

PROIECT – SISTEME DE BAZE DE DATE

Cuprins

A) Prezentarea bazei de date:.....	2
B) Diagrama entitate-relație	3
C) Diagrama conceptuală.....	4
D) Design logic.....	4
E) FN1, FN2, FN3	5
F) Implementarea tabelelor.....	8
G) 15 interogari	14
H) Tabel mesaje	18
I) și J) P L/SQL	18
K) Ștergere date	25

A) Prezentarea bazei de date: baza de date descrie operațiunile specifice unui cofetărie care acceptă doar comenzi online, plătite online. Aceasta are mai mulți furnizori, iar produsele sale pot fi achiziționate de mai mulți clienți.

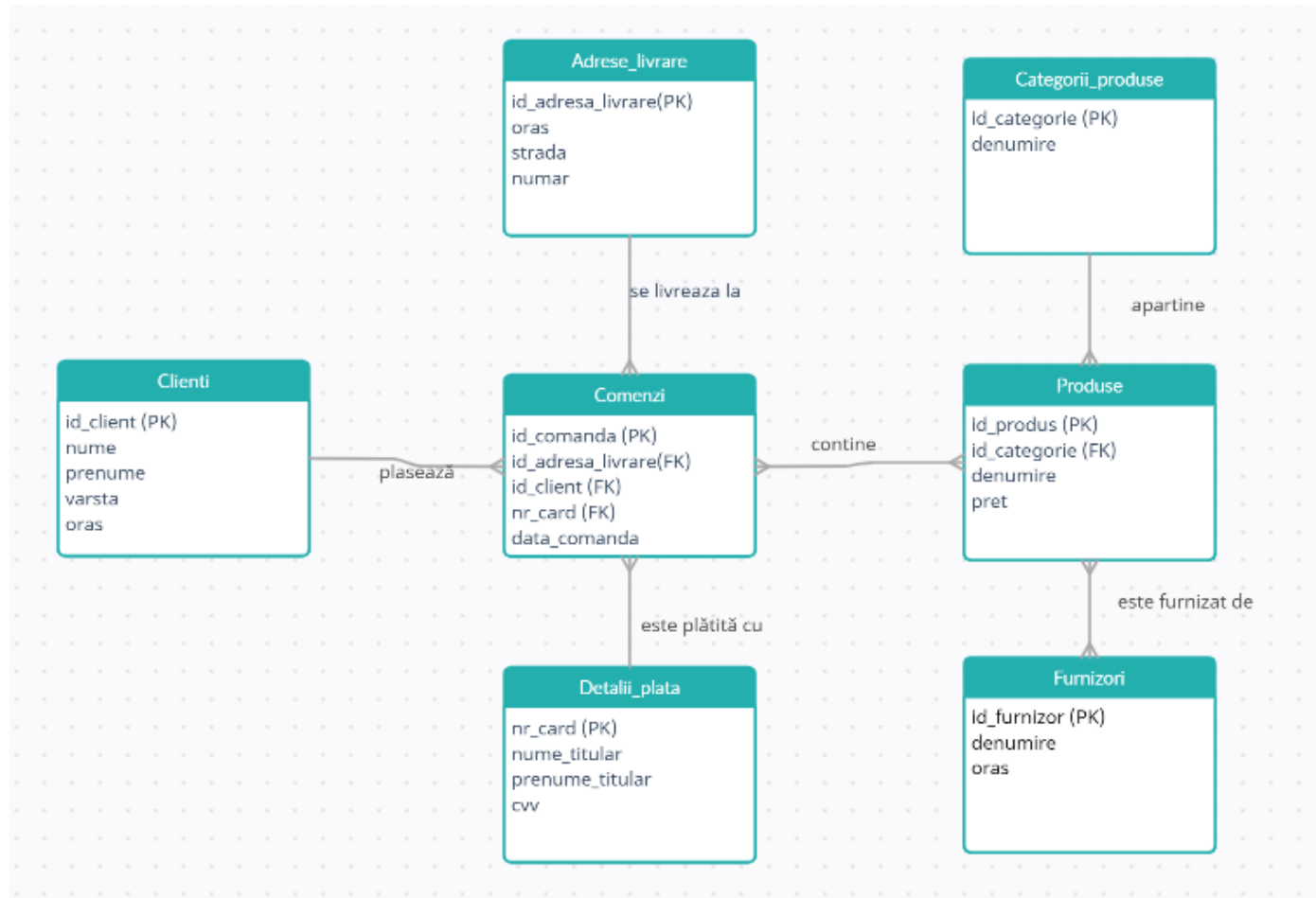
Entități independente:

