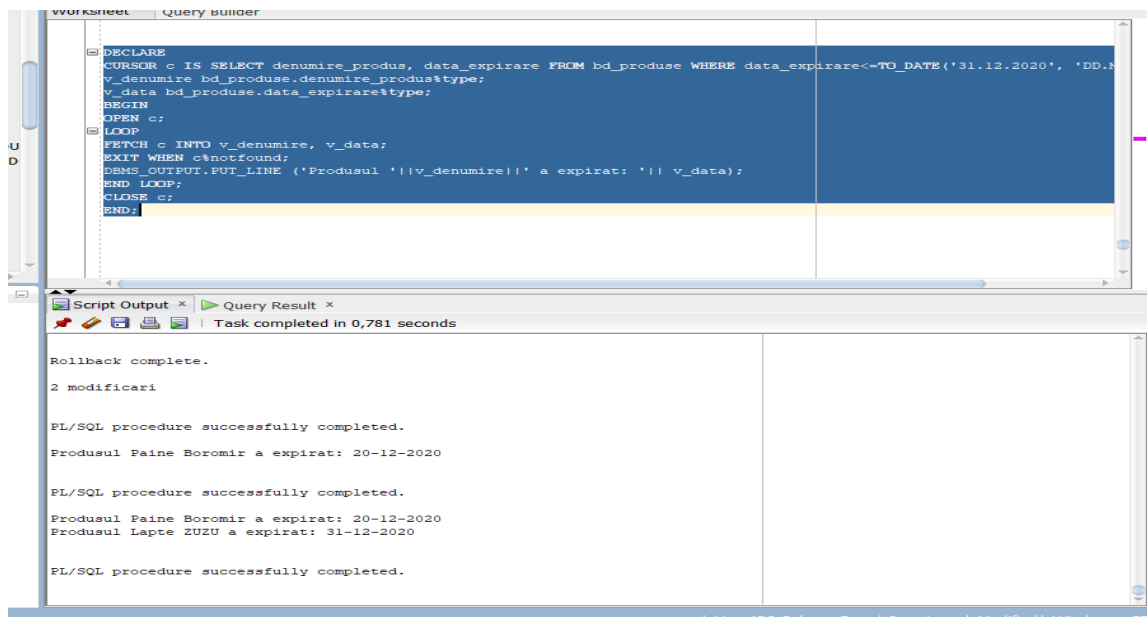
```
OPEN c;
```

```
LOOP
```

```

FETCH c INTO v_denumire, v_data;
EXIT WHEN c%notfound;
DBMS_OUTPUT.PUT_LINE ('Produsul '||v_denumire||' a expirat: '|| v_data);
END LOOP;
CLOSE c;
END;

```



Partea 2. Structuri de control. Cursori impliciti. Cursori expliciti. Exceptii

1. Sa se creasca cantitatea de restock cu 50 de bucati pentru id-ul furnizorului primit de la tastatura. Sa se ridice o exceptie in cazul in care id-ul este introdus gresit sau nu exista.

```

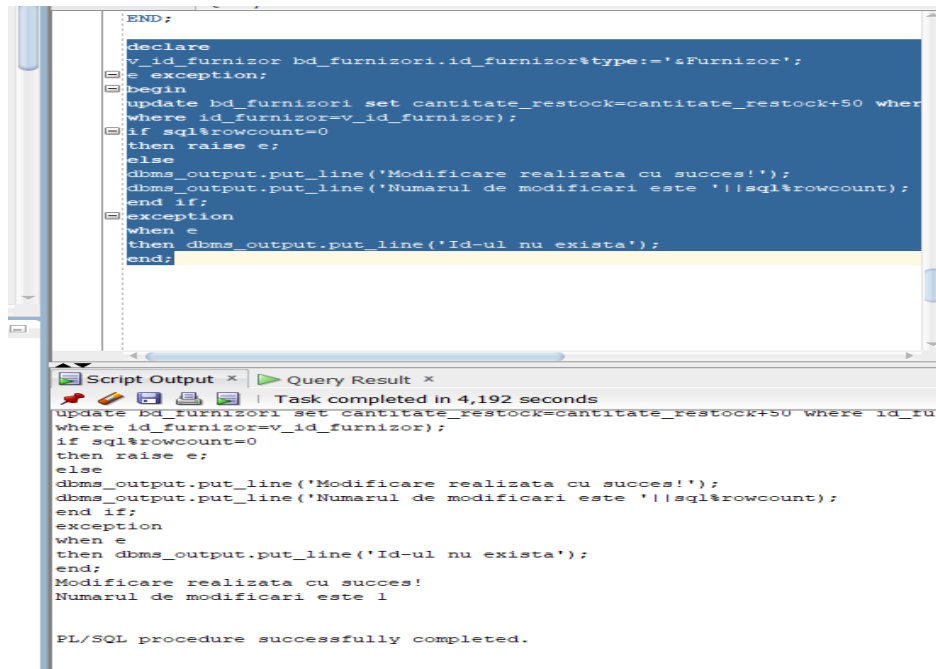
declare
v_id_furnizor bd_furnizori.id_furnizor%type:= '&Furnizor';
e exception;
begin
update bd_furnizori set cantitate_restock=cantitate_restock+50 where id_furnizor in(select id_furnizor from
bd_furnizori
where id_furnizor=v_id_furnizor);
if sql%rowcount=0
then raise e;
else
dbms_output.put_line('Modificare realizata cu succes!');

```

```

dbms_output.put_line('Numarul de modificari este '||sql%rowcount);
end if;
exception
when e
then dbms_output.put_line('Id-ul nu exista');
end;

```



2. Sa se afiseze numele, strada si orasul unui angajat dat de la tastatura. Daca numele nu exista sau sunt mai multi angajati cu acest nume, evidentiati printr-o eroare.

```

DECLARE
v_strada bd_angajati.strada%type;
v_oras bd_angajati.oras%type;
v_nume bd_angajati.nume%type;
BEGIN
select strada, oras, nume into v_strada, v_oras, v_nume from bd_angajati where nume='&n';
dbms_output.put_line('Angajatul '||v_nume||' locuieste in orasul '||v_oras||' pe strada '||v_strada);
EXCEPTION
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE('Exista mai multi salariati cu acest nume!');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu exista un salariat cu acest nume!');
END;

