

# GESTIUNEA UNEI FLORARII

-Proiect SGBD-

INFORMATICA- Anul 2

Dogareci Bianca-Alexandra

Grupa 243

## **Cerinta 1:**

**Prezentati pe scurt baza de date (utilitatea ei).**

### **Descrierea modelului real si al utilitatii lui:**

Pentru acest proiect tema pe care am ales-o este o florarie care se ocupa cu vanzarea de produse online, adica vanzarea unor diferite tipuri de flori, a unor ingrasaminte pentru plante etc. In baza de date a florariei sunt stocate informatii esentiale, importante cu privire la produse, clienti, comenziile plasate de clienti, angajati s.a.m.d.

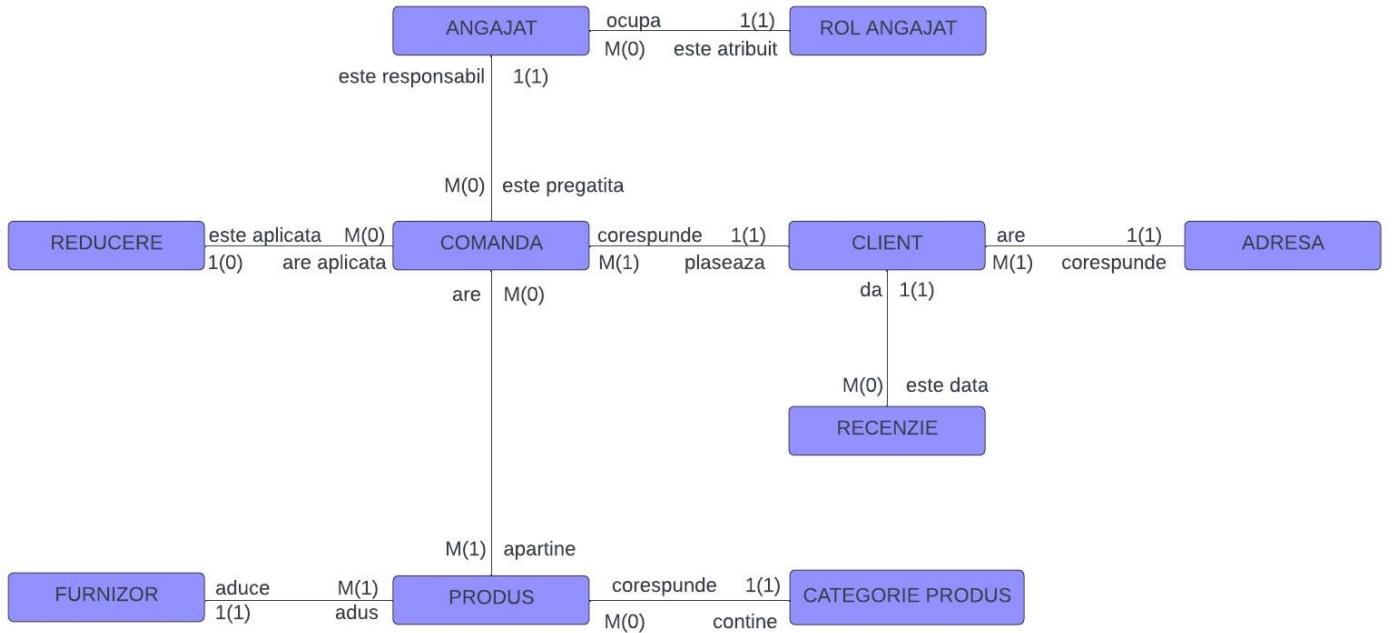
Utilitatea bazei de date consta in capacitatea de a gestiona informatiile, de a putea urmari si analiza comportamentul clientilor, de a gestiona stocurile, de a optimiza procesul de vanzare, de a putea vedea ce angajat se ocupa de o anumita comanda si multe altele.

### **Reguli de functionare:**

- Angajatii pot ocupa un singur rol.
- Roulurile pot fi atribuite mai multor angajati.
- Un angajat poate fi responsabil de mai multe comenzi.
- O comanda este pregatita de un singur angajat.
- Un client poate pesta mai multe comenzi.
- O comanda poate corespunde unui singur client.
- O adresa poate corespunde mai multor clienti.
- Un client poate avea numai o adresa.
- Un client poate da mai multe recenzii.
- O recenzie poate fi a unui singur client.
- O comanda poate avea mai multe produse.
- Un produs poate apartine mai multor comenzi.
- Un furnizor poate aduce mai multe produse.
- Un produs poate fi adus numai de un furnizor.
- Un produs corespunde unei singure categorii de produse.
- O categorie de produse poate contine mai multe produse.
- O comanda poate avea aplicata o singura reducere.
- O reducere poate fi aplicata mai multor comenzi.

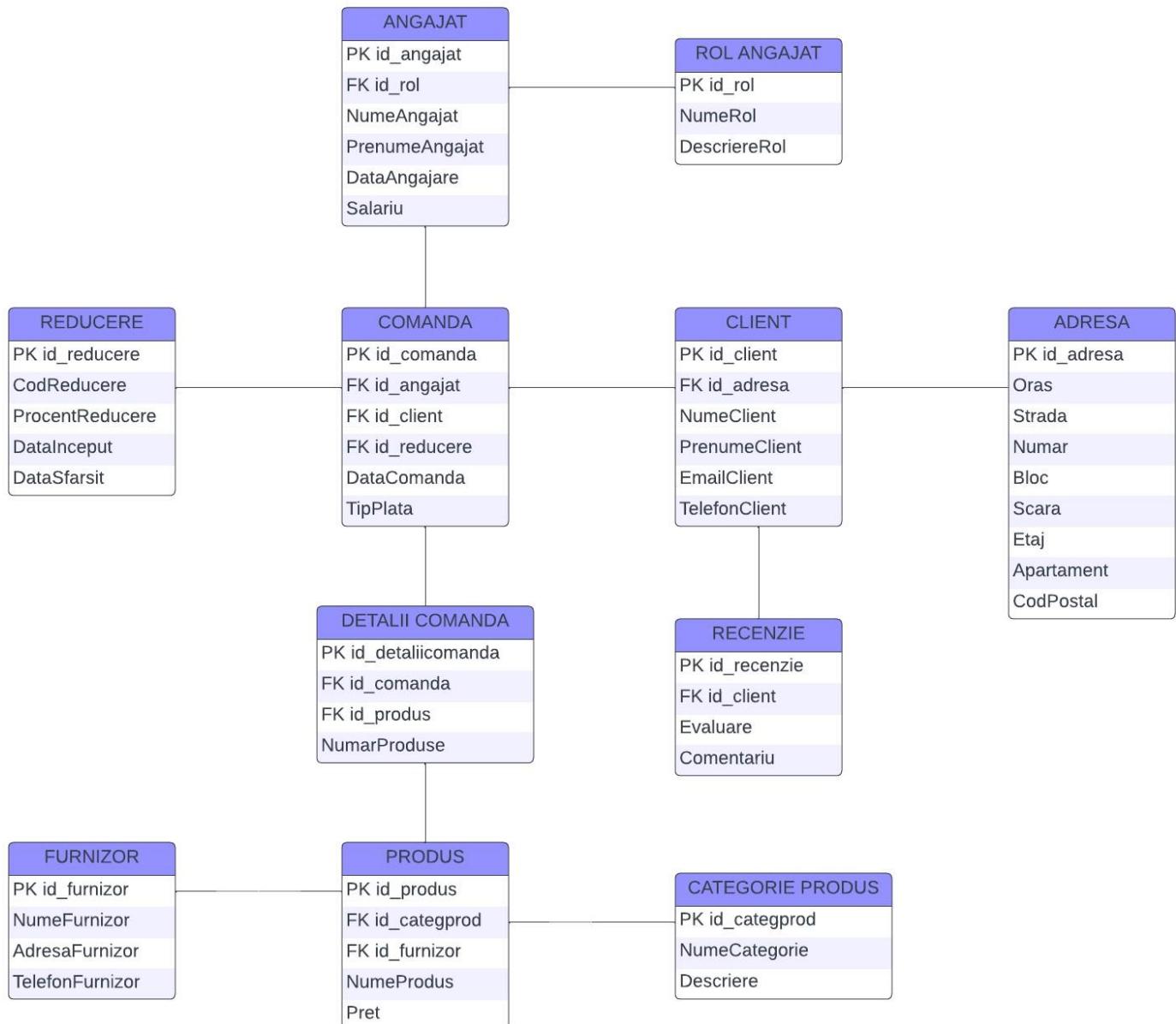
## Cerinta 2:

Realizati diagrama entitate-relatie (ERD): entitatile, relatiile si atributele trebuie definite in limba romana (vezi curs SGBD/ model de diagrama ERD; nu se va accepta alt format).



### Cerinta 3:

Pornind de la diagrama entitate-relatie realizati diagrama conceptuala a modelului propus, integrand toate atributele necesare: entitatile, relatiile si atributele trebuie definite in limba romana.



#### Cerinta 4:

Implementati in Oracle diagrama conceptuala realizata: definiti toate tabelele, definind toate constrangerile de integritate necesare (chei primare, chei externe etc.)

- **TABELA ROL\_ANGAJAT:**

```
create table ROL_ANGAJAT(
    id_rol number(5) constraint pkey_rol primary key,
    NumeRol varchar(20) constraint NumeRol not null,
    DescriereRol varchar(100) constraint DescriereRol null
);
```

```
create table ROL_ANGAJAT(
    id_rol number(5) constraint pkey_rol primary key,
    NumeRol varchar(20) constraint NumeRol not null,
    DescriereRol varchar(100) constraint DescriereRol null
);
```

Table ROL\_ANGAJAT created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ROL	NUMBER(5,0)	No	(null)	1 (null)	
2 NUMEROL	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3 DESCRIEREROL	VARCHAR2(100 BYTE)	Yes	(null)	3 (null)	

- **TABELA ANGAJAT:**

```
create table ANGAJAT(
    id_angajat number(5) constraint pkey_angajat primary key,
    id_rol number(5), constraint fk_angajat_rol foreign key(id_rol) references
ROL_ANGAJAT(id_rol),
    NumeAngajat varchar(30) constraint NumeAngajat not null,
    PrenumeAngajat varchar(30) constraint PrenumeAngajat not null,
    DataAngajare date constraint DataAngajare not null,
    Salariu number(10) constraint Salariu not null
);
```

```

create table ANGAJAT(
    id_angajat number(5) constraint pkey_angajat primary key,
    id_rol number(5), constraint fk_angajat_rol foreign key(id_rol) references ROL_ANGAJAT(id_rol),
    NumeAngajat varchar(30) constraint NumeAngajat not null,
    PrenumeAngajat varchar(30) constraint PrenumeAngajat not null,
    DataAngajare date constraint DataAngajare not null,
    Salariu number(10) constraint Salariu not null
);

```

Table ANGAJAT created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ANGAJAT	NUMBER(5,0)	No	(null)	1	(null)
2 ID_ROL	NUMBER(5,0)	Yes	(null)	2	(null)
3 NUMEANGAJAT	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4 PRENUMEANGAJAT	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5 DATAANGAJARE	DATE	No	(null)	5	(null)
6 SALARIU	NUMBER(10,0)	No	(null)	6	(null)

- **TABELA ADRESA:**

```
create table ADRESA(
```

```

    id_adresa number(5) constraint pkey_adresa primary key,
    Oras varchar(20) constraint Oras not null,
    Strada varchar(30) constraint Strada not null,
    Numar number(5) constraint Numar not null,
    Bloc varchar(20) constraint Bloc null,
    Scara number(5) constraint Scara null,
    Etaj number(5) constraint Etaj null,
    Apartament number(5) constraint Apartament null,
    CodPostal varchar(20) constraint CodPostal not null
);
```

```

create table ADRESA(
    id_adresa number(5) constraint pkey_adresa primary key,
    Oras varchar(20) constraint Oras not null,
    Strada varchar(30) constraint Strada not null,
    Numar number(5) constraint Numar not null,
    Bloc varchar(20) constraint Bloc null,
    Scara number(5) constraint Scara null,
    Etaj number(5) constraint Etaj null,
    Apartament number(5) constraint Apartament null,
    CodPostal varchar(20) constraint CodPostal not null
);
```

Table ADRESA created.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ADRESA	NUMBER(5,0)	No	(null)	1	(null)
2	ORAS	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	STRADA	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4	NUMAR	NUMBER(5,0)	No	(null)	4	(null)
5	BLOC	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
6	SCARA	NUMBER(5,0)	Yes	(null)	6	(null)
7	ETAJ	NUMBER(5,0)	Yes	(null)	7	(null)
8	APARTAMENT	NUMBER(5,0)	Yes	(null)	8	(null)
9	CODPOSTAL	VARCHAR2(20 BYTE)	No	(null)	9	(null)

- **TABELA CLIENT:**

create table CLIENT(

```

    id_client number(5) constraint pkey_client primary key,
    id_adresa number(5), constraint fk_client_adresa foreign key(id_adresa)
references ADRESA(id_adresa),
    NumeClient varchar(30) constraint NumeClient not null,
    PrenumeClient varchar(30) constraint PrenumeClient not null,
    EmailClient varchar(30) constraint EmailClient not null unique,
    TelefonClient varchar(30) constraint TelefonClient not null unique
);
```

```
create table CLIENT(
    id_client number(5) constraint pkey_client primary key,
    id_adresa number(5), constraint fk_client_adresa foreign key(id_adresa) references ADRESA(id_adresa),
    NumeClient varchar(30) constraint NumeClient not null,
    PrenumeClient varchar(30) constraint PrenumeClient not null,
    EmailClient varchar(30) constraint EmailClient not null unique,
    TelefonClient varchar(30) constraint TelefonClient not null unique
);
```

Table CLIENT created.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_CLIENT	NUMBER(5,0)	No	(null)	1	(null)
2	ID_ADRESA	NUMBER(5,0)	Yes	(null)	2	(null)
3	NUMECLIENT	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4	PRENUMECLIENT	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5	EMAILCLIENT	VARCHAR2(30 BYTE)	No	(null)	5	(null)
6	TELEFONCLIENT	VARCHAR2(30 BYTE)	No	(null)	6	(null)

- **TABELA RECENZIE:**

```

create table RECENZIE(
    id_recenzie number(5) constraint pkey_recenzie primary key,
    id_client number(5), constraint fk_recenzie_client foreign key(id_client)
references CLIENT(id_client),
    Evaluare number(2) constraint Evaluare not null,
    Comentariu varchar(100) constraint Comentariu null
);
create table RECENZIE(
    id_recenzie number(5) constraint pkey_recenzie primary key,
    id_client number(5), constraint fk_recenzie_client foreign key(id_client) references CLIENT(id_client),
    Evaluare number(2) constraint Evaluare not null,
    Comentariu varchar(100) constraint Comentariu null
);

```

Table RECENZIE created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_RECENZIE	NUMBER(5,0)	No	(null)	1	(null)
2 ID_CLIENT	NUMBER(5,0)	Yes	(null)	2	(null)
3 EVALUARE	NUMBER(2,0)	No	(null)	3	(null)
4 COMENTARIU	VARCHAR2(100 BYTE)	Yes	(null)	4	(null)

- **TABELA REDUCERE:**

```

CREATE TABLE REDUCERE(
    id_reducere number(5) constraint pkey_reducere primary key,
    CodReducere varchar(255) constraint CodReducere not null unique,
    ProcentReducere decimal(5,2) constraint ProcentReducere not null,
    DataInceput date constraint DataInceput not null,
    DataSfarsit date constraint DataSfarsit not null
);
CREATE TABLE REDUCERE(
    id_reducere number(5) constraint pkey_reducere primary key,
    CodReducere varchar(255) constraint CodReducere not null unique,
    ProcentReducere decimal(5,2) constraint ProcentReducere not null,
    DataInceput date constraint DataInceput not null,
    DataSfarsit date constraint DataSfarsit not null
);