1. Furnizori;
2. Clienți;
3. Adrese_livrare;
4. Detalii_plată;
5. Produse;
6. Categoriile_produce;
7. Comenzi

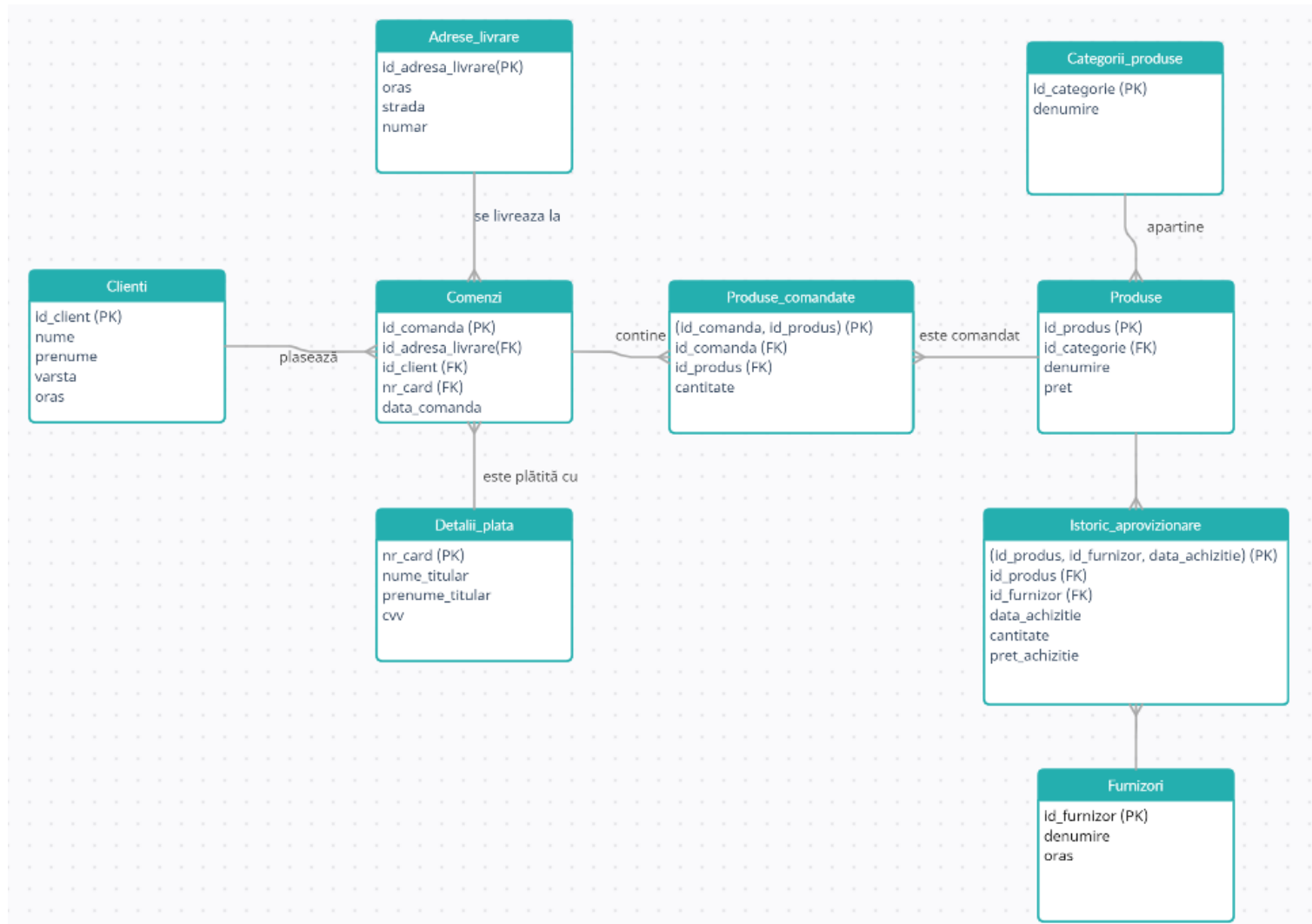
Tabele asociative:

- Produse_comandate (chei externe: id_articol, id_client);
- Istorice_aprovizionare (chei externe: id_furnizor, id_articol);

B) Diagrama entitate-relație



C) Diagrama conceptuală



D) Design logic

Tabelele (atributele care sunt chei primare sunt subliniate, iar cele care sunt chei străine sunt înclinate):

- Furnizori (id_furnizor, denumire, oraș);
- Clienți (id_client, nume, prenume, vârstă, oraș);
- Adrese_livrare (id_adresă_livrare, oraș, stradă, număr);
- Detalii_plată (nr_card, nume_titular, prenume_titular, cvv);
- Produse (id_produș, id_categorie, denumire, pret);
- Categoriile_produce (id_categorie, denumire);
- Comenzi (id_comanda, id_adresa_livrare, id_client, nr_card, data_comanda);

- Produse_comandate (*id_comanda, id_produș, cantitate*);
- Istoric_aprovizionare (*id_produș, id_furnizor, data_achizitie, cantitate, pret_achizitie*);

E) FN1, FN2, FN3

Considerăm 2 tabele:

Comenzi

Id coman dă	Id adresă	adresă	Id client	Nume client	Id prosus	Denu mire produ s	Preț produ s	Id catego rie	Categ orie	cantit ate	Mod de plată	Id furnizo r	Furni zor produ s	Oraș furniz or	Data coman dă
1	1	Bucur ești, str. Acade miei, nr. 14	2	Popesc u Andrei	2	Tort ciocol ată	60	1	Torturi	1	12345 67, 345 Popesc u Andrei	100	Cofetă ria X	Cluj	22.01.2 022
1	1	Bucur ești, str. Acade miei, nr. 14	2	Popesc u Andrei	4	Ecler	40	2	Dulciu ri	4	12345 67, 345 Popesc u Andrei	56	Cofetă ria Y	Bucur ești	22.01.2 022
2	2	Bucur ești, str. Repub lica, nr. 22	4	Vasile Alina	4	Ecler	40	2	Dulciu ni	4	98765 43, 123 Vasile Andrei	56	Cofetă ria Y	Bucur ești	15.01.2 022

Istoric aprovizionare

Id furnizor	Data	Id produs	Cantitate	Preț
100	12.01.2022	2	5	50
56	15.01.2022	4	20	7

FN1

- Coloane cu attribute compuse: adresă, mod_de_plată, nume client.

- Grupuri de attribute repetitive: nu există (un exemplu de tabel cu attribute repetitive ar fi fost unul care să conțină următoarele coloane: id_comandă, id_produs1, id_produs2, id_produs3)

Eliminare attribute compuse:

Id comandă	Id adresă	Oraș	Stradă	Nr	Id client	Pre nume client	Nume client	Id produs	Denumire produs	Preț produs	Id categorie	Categorie	cantitate	Nr. card	CVV	Nume titular card	Pre nume titular card	Id furnizor	Furnizor produs	Oraș furnizor	Data comandă
1	1	București	str. Academiei	14	2	Andrei	Popescu	2	Tort ciocolată	60	1	Torturi	1	1234567	345	Popescu	Andrei	100	Cofetăria X	Cluj	22.01.2022
1	1	București	str. Academiei	14	2	Andrei	Popescu	4	Ecler	40	2	Dulciuri	4	1234567	345	Popescu	Andrei	56	Cofetăria Y	București	22.01.2022
2	2	București	Str. Republicii	22	4	Alina	Vasile	4	Ecler	40	2	Dulciuri	3	9876543	123	Vasile	Andrei	56	Cofetăria Y	București	15.01.2022

FN2

Dependențe parțiale:

- Id_adresă -> oraș, stradă, nr;
- Id_client -> nume, prenume;
- Id_produs -> denumire, preț;
- Id_categorie -> denumire categorie;
- nr_card -> CVV, nume și prenume titular card;
- Id_furnizor -> furnizor produs, oraș furnizor.