```

```

--sa se afiseze strada si orasul angajatului cu numele 'Paraschiv'. Sa se trateze erorile
DECLARE
v_strada bd_angajati.strada%type;
v_oras bd_angajati.oras%type;
v_nume bd_angajati.nume%type;
BEGIN
select strada, oras, nume into v_strada, v_oras, v_nume from bd_angajati where nume='in';
dbms_output.put_line('Angajatul '||v_nume||' locuieste in orasul '||v_oras||' pe strada '||v_strada);
EXCEPTION
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE('Exista mai multi salariatii cu numele '||v_nume||');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu exista un salariat cu numele '||v_nume||');
END;

```

Script Output x

Task completed in 4,545 seconds

```

WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu exista un salariat cu numele '||v_nume||');
END;
Angajatul Padurarul locuieste in orasul Calarasi pe strada Strada Garii

PL/SQL procedure successfully completed.

```

```

INSERT INTO bd_angajati(id_angajat, nume, prenume, strada, oras, telefon, salariu, id functie)
VALUES('11', 'Paraschiv', 'Cosmin', 'Strada Dorobantilor', 'Bucuresti', '0715151515', '2300', '3');

--sa se afiseze strada si orasul angajatului cu numele 'Paraschiv'. Sa se trateze erorile
DECLARE
v_strada bd_angajati.strada%type;
v_oras bd_angajati.oras%type;
v_nume bd_angajati.nume%type;
BEGIN
select strada, oras, nume into v_strada, v_oras, v_nume from bd_angajati where nume='in';
dbms_output.put_line('Angajatul '||v_nume||' locuieste in orasul '||v_oras||' pe strada '||v_strada);
EXCEPTION
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE('Exista mai multi salariatii cu numele '||v_nume||');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu exista un salariat cu numele '||v_nume||');
END;

```

Script Output x

Task completed in 3,665 seconds

```

WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu exista un salariat cu numele '||v_nume||');
END;
Exista mai multi salariatii cu numele Paraschiv!

PL/SQL procedure successfully completed.

```

3. Sa se afiseze denumirea functiilor si salariul maxim in ordine descrescatoare. Sa se verifice, printr-o exceptie, starea cursorului.

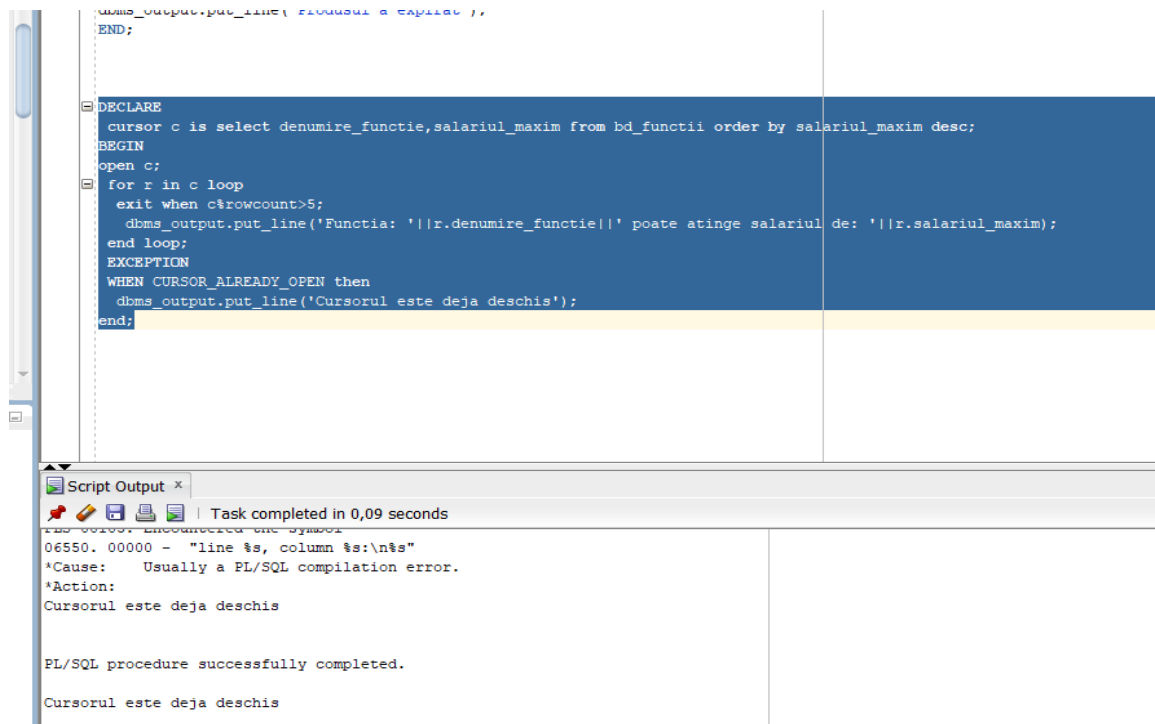
DECLARE

cursor c is select denumire_functie,salariul_maxim from bd_functii order by salariul_maxim desc;


```

BEGIN
open c;
for r in c loop
exit when c%rowcount>5;
dbms_output.put_line('Funcția: '||r.denumire_functie||' poate atinge salariul de: '||r.salariul_maxim);
end loop;
EXCEPTION
WHEN CURSOR_ALREADY_OPEN then
dbms_output.put_line('Cursorul este deja deschis');
end;

```



The screenshot shows a code editor with a PL/SQL script and a 'Script Output' window below it. The script defines a cursor 'c' to select function names and their maximum salaries from a table 'bd_functii', ordered by maximum salary in descending order. It then opens the cursor, loops through the results (exiting after 5 rows), and prints each function name and its maximum salary. An exception is handled for the case where the cursor is already open.

```

-- dbms_output.put_line( 'Cursorul e deschis' );
END;

DECLARE
cursor c is select denumire_functie,salariul_maxim from bd_functii order by salariul_maxim desc;
BEGIN
open c;
for r in c loop
exit when c%rowcount>5;
dbms_output.put_line('Funcția: '||r.denumire_functie||' poate atinge salariul de: '||r.salariul_maxim);
end loop;
EXCEPTION
WHEN CURSOR_ALREADY_OPEN then
dbms_output.put_line('Cursorul este deja deschis');
end;