```

Table REDUCERE created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_REDUCERE	NUMBER(5,0)	No	(null)	1	(null)
2 CODREDUCERE	VARCHAR2(255 BYTE)	No	(null)	2	(null)
3 PROCENTREDUCERE	NUMBER(5,2)	No	(null)	3	(null)
4 DATAINCEPUT	DATE	No	(null)	4	(null)
5 DATASFARSIT	DATE	No	(null)	5	(null)

- **TABELA COMANDA:**

create table COMANDA(

    id\_comanda number(5) constraint pkey\_comanda primary key,  
     id\_angajat number(5), constraint fk\_comanda\_angajat foreign key(id\_angajat)  
     references ANGAJAT(id\_angajat),

    id\_client number(5), constraint fk\_comanda\_client foreign key(id\_client)  
     references CLIENT(id\_client),

    id\_reducere number(5), constraint fk\_comanda\_reducere foreign  
     key(id\_reducere) references REDUCERE(id\_reducere),

    DataComanda date constraint DataComanda not null,

    TipPlata varchar(10) default 'cash' constraint TipPlata not null

);

```
create table COMANDA(
    id_comanda number(5) constraint pkey_comanda primary key,
    id_angajat number(5), constraint fk_comanda_angajat foreign key(id_angajat) references ANGAJAT(id_angajat),
    id_client number(5), constraint fk_comanda_client foreign key(id_client) references CLIENT(id_client),
    id_reducere number(5), constraint fk_comanda_reducere foreign key(id_reducere) references REDUCERE(id_reducere),
    DataComanda date constraint DataComanda not null,
    TipPlata varchar(10) default 'cash' constraint TipPlata not null
);
```

Table COMANDA created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_COMANDA	NUMBER(5,0)	No	(null)	1	(null)
2 ID_ANGAJAT	NUMBER(5,0)	Yes	(null)	2	(null)
3 ID_CLIENT	NUMBER(5,0)	Yes	(null)	3	(null)
4 ID_REDUCERE	NUMBER(5,0)	Yes	(null)	4	(null)
5 DATACOMANDA	DATE	No	(null)	5	(null)
6 TIPPLATA	VARCHAR2(10 BYTE)	No	'cash'	6	(null)

- **TABELA CATEGORIE\_PRODUS:**

```
create table CATEGORIE_PRODUS(
    id_categprod number(5) constraint pkey_categprod primary key,
    NumeCategoria varchar(20) constraint NumeCategoria not null,
    Descriere varchar(100) constraint Descriere null
);
create table CATEGORIE_PRODUS(
    id_categprod number(5) constraint pkey_categprod primary key,
    NumeCategoria varchar(20) constraint NumeCategoria not null,
    Descriere varchar(100) constraint Descriere null
);
```

Table CATEGORIE\_PRODUS created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_CATEGPROD	NUMBER(5,0)	No	(null)	1	(null)
2 NUMECATEGORIE	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3 DESCRIERE	VARCHAR2(100 BYTE)	Yes	(null)	3	(null)

- **TABELA FURNIZOR:**

```
create table FURNIZOR(
    id_furnizor number(5) constraint pkey_furnizor primary key,
    NumeFurnizor varchar(30) constraint NumeFurnizor not null,
    AdresaFurnizor varchar(100) constraint AdresaFurnizor not null unique,
    TelefonFurnizor varchar(10) constraint TelefonFurnizor not null unique
);
create table FURNIZOR(
    id_furnizor number(5) constraint pkey_furnizor primary key,
    NumeFurnizor varchar(30) constraint NumeFurnizor not null,
    AdresaFurnizor varchar(100) constraint AdresaFurnizor not null unique,
    TelefonFurnizor varchar(10) constraint TelefonFurnizor not null unique
);
```

Table FURNIZOR created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_FURNIZOR	NUMBER(5,0)	No	(null)	1	(null)
2 NUMEFURNIZOR	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3 ADRESAFURNIZOR	VARCHAR2(100 BYTE)	No	(null)	3	(null)
4 TELEFONFURNIZOR	VARCHAR2(10 BYTE)	No	(null)	4	(null)

- **TABELA PRODUS:**

```

create table PRODUS(
    id_produs number(5) constraint pkey_produs primary key,
    id_categprod number(5), constraint fk_produs_categprod foreign
key(id_categprod) references CATEGORIE_PRODUS(id_categprod),
    id_furnizor number(5), constraint fk_produs_furnizor foreign key(id_furnizor)
references FURNIZOR(id_furnizor),
    NumeProdus varchar(30) constraint NumeProdus not null unique,
    Pret number(5) constraint Pret not null
);

```

```

create table PRODUS(
    id_produs number(5) constraint pkey_produs primary key,
    id_categprod number(5), constraint fk_produs_categprod foreign key(id_categprod) references CATEGORIE_PRODUS(id_categprod),
    id_furnizor number(5), constraint fk_produs_furnizor foreign key(id_furnizor) references FURNIZOR(id_furnizor),
    NumeProdus varchar(30) constraint NumeProdus not null unique,
    Pret number(5) constraint Pret not null
);

```

Table PRODUS created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_PRODUS	NUMBER(5,0)	No	(null)	1	(null)
2 ID_CATEGPROD	NUMBER(5,0)	Yes	(null)	2	(null)
3 ID_FURNIZOR	NUMBER(5,0)	Yes	(null)	3	(null)
4 NUMEPRODUS	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5 PRET	NUMBER(5,0)	No	(null)	5	(null)

- **TABELA DETALII\_COMANDA:**

```

create table DETALII_COMANDA(
    id_detaliiicomanda number(5) constraint pkey_detaliiicomanda primary key,
    id_comanda number(5), constraint fk_detaliiicomanda_comanda foreign
key(id_comanda) references COMANDA(id_comanda),
    id_produs number(5), constraint fk_detaliiicomanda_produs foreign
key(id_produs) references PRODUS(id_produs),
    NumarProduse number(5) constraint NumarProduse not null
);

```

```

create table DETALII_COMANDA(
    id_detaliiicomanda number(5) constraint pkey_detaliiicomanda primary key,
    id_comanda number(5), constraint fk_detaliiicomanda_comanda foreign key(id_comanda) references COMANDA(id_comanda),
    id_produs number(5), constraint fk_detaliiicomanda_produs foreign key(id_produs) references PRODUS(id_produs),
    NumarProduse number(5) constraint NumarProduse not null
);

```

| Table DETALII\_COMANDA created.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_DETALIICOMANDA	NUMBER(5,0)	No	(null)	1	(null)
2 ID_COMANDA	NUMBER(5,0)	Yes	(null)	2	(null)
3 ID_PRODUS	NUMBER(5,0)	Yes	(null)	3	(null)
4 NUMARPRODUSE	NUMBER(5,0)	No	(null)	4	(null)

### **Cerinta 5:**

Adaugati informatii coerente in tabelele create (minim 5 inregistrari pentru fiecare entitate independenta; minim 10 inregistrari pentru tabela asociativa).

Se creeaza o secventa care va fi utilizata la inserarea inregistrarilor in tabele.

create sequence id\_seq

start with 1

increment by 1

nocache

nocycle;

```
| create sequence id_seq
|   start with 1
|   increment by 1
|   nocache
|   nocycle;
```

| Sequence ID\_SEQ created.

- **TABELA ROL\_ANGAJAT:**

insert into ROL\_ANGAJAT

values(id\_seq.nextval,'Florar','Angajatul responsabil cu aranjarea buchetelor și compozițiilor florale.');

insert into ROL\_ANGAJAT

values(id\_seq.nextval,'Manager','Angajatul cu rol de supervizare și coordonare a operatiunilor zilnice ale florariei.');

insert into ROL\_ANGAJAT

values(id\_seq.nextval,'Specialist design','Angajatul cu expertiza în creatia de aranjamente florale pentru evenimente.');

```
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Vanzator','Angajatul care interactioneaza direct cu clientii.');
```

```
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Ingrijitor','Angajatul responsabil cu curatenia.');
```

```
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Specialist calitate','Angajatul care gestioneaza procesul de  
depozitare al florilor ca ele sa ramana proaspete.');
```

```
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Florar','Angajatul responsabil cu aranjarea buchetelor și componzițiilor florale.');//  
  
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Manager','Angajatul cu rol de supervizare și coordonare a operațiunilor zilnice ale florariei.');//  
  
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Specialist design','Angajatul cu expertiza în creația de aranjamente florale pentru evenimente.');//  
  
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Vanzator','Angajatul care interactioneaza direct cu clientii.');//  
  
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Ingrijitor','Angajatul responsabil cu curatenia.');//  
  
insert into ROL_ANGAJAT  
values(id_seq.nextval,'Specialist calitate','Angajatul care gestioneaza procesul de depozitare al florilor ca ele sa ramana proaspete.');//
```

ID_ROL	NUMEROL	DESCRIEREROL
1	1 Florar	Angajatul responsabil cu aranjarea buchetelor și componzițiilor florale.
2	2 Manager	Angajatul cu rol de supervizare și coordonare a operațiunilor zilnice ale florariei.
3	3 Specialist design	Angajatul cu expertiza în creația de aranjamente florale pentru evenimente.
4	4 Vanzator	Angajatul care interactioneaza direct cu clientii.
5	5 Ingrijitor	Angajatul responsabil cu curatenia.
6	10 Specialist calitate	Angajatul care gestioneaza procesul de depozitare al florilor ca ele sa ramana proaspete.

### • TABELA ANGAJAT:

```
insert into ANGAJAT  
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where  
NumeRol='Florar'),'Ionescu','Anastasia',to_date('16-08-2019','DD-MM-  
YYYY'),5000);
```

```
insert into ANGAJAT  
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where  
NumeRol='Florar'),'Florescu','Letitia',to_date('04-07-2019','DD-MM-YYYY'),5100);
```

```
insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where
NumeRol='Manager'),'Stanescu','Stefan',to_date('01-12-2005','DD-MM-
YYYY'),7000);
```

```
insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where
NumeRol='Specialist design'),'Dumitrache','Rafael',to_date('10-03-2019','DD-MM-
YYYY'),6200);
```

```
insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where
NumeRol='Specialist design'),'Teodorescu','Eleonora',to_date('15-04-2015','DD-
MM-YYYY'),6400);
```

```
insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where
NumeRol='Ingrijitor'),'Popescu','Ana',to_date('16-08-2015','DD-MM-YYYY'),4000);
```

```
insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where
NumeRol='Vanzator'),'Marin','Roxana',to_date('16-08-2020','DD-MM-YYYY'),5000);
```

```
insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Florar'),'Ionescu','Anastasia',to_date('16-08-2019','DD-MM-YYYY'),5000);

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Florar'),'Florescu','Letitia',to_date('04-07-2019','DD-MM-YYYY'),5100);

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Manager'),'Stanescu','Stefan',to_date('01-12-2005','DD-MM-YYYY'),7000);

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Specialist design'),'Dumitrache','Rafael',to_date('10-03-2019','DD-MM-YYYY'),6200);

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Specialist design'),'Teodorescu','Eleonora',to_date('15-04-2015','DD-MM-YYYY'),6400);

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Ingrijitor'),'Popescu','Ana',to_date('16-08-2015','DD-MM-YYYY'),4000);

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where NumeRol='Vanzator'),'Marin','Roxana',to_date('16-08-2020','DD-MM-YYYY'),5000);
```

	ID_ANGAJAT	ID_ROL	NUMEANGAJAT	PRENUMEANGAJAT	DATAANGAJARE	SALARIU
1	11	1	Ionescu	Anastasia	16-AUG-19	5000
2	12	1	Florescu	Letitia	04-JUL-19	5100
3	13	2	Stanescu	Stefan	01-DEC-05	7000
4	14	3	Dumitrache	Rafael	10-MAR-19	6200
5	15	3	Teodorescu	Eleonora	15-APR-15	6400
6	16	5	Popescu	Ana	16-AUG-15	4000
7	17	4	Marin	Roxana	16-AUG-20	5000

- **TABELA ADRESA:**

```
insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Str. Victoriei', 10, null, null, null, null, '010101');
```

```
insert into ADRESA
values (id_seq.nextval, 'Cluj-Napoca', 'Str. Bucuriei', 15, 'A', 2, 3, 7, '400000');
```

```
insert into ADRESA
values (id_seq.nextval, 'Timisoara', 'Str. Mihai Viteazu', 8, 'B3', 1, 6, 60, '300100');
```

```
insert into ADRESA
values (id_seq.nextval, 'Iasi', 'Str. Stefan Cel Mare', 25, null, null, null, null, '700300');
```

```
insert into ADRESA
values (id_seq.nextval, 'Constanta', 'Bd. Mamaia', 20, null, null, null, null, '900001');
```

```
insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Sos. Colentina', 11, 'C', 2, 6, 60, '405110');
```

```
insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Str. Simetriei', 16, null, null, null, null, '794620');
```

```
insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Str. Velintei', 20, null, null, null, null, '794264');
```

```
insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Str. Victoriei', 10, null, null, null, null, '010101');

insert into ADRESA
values (id_seq.nextval, 'Cluj-Napoca', 'Str. Bucuriei', 15, 'A', 2, 3, 7, '400000');

insert into ADRESA
values (id_seq.nextval, 'Timisoara', 'Str. Mihai Viteazu', 8, 'B3', 1, 6, 60, '300100');

insert into ADRESA
values (id_seq.nextval, 'Iasi', 'Str. Stefan Cel Mare', 25, null, null, null, null, '700300');

insert into ADRESA
values (id_seq.nextval, 'Constanta', 'Bd. Mamaia', 20, null, null, null, null, '900001');

insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Sos. Colentina', 11, 'C', 2, 6, 60, '405110');

insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Str. Simetriei', 16, null, null, null, null, '794620');

insert into ADRESA
values (id_seq.nextval, 'Bucuresti', 'Str. Velintei', 20, null, null, null, null, '794264');
```

	ID_ADRESA	ORAS	STRADA	NUMAR	BLOC	SCARA	ETAJ	APARTAMENT	CODPOSTAL
1	18	Bucuresti	Str. Victoriei	10 (null)	(null)	(null)		(null)	010101
2	19	Cluj-Napoca	Str. Bucuriei	15 A		2	3		7 400000
3	20	Timisoara	Str. Mihai Viteazu	8 B3		1	6		60 300100
4	21	Iasi	Str. Stefan Cel Mare	25 (null)	(null)	(null)		(null)	700300
5	22	Constanta	Bd. Mamaia	20 (null)	(null)	(null)		(null)	900001
6	23	Bucuresti	Sos. Colentina	11 C		2	6		60 405110
7	24	Bucuresti	Str. Simetriei	16 (null)	(null)	(null)		(null)	794620
8	25	Bucuresti	Str. Velintei	20 (null)	(null)	(null)		(null)	794264