Rezolvare dependențe parțiale

Adrese livrare

Id adresă	Oraș	Stradă	Nr
1	București	str. Academiei	14
2	București	Str. Republicii	22

Clienți

Id client	Prenume client	Nume client
2	Andrei	Popescu
4	Alina	Vasile

Produce

Id produs	Denumire produs	Preț produs
2	Tort ciocolată	60
4	Ecler	40

Categorii produse

Id categorie	Denumire
1	Torturi
2	Dulciuri

Detalii plată

Nr. card	CVV	Nume titular card	Prenume titular card
1234567	345	Popescu	Andrei
9876543	123	Vasile	Andrei

Furnizori

Id furnizor	Furnizor produs	Oraș furnizor
100	Cofetăria X	Cluj
56	Cofetăria Y	București

Comenzi

Id comandă	Id adresă	Id client	Nr. card	Data comandă
1	1	2	1234567	22.01.2022
2	2	4	9876543	15.01.2022

Produce comandate

Id comandă	Id produs	Cantitate
1	2	1
1	4	4
2	4	3

Istoric aprovizionare

Id furnizor	Data	Id produs	Cantitate	Pret
100	12.01.2022	2	5	50
56	15.01.2022	4	20	7

FN3

Dependente tranzitive: nu există.

F) Implementarea tabelelor

-- Creare tabele si constrangeri

```
create table adrese_livrare_oiv (  
  id_adresa_livrare varchar(15),  
  oras varchar2(30) not null,  
  strada varchar2(30),  
  numar number(4),  
  primary key (id_adresa_livrare)  
);
```

```
create table clienti_oiv (  
  id_client varchar(15),  
  nume varchar2(30) not null,  
  prenume varchar2(30),  
  varsta number(2),  
  oras varchar2(30),  
  primary key (id_client)  
);
```

```
create table categorii_produce_oiv (  
  id_categorie varchar(15),  
  denumire varchar2(30) not null,  
  primary key (id_categorie)  
);
```

```
create table produse_oiv (  
  id_produs varchar(15),  
  denumire varchar2(30) not null,  
  pret number(4),  
  id_categorie varchar(15),  
  primary key (id_produs),  
  foreign key(id_categorie) references categorii_produce_oiv(id_categorie)
```



```
);

create table detalii_plata_oiv (
    nr_card varchar(15),
    nume_titular varchar2(30) not null,
    prenume_titular varchar2(30) not null,
    cvv number(3),
    primary key (nr_card)
);

create table furnizori_oiv (
    id_furnizor varchar(15),
    denumire varchar2(30) not null,
    oras varchar2(30) not null,
    primary key (id_furnizor)
);

create table comenzi_oiv (
    id_comanda varchar(15),
    id_client varchar(15),
    nr_card varchar(15),
    id_adresa_livrare varchar(15),
    data_comanda date default sysdate,
    primary key (id_comanda),
    foreign key(id_client) references clienti_oiv(id_client),
    foreign key(nr_card) references detalii_plata_oiv(nr_card),
    foreign key(id_adresa_livrare) references adrese_livrare_oiv(id_adresa_livrare)
);

create table produse_comandate_oiv (
    id_comanda varchar(15),
    id_produș varchar(15),
    cantitate number(4) not null,
    primary key (id_comanda, id_produș),
    foreign key(id_comanda) references comenzi_oiv(id_comanda),
    foreign key(id_produș) references produse_oiv(id_produș)
);

create table istoric_aprovizionare_oiv (
    id_produș varchar(15),
    id_furnizor varchar(15),
    cantitate number(4) not null,
    data_achizitie date default sysdate,
    pret_achizitie number(6) not null,
    primary key (id_produș, id_furnizor, data_achizitie),
    foreign key(id_produș) references produse_oiv(id_produș),
```

```
foreign key(id_furnizor) references furnizori_oiv(id_furnizor)
);
```