```

The 'Script Output' window shows the following messages:

```

Task completed in 0,09 seconds
PL/SQL: Encountered the symbol
06550. 00000 - "line %s, column %s:\n%s"
*Cause:    Usually a PL/SQL compilation error.
*Action:
Cursorul este deja deschis

PL/SQL procedure successfully completed.
Cursorul este deja deschis

```

4. Să se modifice salariul angajatului cu id-ul 68. Dacă angajatul nu există sau dacă o altă eroare apare să fie ridicată o excepție.

```

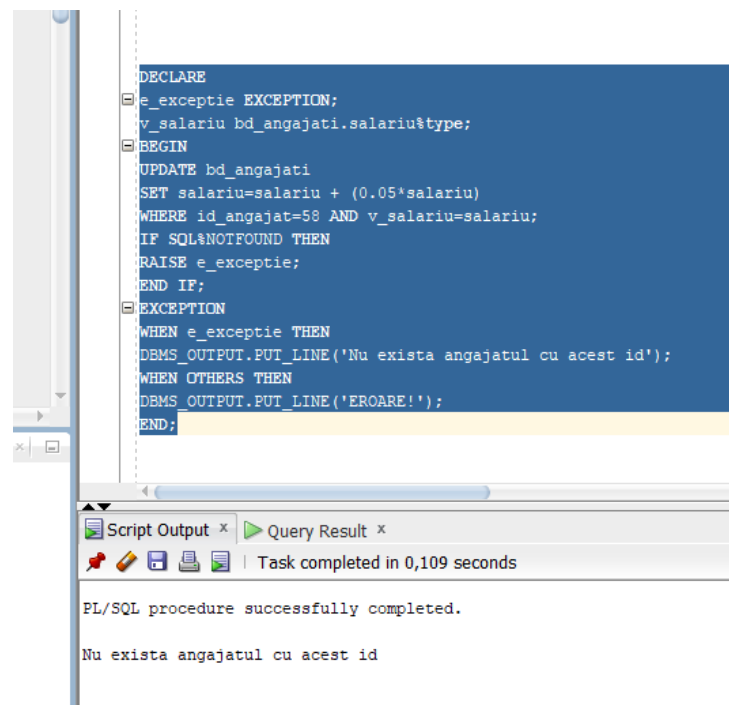
DECLARE
e_excepție EXCEPTION;
v_salariu bd_angajati.salariu%type;
BEGIN
UPDATE bd_angajati
SET salariu=salariu + (0.05*salariu)
WHERE id_angajat=58 AND v_salariu=salariu;
IF SQL%NOTFOUND THEN
RAISE e_excepție;
END IF;

```

```

EXCEPTION
WHEN e_exceptie THEN
DBMS_OUTPUT.PUT_LINE('Nu exista angajatul cu acest id');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('EROARE!');
END;

```



Partea 3. Functii. Proceduri. Pachete.

1. Creati o functie care sa returneze valoarea totala a produselor furnizate de catre un furnizor.

create or replace function valoare_totala_furnizor

(p_id_furnizor in bd_furnizori.id_furnizor%type)

return number

is

v_pret bd_produce.pret%type;

furn_invalid exception; begin

select sum(p.pret*p.cantitate) into v_pret from bd_furnizori f, bd_produce p where
f.id_furnizor=p.id_furnizor and f.id_furnizor=p_id_furnizor group by f.id_furnizor;

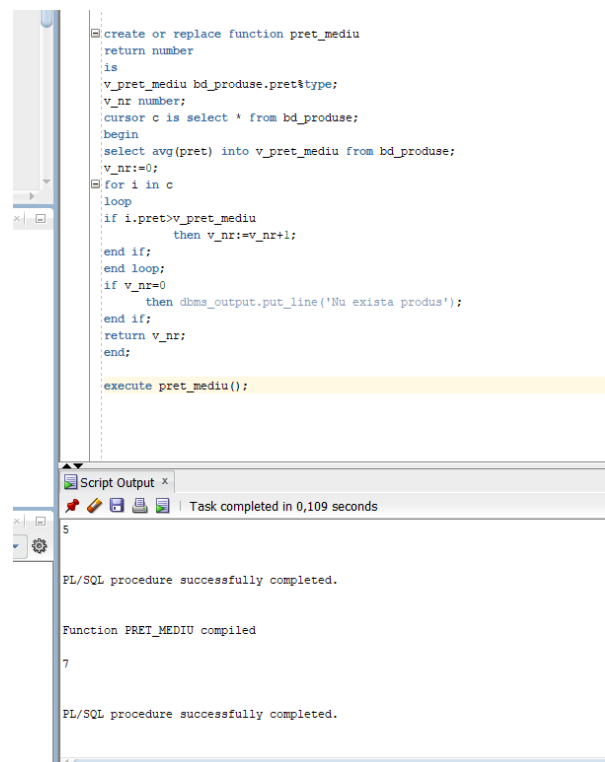
if v_pret=0


```

        then v_nr:=v_nr+1;
    end if;
end loop;
if v_nr=0
    then dbms_output.put_line('Nu exista produs');
end if;
return v_nr;
end;

execute dbms_output.put_line(pret_mediu());

```



The screenshot shows the Oracle SQL Developer environment. The main window displays a PL/SQL script for a function named `pret_mediu`. The script defines a function that takes a product name as input and returns its average price. It uses a cursor to iterate through the `bd_produce` table, calculating the average price and counting the number of products. The function is then executed, and the output is displayed in the 'Script Output' window.

```

create or replace function pret_mediu
return number
is
    v_pret_mediu bd_produce.pret%type;
    v_nr number;
    cursor c is select * from bd_produce;
begin
    select avg(pret) into v_pret_mediu from bd_produce;
    v_nr:=0;
    for i in c
    loop
        if i.pret>v_pret_mediu
            then v_nr:=v_nr+1;
        end if;
    end loop;
    if v_nr=0
        then dbms_output.put_line('Nu exista produs');
    end if;
    return v_nr;
end;

execute pret_mediu();

```

The 'Script Output' window shows the following messages:

```

5
PL/SQL procedure successfully completed.

Function PRET_MEDIU compiled

7
PL/SQL procedure successfully completed.