### • TABELA CLIENT:

```
insert into CLIENT
values
(id_seq.nextval,18,'Popescu','Ioana','ioanapopescu@gmail.com','0744322188');
```

```
insert into CLIENT
values
(id_seq.nextval,19,'Apostol','Teodora','teodoraapostol@yahoo.com','0744325438')
;
```

```
insert into CLIENT
values
(id_seq.nextval,20,'Olteanu','Stefan','stefanolteanu@gmail.com','0734522188');
```

```
insert into CLIENT
values
(id_seq.nextval,21,'Tintea','Alexandru','alextintea@gmail.com','0744678388');
```

```
insert into CLIENT
values (id_seq.nextval,22,'Nica','Ana','ananica@ymail.com','0722722722');
```

```
insert into CLIENT
values
(id_seq.nextval,23,'Feurdean','Mihai','mihaifeur@gmail.com','0747777788');
```

```
insert into CLIENT
values
(id_seq.nextval,24,'Marinescu','Andrei','andreimarinescu@gmail.com','074734299
9');
```

```
insert into CLIENT
values
(id_seq.nextval,25,'Dogareci','Bianca','biancadogareci@gmail.com','0740017007');
```

```
insert into CLIENT
values
(id_seq.nextval,18,'Ionescu','Marian','ionescumarian@gmail.com','0776345967');
```

```

insert into CLIENT
values (id_seq.nextval,18,'Popescu','Ioana','ioanapopescu@gmail.com','0744322188');

insert into CLIENT
values (id_seq.nextval,19,'Apostol','Teodora','teodoraapostol@yahoo.com','0744325438');

insert into CLIENT
values (id_seq.nextval,20,'Olteanu','Stefan','stefanolteanu@gmail.com','0734522188');

insert into CLIENT
values (id_seq.nextval,21,'Tintea','Alexandru','alextintea@gmail.com','0744678388');

insert into CLIENT
values (id_seq.nextval,22,'Nica','Ana','ananica@ymail.com','0722722722');

insert into CLIENT
values (id_seq.nextval,23,'Feurdean','Mihai','mihaifeur@gmail.com','0747777788');

insert into CLIENT
values (id_seq.nextval,24,'Marinescu','Andrei','andreimarinescu@gmail.com','0747342999');

insert into CLIENT
values (id_seq.nextval,25,'Dogareci','Bianca','biancadogareci@gmail.com','0740017007');

insert into CLIENT
values (id_seq.nextval,18,'Ionescu','Marian','ionescumarian@gmail.com','0776345967');

```

	ID_CLIENT	ID_ADRESA	NUMECLIENT	PRENUMECLIENT	EMAILCLIENT	TELEFONCLIENT
1	26	18 Popescu	Ioana		ioanapopescu@gmail.com	0744322188
2	27	19 Apostol	Teodora		teodoraapostol@yahoo.com	0744325438
3	28	20 Olteanu	Stefan		stefanolteanu@gmail.com	0734522188
4	29	21 Tintea	Alexandru		alextintea@gmail.com	0744678388
5	30	22 Nica	Ana		ananica@ymail.com	0722722722
6	31	23 Feurdean	Mihai		mihaifeur@gmail.com	0747777788
7	32	24 Marinescu	Andrei		andreimarinescu@gmail.com	0747342999
8	33	25 Dogareci	Bianca		biancadogareci@gmail.com	0740017007
9	34	18 Ionescu	Marian		ionescumarian@gmail.com	0776345967

### • TABELA RECENZIE:

```

insert into RECENZIE
values (id_seq.nextval,(select id_client from CLIENT where
EmailClient='mihaifeur@gmail.com'),10,'Foarte multumit de servicii, personal
amabil.');

```

```
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where  
EmailClient='mihaifeur@gmail.com'),9,null);
```

```
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where  
EmailClient='ananica@ymail.com'),10,'Produse de calitate!');
```

```
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where  
EmailClient='ioanapopescu@gmail.com'),5,'Nu am fost impresionata de calitatea  
produselor.');
```

```
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where  
EmailClient='alextintea@gmail.com'),10,'Recomand cu incredere, flori  
proaspete.');
```

```
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where  
EmailClient='biancadogareci@gmail.com'),10,'Aranjamente foarte frumoase!');
```

```
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where EmailClient='mihaifeur@gmail.com'),10,'Foarte multumit de servicii, personal amabil.');//  
  
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where EmailClient='mihaifeur@gmail.com'),9,null);  
  
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where EmailClient='ananica@ymail.com'),10,'Produse de calitate!');//  
  
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where EmailClient='ioanapopescu@gmail.com'),5,'Nu am fost impresionata de calitatea produselor.');//  
  
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where EmailClient='alextintea@gmail.com'),10,'Recomand cu incredere, flori proaspete.');//  
  
insert into RECENZIE  
values (id_seq.nextval,(select id_client from CLIENT where EmailClient='biancadogareci@gmail.com'),10,'Aranjamente foarte frumoase!');
```

ID_RECENZIE	ID_CLIENT	EVALUARE	COMENTARIU
1	35	31	10 Foarte multumit de servicii, personal amabil.
2	36	31	9 (null)
3	37	30	10 Produse de calitate!
4	38	26	5 Nu am fost impresionata de calitatea produselor.
5	39	29	10 Recomand cu incredere, flori proaspete.
6	40	33	10 Aranjamente foarte frumoase!

- **TABELA REDUCERE:**

```

insert into REDUCERE
values (id_seq.nextval, 'aj4id6iwk', 10.5, to_date('01-01-2023','DD-MM-YYYY'),
to_date('31-12-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'cod235786', 5.0, to_date('01-06-2023','DD-MM-YYYY'),
to_date('31-08-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'codcraciun2', 15.0, to_date('01-12-2023','DD-MM-YYYY'),
to_date('31-12-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'BlackFriday!3', 20.0, to_date('01-11-2023','DD-MM-YYYY'),
to_date('30-11-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'red453466', 5.0, to_date('01-03-2023','DD-MM-YYYY'),
to_date('31-08-2023','DD-MM-YYYY'));

```

```

insert into REDUCERE
values (id_seq.nextval, 'aj4id6iwk', 10.5, to_date('01-01-2023','DD-MM-YYYY'), to_date('31-12-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'cod235786', 5.0, to_date('01-06-2023','DD-MM-YYYY'), to_date('31-08-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'codcraciun2', 15.0, to_date('01-12-2023','DD-MM-YYYY'), to_date('31-12-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'BlackFriday!3', 20.0, to_date('01-11-2023','DD-MM-YYYY'), to_date('30-11-2023','DD-MM-YYYY'));

insert into REDUCERE
values (id_seq.nextval, 'red453466', 5.0, to_date('01-03-2023','DD-MM-YYYY'), to_date('31-08-2023','DD-MM-YYYY'));

```

ID_REDUCERE	CODREDUCERE	PROCENTREDUCERE	DATAINCEPUT	DATASFARSIT
1	41 aj4id6iwk	10.5	01-JAN-23	31-DEC-23
2	42 cod235786	5	01-JUN-23	31-AUG-23
3	43 codcraciun2	15	01-DEC-23	31-DEC-23
4	44 BlackFriday!3	20	01-NOV-23	30-NOV-23
5	45 red453466	5	01-MAR-23	31-AUG-23

- **TABELA COMANDA:**

```
insert into COMANDA
values (id_seq.nextval,11,(select id_client from CLIENT where
EmailClient='mihaifeur@gmail.com'), 41,to_date('15-12-2023','DD-MM-
YYYY'),'card');
```

```
insert into COMANDA
values (id_seq.nextval,12,(select id_client from CLIENT where
EmailClient='ananica@ymail.com'), null,to_date('05-05-2022','DD-MM-
YYYY'),'card');
```

```
insert into COMANDA
values (id_seq.nextval,14,(select id_client from CLIENT where
EmailClient='ananica@ymail.com'), null,to_date('27-11-2022','DD-MM-
YYYY'),'cash');
```

```
insert into COMANDA
values (id_seq.nextval,15,(select id_client from CLIENT where
EmailClient='ioanapopescu@gmail.com'), 44,to_date('18-11-2023','DD-MM-
YYYY'),default);
```

```
insert into COMANDA
values (id_seq.nextval,11,(select id_client from CLIENT where
EmailClient='alextintea@gmail.com'), 45,to_date('04-05-2023','DD-MM-
YYYY'),'cash');
```

```
insert into COMANDA
values (id_seq.nextval,12,(select id_client from CLIENT where
EmailClient='teodoraapostol@yahoo.com'), null,to_date('05-03-2023','DD-MM-
YYYY'),'card');
```

```
insert into COMANDA
values (id_seq.nextval,12,(select id_client from CLIENT where
EmailClient='stefanolteanu@gmail.com'), null,to_date('14-07-2023','DD-MM-
YYYY'),'cash');
```

```
insert into COMANDA
values (id_seq.nextval,14,(select id_client from CLIENT where
EmailClient='andreimarinescu@gmail.com'), 42,to_date('04-07-2023','DD-MM-
YYYY'),'cash');
```

```
insert into COMANDA
values (id_seq.nextval,11,(select id_client from CLIENT where
EmailClient='biancadogareci@gmail.com'), null,to_date('16-02-2023','DD-MM-
YYYY'),'card');
```

```
insert into COMANDA
values (id_seq.nextval,15,(select id_client from CLIENT where
EmailClient='ionescumarian@gmail.com'), null,to_date('08-03-2023','DD-MM-
YYYY'),'cash');
```

```
insert into COMANDA
values (id_seq.nextval,11,(select id_client from CLIENT where EmailClient='mihaifeur@gmail.com'), 41,to_date('15-12-2023','DD-MM-YYYY'),'card');

insert into COMANDA
values (id_seq.nextval,12,(select id_client from CLIENT where EmailClient='ananica@ymail.com'), null,to_date('05-05-2022','DD-MM-YYYY'),'card');

insert into COMANDA
values (id_seq.nextval,14,(select id_client from CLIENT where EmailClient='ananica@ymail.com'), null,to_date('27-11-2022','DD-MM-YYYY'),'cash');

insert into COMANDA
values (id_seq.nextval,15,(select id_client from CLIENT where EmailClient='ioanapopescu@gmail.com'), 44,to_date('18-11-2023','DD-MM-YYYY'),default);

insert into COMANDA
values (id_seq.nextval,11,(select id_client from CLIENT where EmailClient='alextintea@gmail.com'), 45,to_date('04-05-2023','DD-MM-YYYY'),'cash');

insert into COMANDA
values (id_seq.nextval,12,(select id_client from CLIENT where EmailClient='teodoraapostol@yahoo.com'), null,to_date('05-03-2023','DD-MM-YYYY'),'card');

insert into COMANDA
values (id_seq.nextval,12,(select id_client from CLIENT where EmailClient='stefanolteanu@gmail.com'), null,to_date('14-07-2023','DD-MM-YYYY'),'cash');

insert into COMANDA
values (id_seq.nextval,14,(select id_client from CLIENT where EmailClient='andreimarinescu@gmail.com'), 42,to_date('04-07-2023','DD-MM-YYYY'),'cash');

insert into COMANDA
values (id_seq.nextval,11,(select id_client from CLIENT where EmailClient='biancadogareci@gmail.com'), null,to_date('16-02-2023','DD-MM-YYYY'),'card');

insert into COMANDA
values (id_seq.nextval,15,(select id_client from CLIENT where EmailClient='ionescumarian@gmail.com'), null,to_date('08-03-2023','DD-MM-YYYY'),'cash');
```

	ID_COMANDA	ID_ANGAJAT	ID_CLIENT	ID_REDUCERE	DATA COMANDA	TIP PLATA
1	46	11	31	41	15-DEC-23	card
2	47	12	30	(null)	05-MAY-22	card
3	48	14	30	(null)	27-NOV-22	cash
4	49	15	26	44	18-NOV-23	cash
5	50	11	29	45	04-MAY-23	cash
6	51	12	27	(null)	05-MAR-23	card
7	52	12	28	(null)	14-JUL-23	cash
8	53	14	32	42	04-JUL-23	cash
9	54	11	33	(null)	16-FEB-23	card
10	55	15	34	(null)	08-MAR-23	cash

- **TABELA CATEGORIE\_PRODUS:**

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Buchete','Buchete de flori proaspete și aranjamente florale elegante.');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Plante de interior',null);
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Plante de exterior','Plante de exterior perene și anuale');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Accesorii','Ghivece, ambalaje, felicitari, ingrasaminte, decoratiuni etc.');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'La bucata','Vanzare la fir.');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Buchete','Buchete de flori proaspete și aranjamente florale elegante.');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Plante de interior',null);
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Plante de exterior','Plante de exterior perene și anuale');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'Accesorii','Ghivece, ambalaje, felicitari, ingrasaminte, decoratiuni etc.');
```

```
insert into CATEGORIE_PRODUS
```

```
values (id_seq.nextval,'La bucata','Vanzare la fir.');
```

	ID_CATEGPROD	NUMECATEGORIE	DESCRIERE
1	56	Buchete	Buchete de flori proaspete și aranjamente florale elegante.
2	57	Plante de interior (null)	
3	58	Plante de exterior	Plante de exterior perene și anuale
4	59	Accesorii	Ghivece, ambalaje, felicitari, ingrasaminte, decoratiuni etc.
5	60	La bucată	Vanzare la fir.

- **TABELA FURNIZOR:**

```
insert into FURNIZOR
values (id_seq.nextval,'Fresco Verde','Strada Trandafirilor 10, București, 010101,
România','0721123456');
```

```
insert into FURNIZOR
values (id_seq.nextval,'Happy Flower Holland','Aleea Florilor 5, București, 400000,
România','0765987654');
```

```
insert into FURNIZOR
values (id_seq.nextval,'GreenLeaf Plants','Bulevardul Liliacului 25, București,
700000, România','0744555888');
```

```
insert into FURNIZOR
values (id_seq.nextval,'Natures Garden Supplies','Strada Irisului 8, București,
300001, România','0782111222');
```

```
insert into FURNIZOR
values (id_seq.nextval,'Bloom Nursery Co.','Splaiul Crizantemelor 3, București,
500100, România','0733999000');
```

```

insert into FURNIZOR
values (id_seq.nextval,'Fresco Verde','Strada Trandafirilor 10, Bucureşti, 010101, România','0721123456');

insert into FURNIZOR
values (id_seq.nextval,'Happy Flower Holland','Aleea Florilor 5, Bucureşti, 400000, România','0765987654');

insert into FURNIZOR
values (id_seq.nextval,'GreenLeaf Plants','Bulevardul Liliacului 25, Bucureşti, 700000, România','0744555888');

insert into FURNIZOR
values (id_seq.nextval,'Natures Garden Supplies','Strada Irisului 8, Bucureşti, 300001, România','0782111222');

insert into FURNIZOR
values (id_seq.nextval,'Bloom Nursery Co.','Splaiul Crizantemelor 3, Bucureşti, 500100, România','0733999000');

```

	ID_FURNIZOR	NUMEFURNIZOR	ADRESAFURNIZOR	TELEFONFURNIZOR
1	61	Fresco Verde	Strada Trandafirilor 10, Bucureşti, 010101, România	0721123456
2	62	Happy Flower Holland	Aleea Florilor 5, Bucureşti, 400000, România	0765987654
3	63	GreenLeaf Plants	Bulevardul Liliacului 25, Bucureşti, 700000, România	0744555888
4	64	Natures Garden Supplies	Strada Irisului 8, Bucureşti, 300001, România	0782111222
5	65	Bloom Nursery Co.	Splaiul Crizantemelor 3, Bucureşti, 500100, România	0733999000

- **TABELA PRODUS:**

```

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Fresco Verde'), 'Buchet trandafiri', 100);

```

```

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Fresco Verde'), 'Buchet de vara', 80);

```