Adăugare informații

```
-- Creare secvente pentru a genera cheile primare
create sequence s_id_adresa_livrare;
create sequence s_id_client;
create sequence s_id_furnizor;
create sequence s_id_categorie;
create sequence s_id_produș;
create sequence s_id_comanda;

-- inserare valori in tabelul detalii de plata
insert into detalii_plata_oiv (nr_card, nume_titular, prenume_titular, cvv)
values (123456, 'Popescu', 'Andrei', 123);
insert into detalii_plata_oiv (nr_card, nume_titular, prenume_titular, cvv)
values (123457, 'Ionescu', 'Maria', 455);
insert into detalii_plata_oiv (nr_card, nume_titular, prenume_titular, cvv)
values (123458, 'Matei', 'David', 678);
insert into detalii_plata_oiv (nr_card, nume_titular, prenume_titular, cvv)
values (123459, 'Dramescu', 'Ana', 123);
insert into detalii_plata_oiv (nr_card, nume_titular, prenume_titular, cvv)
values (123490, 'Vasilescu', 'Mihai', 456);

savepoint adaugat_plati;

-- inserare valori in tabelul adrese_livrare_oiv
insert into adrese_livrare_oiv (id_adresa_livrare, oras, strada, numar)
values ('adr_' || s_id_adresa_livrare.nextval, 'Bucuresti', 'Iuliu Maniu', 46);
insert into adrese_livrare_oiv (id_adresa_livrare, oras, strada, numar)
values ('adr_' || s_id_adresa_livrare.nextval, 'Bucuresti', 'Mihai Eminescu', 3);
insert into adrese_livrare_oiv (id_adresa_livrare, oras, strada, numar)
values ('adr_' || s_id_adresa_livrare.nextval, 'Cluj', 'Ion Creanga', 16);
insert into adrese_livrare_oiv (id_adresa_livrare, oras, strada, numar)
values ('adr_' || s_id_adresa_livrare.nextval, 'Craiova', 'Tudor Vladimirescu', 4);
insert into adrese_livrare_oiv (id_adresa_livrare, oras, strada)
values ('adr_' || s_id_adresa_livrare.nextval, 'Titu', 'Unirea');

savepoint adaugat_adrese_livrare;

-- inserare valori in tabelul clienti
insert into clienti_oiv (id_client, nume, prenume, varsta, oras)
```

```
values ('client_'|| s_id_client.nextval, 'Vasile', 'Ionut', 22, 'Bucuresti');
insert into clienti_oiv (id_client, nume, prenume, varsta)
values ('client_'|| s_id_client.nextval, 'Daniel', 'Robert', 36);
insert into clienti_oiv (id_client, nume, prenume, oras, varsta)
values ('client_'|| s_id_client.nextval, 'Mihai', 'Maria', 'Targoviste', 34);
insert into clienti_oiv (id_client, nume, prenume, varsta)
values ('client_'|| s_id_client.nextval, 'Denis', 'Andrei', 23);
insert into clienti_oiv (id_client, nume, prenume, oras, varsta)
values ('client_'|| s_id_client.nextval, 'Dinu', 'Ana', 'Constanta', 37);
insert into clienti_oiv (id_client, nume, prenume, varsta)
values ('client_'|| s_id_client.nextval, 'Marius', 'Raluca', 40);
insert into clienti_oiv (id_client, nume, prenume, varsta)
values ('client_'|| s_id_client.nextval, 'Constantin', 'Denisa', 29);
```

```
savepoint adaugat_clienti;
```

```
-- inserare valori in tabelul furnizori
```

```
insert into furnizori_oiv (id_furnizor, denumire, oras)
values ('fz_'|| s_id_furnizor.nextval, 'Cofetaria Matilda', 'Girgiu');
insert into furnizori_oiv (id_furnizor, denumire, oras)
values ('fz_'|| s_id_furnizor.nextval, 'Cofetaria A', 'Bucuresti');
insert into furnizori_oiv (id_furnizor, denumire, oras)
values ('fz_'|| s_id_furnizor.nextval, 'Cofetaria B', 'Bucuresti');
insert into furnizori_oiv (id_furnizor, denumire, oras)
values ('fz_'|| s_id_furnizor.nextval, 'Cofetaria X', 'Cluj');
insert into furnizori_oiv (id_furnizor, denumire, oras)
values ('fz_'|| s_id_furnizor.nextval, 'Cofetaria Y', 'Cluj');
```

```
savepoint adaugat_furnizori;
```

```
-- inserare valori in tabelul categorii_produce
```

```
insert into categorii_produce_oiv (id_categorie, denumire)
values ('cat_'|| s_id_categorie.nextval, 'Prajituri clasice');
insert into categorii_produce_oiv (id_categorie, denumire)
values ('cat_'|| s_id_categorie.nextval, 'Prajituri italiene');
insert into categorii_produce_oiv (id_categorie, denumire)
values ('cat_'|| s_id_categorie.nextval, 'Prajituri cu frisca');
insert into categorii_produce_oiv (id_categorie, denumire)
values ('cat_'|| s_id_categorie.nextval, 'Prajituri cu branza');
insert into categorii_produce_oiv (id_categorie, denumire)
values ('cat_'|| s_id_categorie.nextval, 'Torturi');
```

```
savepoint adaugat_categorii_produce;
```

```
-- inserare valori in tabelul produse
```

```
insert into produse_oiv (id_produs, denumire, pret, id_categorie)
```

```
values ('prod_'|| s_id_produs.nextval, 'Ecler', 10, 'cat_1');
insert into produse_oiv (id_produs, denumire, pret, id_categorie)
values ('prod_'|| s_id_produs.nextval, 'Tort diplomat', 60, 'cat_5');
insert into produse_oiv (id_produs, denumire, pret)
values ('prod_'|| s_id_produs.nextval, 'Prajitura ciocolata', 45);
insert into produse_oiv (id_produs, denumire, pret, id_categorie)
values ('prod_'|| s_id_produs.nextval, 'Amandina', 10, 'cat_1');
insert into produse_oiv (id_produs, denumire, pret, id_categorie)
values ('prod_'|| s_id_produs.nextval, 'Savarina', 15, 'cat_3');
insert into produse_oiv (id_produs, denumire, pret, id_categorie)
values ('prod_'|| s_id_produs.nextval, 'Foietaj branza', 7, 'cat_4');
insert into produse_oiv (id_produs, denumire, pret)
values ('prod_'|| s_id_produs.nextval, 'Tarta fructe', 25);
insert into produse_oiv (id_produs, denumire, pret, id_categorie)
values ('prod_'|| s_id_produs.nextval, 'Tiramisu', 55, 'cat_2');

savepoint adaugat_produse;

-- inserare valori in tabelul istoric_aprovizionare_oiv
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_1', 'fz_1', 10, to_date('20-08-2020', 'dd-mm-yyyy'), 15);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_1', 'fz_3', 90, to_date('12-09-2021', 'dd-mm-yyyy'), 17);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_2', 'fz_2', 60, to_date('20-09-2021', 'dd-mm-yyyy'), 50);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_6', 'fz_2', 100, to_date('20-11-2021', 'dd-mm-yyyy'), 39);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_7', 'fz_5', 60, to_date('29-10-2021', 'dd-mm-yyyy'), 17);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_8', 'fz_5', 70, to_date('29-11-2021', 'dd-mm-yyyy'), 10);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_3', 'fz_4', 20, to_date('15-12-2021', 'dd-mm-yyyy'), 35);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_4', 'fz_5', 20, to_date('18-12-2021', 'dd-mm-yyyy'), 15);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_1', 'fz_1', 20, to_date('20-12-2021', 'dd-mm-yyyy'), 30);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_7', 'fz_3', 20, to_date('02-12-2021', 'dd-mm-yyyy'), 32);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, pret_achizitie)
values ('prod_2', 'fz_3', 20, 55);
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, pret_achizitie)
values ('prod_6', 'fz_2', 10, 49);

savepoint adaugat_istoric_aprovizionare;
```

```
-- inserare valori in tabelul comenzi_oiv
insert into comenzi_oiv (id_comanda, id_client, nr_card, id_adresa_livrare, data_comanda)
values ('comanda_'|| s_id_comanda.nextval, 'client_1', '123490', 'adr_1', to_date('03-12-2021', 'dd-mm-
yyyy'));
insert into comenzi_oiv (id_comanda, id_client, nr_card, id_adresa_livrare, data_comanda)
values ('comanda_'|| s_id_comanda.nextval, 'client_1', '123456', 'adr_4', to_date('12-01-2022', 'dd-mm-
yyyy'));
insert into comenzi_oiv (id_comanda, id_client, nr_card, id_adresa_livrare, data_comanda)
values ('comanda_'|| s_id_comanda.nextval, 'client_1', '123457', 'adr_3', to_date('17-01-2022', 'dd-mm-
yyyy'));
insert into comenzi_oiv (id_comanda, id_client, nr_card, id_adresa_livrare, data_comanda)
values ('comanda_'|| s_id_comanda.nextval, 'client_3', '123459', 'adr_1', to_date('19-12-2021', 'dd-mm-
yyyy'));
insert into comenzi_oiv (id_comanda, id_client, nr_card, id_adresa_livrare) -- data_comenzii va lua valoarea
default
values ('comanda_'|| s_id_comanda.nextval, 'client_2', '123458', 'adr_2');
insert into comenzi_oiv (id_comanda, id_client, nr_card, id_adresa_livrare) -- data_comenzii va lua valoarea
default
values ('comanda_'|| s_id_comanda.nextval, 'client_1', '123490', 'adr_2');

savepoint adaugat_comenzi;

-- inserare in tabelul produse_comandate_oiv
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_1', 'prod_2', 12);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_1', 'prod_3', 20);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_2', 'prod_1', 30);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_2', 'prod_7', 24);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_2', 'prod_5', 10);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_3', 'prod_5', 25);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_4', 'prod_4', 22);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_5', 'prod_2', 40);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_5', 'prod_6', 20);
insert into produse_comandate_oiv (id_comanda, id_produc, cantitate)
values ('comanda_6', 'prod_7', 10);

savepoint adaugat_produse_comandate;
```

```
commit;
```