```

3. Creati o procedura prin care sa se afiseze numele si salariul pentru angajatii care au salariul mai mare decat media dintre salariul minim si salariul maxim. Procedura primeste ca parametri de tip IN id-urile a 2 angajati, reprezentand limita inferioara si cea superioara a intervalului de cautare. Folositi un cursor explicit pentru a parcurge angajatii. Tratati si cazul in care intervalul nu cuprinde astfel de angajati.

```

create or replace procedure interval_salariu

```

```

(p_id_angajat_1 in bd_angajati.id_angajat%type, p_id_angajat_2 in bd_angajati.id_angajat%type) is

```

```

e exception;

```

```

cursor c is (select nume, salariu from bd_angajati a, bd_functii f where id_angajat between p_id_angajat_1
and p_id_angajat_2 and a.id_functie=f.id_functie and salariu>(salariul_maxim+salariul_minim)/2);

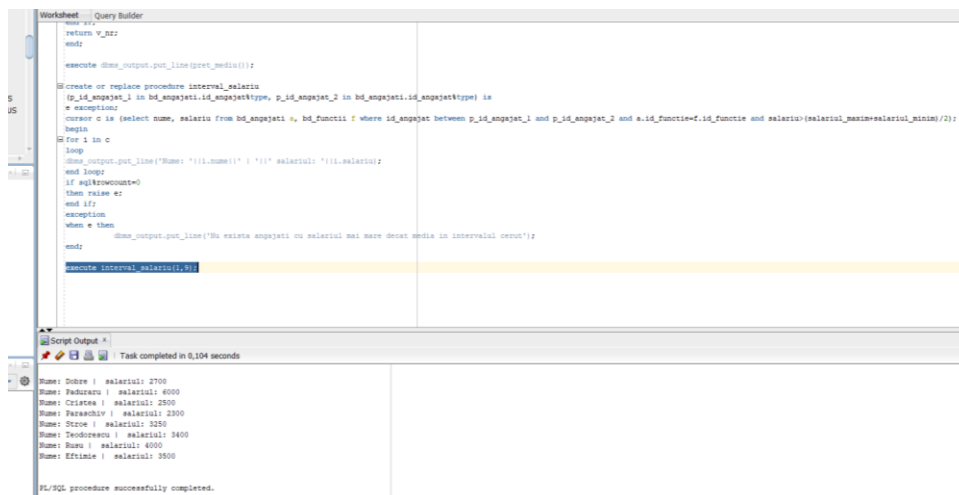
```

```

begin
for i in c
loop
dbms_output.put_line('Nume: '||i.nume||' salariul: '||i.salariu);
end loop;
if sql%rowcount=0
then raise e;
end if;
exception
when e then
    dbms_output.put_line('Nu exista angajati cu salariul mai mare decat media in intervalul cerut');
end;

execute interval_salariu(1,9);

```



4. Creati o procedura care primeste ca parametru de intrare id-ul unui client si returneaza prin parametru de tip out emailul acestuia.

create or replace procedure informatie_client

(p_id_client in bd_clienti.id_client%type, p_email out bd_clienti.email%type) is

begin

select email into p_email from bd_clienti where id_client=p_id_client;

dbms_output.put_line('Clientul cu id-ul: '||p_id_client||' are emailul: '||p_email);

EXCEPTION

when NO_DATA_FOUND

```

THEN dbms_output.put_line('Clientul cautat nu exista');

end;

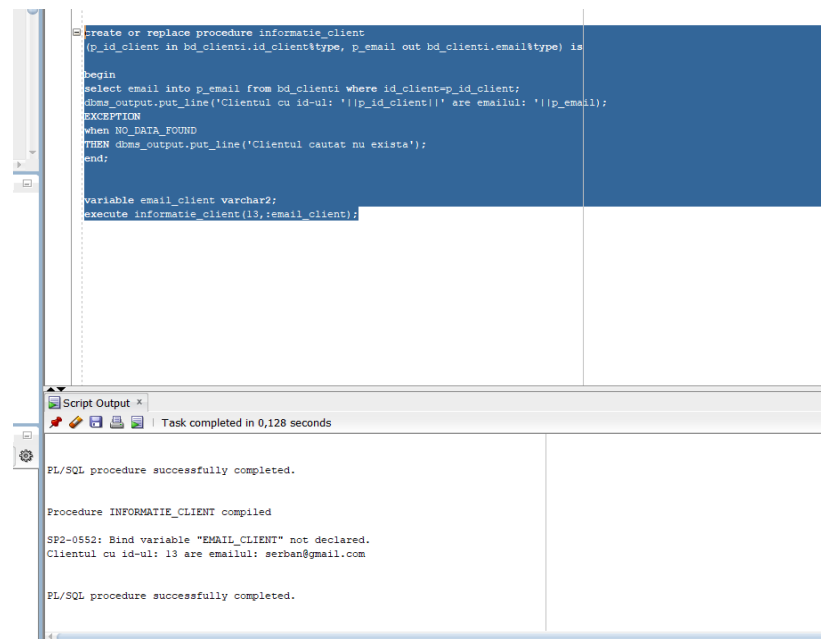
```

```

variable email_client varchar2;

execute informatie_client(13,:email_client);

```



5. Creati un pachet care sa cuprinda:

- o functie care sa returneze valoarea totala a vanzarilor unui produs. Functia primeste ca parametru id-ul produsului.
- o procedura care se foloseste de functia de mai sus pentru a afisa primele 5 cele mai bine vandute produse.

```

create or replace package projpack is

```

```

function val_vanz_prod(p_id_produs in bd_produce.id_produs%type) return number;