```

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Fresco Verde'), 'Buchet de primavara', 120);

```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Plante de interior'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Natures Garden Supplies'), 'Dracena', 50);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Plante de interior'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Happy Flower Holland'), 'Ficus lyrata', 30);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Plante de interior'), (select id_furnizor from FURNIZOR where
NumeFurnizor='GreenLeaf Plants'), 'Trandafir japonez', 50);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Plante de exterior'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Bloom Nursery Co.'), 'Trandafir englezesc', 60);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Plante de exterior'), (select id_furnizor from FURNIZOR where
NumeFurnizor='GreenLeaf Plants'), 'Cupressus sempervirens', 80);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Plante de exterior'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Natures Garden Supplies'), 'Bujor', 24);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Accesorii'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Happy Flower Holland'), 'Ghiveci ceramic', 25);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Accesorii'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Natures Garden Supplies'), 'Vaza de sticla', 25);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='Accesorii'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Bloom Nursery Co.'), 'Felicitare', 15);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='La bucata'), (select id_furnizor from FURNIZOR where
NumeFurnizor='GreenLeaf Plants'), 'Trandafir roz', 18);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='La bucata'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Natures Garden Supplies'), 'Crin alb', 14);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where
NumeCategoria='La bucata'), (select id_furnizor from FURNIZOR where
NumeFurnizor='Happy Flower Holland'), 'Frezie galbenă', 10);
```

```
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where NumeFurnizor='Fresco Verde'), 'Buchet trandafiri', 100);
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where NumeFurnizor='Fresco Verde'), 'Buchet de vara', 80);
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where NumeFurnizor='Fresco Verde'), 'Buchet de primavara', 120);
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Plante de interior'), (select id_furnizor from FURNIZOR where NumeFurnizor='Natures Garden Supplies'), 'Dracena', 50);
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Plante de interior'), (select id_furnizor from FURNIZOR where NumeFurnizor='Happy Flower Holland'), 'Ficus lyrata', 30);
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Plante de interior'), (select id_furnizor from FURNIZOR where NumeFurnizor='GreenLeaf Plants'), 'Trandafir japonez', 5);
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Plante de exterior'), (select id_furnizor from FURNIZOR where NumeFurnizor='Bloom Nursery Co.'), 'Trandafir englezesc');
insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Plante de exterior'), (select id_furnizor from FURNIZOR where NumeFurnizor='GreenLeaf Plants'), 'Cupressus sempervirens');
```

```

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Plante de exterior'), (select id_furnizor from FURNIZOR where NumeFurnizor='Natures Garden Supplies'), 'Bujor', 24);

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Accesorii'), (select id_furnizor from FURNIZOR where NumeFurnizor='Happy Flower Holland'), 'Ghiveci ceramic', 25);

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Accesorii'), (select id_furnizor from FURNIZOR where NumeFurnizor='Natures Garden Supplies'), 'Vaza de sticla', 25);

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Accesorii'), (select id_furnizor from FURNIZOR where NumeFurnizor='Bloom Nursery Co.'), 'Felicitare', 15);

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='La bucata'), (select id_furnizor from FURNIZOR where NumeFurnizor='GreenLeaf Plants'), 'Trandafir roz', 18);

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='La bucata'), (select id_furnizor from FURNIZOR where NumeFurnizor='Natures Garden Supplies'), 'Crin alb', 14);

insert into PRODUS
values (id_seq.nextval, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='La bucata'), (select id_furnizor from FURNIZOR where NumeFurnizor='Happy Flower Holland'), 'Frezie galbenă', 10);

```

ID_PRODUS	ID_CATEGPROD	ID_FURNIZOR	NUMEPRODUS	PRET
1	66	56	61 Buchet trandafiri	100
2	67	56	61 Buchet de vara	80
3	68	56	61 Buchet de primavara	120
4	69	57	64 Dracena	50
5	70	57	62 Ficus lyrata	30
6	71	57	63 Trandafir japonez	50
7	72	58	65 Trandafir englezesc	60
8	73	58	63 Cupressus sempervirens	80
9	74	58	64 Bujor	24
10	75	59	62 Ghiveci ceramic	25
11	76	59	64 Vaza de sticla	25
12	77	59	65 Felicitare	15
13	78	60	63 Trandafir roz	18
14	79	60	64 Crin alb	14
15	80	60	62 Frezie galbenă	10

### • TABELA DETALII\_COMANDA:

```

insert into DETALII_COMANDA
values (id_seq.nextval,46,(select id_produs from PRODUS where
NumeProdus='Cupressus sempervirens'),1);

```

```

insert into DETALII_COMANDA
values (id_seq.nextval,46,(select id_produs from PRODUS where
NumeProdus='Ghiveci ceramic'),1);

```

```

insert into DETALII_COMANDA
values (id_seq.nextval,48,(select id_produs from PRODUS where
NumeProdus='Buchet trandafiri'),1);

```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,47,(select id_produs from PRODUS where  
NumeProdus='Frezie galbenă'),15);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,47,(select id_produs from PRODUS where  
NumeProdus='Vaza de sticla'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,47,(select id_produs from PRODUS where  
NumeProdus='Felicitare'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,50,(select id_produs from PRODUS where  
NumeProdus='Ghiveci ceramic'),3);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,50,(select id_produs from PRODUS where  
NumeProdus='Trandafir englezesc'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,51,(select id_produs from PRODUS where  
NumeProdus='Dracena'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,51,(select id_produs from PRODUS where  
NumeProdus='Bujor'),2);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,49,(select id_produs from PRODUS where  
NumeProdus='Trandafir roz'),50);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,49,(select id_produs from PRODUS where  
NumeProdus='Felicitare'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,52,(select id_produs from PRODUS where  
NumeProdus='Ficus lyrata'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,53,(select id_produs from PRODUS where  
NumeProdus='Buchet trandafiri'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,53,(select id_produs from PRODUS where  
NumeProdus='Vaza de sticla'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,53,(select id_produs from PRODUS where  
NumeProdus='Felicitare'),1);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,54,(select id_produs from PRODUS where  
NumeProdus='Bujor'),3);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,55,(select id_produs from PRODUS where  
NumeProdus='Crin alb'),15);
```

```
insert into DETALII_COMANDA  
values (id_seq.nextval,55,(select id_produs from PRODUS where  
NumeProdus='Felicitare'),1);
```

```
insert into DETALII_COMANDA
values (id_seq.nextval,46,(select id_produs from PRODUS where NumeProdus='Cupressus sempervirens'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,46,(select id_produs from PRODUS where NumeProdus='Ghiveci ceramic'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,48,(select id_produs from PRODUS where NumeProdus='Buchet trandafiri'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,47,(select id_produs from PRODUS where NumeProdus='Frezie galbenă'),15);

insert into DETALII_COMANDA
values (id_seq.nextval,47,(select id_produs from PRODUS where NumeProdus='Vaza de sticla'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,47,(select id_produs from PRODUS where NumeProdus='Felicitare'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,50,(select id_produs from PRODUS where NumeProdus='Ghiveci ceramic'),3);

insert into DETALII_COMANDA
values (id_seq.nextval,50,(select id_produs from PRODUS where NumeProdus='Trandafir englezesc'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,51,(select id_produs from PRODUS where NumeProdus='Dracena'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,51,(select id_produs from PRODUS where NumeProdus='Bujor'),2);

insert into DETALII_COMANDA
values (id_seq.nextval,49,(select id_produs from PRODUS where NumeProdus='Trandafir roz'),50);

insert into DETALII_COMANDA
values (id_seq.nextval,49,(select id_produs from PRODUS where NumeProdus='Felicitare'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,52,(select id_produs from PRODUS where NumeProdus='Ficus lyrata'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,53,(select id_produs from PRODUS where NumeProdus='Buchet trandafiri'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,53,(select id_produs from PRODUS where NumeProdus='Vaza de sticla'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,53,(select id_produs from PRODUS where NumeProdus='Felicitare'),1);

insert into DETALII_COMANDA
values (id_seq.nextval,54,(select id_produs from PRODUS where NumeProdus='Bujor'),3);

insert into DETALII_COMANDA
values (id_seq.nextval,55,(select id_produs from PRODUS where NumeProdus='Crin alb'),15);

insert into DETALII_COMANDA
values (id_seq.nextval,55,(select id_produs from PRODUS where NumeProdus='Felicitare'),1);
```

	ID_DETALIICOMANDA	ID_COMANDA	ID_PRODUS	NUMARPRODUSE
1	81	46	73	1
2	82	46	75	1
3	83	48	66	1
4	84	47	80	15
5	85	47	76	1
6	86	47	77	1
7	87	50	75	3
8	88	50	72	1
9	89	51	69	1
10	90	51	74	2
11	91	49	78	50
12	92	49	77	1
13	93	52	70	1
14	94	53	66	1
15	95	53	76	1
16	96	53	77	1
17	97	54	74	3
18	98	55	79	15
19	99	55	77	1

### **Cerinta 6:**

Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze toate cele 3 tipuri de colectii studiate. Apelati subprogramul.

### **Enuntul problemei:**

Implementati functionalitatea de adaugare a unui nou produs in baza de date prin definirea unui subprogram stocat independent. Scopul problemei este de a facilita introducerea eficienta si corecta a unui nou produs in sistem. Stim ca tabela ‘PRODUS’ are urmatoarele atribute:

- ‘id\_produs’: cheie primara deci unica;
- ‘id\_categprod’: cheie externa deci trebuie sa existe in tabela ‘CATEGORIE\_PRODUS’ o categorie cu id-ul respectiv;
- ‘id\_furnizor’: cheie externa deci trebuie sa existe in tabela ‘FURNIZOR’ un furnizor cu id-ul respectiv;
- ‘numeprodus’: este unic;
- ‘pret’.

In rezolvarea problemei tineti cont de restrictiile mentionate anterior.

**Rezolvare:**

```
create or replace procedure AdaugaProdus
(v_id_produs produs.id_produs%type,
 v_numecategorie categorie_produs.numecategorie%type,
 v_numefurnizor furnizor.numefurnizor%type,
 v_numeprodus produs.numeprodus%type,
 v_pret produs.pret%type)
is
type tablou_indexat is table of varchar2(255) index by pls_integer;
type tablou_imbricat is table of varchar2(255);
type vector is varray(50) of varchar2(255);

tablou_categprod tablou_indexat;
tablou_furnizori tablou_imbricat := tablou_imbricat();
vector_produse vector := vector();

v_categprod_exista boolean := false;
v_furnizor_exista boolean := false;
v_produs_exista boolean := false;
v_id_produs_exista number;

begin
select NumeCategoria
bulk collect into tablou_categprod
from categorie_produs;

for i in tablou_categprod.first..tablou_categprod.last loop
if tablou_categprod(i) = v_numecategorie then
    v_categprod_exista := true;
    exit;
end if;
end loop;

select NumeFurnizor
bulk collect into tablou_furnizori
```

```
from furnizor;

for i in tablou_furnizori.first..tablou_furnizori.last loop
    if tablou_furnizori(i) = v_numefurnizor then
        v_furnizor_exista := true;
        exit;
    end if;
end loop;
```

```
select NumeProdus
bulk collect into vector_produse
from produs;
```

```
for i in vector_produse.first..vector_produse.last loop
    if vector_produse(i) = v_numeprodus then
        v_produs_exista := true;
        exit;
    end if;
end loop;
```

```
select count(*)
into v_id_produs_exista
from produs
where id_produs = v_id_produs;
```

```
if v_categprod_exista = false then
    dbms_output.put_line('Categoria de produs nu exista.');
end if;
```

```
if v_furnizor_exista = false then
    dbms_output.put_line('Furnizorul nu exista.');
end if;
```

```
if v_produs_exista = true then
```

```
    dbms_output.put_line('Produsul deja exista.');
end if;

if v_id_produs_exista > 0 then
    dbms_output.put_line('ID-ul produsului nu este unic.');
end if;

if v_categprod_exista = false or v_furnizor_exista = false or v_produs_exista =
true or v_id_produs_exista > 0 then
    return;
end if;

insert into produs
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie =
v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
    raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/
```

```
create or replace procedure AdaugaProdus
    (v_id_produs produs.id_produs%type,
     v_numecategorie categorie_produs.numecategorie%type,
     v_numefurnizor furnizor.numefurnizor%type,
     v_numeprodus produs.numeprodus%type,
     v_pret produs.pret%type)
is
begin
    type tablou_indexat is table of varchar2(255) index by pls_integer;
    type tablou_imbricat is table of varchar2(255);
    type vector is varray(50) of varchar2(255);

    tablou_categprod tablou_indexat;
    tablou_furnizori tablou_imbricat := tablou_imbricat();
    vector_produse vector := vector();

    v_categprod_exista boolean := false;
    v_furnizor_exista boolean := false;
    v_produs_exista boolean := false;
    v_id_produs_exista number;

    select NumeCategoria
    bulk collect into tablou_categprod
    from categorie_produs;

    for i in tablou_categprod.first..tablou_categprod.last loop
        if tablou_categprod(i) = v_numecategorie then
            v_categprod_exista := true;
            exit;
        end if;
    end loop;

    select NumeFurnizor
    bulk collect into tablou_furnizori
    from furnizor;

    for i in tablou_furnizori.first..tablou_furnizori.last loop
        if tablou_furnizori(i) = v_numefurnizor then
            v_furnizor_exista := true;
            exit;
        end if;
    end loop;

    select NumeProdus
    bulk collect into vector_produse
    from produs;

    for i in vector_produse.first..vector_produse.last loop
        if vector_produse(i) = v_numeprodus then
            v_produs_exista := true;
            exit;
        end if;
    end loop;

    select count(*)
    into v_id_produs_exista
    from produs
    where id_produs = v_id_produs;
```

```

if v_categprod_exista = false then
    dbms_output.put_line('Categoria de produs nu exista.');
end if;

if v_furnizor_exista = false then
    dbms_output.put_line('Furnizorul nu exista.');
end if;

if v_produs_exista = true then
    dbms_output.put_line('Produsul deja exista.');
end if;