G) 15 interogari

```
-- Interogarea 1
-- Sa se afiseze denumirea si pretul fiecarui produs comandat din comanda a carui numar este dat de la
tastatura
select denumire, pret
from produse_oiv
where id_produs in (select id_produs
                    from produse_comandate_oiv
                    where id_comanda = 'comanda_' || &nr_comanda);
-- subcerere in where

-- Interogarea 2
-- Sa se afiseze numarul comenzilor a caror valoare este egala sau depaseste o suma data de la tastatura.
select sum(cantitate * pret) suma, id_comanda
from produse_comandate_oiv prod_com join produse_oiv prod on (prod_com.id_produs = prod.id_produs)
group by id_comanda
having sum(cantitate * pret) >= &suma_minima;
-- sum, having, group by

-- Interogarea 3
-- Sa se afiseze alfabetic dupa numele de familie date despre clientii care au plasat comenzi in anul curent.
select id_client ID, nume Nume, nvl(prenume, '-') Prenume, nvl(varsta, 0) Varsta, nvl(orasi, '-') Oras
from clienti_oiv
where id_client in (select id_client
                   from comenzi_oiv
                   where to_char(data_comanda, 'YEAR') = to_char(sysdate, 'YEAR'))
order by nume;
-- nvl, subcerere in where, sysdate, to_char

-- Interogarea 4
-- Sa se afiseze denumirea produsului cu cea mai mare cantitate comandata.
select denumire "Cel mai comandat produs"
from produse_oiv
where id_produs in (
    select id_produs
    from produse_comandate_oiv
    where cantitate in (select max(cantitate) from produse_comandate_oiv)
);
-- max, subcerere in where
```

-- Interogarea 5

-- Sa se afiseze strada si numarul adreselor de livrare din Bucuresti si adresa completa pentru cele din alte orase

```
select id_adresa_livrare, decode(upper(oras), 'BUCURESTI', strada || decode(nvl(numar, -1), -1, ' (nu se cunoaste numarul strazii)', ', nr. ' || numar),  
      oras || ', str. ' || strada || decode(nvl(numar, -1), -1, ' (nu se cunoaste numarul strazii)', ', nr. ' || numar)) Adresa  
from adrese_livrare_oiv;  
-- decode, upper, nvl
```

-- Interogarea 6

-- Sa se afiseze date despre produsele cu pretul pretul mediu de achizitionare mai mic decat 50

-- de lei, excluzandu-le pe cele din categoria "Prajituri cu branza" sau pe cele fara categorie

```
select id_produc "ID produs", prod.denumire "Denumire produs", round(avg(pret_achizitie), 2) "Pret de achizitie mediu", cat.denumire "Denumire categorie"
```

from

```
(select id_produc  
from istoric_aprovizionare_oiv  
group by id_produc  
having round(avg(pret_achizitie), 2) < 50
```

minus

```
select id_produc  
from produse_oiv  
where id_categorie = (select unique id_categorie  
      from categorii_produce_oiv  
      where lower(denumire) = 'prajituri cu branza')  
or id_categorie is null) query_cerinta
```

```
join istoric_aprovizionare_oiv ist on (query_cerinta.id_produc = ist.id_produc)
```

```
join produse_oiv prod on (query_cerinta.id_produc = prod.id_produc)
```

```
join categorii_produce_oiv cat on (prod.id_categorie = cat.id_categorie)
```

```
group by id_produc, prod.denumire, cat.denumire;
```