```

```

procedure top_3_prod;

```

```

end;

```

```

create or replace package body projpack is

```

```

function val_vanz_prod

```

```

(p_id_produs in bd_produce.id_produs%type)

```

```

return number

```

is

v_val number;

begin

select sum(p.pret*r.cantitate_comandata) into v_val from bd_rand_comenzi r, bd_produce p where
r.id_produș=p.id_produș;

if v_val=0

then return 0;

else return v_val;

end if;

end;

procedure top_3_prod is

cursor c1 is select id_produș, denumire_produș, val_vanz_prod(id_produș) as nr_t from bd_produce
order by 2 desc;

begin

for i in c1 loop

exit when c1%rowcount=4;

dbms_output.put_line('Produsul '||i.id_produș||' '||i.denumire_produș||' a avut vanzari de '||i.nr_t);

end loop;

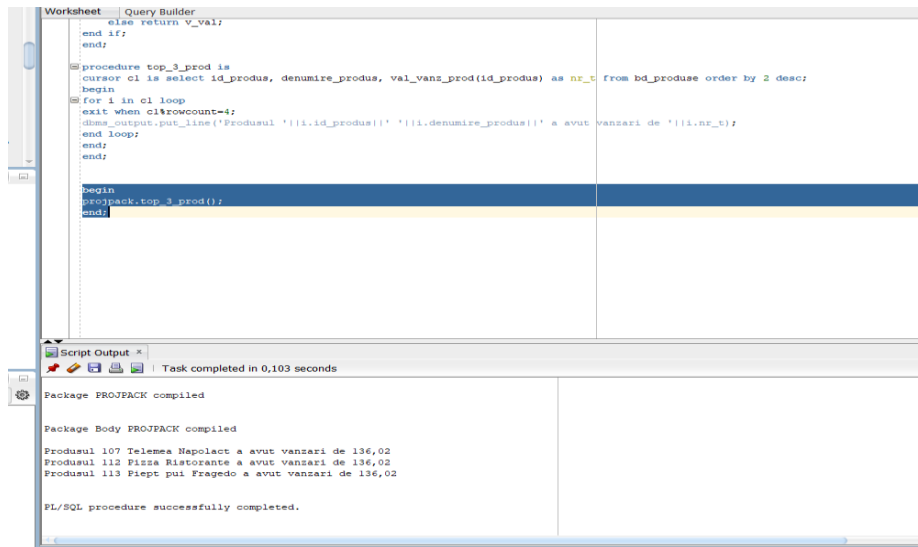
end;

end;

begin

projpack.top_3_prod();

end;



6. Sa se creeze un pachet care sa ajute la monitorizarea facturilor venite de la furnizori.
Aceasta va contine:

- O procedura pentru adaugarea unei noi facturi;
- O procedura pentru a modifica data la care a fost emisa factura;
- O procedura pentru a sterge o factura.

create or replace package pack_facturi as

procedure adauga_factura(nr_fact number, data_fact DATE, id_furn number);

procedure modifica_factura(nr_fact number,new_data DATE);

procedure sterge_factura(nr_fact number);

end pack_facturi;

/

create or replace package body pack_facturi as

procedure adauga_factura(nr_fact number, data_fact DATE, id_furn number) as

v_nr_fact number;

ex exception;

begin

select count(nr_fact) into v_nr_fact from bd_facturi where nr_factura=nr_fact;

if (v_nr_fact>0) then

raise ex;

else

insert into bd_facturi values(nr_fact,data_fact,id_furn);


```
end if;

exception

when ex then

dbms_output.put_line('Exista deja o factura cu acel numar de inregistrare');

end;
```

```
procedure modifica_factura(nr_fact number,new_data DATE) is

nr number;

ex exception;

begin

select count(nr_fact) into nr from bd_facturi where nr_fact=nr_factura;

if (nr=0) then

raise ex;

else

update bd_facturi

set data_factura=new_data

where nr_factura=nr_fact;

end if;

exception

when ex then

dbms_output.put_line('Nu exista factura cu numarul de inregistrare specificat');

end;
```

```
procedure sterge_factura(nr_fact number) is

nr number;

ex exception;

begin
```

```
select count(nr_fact) into nr from bd_facturi where nr_factura=nr_fact;

if (nr=0) then

    raise ex;

else

    delete from bd_facturi where nr_factura=nr_fact;

end if;

exception

when ex then

    dbms_output.put_line('Nu exista factura cu numarul de inregistrare specificat');

end;

end pack_facturi;

execute pack_facturi.adauga_factura(1115,to_date('16.06.2021', 'dd.mm.yyyy'), 1);

select * from bd_facturi where nr_factura=1115;


execute pack_facturi.modifica_factura(1115,to_date('20.06.2021', 'dd.mm.yyyy'));

select * from bd_facturi where nr_factura=1115;


execute pack_facturi.sterge_factura(1115);

select * from bd_facturi where nr_factura=1115;
```

Worksheet Query Builder

```

when ex then
    dbms_output.put_line('Exista deja o factura cu acel numar de inregistrare');
end;

procedure modifica_factura(nr_fact number,new_data DATE) is
    nr number;
    ex exception;
begin
    select count(nr_fact) into nr from bd_facturi where nr_fact=nr_factura;
    if (nr=0) then
        raise ex;
    else
        update bd_facturi
        set data_factura=new_data
        where nr_factura=nr_fact;
    end if;
exception
    when ex then
        dbms_output.put_line('Nu exista factura cu numarul de inregistrare specificat');
end;

procedure sterge_factura(nr_fact number) is
    nr number;
    ex exception;
begin
    select count(nr_fact) into nr from bd_facturi where nr_factura=nr_fact;
    if (nr=0) then
        raise ex;
    end if;
end;

```

Script Output x Query Result x

Task completed in 0,106 seconds

Cause: Usually a PL/SQL compilation error.

Action:

Package PACK_FACTURI compiled

Package Body PACK_FACTURI compiled

PL/SQL procedure successfully completed.