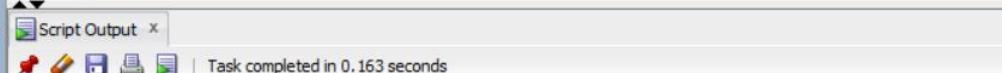
if v_id_produs_exista > 0 then
    dbms_output.put_line('ID-ul produsului nu este unic.');
end if;

if v_categprod_exista = false or v_furnizor_exista = false or v_produs_exista = true or v_id_produs_exista > 0 then
    return;
end if;

insert into produs
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie = v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
    raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/

```

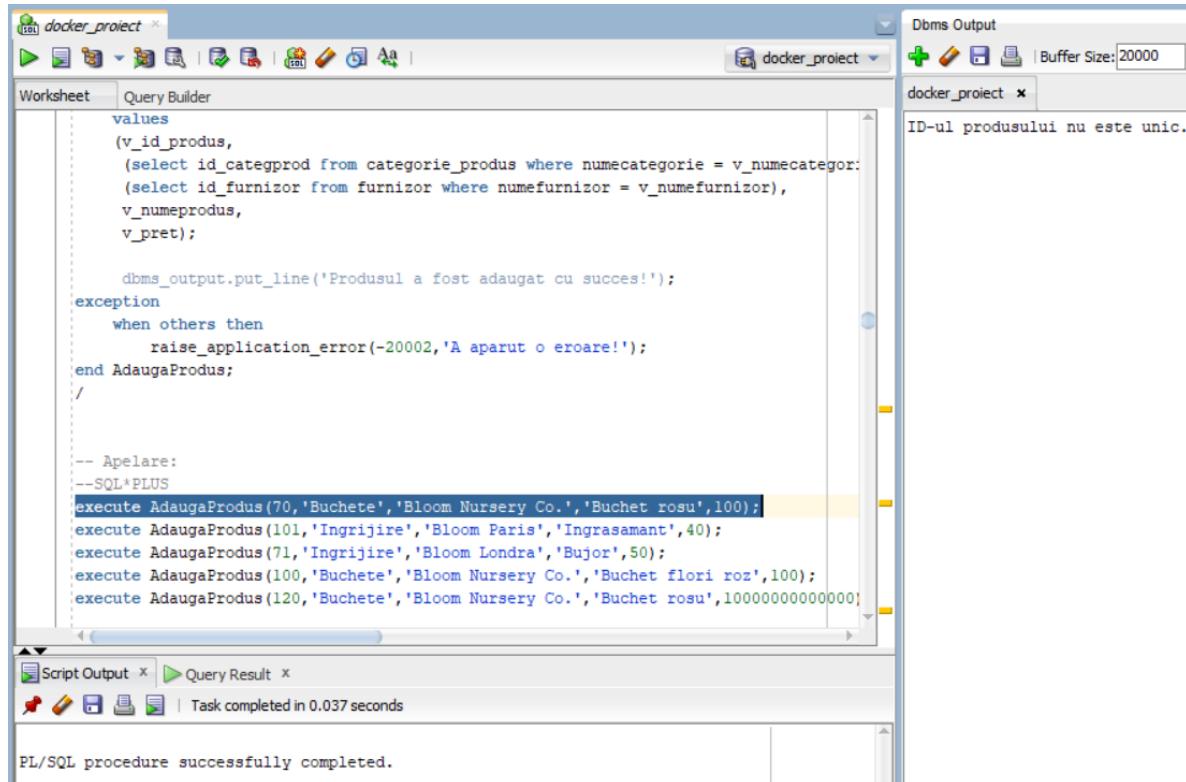


Procedure ADAUGAPRODUS compiled

## Apelarea subprogramului (SQL\*PLUS), exemple:

- cazuri in care produsul nu se introduce in baza de date deoarece avem unul sau mai multe campuri cu informatii gresite;

execute AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);

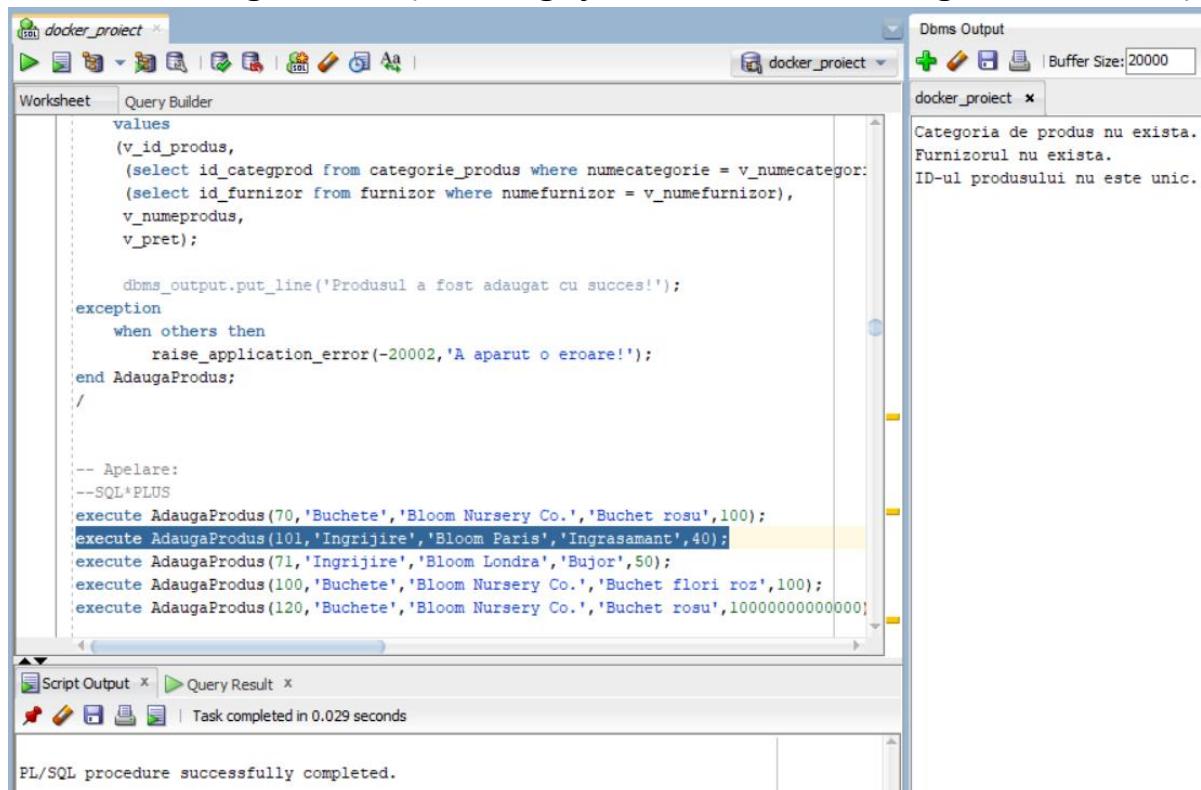


```
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie = v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/


-- Apelare:
--SQL^PLUS
execute AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);
execute AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);
execute AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
execute AdaugaProdus(100,'Buchete','Bloom Nursery Co.','Buchet flori roz',100);
execute AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',10000000000000)
```

execute AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);



```
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie = v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/


-- Apelare:
--SQL^PLUS
execute AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);
execute AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);
execute AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
execute AdaugaProdus(100,'Buchete','Bloom Nursery Co.','Buchet flori roz',100);
execute AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',10000000000000)
```

execute AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);

```
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie = v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/


-- Apelare:
--SQL*PLUS
execute AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);
execute AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);
execute AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
execute AdaugaProdus(100,'Buchete','Bloom Nursery Co.','Buchet flori roz',100);
execute AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',10000000000000)
```

Script Output x Query Result x  
Task completed in 0.03 seconds

PL/SQL procedure successfully completed.

Dbsm Output  
Categoría de produs nu există.  
Furnizorul nu există.  
Produsul deja există.  
ID-ul produsului nu este unic.

- cazul in care produsul se introduce in baza de date deoarece toate campurile contin informatii corecte;

execute AdaugaProdus(100,'Buchete','Bloom Nursery Co.','Buchet flori roz',100);

```
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie = v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/


-- Apelare:
--SQL*PLUS
execute AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);
execute AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);
execute AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
execute AdaugaProdus(100,'Buchete','Bloom Nursery Co.','Buchet flori roz',100);
execute AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',10000000000000)
```

Script Output x Query Result x  
Task completed in 0.037 seconds

PL/SQL procedure successfully completed.

Dbsm Output  
Produsul a fost adaugat cu succes!

Observam ca produsul a fost adaugat in baza de date cand datele introduse sunt corecte, lui ii corespunde ultima linie (linia 16):

ID_PRODUS	ID_CATEGPROD	ID_FURNIZOR	NUMEPRODUS	PRET
1	66	56	61 Buchet trandafiri	100
2	67	56	61 Buchet de vara	80
3	68	56	61 Buchet de primavara	120
4	69	57	64 Dracena	50
5	70	57	62 Ficus lyrata	30
6	71	57	63 Trandafir japonez	50
7	72	58	65 Trandafir englezesc	60
8	73	58	63 Cupressus sempervirens	80
9	74	58	64 Bujor	24
10	75	59	62 Ghiveci ceramic	25
11	76	59	64 Vaza de sticla	25
12	77	59	65 Felicitare	15
13	78	60	63 Trandafir roz	18
14	79	60	64 Crin alb	14
15	80	60	62 Frezie galbenă	10
16	100	56	65 Buchet flori roz	100

- cazul in care produsul nu se introduce in baza de date, deoarece blocul 'when others' prinde o exceptie care nu este gestionata explicit in blocul 'begin...end'.

execute AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',1000000000000000);

```

values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie = v_numecategorie,
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/


-- Apelare:
--SQL*PLUS
execute AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);
execute AdaugaProdus(101,'Ingrjire','Bloom Paris','Ingrasamant',40);
execute AdaugaProdus(71,'Ingrjire','Bloom Londra','Bujor',50);
execute AdaugaProdus(100,'Buchete','Bloom Nursery Co.','Buchet flori roz',100);
execute AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',1000000000000000)

```

Script Output | Query Result | Task completed in 0.032 seconds

```

Error starting at line : 511 in command -
BEGIN AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',1000000000000000); END;
Error report -
ORA-20002: A aparut o eroare!
ORA-06512: at "PROIECT.ADAUGAPRODUS", line 90
ORA-06512: at line 1

```

## Apelarea subprogramului (PLSQL), exemple:

- cazuri in care produsul nu se introduce in baza de date deoarece avem unul sau mai multe campuri cu informatii gresite;

begin

    AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);

end;

/

The screenshot shows the Oracle SQL Developer interface with a PL/SQL script being run. The 'Worksheet' tab contains the following code:

```
begin
    AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);
end;
/
begin
```

The line 'AdaugaProdus(70,'Buchete','Bloom Nursery Co.','Buchet rosu',100);' is highlighted in blue, indicating it is currently being executed. The 'Dbms Output' tab shows the error message: 'ID-ul produsului nu este unic.' (The product ID is not unique). The 'Script Output' tab shows the successful completion message: 'PL/SQL procedure successfully completed.'

```
begin
```

```
    AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);  
end;
```

```
/
```

The screenshot shows the Oracle SQL Developer interface during the execution of a PL/SQL script. The main window displays the script code:

```
begin  
    AdaugaProdus(101,'Ingrijire','Bloom Paris','Ingrasamant',40);  
end;  
/

```
Script Output x | Query Result x
```



The 'Script Output' tab shows the message: "PL/SQL procedure successfully completed." The 'Query Result' tab shows the message: "Task completed in 0.037 seconds".



In the top right corner, the 'Dbms Output' window displays error messages:  
Categoría de produs nu există.  
Furnizorul nu există.  
ID-ul produsului nu este unic.


```

```
begin
```

```
    AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
```

```
end;
```

```
/
```

```
begin
    AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
end;
/
begin
```

Dbms Output

Categoria de produs nu exista.  
Furnizorul nu exista.  
Produsul deja exista.  
ID-ul produsului nu este unic.

Script Output | Task completed in 0.038 seconds

PL/SQL procedure successfully completed.

- cazul in care produsul se introduce in baza de date deoarece toate campurile contin informatii corecte;

```
begin
```

```
    AdaugaProdus(101,'Buchete','Bloom Nursery Co.','Buchet flori galbene',100);
```

```
end;
```

```
/
```

Screenshot of Oracle SQL Developer showing a PL/SQL script being run.

```

    end if;

    insert into produs
    values
    (v_id_produs,
     (select id_categprod from categorie_produs where numecategorie = v_numecategorie),
     (select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
     v_numeprodus,
     v_pret);

    dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
    raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;
/
begin
    AdaugaProdus(101,'Buchete','Bloom Nursery Co.','Buchet flori galbene',100);
end;
/
begin

```

Script Output:

```

PL/SQL procedure successfully completed.

```

Dbms Output:

```

Produsul a fost adaugat cu succes!

```

Observam ca produsul a fost adaugat in baza de date cand datele introduse sunt corecte, lui ii corespunde ultima linie (linia 17):

Screenshot of Oracle Database Navigator showing the PRODUS table.

Table Structure:

ID_PRODUS	ID_CATEGPROD	ID_FURNIZOR	NUMEPRODUS	PRET
1	66	56	61 Buchet trandafiri	100
2	67	56	61 Buchet de vara	80
3	68	56	61 Buchet de primavara	120
4	69	57	64 Dracena	50
5	70	57	62 Ficus lyrata	30
6	71	57	63 Trandafir japonez	50
7	72	58	65 Trandafir englezesc	60
8	73	58	63 Cupressus sempervirens	80
9	74	58	64 Bujor	24
10	75	59	62 Ghiveci ceramic	25
11	76	59	64 Vaza de sticla	25
12	77	59	65 Felicitare	15
13	78	60	63 Trandafir roz	18
14	79	60	64 Crin alb	14
15	80	60	62 Frezie galbenă	10
16	100	56	65 Buchet flori roz	100
17	101	56	65 Buchet flori galbene	100

- cazul in care produsul nu se introduce in baza de date, deoarece blocul 'when others' prinde o exceptie care nu este gestionata explicit in blocul 'begin...end'.

begin

```
AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet
rosu',100000000000000);
end;
/
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Worksheet Tab:** Contains the PL/SQL code. The code includes a block of code starting with 'begin', followed by a call to 'AdaugaProdus', and ends with 'end;'. A slash (/) is present at the end of the block. The code is highlighted in blue, indicating it is a PL/SQL block.
- Script Output Tab:** Shows the execution results. It displays the error message: "Error starting at line : 504 in command - begin AdaugaProdus(120,'Buchete','Bloom Nursery Co.','Buchet rosu',100000000000000); end; Error report - ORA-20002: A aparut o eroare! ORA-06512: at "PROIECT.ADAUGAPRODUS", line 90 ORA-06512: at line 2".
- Dbms Output Tab:** Shows the output of the DBMS\_OUTPUT.PUT\_LINE statement from the code, which reads "Produsul a fost adaugat cu succes!".

## **Cerinta 7:**

Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze doua tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celalalt cursor. Apelati subprogramul.

### **Enuntul problemei:**

Sa se afiseze pentru fiecare categorie de produse o lista cu toate produsele ce fac parte din aceasta si id-ul fiecarui produs in paranteza dupa numele produsului.

### **Rezolvare:**

```
create or replace procedure AfiseazaProduseDupaCategorie is
```

```
cursor CursorCategorii is
```

```
    select id_categprod id_categorie, numecategorie nume_categorie  
    from CATEGORIE_PRODUS;
```

```
cursor CursorProduse (v_id_categprod categorie_produs.id_categprod%type) is  
    select numeprodus nume_produs, id_produs idprodus  
    from PRODUS  
    where id_categprod = v_id_categprod;
```

```
var_nume_produs produs.numeprodus%type;
```

```
var_id_produs produs.id_produs%type;
```

```
begin
```

```
for i in CursorCategorii loop
```

```
    dbms_output.put_line('Categorie: ' || i.nume_categorie);
```

```
    open CursorProduse(i.id_categorie);
```

```
    loop
```

```
        fetch CursorProduse into var_nume_produs, var_id_produs;
```

```
        exit when CursorProduse%notfound;
```

```
        dbms_output.put_line(' Produs: ' || var_nume_produs || '(id = ' ||  
        var_id_produs || ')');
```

```
    end loop;
```

```

dbms_output.put_line(' ');
close CursorProduse;
end loop;
end AfiseazaProduseDupaCategorie;
/

```

```

create or replace procedure AfiseazaProduseDupaCategorie is
  cursor CursorCategorii is
    select id_categprod id_categorie, numecategorie nume_categorie
    from CATEGORIE_PRODUS;

  cursor CursorProduse (v_id_categprod categorie_produs.id_categprod%type) is
    select numeprodus nume_produs, id_produs idprodus
    from PRODUS
    where id_categprod = v_id_categprod;

  var_nume_produs produs.numeprodus%type;
  var_id_produs produs.id_produs%type;
begin
  for i in CursorCategorii loop
    dbms_output.put_line('Categorie: ' || i.nume_categorie);
    open CursorProduse(i.id_categorie);
    loop
      fetch CursorProduse into var_nume_produs, var_id_produs;
      exit when CursorProduse%notfound;
      dbms_output.put_line('  Produs: ' || var_nume_produs || ' (id = ' || var_id_produs || ')');
    end loop;
    dbms_output.put_line('');
    close CursorProduse;
  end loop;
end AfiseazaProduseDupaCategorie;
/

```

ts   |> Script Output x  
 ts   |> Task completed in 0.035 seconds

ts   Procedure AFISEAZAPRODUSEDUPACATEGORIE compiled

Am creat o procedura ('AfiseazaProduseDupaCategorie') care utilizeaza doua cursoare de tipuri diferite. Un ciclu cursor ('CursorCategorii') si un cursor clasic explicit parametrizat ('CursorProduse') care depinde de rezultatele primului cursor.