-- round, avg, subcerere in from si in where, group by, lower, join

-- Interogare 7

-- Sa se afiseze denumirea furnizorilor de la care s-au comandat mai mult de 100 de produse

```
select id_furnizor "ID furnizor", sum(cantitate) "Total cantitate furnizata", (select denumire from  
furnizori_oiv where id_furnizor = ist.id_furnizor) Denumire  
from istoric_aprovizionare_oiv ist
```

```
group by id_furnizor
having sum(cantitate) > 100
order by sum(cantitate) desc;
-- subcerere in select, functia sum, order by

-- Interogarea 8
-- Sa se afiseze informatii despre clientii care locuiesc intr-un oras unde se afla si
-- unul din furnizorii magazinului sau care au varsta mai mare de decat media de varsta
-- a clientilor
select *
from clienti_oiv client
where oras in (select oras from furnizori_oiv where oras = client.oras)
union
select *
from clienti_oiv
where varsta > (select avg(varsta) from clienti_oiv);
-- union, subcerere in where, avg

-- Interogare 9
-- Sa se afiseze date despre produsele achizitionate de magazin in luna decembrie 2021
select (select denumire from furnizori_oiv where id_furnizor = ist.id_furnizor) "Denumire furnizor",
       prod.denumire "Denumire produs", ist.*
from istoric_aprovizionare_oiv ist join produse_oiv prod on (ist.id_produs = prod.id_produs)
where data_achizitie between to_date('01.12.2021', 'dd-mm-yyyy') and to_date('31.12.2021', 'dd-mm-yyyy');
-- subcerere in select, join, between dates, to_date

-- Interogare 10
-- Pentru toate produsele sa se afiseze pretul de vanzare, pentru produsul cu id-ul "prod_1"
-- sa se calculeze potentialul profit pe unitate de produs raportat la pretul de achizitie
-- maxim, iar pentru celelalte produse raportat la pretul de achizitie mediu
select pret, pret - nvl((select case(upper(id_produs))
                        when 'PROD_1' then max(pret_achizitie)
                        else
                        avg(pret_achizitie)
                        end
                        from istoric_aprovizionare_oiv
                        where id_produs = prod_ext.id_produs
                        group by id_produs), 0) Potential_profit
from produse_oiv prod_ext;
-- nvl, case, uper, max, avg, group by

-- Interogare 11
-- Sa se afiseze detalii despre cea mai recenta comanda.
```



```
select data_comanda,
      (select prenume || ' ' || upper(ume) from clienti_oiv where id_client = detalii_comanda.id_client) "Nume
client",
      (select oras || ', strada ' || strada || ', nr ' || numar from adrese_livrare_oiv where id_adresa_livrare =
detalii_comanda.id_adresa_livrare) Adresa
from (select *
      from comenzi_oiv
      order by data_comanda desc) detalii_comanda
where rownum < 2;
-- subcereri in select si from, upper, rownum

-- Interogarea 12
-- Sa se afiseze denumirea si pretul de vanzare pentru cel mai comandat produs de la furnizori.
select id_produc, denumire, pret
from (select id_produc, count(*) Nr_aparitii
      from istoric_aprovizionare_oiv
      group by id_produc) join produse_oiv using (id_produc)
where Nr_aparitii = (select max(Nr_aparitii)
                    from (select id_produc, count(*) Nr_aparitii
                          from istoric_aprovizionare_oiv
                          group by id_produc));
-- count, subcereri in from si where, max, group by

-- Interogarea 13
-- Sa se afiseze catitatea de produse ramase pe stoc dupa ce se vor onora tate comenzile
with produse_existente as (select id_produc, sum(cantitate) cantitate_existentia
                           from istoric_aprovizionare_oiv
                           group by id_produc),
produse_comandate as (select id_produc, sum(cantitate) cantitate_ceruta
                      from produse_comandate_oiv
                      group by id_produc)
select existente.id_produc, cantitate_existentia, cantitate_ceruta, (cantitate_existentia - cantitate_ceruta) "Stoc
potential"
from produse_existente existente join produse_comandate comandate on (existente.id_produc =
comandate.id_produc);
-- with, sum, group by, join

-- Interogarea 14
-- Sa se afiseze toate comenzile plasate in ultima luna, precum si numele clientului care le-a initiat
select data_comanda, (select prenume || ' ' || upper(ume)
                      from clienti_oiv
                      where id_client = comenzi.id_client) Client
from comenzi_oiv comenzi
```

```
where data_comanda between (add_months(sysdate, -1)) and sysdate;
-- subcerere in select, between dates, add_months, sysdate

-- Interogarea 15
-- Pentru fiecare comanda sa se afiseze suma totala.
select id_comanda, sum(cantitate * pret) total_comanda
from produse_comandate_oiv prod_com join produse_oiv prod on (prod_com.id_produc = prod.id_produc)
group by id_comanda;
-- sum, join

Interogarea 16
-- Afisarea id-urile tuturor produselor care au fost cumparate de la furnizori din Bucuresti.
select distinct id_produc
from istoric_aprovizionare_oiv ist1
where not exists (select 1
                  from furnizori_oiv fz
                  where upper(oras) = 'BUCURESTI'
                  and not exists (select 'x'
                                  from istoric_aprovizionare_oiv ist2
                                  where fz.id_furnizor = ist2.id_furnizor and ist2.id_produc = ist1.id_produc));
-- division, not exists, subcerere in where, upper, distinct
```

H) Tabel mesaje

```
create table messages_oiv (
  message_id number(5),
  message varchar2(255),
  message_type varchar2(1),
  created_by varchar2(40) not null,
  created_at date not null,
  primary key (message_id)
);
```

I) și J) P L/SQL

```
SET SERVEROUTPUT ON

-- subprogram stocat care sa utilizeze 2 tipuri de colectii
-- Sa se creeze o procedura care primeste un id de categorie si insereaza in produse_oiv
-- duplicate ale produselor din respectiva categorie cu numele actualizat. Se vor afisa
-- numele produselor initiale, precum si numele celor inserate cu nume actualizat.

create or replace procedure afisare_denum_prod_oiv (p_categorie categorii_produse_oiv.id_categorie%type)
is
  type denumiri_prajituri_vector is table of produse_oiv.denumire%type index by BINARY_INTEGER;
```

```

denumiri_prajituri denumiri_prajituri_vector;

type prajituri_vector is table of produse_oiv%rowtype index by BINARY_INTEGER;
prajituri prajituri_vector;
begin
select denumire bulk collect into denumiri_prajituri
from produse_oiv
where id_categorie = p_categorie;

dbms_output.put_line('Prajiturile din categoria ' || p_categorie || ': ');
for i in denumiri_prajituri.first..denumiri_prajituri.last loop
    dbms_output.put_line(denumiri_prajituri(i));
end loop;

select * bulk collect into prajituri
from produse_oiv
where id_categorie = p_categorie;

dbms_output.put_line('Prajiturile din categoria ' || p_categorie || ' cu numele actualizat: ');
for i in prajituri.first..prajituri.last loop
    prajituri(i).denumire := prajituri(i).denumire || '_updated';

insert into produse_oiv (id_produs, denumire, pret, id_categorie)
values ('prod_'|| s_id_produs.nextval, prajituri(i).denumire, prajituri(i).pret, p_categorie);

    dbms_output.put_line(prajituri(i).denumire);
end loop;
end;
/

-- apelare
begin
    afisare_denum_prod_oiv('cat_1');
end;
/

-- verificare
select count(*) from produse_oiv where denumire like '%_updated';

-- subprogram stocat care sa foloseasca cursor
-- Procedura care maresta pretul produselor,
-- care apartin unei anumite categorii cu id-ul dat ca parametru,
-- cu un anumit procent dat ca parametru

create or replace procedure cursor_oiv (p_categorie categorii_produce_oiv.id_categorie%type,
p_procent number)