Error starting at line : 243 in command -
 BEGIN pack2.modifica_factura(1115,to_date('20.06.2021', 'dd.mm.yyyy')); END;

Worksheet Query Builder

```

procedure sterge_factura(nr_fact number) is
    nr number;
    ex exception;
begin
    select count(nr_fact) into nr from bd_facturi where nr_factura=nr_fact;
    if (nr=0) then
        raise ex;
    else
        delete from bd_facturi where nr_factura=nr_fact;
    end if;
exception
    when ex then
        dbms_output.put_line('Nu exista factura cu numarul de inregistrare specificat');
end;

end pack_facturi;

execute pack_facturi.adauga_factura(1115,to_date('16.06.2021', 'dd.mm.yyyy'), 1);
select * from bd_facturi where nr_factura=1115;

execute pack_facturi.modifica_factura(1115,to_date('20.06.2021', 'dd.mm.yyyy'));
select * from bd_facturi where nr_factura=1115;

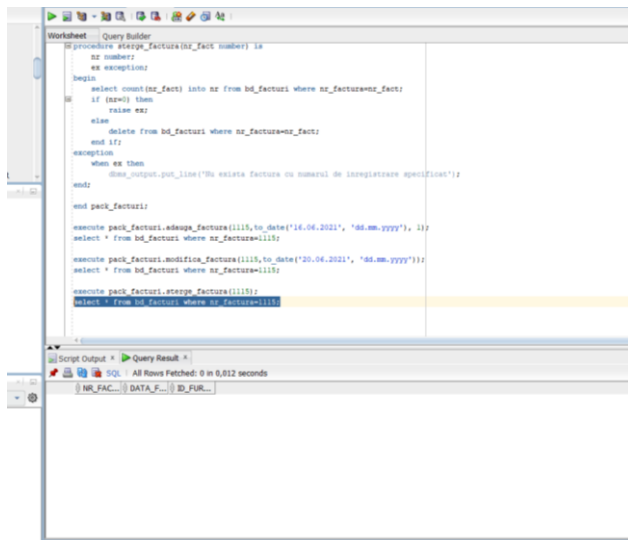
execute pack_facturi.sterge_factura(1115);
select * from bd_facturi where nr_factura=1115;

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0,018 seconds

NR_FACTURA	DATA_FACTURA	ID_FURNIZOR
1	1115 16-06-2021	1



7. Sa se creeze o functie care sa returneze numarul de comenzi efectuate de un client, avand id_client dat ca parametru al functiei.

create or replace function numar_comenzi(id number) return number as

nr_comanda number;

nr_client number;

client_inexistent exception;

client_fara_comanda exception;

begin

select count(id_client) into nr_client from bd_clienti where id_client=id;

if (nr_client = 0) then

raise client_inexistent;

end if;

select count(id_client) into nr_comanda from bd_comenzi where id_client=id;

if (nr_comanda = 0) then

raise client_fara_comanda;

else

return nr_comanda;

end if;

exception

when client_inexistent then

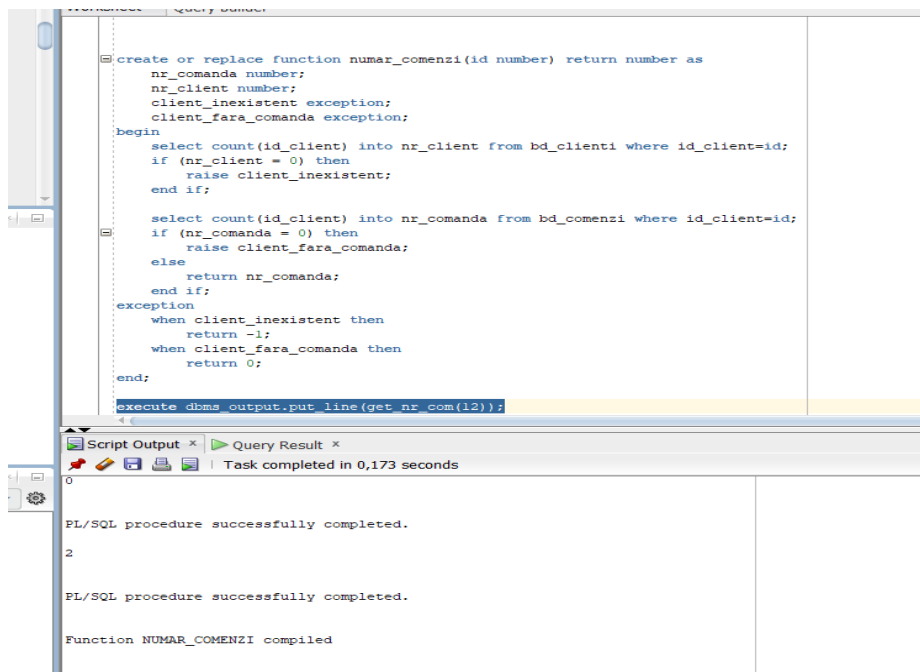
return -1;

when client_fara_comanda then

return 0;

end;

execute dbms_output.put_line(get_nr_com(12));



8. Sa se creeze o procedura prin care sa se mărească cu 30% prețul produselor primite de la furnizorii din orasul Brasov. Procentul și localitatea sunt parametrii procedurii. Trateaza eventualele exceptii.

```
CREATE OR REPLACE PROCEDURE marire_pret(p_procent IN NUMBER, p_localitate IN
bd_furnizori.localitate%type)
IS
nu_exista_produs EXCEPTION;
BEGIN