## Apelarea subprogramului (SQL\*PLUS): execute AfiseazaProduseDupaCategorie;

The screenshot shows the Oracle SQL\*Plus interface. In the Worksheet window, a PL/SQL procedure named 'AfiseazaProduseDupaCategorie' is being executed. The code uses a cursor to fetch products from a category and prints them to the screen. The output window shows the results categorized by type: Buchete, Plante de interior, Plante de exterior, Accesorii, and La bucata, each listing various products with their IDs.

```
open CursorProduse(i.id_categoria);
loop
    fetch CursorProduse into var_nume_produs, var_id_produs;
    exit when CursorProduse%notfound;
    dbms_output.put_line(' Produs: ' || var_nume_produs || ' (id = ' || var_id_produs || ')');
end loop;
close CursorProduse;
end AfiseazaProduseDupaCategorie;
/
-- Apelarea subprogramului
begin
    AfiseazaProduseDupaCategorie;
end;
/
execute AfiseazaProduseDupaCategorie;
```

Script Output | Task completed in 0.032 seconds

PL/SQL procedure successfully completed.

Dbsm Output

Categorie: Buchete  
Produs: Buchet trandafiri (id = 66)  
Produs: Buchet de vara (id = 67)  
Produs: Buchet de primavara (id = 68)  
Produs: Buchet flori roz (id = 100)  
Produs: Buchet flori galbene (id = 101)

Categorie: Plante de interior  
Produs: Dracena (id = 69)  
Produs: Ficus lyrate (id = 70)  
Produs: Trandafir japonez (id = 71)

Categorie: Plante de exterior  
Produs: Trandafir englezesc (id = 72)  
Produs: Cupressus sempervirens (id = 73)  
Produs: Bujor (id = 74)

Categorie: Accesorii  
Produs: Ghiveci ceramic (id = 75)  
Produs: Vaza de sticla (id = 76)  
Produs: Felicitare (id = 77)

Categorie: La bucata  
Produs: Trandafir roz (id = 78)  
Produs: Crin alb (id = 79)  
Produs: Frezie galbenă (id = 80)

## Apelarea subprogramului (PLSQL): begin

```
AfiseazaProduseDupaCategorie;
```

```
end;
```

```
/
```

The screenshot shows the Oracle SQL\*Plus interface. A PL/SQL block is being executed. It calls the procedure 'AfiseazaProduseDupaCategorie' which prints product information. The output window shows the same categorized list of products as the previous screenshot.

```
open CursorProduse(i.id_categoria);
loop
    fetch CursorProduse into var_nume_produs, var_id_produs;
    exit when CursorProduse%notfound;
    dbms_output.put_line(' Produs: ' || var_nume_produs || ' (id = ' || var_id_produs || ')');
end loop;
close CursorProduse;
end AfiseazaProduseDupaCategorie;
/
-- Apelarea subprogramului
begin
    AfiseazaProduseDupaCategorie;
end;
/
execute AfiseazaProduseDupaCategorie;
```

Script Output | Task completed in 0.219 seconds

PL/SQL procedure successfully completed.

Dbsm Output

Categorie: Buchete  
Produs: Buchet trandafiri (id = 66)  
Produs: Buchet de vara (id = 67)  
Produs: Buchet de primavara (id = 68)  
Produs: Buchet flori roz (id = 100)  
Produs: Buchet flori galbene (id = 101)

Categorie: Plante de interior  
Produs: Dracena (id = 69)  
Produs: Ficus lyrate (id = 70)  
Produs: Trandafir japonez (id = 71)

Categorie: Plante de exterior  
Produs: Trandafir englezesc (id = 72)  
Produs: Cupressus sempervirens (id = 73)  
Produs: Bujor (id = 74)

Categorie: Accesorii  
Produs: Ghiveci ceramic (id = 75)  
Produs: Vaza de sticla (id = 76)  
Produs: Felicitare (id = 77)

Categorie: La bucata  
Produs: Trandafir roz (id = 78)  
Produs: Crin alb (id = 79)  
Produs: Frezie galbenă (id = 80)

### **Cerinta 8:**

Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip functie care sa utilizeze intr-o singura comanda SQL 3 dintre tabelele definite. Definiti minim 2 exceptii proprii. Apelati subprogramul astfel incat sa evidentiati toate cazurile definite si tratate.

### **Enuntul problemei:**

Afisati numarul de comenzi procesate de un anumit angajat (numar returnat de functie), numele complet al clientilor care au plasat respectivele comenzi si numarul comenzilor care au aplicat un cod de reducere asupra lor, stiind ca ni se dau numele si prenumele angajatului. Sa se defineasca exceptii pentru atunci cand angajatul nu exista, exista dar nu se ocupa de vreo comanda sau exista dar are un rol care nu ii permite sa se ocupe de comenzi.

### **Rezolvare:**

```
create or replace function ComenziAngajat(
    p_nume_angajat angajat.numeangajat%type,
    p_prenume_angajat angajat.prenumeangajat%type
) return number is
    cursor CursorClienti (id_ang comanda.id_angajat%type) is
        select id_client id_client_
        from comanda
        where id_angajat = id_ang;

    var_id_client client.id_client%type;
    v_id_angajat angajat.id_angajat%type;
    nr_comenzi number;
    nume client.numeclient%type;
    prenume client.prenumeclient%type;
    angajat_unic number;
    id_rol_angajat number;
    comenzi_reduse number;

    --exceptii
```

```
angajatul_nu_exista exception;
angajatul_nu_are_comenzi exception;
angajat_care_nu_e_responsabil_de_comenzi exception;
begin
    select count(*) into angajat_unic
    from angajat
    where NumeAngajat = p_num_angajat and PrenumeAngajat =
p_prenume_angajat;

    if angajat_unic < 1 then
        raise angajatul_nu_exista;
    end if;

    select id_angajat into v_id_angajat
    from angajat
    where NumeAngajat = p_num_angajat and PrenumeAngajat =
p_prenume_angajat;

    select id_rol into id_rol_angajat
    from angajat
    where id_angajat = v_id_angajat;

    if id_rol_angajat not in (1,3) then
        raise angajat_care_nu_e_responsabil_de_comenzi;
    end if;

    select count(*) into nr_comenzi
    from comanda
    where id_angajat = v_id_angajat;

    if nr_comenzi = 0 then
        raise angajatul_nu_are_comenzi;
    end if;
```

```

dbms_output.put_line('Numele clientilor care au plasat comenzi:');
open CursorClienti(v_id_angajat);
loop
  fetch CursorClienti into var_id_client;
  exit when CursorClienti%notfound;
  select numeclient, prenumeclient into nume, prenume
  from client
  where id_client = var_id_client;
  dbms_output.put_line(' ' || nume || '' || prenume);
end loop;
close CursorClienti;

select count (*) into comenzi_reduse
from comanda c
left join reducere r on c.id_reducere = r.id_reducere
left join angajat a on a.id_angajat = c.id_angajat
where a.numeangajat = p_nume_angajat and a.prenumeangajat =
p_prenume_angajat and r.codreducere is not null;

dbms_output.put_line('Numar comenzi reduse:');
dbms_output.put_line(comenzi_reduse);

dbms_output.put_line('Numar comenzi:');
return nr_comenzi;
exception
when angajatul_nu_exista then
  dbms_output.put_line('Acest angajat nu exista');
  return -1;
when angajatul_nu_are_comenzi then
  dbms_output.put_line('Angajatului nu ii corespunde nicio comanda');
  return -1;
when angajat_care_nu_e_responsabil_de_comenzi then
  dbms_output.put_line('Angajatul nu poate sa se ocupe de comenzi deoarece
are un rol care nu ii permite sa faca acest lucru');

```

```

    return -1;
when others then
    dbms_output.put_line('Alta eroare: ' || SQLERRM);
    return -1;
end;
/

```

```

create or replace function ComenziAngajat(
    p_nume_angajat angajat.numeangajat%type,
    p_prenume_angajat angajat.prenumeangajat%type
) return number is
    cursor CursorClienti (id_ang comanda.id_angajat%type) is
        select id_client id_client_
        from comanda
        where id_angajat = id_ang;

    var_id_client client.id_client%type;
    v_id_angajat angajat.id_angajat%type;
    nr_comenzi number;
    nume client.numeclient%type;
    prenume client.prenumeclient%type;
    angajat_unic number;
    id_rol_angajat number;
    comenzi_reduse number;

    --exceptii
    angajatul_nu_exista exception;
    angajatul_nu_are_comenzi exception;
    angajat_care_nu_e_responsabil_de_comenzi exception;
begin
    select count(*) into angajat_unic
    from angajat
    where NumeAngajat = p_nume_angajat and PrenumeAngajat = p_prenume_angajat;

    if angajat_unic < 1 then
        raise angajatul_nu_exista;
    end if;

    select id_angajat into v_id_angajat
    from angajat
    where NumeAngajat = p_nume_angajat and PrenumeAngajat = p_prenume_angajat;

```

```

select id_rol into id_rol_angajat
from angajat
where id_angajat = v_id_angajat;

if id_rol_angajat not in (1,3) then
    raise angajat_care_nu_e_responsabil_de_comenzi;
end if;

select count(*) into nr_comenzi
from comanda
where id_angajat = v_id_angajat;

if nr_comenzi = 0 then
    raise angajatul_nu_are_comenzi;
end if;

dbms_output.put_line('Numele clientilor care au plasat comenzile:');
open CursorClienti(v_id_angajat);
loop
    fetch CursorClienti into var_id_client;
    exit when CursorClienti%notfound;
    select numeclient, prenumeclient into nume, prenume
    from client
    where id_client = var_id_client;
    dbms_output.put_line(' ' || nume || ' ' || prenume);
end loop;
close CursorClienti;

select count (*) into comenzi_reduse
from comanda c
left join reducere r on c.id_reducere = r.id_reducere
left join angajat a on a.id_angajat = c.id_angajat
where a.numeangajat = p_nume_angajat and a.prenumeangajat = p_prenume_angajat and r.codreducere is not null;

dbms_output.put_line('Numar comenzi reduse:');
dbms_output.put_line(comenzi_reduse);

dbms_output.put_line('Numar comenzi:');
return nr_comenzi;
exception
    when angajatul_nu_exista then
        dbms_output.put_line('Acest angajat nu există');
        return -1;
    when angajatul_nu_are_comenzi then
        dbms_output.put_line('Angajatului nu ii corespunde nicio comanda');
        return -1;
    when angajat_care_nu_e_responsabil_de_comenzi then
        dbms_output.put_line('Angajatul nu poate sa se ocupe de comenzi deoarece are un rol care nu ii permite sa faca acest lucru');
        return -1;
    when others then
        dbms_output.put_line('Alta eroare: ' || SQLERRM);
        return -1;
end;
/

```

Script Output x  
 Task completed in 0.04 seconds

Function COMENZIANGAJAT compiled

## Apelarea subprogramului (PLSQL), exemple:

begin

```
    dbms_output.put_line(ComenziAngajat('Ionescu','Anastasia'));
```

end;

/

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a 'Worksheet' tab with the following PL/SQL code:

```
when angajatul_nu_exista then
    dbms_output.put_line('Acest angajat nu există');
    return -1;
when angajatul_nu_are_comenzi then
    dbms_output.put_line('Angajatului nu îi corespunde nicio comandă');
    return -1;
when angajat_care_nu_e_responsabil_de_comenzi then
    dbms_output.put_line('Angajatul nu poate să se ocupe de comenzi deoarece are un rol care nu îi permite să facă acest lucru');
    return -1;
when others then
    dbms_output.put_line('Alta eroare: ' || SQLERRM);
    return -1;
end;
/
--Apelare:
--PLSQL
begin
    dbms_output.put_line(ComenziAngajat('Ionescu','Anastasia'));
end;
/
```

In the bottom-left pane, the status bar says "PL/SQL procedure successfully completed." In the top-right pane, titled "Dbms Output", the output is:

```
Numele clientilor care au plasat comenzi:
Feurdean Mihai
Tintea Alexandru
Dogareci Bianca
Numar comenzi reduse:
2
Numar comenzi:
3
```

begin

```
    dbms_output.put_line(ComenziAngajat('Popescu','Anastasia'));
```

end;

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a 'Worksheet' tab with the following PL/SQL code:

```
when angajatul_nu_are_comenzi then
    dbms_output.put_line('Angajatului nu îi corespunde nicio comandă');
    return -1;
when angajat_care_nu_e_responsabil_de_comenzi then
    dbms_output.put_line('Angajatul nu poate să se ocupe de comenzi deoarece are un rol care nu îi permite să facă acest lucru');
    return -1;
when others then
    dbms_output.put_line('Alta eroare: ' || SQLERRM);
    return -1;
end;
/
--Apelare:
--PLSQL
begin
    dbms_output.put_line(ComenziAngajat('Ionescu','Anastasia'));
end;
/
begin
    dbms_output.put_line(ComenziAngajat('Popescu','Anastasia'));
end;
```

In the bottom-left pane, the status bar says "PL/SQL procedure successfully completed." In the top-right pane, titled "Dbms Output", the output is:

```
Acest angajat nu există
-1
```

```

begin
    dbms_output.put_line(ComenziAngajat('Popescu','Ana'));
end;
/

```

```

dbms_output.put_line('Numar comenzi:');
return nr_comenzi;
exception
when angajatul_nu_exista then
    dbms_output.put_line('Acest angajat nu exista');
    return -1;
when angajatul_nu_are_comenzi then
    dbms_output.put_line('Angajatului nu ii corespunde nicio comanda');
    return -1;
when angajat_care_nu_e_responsabil_de_comenzi then
    dbms_output.put_line('Angajatul nu poate sa se ocupe de comenzi deoarece are un rol care nu ii permite sa faca acest lucru -1');
when others then
    dbms_output.put_line('Alta eroare: ' || SQLERRM);
    return -1;
end;
/

begin
    dbms_output.put_line(ComenziAngajat('Popescu','Ana'));
end;

```

PL/SQL procedure successfully completed.

Pentru a putea vedea si cum functioneaza exceptia 'angajatul\_nu\_are\_comenzi' inseram un nou angajat care se poate ocupa de comenzi, dar care nu are deocamdata nicio comanda atribuita.

```

insert into ANGAJAT
values(id_seq.nextval,(select id_rol from ROL_ANGAJAT where
NumeRol='Florar'),'Munteanu','Iulia',to_date('16-08-2023','DD-MM-YYYY'),5000);

```

```

begin
    dbms_output.put_line(ComenziAngajat('Munteanu','Iulia'));
end;
/

```

```

    dbms_output.put_line('Numar comenzi:');
    return nr_comenzi;
exception
  when angajatul_nu_exista then
    dbms_output.put_line('Acest angajat nu există');
    return -1;
  when angajatul_nu_are_comenzi then
    dbms_output.put_line('Angajatului nu ii corespunde nicio comandă');
    return -1;
  when angajat_care_nu_e_responsabil_de_comenzi then
    dbms_output.put_line('Angajatul nu poate să se ocupe de comenzi deoarece are');
    return -1;
  when others then
    dbms_output.put_line('Alta eroare: ' || SQLERRM);
    return -1;
end;
/

begin
  dbms_output.put_line(ComenziAngajat('Munteanu', 'Iulia'));
end;

```

Script Output | Query Result | Task completed in 0.03 seconds

PL/SQL procedure successfully completed.