```

```
is
  CURSOR c_id_produce is (select id_produc from produse_oiv where id_categorie = p_categorie);
  v_id_produc_curent produse_oiv.id_produc%type;
begin
  open c_id_produce;

  loop
    fetch c_id_produce into v_id_produc_curent;
    exit when c_id_produce%notfound;

    update produse_oiv
    set pret = pret * (1 + p_procent)
    where id_produc = v_id_produc_curent;
  end loop;

  close c_id_produce;
end;
/

-- apelare
begin
  cursor_oiv('cat_1', 0.5);
end;
/

-- functie care sa contina o comanda SQL care foloseste 3 tabele + tratare exceptii
-- numarul clientilor care sunt in categoria de nume 'V' si care au avut comenzi mai mari
-- de o valoare data ca parametru.
create or replace function nr_clienti_oiv (p_litera varchar, p_suma_minima number) return number
is
  v_nr_clienti number(3);

  exp_litera exception;
  exp_suma_minima exception;

  eroare_cod varchar2(1);
  eroare_mesaj varchar2(255);
begin

  if (length(p_litera) <> 1) then
    raise exp_litera;
  end if;

  if (p_suma_minima <= 0) then
    raise exp_suma_minima;
  end if;
```

```
select count(*) into v_nr_clienti
from comenzi_oiv com join clienti_oiv cl on (com.id_client = cl.id_client)
  join (select id_comanda, sum(cantitate * pret) suma
        from produse_comandate_oiv prod_com join produse_oiv prod on (prod_com.id_produ =
prod.id_produ)
        group by id_comanda) total_pret on (com.id_comanda = total_pret.id_comanda)
where upper(cl.num) like (upper(p_litera) || '%') and total_pret.suma > p_suma_minima;

return v_nr_clienti;
exception
when exp_litera then
  eroare_cod := 'E';
  eroare_mesaj := 'Introduceti doar prima litera pentru numele clientilor';
  insert into messages_oiv (message_id, message, message_type, created_by, created_at)
  values(s_id_message.nextval, eroare_mesaj, eroare_cod, user, sysdate);
  return -1;
when exp_suma_minima then
  eroare_cod := 'E';
  eroare_mesaj := 'Suma minima trebuie sa fie mai mare decat zero';
  insert into messages_oiv (message_id, message, message_type, created_by, created_at)
  values(s_id_message.nextval, eroare_mesaj, eroare_cod, user, sysdate);
  return -1;
when others then
  eroare_cod := sqlcode;
  eroare_mesaj := substr(sqlerrm, 1, 100);
  insert into messages_oiv (message_id, message, message_type, created_by, created_at)
  values(s_id_message.nextval, eroare_mesaj, eroare_cod, user, sysdate);
  return -1;
end;
/

-- apelare
begin
  dbms_output.put_line(nr_clienti_oiv('Va', 400));
  dbms_output.put_line(nr_clienti_oiv('V', -400));
  dbms_output.put_line(nr_clienti_oiv('V', 400));
end;
/

-- Trigger LMD la nivel de comanda
-- Magazinul se aprovizioneaza doar in zilele de luni si miercuri, in intervalul
-- orar 09:00 - 12:00. Inainte de inserarea de randuri in tabelul istoric_aprovizionare_oiv
-- sa se verifice ca este respectata aceasta constrangere.
create or replace trigger verific_data_oiv
before insert on istoric_aprovizionare_oiv
```

```
begin
  if ((to_char(sysdate, 'd') not in ('2', '4')) or (to_char(sysdate, 'hh24') not between '09' and '13')) then
    insert into messages_oiv (message_id, message, message_type, created_by, created_at)
      values(s_id_message.nextval, to_char(sysdate, 'dd-mm-yyyy, hh') || ' - In acest moment nu se pot realiza
achizitii', 'E', user, sysdate);
    raise_application_error(-20500, 'In acest moment nu se pot realiza achizitii');
  end if;
end;
/

-- verificare
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, pret_achizitie)
values ('prod_1', 'fz_1', 10, 1);

-- Trigger LMD la nivel de linie
-- La inserarea sau actualizare in tabelul istoric_aprovizionare_oiv sa se
-- verifice ca pretul de achizitie nu este mai mare decat pretul de vanzare
create or replace trigger verific_pret_achizitie_oiv
before insert on istoric_aprovizionare_oiv
for each row
declare
  v_pret_vanzare produse_oiv.pret%type;
begin
  select pret into v_pret_vanzare
  from produse_oiv
  where id_produs = :NEW.id_produs;

  if (:NEW.pret_achizitie >= v_pret_vanzare) then
    insert into messages_oiv (message_id, message, message_type, created_by, created_at)
      values(s_id_message.nextval, 'Eroare insert: pret achizitie > pret vanzare', 'E', user, sysdate);

    raise_application_error(-20500, 'Pretul de achizitie este mai mare decat cel de vanzare pentru produsul cu
id-ul ' || :NEW.id_produs);
  end if;
end;
/

-- verificare
insert into istoric_aprovizionare_oiv (id_produs, id_furnizor, cantitate, data_achizitie, pret_achizitie)
values ('prod_1', 'fz_1', 10, to_date('20-12-2021', 'dd-mm-yyyy'), 15);

-- Trigger LDD
-- Sa se implementeze functionalitatea ca niciun tabel nu poate fi sters in zilele de weekend

create or replace trigger ldd_trigger_oiv
```

```
before drop on database
begin
  if (to_char(sysdate, 'd') in ('7', '1')) then
    dbms_output.put_line('Nu se pot sterge tabele in weekend.');
```