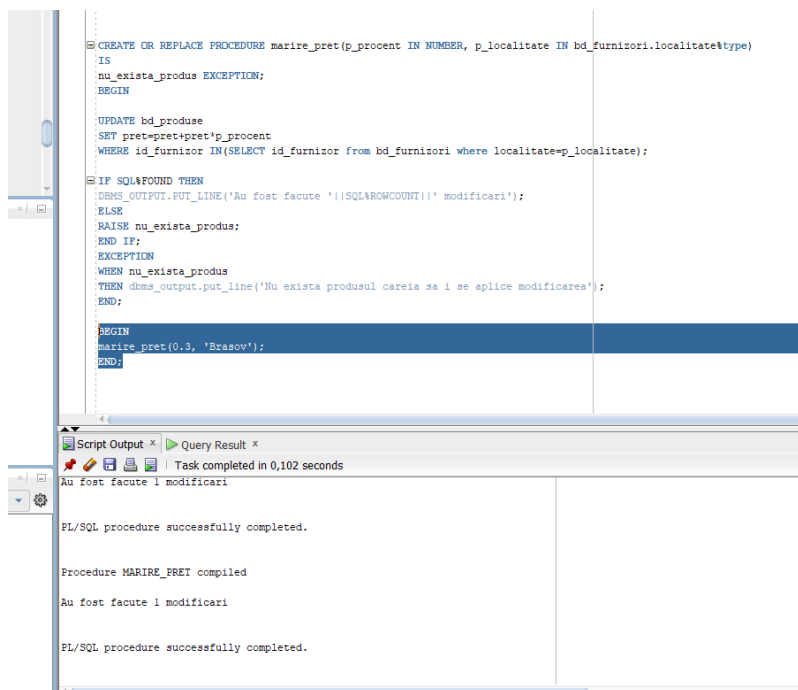
UPDATE bd_produce
SET pret=pret+pret*p_procent
WHERE id_furnizor IN(SELECT id_furnizor from bd_furnizori where localitate=p_localitate);
```

```

IF SQL%FOUND THEN
DBMS_OUTPUT.PUT_LINE('Au fost facute '||SQL%ROWCOUNT||' modificari');
ELSE
RAISE nu_exista_produc;
END IF;
EXCEPTION
WHEN nu_exista_produc
THEN dbms_output.put_line('Nu exista produsul careia sa i se aplice modificarea');
END;

BEGIN
marire_pret(0.3, 'Brasov');
END;

```



9. Sa se creeze o procedura care sa afiseze detalii despre comenzile care contin produse cu data de expirare mai mica decat 23.05.2021.

```
CREATE OR REPLACE PROCEDURE afiseaza_comenzi AS
```

```

CURSOR c IS SELECT c.data_comanda, c.modalitate_plata, c.status_comanda, r.cantitate_comandata,
p.data_expirare

```

```

FROM bd_comenzi c, bd_rand_comenzi r, bd_produce p WHERE c.id_comanda = r.id_comanda
AND p.id_produc=r.id_produc ;

```

```

rec_c c%ROWTYPE;

```

```
BEGIN
```

```

OPEN c;

```

```
LOOP
```

```

FETCH c INTO rec_c;

```

```

EXIT WHEN c%NOTFOUND;

IF rec_c.data_expirare > TO_DATE('23-05-2021', 'DD-MM-YYYY') THEN

    DBMS_OUTPUT.PUT_LINE(rec_c.data_comanda||' ' || rec_c.modalitate_plata|| ' ' ||
rec_c.status_comanda||' '||rec_c.cantitate_comandata);

END IF;

END LOOP;

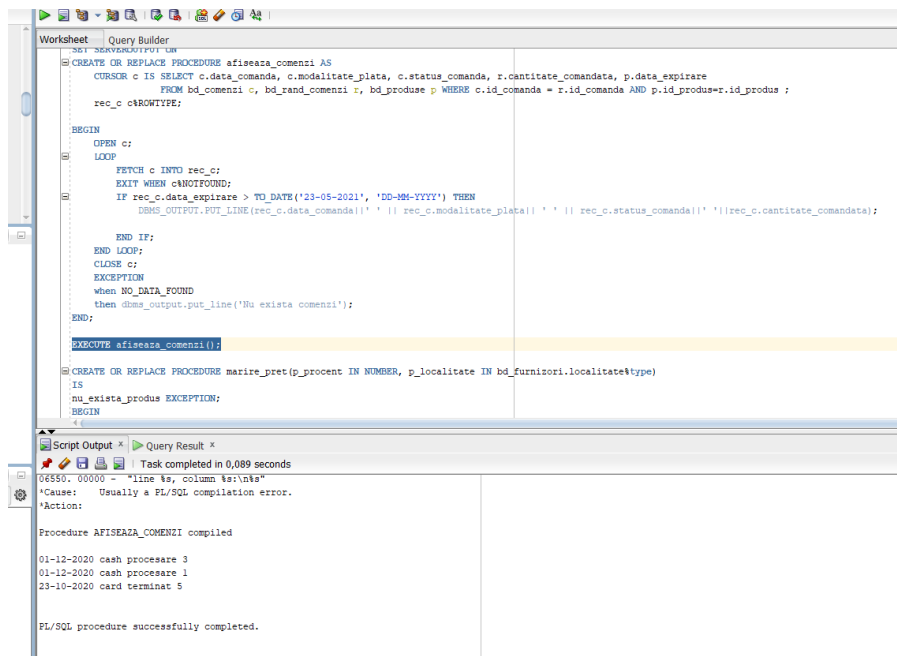
CLOSE c;

EXCEPTION

when NO_DATA_FOUND

then dbms_output.put_line('Nu exista comenzi');