## Apelarea subprogramului (SQL), exemple:

```
select ComenziAngajat('Ionescu','Anastasia')
from dual;
```

```
--SQL
select ComenziAngajat('Ionescu','Anastasia')
from dual;
```

Script Output | Query Result | All Rows Fetched: 1 in 0.004 seconds

	COMENZIANGAJAT('IONESCU','ANASTASIA')
1	3

```
select ComenziAngajat('Popescu','Anastasia')
from dual;
```

```
select ComenziAngajat('Popescu','Anastasia')  
from dual;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left query editor, the following PL/SQL code is written:

```
select ComenziAngajat('Popescu','Anastasia')  
from dual;
```

The code is highlighted with syntax coloring. Below the editor, the tool bar has icons for Script, Output, Query, and Result, with the SQL tab selected. A status message indicates "All Rows Fetched: 1 in 0.005 seconds". The bottom part of the window displays the results in a table:

COMENZIANGAJAT('POPESCU','ANASTASIA')	
1	-1

```
select ComenziAngajat('Popescu','Ana')  
from dual;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left query editor, the following PL/SQL code is written:

```
select ComenziAngajat('Popescu','Ana')  
from dual;
```

The code is highlighted with syntax coloring. Below the editor, the tool bar has icons for Script, Output, Query, and Result, with the SQL tab selected. A status message indicates "All Rows Fetched: 1 in 0.006 seconds". The bottom part of the window displays the results in a table:

COMENZIANGAJAT('POPESCU','ANA')	
1	-1

## **Cerinta 9:**

Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip procedura care sa utilizeze intr-o singura comanda SQL 5 dintre tabelele definite. Tratati toate exceptiile care pot aparea, incluzand exceptiile NO\_DATA\_FOUND si TOO\_MANY\_ROWS. Apelati subprogramul astfel incat sa evidentiati toate cazurile tratate.

### **Enuntul problemei:**

Se da numele unui client. Sa se afiseze pentru el numele tuturor produselor pe care le-a comandat, pretul pentru fiecare produs in functie de numarul de bucati comandate si numele furnizorului care aprovizeaza floraria cu respectivul produs. Daca el a comandat acelasi produs de doua ori, dar in comenzi diferite se vor afisa de doua ori produsul, pretul si furnizorul.

### **Rezolvare:**

```
create or replace procedure DateComenzi
```

```
    (nume client.numeclient%type)
```

```
is
```

```
    type tabel_info_rec is record(
        numar_produse number,
        nume_produs varchar2(255),
        pret number,
        nume_furnizor varchar2(255)
    );
```

```
    type tabel_info is table of tabel_info_rec index by pls_integer;
```

```
    type tabel_verificare is table of client%rowtype index by pls_integer;
```

```
    tabel_date tabel_info;
```

```
    tabel_client tabel_verificare;
```

```
    no_data_found1 exception;
```

```
    no_data_found2 exception;
```

```
too_many_rows exception;
begin
    select *
    bulk collect into tabel_client
    from client
    where numeclient = nume;

    if sql%notfound then
        raise no_data_found1;
    end if;

    if tabel_client.count >=2 then
        raise too_many_rows;
    end if;

    select dc.numarproduse, p.numeprodus, p.pret, f.numefurnizor
    bulk collect into tabel_date
    from client cl
    join comanda co on cl.id_client = co.id_client
    join detalii_comanda dc on co.id_comanda = dc.id_comanda
    join produs p on dc.id_produs = p.id_produs
    join furnizor f on p.id_furnizor = f.id_furnizor
    where cl.numeclient = nume;

    if sql%notfound then
        raise no_data_found2;
    end if;

    for i in tabel_date.first..tabel_date.last loop
        dbms_output.put_line('Pret pentru numarul de articole comandate: ' ||
tabel_date(i).numar_produse * tabel_date(i).pret);
        dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
        dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
        dbms_output.new_line;
```

```

end loop;

exception
  when no_data_found1 then
    dbms_output.put_line('Nu exista clientul cu acel nume.');
  when no_data_found2 then
    dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are
atribuita vreo comanda.');
  when too_many_rows then
    dbms_output.put_line('Exista mai multi clienti cu acest nume.');
  when others then
    dbms_output.put_line('Codul de eroare: ' || SQLCODE);
    dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

```

```

create or replace procedure DateComenzi
  (nume client.numeclient%type)
is
  type tabel_info_rec is record(
    numar_produse number,
    nume_produs varchar2(255),
    pret number,
    nume_furnizor varchar2(255)
  );

  type tabel_info is table of tabel_info_rec index by pls_integer;
  type tabel_verificare is table of client%rowtype index by pls_integer;

  tabel_date tabel_info;
  tabel_client tabel_verificare;

  no_data_found1 exception;
  no_data_found2 exception;
  too_many_rows exception;
begin
  select *
  bulk collect into tabel_client
  from client
  where numeclient = nume;

  if sql%notfound then
    raise no_data_found1;
  end if;

  if tabel_client.count >=2 then
    raise too_many_rows;
  end if;

```

```

select dc.numarproduse, p.nume_produs, p.pret, f.nume_furnizor
bulk collect into tabel_date
from client cl
join comanda co on cl.id_client = co.id_client
join detalii_comanda dc on co.id_comanda = dc.id_comanda
join produs p on dc.id_produs = p.id_produs
join furnizor f on p.id_furnizor = f.id_furnizor
where cl.numeclient = nume;

if sql%notfound then
    raise no_data_found2;
end if;

for i in tabel_date.first..tabel_date.last loop
    dbms_output.put_line('Pret pentru numarul de articole comandate: ' || tabel_date(i).numar_produse * tabel_date(i).pret);
    dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
    dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
    dbms_output.new_line;
end loop;

exception
    when no_data_found1 then
        dbms_output.put_line('Nu exista clientul cu acel nume.');
    when no_data_found2 then
        dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
    when too_many_rows then
        dbms_output.put_line('Există mai multi clienti cu acest nume.');
    when others then
        dbms_output.put_line('Codul de eroare: ' || SQLCODE);
        dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

```

Script Output x Query Result x  
✖ ✖ ✖ ✖ | Task completed in 0.106 seconds

Procedure DATECOMENZI compiled

## Apelarea subprogramului (SQL\*PLUS), exemple:

execute DateComenzi('Nica');

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the PL/SQL code for the `DateComenzi` procedure. The 'Dbms Output' tab shows the results of executing the procedure with the parameter 'Nica'. The output includes product details for 'Buchet trandafiri' and 'Fresco Verde' at price 100, product details for 'Frezie galbenă' and 'Happy Flower Holland' at price 150, and product details for 'Vaza de sticla' and 'Natures Garden Supplies' at price 25. The 'Script Output' and 'Query Result' tabs show the message 'PL/SQL procedure successfully completed.'

```
dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
dbms_output.new_line;
end loop;

exception
when no_data_found1 then
dbms_output.put_line('Nu exista clientul cu acel nume.');
when no_data_found2 then
dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
when too_many_rows then
dbms_output.put_line('Exista mai multi clienti cu acest nume.');
when others then
dbms_output.put_line('Codul de eroare: ' || SQLCODE);
dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

--Apelare:
--SQL*PLUS
execute DateComenzi('Nica');
execute DateComenzi('Lisman');
```

PL/SQL procedure successfully completed.

Pret pentru numarul de articole comandate: 100  
Nume produs: Buchet trandafiri  
Nume furnizor: Fresco Verde

Pret pentru numarul de articole comandate: 150  
Nume produs: Frezie galbenă  
Nume furnizor: Happy Flower Holland

Pret pentru numarul de articole comandate: 25  
Nume produs: Vaza de sticla  
Nume furnizor: Natures Garden Supplies

Pret pentru numarul de articole comandate: 15  
Nume produs: Felicitare  
Nume furnizor: Bloom Nursery Co.

execute DateComenzi('Lisman');

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the PL/SQL code for the `DateComenzi` procedure. The 'Dbms Output' tab shows the result of executing the procedure with the parameter 'Lisman', which outputs the message 'Nu exista clientul cu acel nume.' The 'Script Output' and 'Query Result' tabs show the message 'PL/SQL procedure successfully completed.'

```
dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
dbms_output.new_line;
end loop;

exception
when no_data_found1 then
dbms_output.put_line('Nu exista clientul cu acel nume.');
when no_data_found2 then
dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
when too_many_rows then
dbms_output.put_line('Exista mai multi clienti cu acest nume.');
when others then
dbms_output.put_line('Codul de eroare: ' || SQLCODE);
dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

--Apelare:
--SQL*PLUS
execute DateComenzi('Nica');
execute DateComenzi('Lisman');
```

PL/SQL procedure successfully completed.

Nu exista clientul cu acel nume.

Pentru a putea vedea un exemplu in care se intra pe exceptia 'too\_many\_rows' am modificat numele unui client astfel incat sa avem doi clienti cu numele 'Apostol'.

UPDATE client

SET numeclient = 'Apostol'

WHERE numeclient = 'Popescu';

execute DateComenzi('Apostol');

The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, there is a PL/SQL procedure named DateComenzi. The procedure loops through a table of orders, printing the price, quantity, and supplier information. It includes exception handling for cases where no data is found or too many rows are found. The final line of the procedure is an execute statement for DateComenzi('Apostol').

```
for i in tabel_date.first..tabel_date.last loop
    dbms_output.put_line('Pret pentru numarul de articole comandate: ' || tabel_date(i).numar_produse * t
    dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
    dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
    dbms_output.new_line;
end loop;

exception
  when no_data_found1 then
    dbms_output.put_line('Nu exista clientul cu acel nume.');
  when no_data_found2 then
    dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
  when too_many_rows then
    dbms_output.put_line('Există mai multi clienti cu acest nume.');
  when others then
    dbms_output.put_line('Codul de eroare: ' || SQLCODE);
    dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

execute DateComenzi('Apostol');
```

In the Dbms Output tab, the message "Exista mai multi clienti cu acest nume." is displayed, indicating that the exception was triggered. In the Script Output tab, the message "PL/SQL procedure successfully completed." is shown, indicating the task was completed in 0.029 seconds.

Pentru a putea vedea un exemplu in care se intra pe exceptia 'no\_data\_found2' inseram un nou client care nu are asociata vreo comanda.

insert into CLIENT

values (35, 20, 'Dogaru', 'Mihai', 'mihaidogaru@gmail.com', '0758564358');

execute DateComenzi('Dogaru');

```

    for i in tabel_date.first..tabel_date.last loop
        dbms_output.put_line('Pret pentru numarul de articole comandate: ' || tabel_date(i).numar_produce * t);
        dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
        dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
        dbms_output.new_line;
    end loop;

    exception
        when no_data_found1 then
            dbms_output.put_line('Nu exista clientul cu acel nume.');
        when no_data_found2 then
            dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
        when too_many_rows then
            dbms_output.put_line('Exista mai multi clienti cu acest nume.');
        when others then
            dbms_output.put_line('Codul de eroare: ' || SQLCODE);
            dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
    end DateComenzi;
/

execute DateComenzi('Dogaru');

```

Script Output | Query Result | Task completed in 0.036 seconds

PL/SQL procedure successfully completed.

## Apelarea subprogramului (PLSQL), exemple (aceleasi ca mai sus):

begin

    DateComenzi('Nica');

end;

/

```

    end loop;

    exception
        when no_data_found1 then
            dbms_output.put_line('Nu exista clientul cu acel nume.');
        when no_data_found2 then
            dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
        when too_many_rows then
            dbms_output.put_line('Exista mai multi clienti cu acest nume.');
        when others then
            dbms_output.put_line('Codul de eroare: ' || SQLCODE);
            dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
    end DateComenzi;
/

--PLSQL
begin
    DateComenzi('Nica');
end;
/

begin
    DateComenzi('Lisman');

```

Script Output | Query Result | Task completed in 0.038 seconds

PL/SQL procedure successfully completed.

Detailed Output:

- Pret pentru numarul de articole comandate: 100  
Nume produs: Buchet trandafiri  
Nume furnizor: Fresco Verde
- Pret pentru numarul de articole comandate: 150  
Nume produs: Frezie galbenă  
Nume furnizor: Happy Flower Holland
- Pret pentru numarul de articole comandate: 25  
Nume produs: Vaza de sticla  
Nume furnizor: Natures Garden Supplies
- Pret pentru numarul de articole comandate: 15  
Nume produs: Felicitare  
Nume furnizor: Bloom Nursery Co.

```

begin
    DateComenzi('Lisman');
end;
/

```

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is:

```

end loop;

exception
    when no_data_found1 then
        dbms_output.put_line('Nu exista clientul cu acel nume.');
    when no_data_found2 then
        dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreodata comanda.');
    when too_many_rows then
        dbms_output.put_line('Există mai multi clienti cu acest nume.');
    when others then
        dbms_output.put_line('Codul de eroare: ' || SQLCODE);
        dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

--PLSQL
begin
    DateComenzi('Lisman');
end;
/
begin
    DateComenzi('Apostol');

```

The output window on the right shows the message: "Nu există clientul cu acel nume."

```

begin
    DateComenzi('Apostol');
end;
/

```

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is identical to the previous one but for 'Apostol':

```

end loop;

exception
    when no_data_found1 then
        dbms_output.put_line('Nu există clientul cu acel nume.');
    when no_data_found2 then
        dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreodata comanda.');
    when too_many_rows then
        dbms_output.put_line('Există mai multi clienti cu acest nume.');
    when others then
        dbms_output.put_line('Codul de eroare: ' || SQLCODE);
        dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

--PLSQL
begin
    DateComenzi('Apostol');
end;
/
begin
    DateComenzi('Apostol');

```

The output window on the right shows the message: "Există mai multi clienti cu acest nume."

```

begin
    DateComenzi('Dogaru');
end;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top-left, there's a toolbar with various icons. Below it is a tab bar with 'Worksheet' and 'Query Builder' selected. The main area is a large text editor containing PL/SQL code. The code includes a procedure 'DateComenzi' that prints a message if no data is found or if there are too many rows. It also includes a PL/SQL block that calls this procedure. The bottom of the editor shows a status bar with 'Task completed in 0.023 seconds'. To the right of the editor is a 'Doms Output' window titled 'docker\_project'. It displays the message: 'Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.' (The client remained in the database even though no command was assigned). At the very bottom of the interface, there's a 'Script Output' tab.

```

dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
dbms_output.new_line;
end loop;

exception
when no_data_found1 then
dbms_output.put_line('Nu exista clientul cu acel nume.');
when no_data_found2 then
dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
when too_many_rows then
dbms_output.put_line('Exista mai multi clienti cu acest nume.');
when others then
dbms_output.put_line('Codul de eroare: ' || SQLCODE);
dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
/

--PLSQL
begin
    DateComenzi('Dogaru');
end;
/

```

## Cerinta 10:

Definiti un trigger de tip LMD la nivel de comanda. Declansati trigger-ul.

### **Enuntul problemei:**

Definiti un trigger care sa permita inserarea, stergerea sau update-ul datelor din tabela 'PRODUS' doar de luni pana vineri intre orele 12:00 - 24:00. De Craciun nu se poate executa niciuna dintre aceste comenzi indiferent de ora.

### **Rezolvare:**

create or replace trigger program\_lucru

before insert or update or delete on produs

begin

if (to\_char(sysdate, 'D') in (7,1)) or (to\_char(sysdate,'HH24') not between 12 and 24) or (to\_char(sysdate, 'DD-MM') = '25-12') then

if inserting then

```

    raise_application_error(-20001, 'Inserarea e permisa doar de luni pana
vineri intre orele 12 si 24 (de Craciun nu se pot face inserari!)');
    elsif deleting then
        raise_application_error(-20001, 'Stergerea e permisa doar de luni pana
vineri intre orele 12 si 24 (de Craciun nu se pot sterge date!)');
    else
        raise_application_error(-20001, 'Update-urile sunt permise doar de luni
pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)');
    end if;
end if;
end;
/

```

The screenshot shows the SQL code for creating a trigger named 'program\_lucru'. The code uses PL/SQL syntax to check the day of the week (sysdate) and time (HH24) against specific constraints. It handles three cases: inserting, deleting, or updating. If any of these actions occur on Saturday (DD-MM) at 25-12, it raises an application error with a message indicating the restriction.