end if;

```
end;
/

-- Pachet c
create or replace package magazin_pkg_oiv is
  procedure afisare_denum_prod_oiv (p_categorie categorii_produce_oiv.id_categorie%type);
  procedure cursor_oiv (p_categorie categorii_produce_oiv.id_categorie%type, p_procent number);
  function nr_clienti_oiv (p_litera varchar, p_suma_minima number) return number;
end magazin_pkg_oiv;
/

create or replace package body magazin_pkg_oiv is

  procedure afisare_denum_prod_oiv (p_categorie categorii_produce_oiv.id_categorie%type)
  is
    type denumiri_prajituri_vector is table of produse_oiv.denumire%type index by BINARY_INTEGER;
    denumiri_prajituri denumiri_prajituri_vector;

    type prajituri_vector is table of produse_oiv%rowtype index by BINARY_INTEGER;
    prajituri prajituri_vector;
  begin
    select denumire bulk collect into denumiri_prajituri
    from produse_oiv
    where id_categorie = p_categorie;

    dbms_output.put_line('Prajiturile din categoria ' || p_categorie || ': ');
    for i in denumiri_prajituri.first..denumiri_prajituri.last loop
      dbms_output.put_line(denumiri_prajituri(i));
    end loop;

    select * bulk collect into prajituri
    from produse_oiv
    where id_categorie = p_categorie;

    dbms_output.put_line('Prajiturile din categoria ' || p_categorie || ' cu numele actualizat: ');
    for i in prajituri.first..prajituri.last loop
      prajituri(i).denumire := prajituri(i).denumire || '_updated';

      insert into produse_oiv (id_produs, denumire, pret, id_categorie)
      values ('prod_' || s_id_produs.nextval, prajituri(i).denumire, prajituri(i).pret, p_categorie);
```

```
        dbms_output.put_line(prajituri(i).denumire);
    end loop;
end afisare_denum_prod_oiv;

procedure cursor_oiv (p_categorie categorii_produce_oiv.id_categorie%type,
                     p_procent number)
is
    CURSOR c_id_produce is (select id_produc from produce_oiv where id_categorie = p_categorie);
    v_id_produc_curent produce_oiv.id_produc%type;
begin
    open c_id_produce;

    loop
        fetch c_id_produce into v_id_produc_curent;
        exit when c_id_produce%notfound;

        update produce_oiv
        set pret = pret * (1 + p_procent)
        where id_produc = v_id_produc_curent;
    end loop;

    close c_id_produce;
end cursor_oiv;

function nr_clienti_oiv (p_litera varchar, p_suma_minima number) return number
is
    v_nr_clienti number(3);

    exp_litera exception;
    exp_suma_minima exception;

    eroare_cod varchar2(1);
    eroare_mesaj varchar2(255);
begin

    if (length(p_litera) <> 1) then
        raise exp_litera;
    end if;

    if (p_suma_minima <= 0) then
        raise exp_suma_minima;
    end if;

    select count(*) into v_nr_clienti
    from comenzi_oiv com join clienti_oiv cl on (com.id_client = cl.id_client)
    join (select id_comanda, sum(cantitate * pret) suma
```



```

        from produse_comandate_oiv prod_com join produse_oiv prod on (prod_com.id_produs =
prod.id_produs)
        group by id_comanda) total_pret on (com.id_comanda = total_pret.id_comanda)
where upper(cl.nume) like (upper(p_litera) || '%') and total_pret.suma > p_suma_minima;

return v_nr_clienti;
exception
when exp_litera then
    eroare_cod := 'E';
    eroare_mesaj := 'Introduceti doar prima litera pentru numele clientilor';
    insert into messages_oiv (message_id, message, message_type, created_by, created_at)
    values(s_id_message.nextval, eroare_mesaj, eroare_cod, user, sysdate);
    return -1;
when exp_suma_minima then
    eroare_cod := 'E';
    eroare_mesaj := 'Suma minima trebuie sa fie mai mare decat zero';
    insert into messages_oiv (message_id, message, message_type, created_by, created_at)
    values(s_id_message.nextval, eroare_mesaj, eroare_cod, user, sysdate);
    return -1;
when others then
    eroare_cod := sqlcode;
    eroare_mesaj := substr(sqlerrm, 1, 100);
    insert into messages_oiv (message_id, message, message_type, created_by, created_at)
    values(s_id_message.nextval, eroare_mesaj, eroare_cod, user, sysdate);
    return -1;
end nr_clienti_oiv;

end magazin_pkg_oiv;
/

-- verificare
execute magazin_pkg_oiv.afisare_denum_prod_oiv('cat_1');
execute magazin_pkg_oiv.cursor_oiv('cat_1', 0.5);
execute dbms_output.put_line(magazin_pkg_oiv.nr_clienti_oiv('Va', 400));
execute dbms_output.put_line(magazin_pkg_oiv.nr_clienti_oiv('V', -400));
execute dbms_output.put_line(magazin_pkg_oiv.nr_clienti_oiv('V', 400));

```

K) Ștergere date

```

-- Ștergere secvente
--drop sequence s_id_adresa_livrare;
--drop sequence s_id_client;
--drop sequence s_id_furnizor;
--drop sequence s_id_produs;

```

```
--drop sequence s_id_categorie;  
--drop sequence s_id_comanda;  
  
-- Stergere tabele  
--drop table istoric_aprovizionare_oiv;  
--drop table produse_comandate_oiv;  
--drop table comenzi_oiv;  
  
--drop table adrese_livrare_oiv;  
--drop table clienti_oiv;  
--drop table produse_oiv;  
--drop table categorii_produce_oiv;  
--drop table detalii_plata_oiv;  
--drop table furnizori_oiv;  
  
--drop package magazin_pkg_oiv;  
--drop procedure afisare_denum_prod_oiv;  
--drop procedure cursor_oiv;  
--drop function nr_clienti_oiv;  
--drop trigger verif_data_oiv;  
--drop trigger verif_pret_achizitie_oiv;  
--drop trigger ldd_trigger_oiv;
```