END;
```



Partea 4. Declansatori

1. Un trigger care sa se declanseze in momentul adaugarii unui produs deja existent sau la modificarea pretului unui produs deja existent.

create or replace trigger valabilitate_mica

before update of pret on bd_produce

for each row

begin

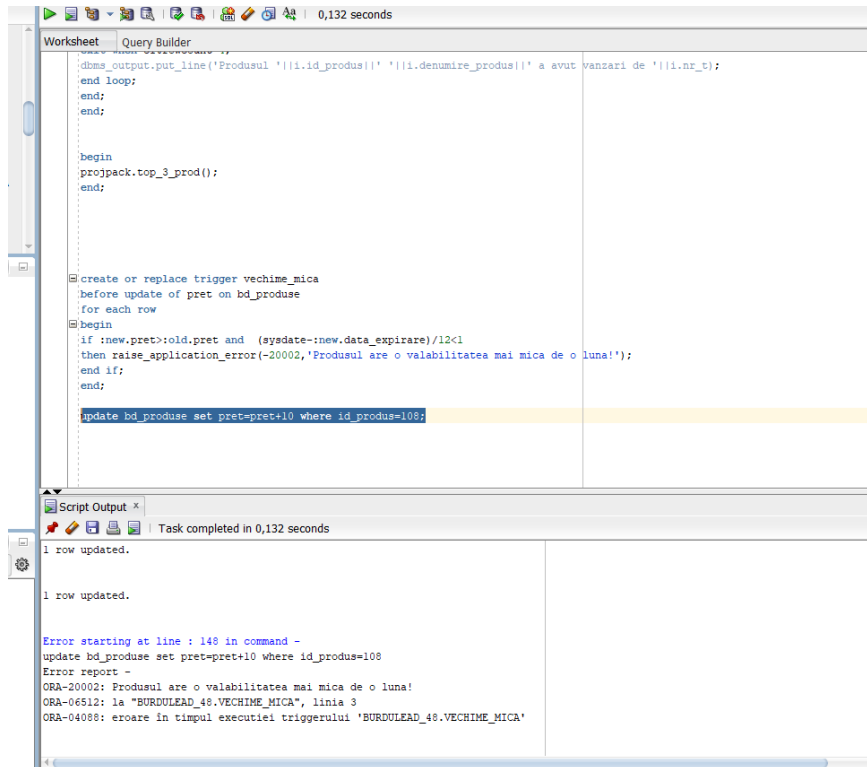
if :new.pret>:old.pret and (sysdate-:new.data_expirare)/12<1

```
then raise_application_error(-20002,'Produsul are o valabilitatea mai mica de o luna!');
```

```
end if;
```

```
end;
```

```
update bd_produce set pret=pret+10 where id_produs=108;
```



2. Un trigger care sa se declanseze inainte de modificarea stocului unui produs din depozit si care sa afiseze un mesaj.

```
create or replace trigger modificare_stoc
```

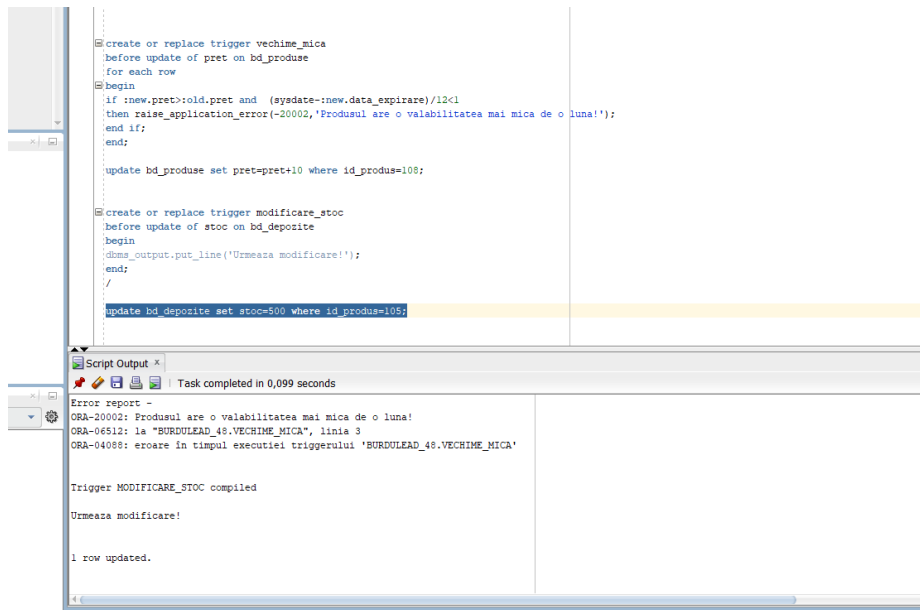
```
before update of stoc on bd_depozite
```

```
begin
```

```
dbms_output.put_line('Urmeaza modificare!');
```

```
end;
```

```
update bd_depozite set stoc=500 where id_produs=105;
```

3. Sa se realizeze un declansator care sa nu permita introducerea unui produs care are data expirarii mai mica decat data actuala.

CREATE OR REPLACE TRIGGER data_expirare_mica

BEFORE INSERT OR UPDATE OF data_expirare on bd_produce

FOR EACH ROW

BEGIN

IF(:new.data_expirare<SYSDATE) THEN

RAISE_APPLICATION_ERROR(-20000, 'Produsul a expirat deja!');

END IF;

END;

INSERT INTO bd_produce(id_produc, denumire_produc, categorie_produc, pret, data_expirare, id_furnizor)

VALUES('114', 'Piept Curcan Fragedo', 'Carne', '24', to_date('02.06.2005', 'dd.mm.yyyy'), '1244');

```
--Sa se realizeze und eclansator care sa nu permita introducerea unui produs care are data expirarii mai mica decat SYSDATE

CREATE OR REPLACE TRIGGER data_expirare_mina
BEFORE INSERT OR UPDATE ON data_expirare ON bd_produce
FOR EACH ROW
BEGIN
    IF (new.data_expirare < SYSDATE) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Produsul a expirat deja!');
    END IF;
END;

INSERT INTO bd_produce(id_produce, denumire_produce, categorie_produce, pret, data_expirare, id_furnizor)
VALUES('114', 'Piept Curcan Frazedo', 'Carne', '24', to_date('02.06.2005', 'dd.mm-yyyy'), '1244');
```

Script Output X

Task completed in 0,121 seconds

Trigger DATA_EXPIRARE_MINA compiled

Error starting at line : 172 in command -

```
INSERT INTO bd_produce(id_produce, denumire_produce, categorie_produce, pret, data_expirare, id_furnizor)
VALUES('114', 'Piept Curcan Frazedo', 'Carne', '24', to_date('02.06.2005', 'dd.mm-yyyy'), '1244')
Error report -
ORA-20000: Produsul a expirat deja!
ORA-06512: la "BURDULEA_46.DATA_EXPIRARE_MINA", linia 3
ORA-04088: eroare în timpul executiei triggerului 'BURDULEA_46.DATA_EXPIRARE_MINA'
```

LINK APEX:

<https://apex.oracle.com/pls/apex/dianaburdulea/r/sghd-burdulea-diana-student1/home?session=116216489926211>