```

create or replace trigger program_lucru
before insert or update or delete on produs
begin
if (to_char(sysdate, 'D') in (7,1)) or (to_char(sysdate,'HH24') not between 12 and 24) or (to_char(sysdate, 'DD-MM') = '25-12') then
    if inserting then
        raise_application_error(-20001, 'Inserarea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face inserari!)');
    elsif deleting then
        raise_application_error(-20001, 'Stergerea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot sterge date!)');
    else
        raise_application_error(-20001, 'Update-urile sunt permise doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)');
    end if;
end if;
end;
/

```

Below the code, the 'Script Output' tab shows the message: 'Trigger PROGRAM\_LUCRU compiled'.

## Declansare trigger:

- Incercam sa inseram ceva in tabel cand nu suntem in timpul programului
- `insert into PRODUS`

`values (102, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where NumeFurnizor='Fresco Verde'), 'Buchet flori albe', 100);`

The screenshot shows the Oracle SQL Developer interface with a tab titled "docker\_project". The "Worksheet" tab is active, displaying a PL/SQL script to create a trigger named "program\_lucru". The script includes logic to check the day of the month (between 12 and 24) and the day of the week (not Saturday or Sunday). It handles insertions, deletions, and updates, raising application errors if the conditions are not met. Below the script, there is a "values" clause that attempts to insert a row into the "PRODUS" table. The "Script Output" tab at the bottom shows the execution results, indicating an error starting at line 24 due to a constraint violation related to the date range.

```
--Cerinta 10
create or replace trigger program_lucru
  before insert or update or delete on produs
begin
  if (to_char(sysdate, 'D') in (7,1)) or (to_char(sysdate,'HH24') not between 12 and 24) or (to_char(sysdate, 'DD-MM') = '25-12') then
    if inserting then
      raise_application_error(-20001, 'Inserarea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face inserari!)');
    elsif deleting then
      raise_application_error(-20001, 'Stergerea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot sterge date!)');
    else
      raise_application_error(-20001, 'Update-uriile sunt permise doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)');
    end if;
  end if;
end;
/

insert into PRODUS
values (102, (select id_categprod from CATEGORIE_PRODUS where NumeCategoria='Buchete'), (select id_furnizor from FURNIZOR where NumeFurnizor='Fresco Verde'), 'Buchet flori albe', 100)
```

- Incercam sa stergem ceva din tabel cand nu suntem in timpul programului  
delete from PRODUS  
where id\_produs = 101;

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Worksheet" and contains a PL/SQL script. The script begins with a trigger body for a table named PRODUS, which restricts updates between 12:00 and 24:00. It then attempts to update the table with a set of values and delete a row where id\_produs = 101. The bottom window, titled "Script Output", shows the execution results, including an error message indicating that the deletion is not allowed because it would violate the trigger's constraint.

```
begin
  if (to_char(sysdate, 'D') in (7,1)) or (to_char(sysdate,'HH24') not between 12 and 24) or (to_char(sysdate, 'D') in (6,8))
    if inserting then
      raise_application_error(-20001, 'Inserarea e permisa doar de luni pana vineri intre orele 12 si 24');
    elsif deleting then
      raise_application_error(-20001, 'Stergerea e permisa doar de luni pana vineri intre orele 12 si 24');
    else
      raise_application_error(-20001, 'Update-urile sunt permise doar de luni pana vineri intre orele 12 si 24');
    end if;
  end if;
end;
/
update PRODUS
set Pret = 2
where id_produs = 101;

delete from PRODUS
where id_produs = 101;
```

Script Output | Query Result | Task completed in 0.041 seconds

```
Error starting at line : 27 in command -
delete from PRODUS
where id_produs = 101
Error report -
ORA-20001: Stergerea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot sterge date!)
ORA-06512: at "PROIECT.PROGRAM_LUCRU", line 6
ORA-04088: error during execution of trigger 'PROIECT.PROGRAM_LUCRU'
```

- Incercam sa facem update cand nu suntem in timpul programului update PRODUS

set Pret = 2

where id\_produs = 101;

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, there is a large block of PL/SQL code defining a trigger named 'program\_lucru'. The trigger handles insert, update, and delete operations on the 'produs' table. It checks if the current date is between 12:00 and 24:00. If so, it raises an application error (-20001) with a message in Romanian. If the operation is a delete, it also checks if the day is Monday through Friday. The trigger body ends with a comment 'de Craciun nu se pot face update-uri!'.

```

before insert or update or delete on produs
begin
  if (to_char(sysdate, 'D') in (7,1)) or (to_char(sysdate,'HH24') not between 12 and 24) or (to_char(sysdate, 'DD-MM'))
    if inserting then
      raise_application_error(-20001, 'Inserarea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)');
    elsif deleting then
      raise_application_error(-20001, 'Stergerea e permisa doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)');
    else
      raise_application_error(-20001, 'Update-urile sunt permise doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)');
    end if;
  end if;
end;
/

```

update PRODUS  
set Pret = 2  
where id\_produs = 101;

In the 'Script Output' tab, the command is run, resulting in an error:

```

Error starting at line : 23 in command -
update PRODUS
set Pret = 2
where id_produs = 101
Error report -
ORA-20001: Update-urile sunt permise doar de luni pana vineri intre orele 12 si 24 (de Craciun nu se pot face update-uri!)
ORA-06512: at "PROIECT.PROGRAM_LUCRU", line 8
ORA-04088: error during execution of trigger 'PROIECT.PROGRAM_LUCRU'

```

## Stergere trigger:

drop trigger program\_lucru;

```

-----  

raise_application_error  

elsif deleting then  

  raise_application_error  

else  

  raise_application_error  

end if;  

end if;  

end;  

/  


```

drop trigger program\_lucru;

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, the 'drop trigger program\_lucru;' command is typed. In the 'Script Output' tab, the command is executed successfully, with the message 'Trigger PROGRAM\_LUCRU dropped.' displayed.

Trigger PROGRAM\_LUCRU dropped.

## Cerinta 11:

Definiti un trigger de tip LMD la nivel de linie. Declansati trigger-ul.

### **Enuntul problemei:**

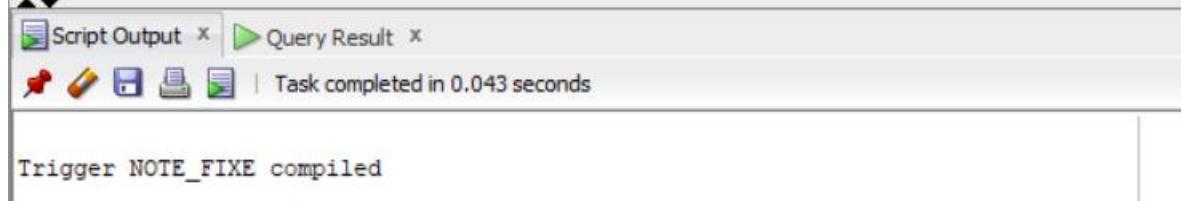
Definiti un trigger care sa impiedice scaderea sau cresterea evaluilor date de client, ele fiind memorate in tabela 'RECENZIE' in atributul 'evaluare'.

### **Rezolvare:**

--Varianta 1:

```
create or replace trigger note_fixe
  before update of evaluare on recenzie
  for each row
begin
  if(:new.evaluare < :old.evaluare) then
    raise_application_error(-20002, 'Evaluarea nu poate fi micsorata');
  end if;
  if(:new.evaluare > :old.evaluare) then
    raise_application_error(-20002, 'Evaluarea nu poate fi marita');
  end if;
end;
/
```

```
create or replace trigger note_fixe
  before update of evaluare on recenzie
  for each row
begin
  if(:new.evaluare < :old.evaluare) then
    raise_application_error(-20002, 'Evaluarea nu poate fi micsorata');
  end if;
  if(:new.evaluare > :old.evaluare) then
    raise_application_error(-20002, 'Evaluarea nu poate fi marita');
  end if;
end;
/
```



## Declansare trigger:

- Incercam sa crestem nota care corespunde recenziei cu id-ul 38

update RECENZIE

set Evaluare = Evaluare + 1

where id\_recenzie = 38;

The screenshot shows the Oracle SQL Developer interface. The top window is a Worksheet containing PL/SQL code. The bottom window is a Script Output window showing the results of the executed command.

**Worksheet Content:**

```
--VARIANTA 1
create or replace trigger note_fixe
  before update of evaluare on recenzie
  for each row
begin
  if(:new.evaluare < :old.evaluare) then
    raise_application_error(-20002, 'Evaluarea nu poate fi micsorat');
  end if;
  if(:new.evaluare > :old.evaluare) then
    raise_application_error(-20002, 'Evaluarea nu poate fi marita');
  end if;
end;
/

update RECENZIE
set Evaluare = Evaluare + 1
where id_recenzie = 38;
```

**Script Output Content:**

```
Error starting at line : 22 in command -
update RECENZIE
set Evaluare = Evaluare + 1
where id_recenzie = 38
Error report -
ORA-20002: Evaluarea nu poate fi marita
ORA-06512: at "PROIECT.NOTE_FIXE", line 6
ORA-04088: error during execution of trigger 'PROIECT.NOTE_FIXE'
```

- Incercam sa scadem nota care corespunde recenziei cu id-ul 38

update RECENZIE

set Evaluare = Evaluare - 1

where id\_recenzie = 38;

docker\_project

Worksheet Query Builder

```
--VARIANTA 1
create or replace trigger note_fixe
    before update of evaluare on recenzie
    for each row
begin
    if(:new.evaluare < :old.evaluare) then
        raise_application_error(-20002, 'Evaluarea nu poate fi micsorata');
    end if;
    if(:new.evaluare > :old.evaluare) then
        raise_application_error(-20002, 'Evaluarea nu poate fi marita');
    end if;
end;
/

update RECENTIE
set Evaluare = Evaluare - 1
where id_recenzie = 38;
```

Script Output    Query Result

Task completed in 0.039 seconds

```
Error starting at line : 22 in command -
update RECENTIE
set Evaluare = Evaluare - 1
where id_recenzie = 38
Error report -
ORA-20002: Evaluarea nu poate fi micsorata
ORA-06512: at "PROIECT.NOTE_FIXE", line 3
ORA-04088: error during execution of trigger 'PROIECT.NOTE_FIXE'
```

## Stergere trigger:

```
drop trigger note_fixe;
```

```
if(:new.evaluare < :old.evaluare)
    raise_application_error(-20002, 'Evaluarea nu poate fi micsorata');
end if;
if(:new.evaluare > :old.evaluare)
    raise_application_error(-20002, 'Evaluarea nu poate fi marita');
end if;
end;
/

drop trigger note_fixe;
```

Script Output    Query Result

Task completed in 0.028 seconds

```
Trigger NOTE_FIXE dropped.
```

## **Rezolvare:**

--Varianta 2

create or replace trigger note\_fixe

    before update of evaluare on recenzie

    for each row

        when (new.evaluare < old.evaluare or new.evaluare > old.evaluare)

begin

        raise\_application\_error(-20002, 'Evaluarea nu poate fi modificata!');

end;

/

```
create or replace trigger note_fixe
    before update of evaluare on recenzie
    for each row
        when (new.evaluare < old.evaluare or new.evaluare > old.evaluare)
begin
    raise_application_error(-20002, 'Evaluarea nu poate fi modificata!');
end;
/
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the trigger. Below the code editor is a toolbar with icons for script output, query result, and other database operations. A status bar at the bottom indicates "Task completed in 0.037 seconds". In the bottom-left corner of the main workspace, the message "Trigger NOTE\_FIXE compiled" is displayed.

## **Declansare trigger:**

update RECENZIE

set Evaluare = Evaluare + 1

where id\_recenzie = 38;

SQL docker\_project

Worksheet Query Builder

```
--VARIANTA 2
create or replace trigger note_fixe
    before update of evaluare on recenzie
    for each row
    when (new.evaluare < old.evaluare or new.evaluare > old.evaluare)
begin
    raise_application_error(-20002, 'Evaluarea nu poate fi modificata!');
end;
/

update RECENTIE
set Evaluare = Evaluare + 1
where id_recenzie = 38;

drop trigger note_fixe;
```

Script Output X Query Result X

| Task completed in 0.035 seconds

```
Error starting at line : 18 in command -
update RECENTIE
set Evaluare = Evaluare + 1
where id_recenzie = 38
Error report -
ORA-20002: Evaluarea nu poate fi modificata!
ORA-06512: at "PROIECT.NOTE_FIXE", line 2
ORA-04088: error during execution of trigger 'PROIECT.NOTE_FIXE'
```

## Stergere trigger:

```
drop trigger note_fixe;
    when new.evaluare < old
begin
    raise_application_error(
end;
/

drop trigger note_fixe;
```

Script Output X Query Result X

| Task completed in 0.01

Trigger NOTE\_FIXE dropped.

## Rezolvare:

--Varianta 3

create or replace procedure procedura

is

begin

    raise\_application\_error(-20000, 'Evaluarea nu poate fi modificata!');

end;

/

create or replace trigger note\_fixe

    before update of evaluare on recenzie

    for each row

        when (new.evaluare <> old.evaluare)

begin

    procedura;

end;

/

```
--VARIANTA 3
create or replace procedure procedura
is
begin
    raise_application_error(-20000, 'Evaluarea nu poate fi modificata!');
end;
/

create or replace trigger note_fixe
    before update of evaluare on recenzie
    for each row
        when (new.evaluare <> old.evaluare)
begin
    procedura;
end;
/
```

Script Output X Query Result X



| Task completed in 0.065 seconds

Procedure PROCEDURA compiled

Trigger NOTE\_FIXE compiled

## Declansare trigger:

```
update RECENZIE  
set Evaluare = Evaluare - 1  
where id_recenzie = 38;
```

The screenshot shows the Oracle SQL Developer interface. The top window is a 'Worksheet' tab where a PL/SQL script is being written. The script creates a trigger named 'note\_fixe' that fires before an update on the 'RECENZIE' table. If the new evaluation value is different from the old one, it raises an application error. The update statement within the trigger body attempts to decrease the evaluation by 1 for a specific record (id\_recenzie = 38). The bottom window is a 'Script Output' tab showing the execution results, which include an error message indicating that evaluation cannot be modified.

```
begin  
    raise_application_error(-20000, 'Evaluarea nu poate fi modificata!');  
end;  
/  
  
create or replace trigger note_fixe  
    before update of evaluare on recenzie  
    for each row  
    when (new.evaluare <> old.evaluare)  
begin  
    procedura;  
end;  
/  
  
update RECENZIE  
set Evaluare = Evaluare - 1  
where id_recenzie = 38;  
  
drop trigger note_fixe;
```

```
Error starting at line : 27 in command -  
update RECENZIE  
set Evaluare = Evaluare - 1  
where id_recenzie = 38  
Error report -  
ORA-20000: Evaluarea nu poate fi modificata!  
ORA-06512: at "PROIECT.PROCEDURA", line 4  
ORA-06512: at "PROIECT.NOTE_FIXE", line 2  
ORA-04088: error during execution of trigger 'PROIECT.NOTE_FIXE'
```

## Stergere trigger:

```
drop trigger note_fixe;
    for each row
        when (new.evaluare <> old.
begin
    procedura;
end;
/
drop trigger note_fixe;
```

The screenshot shows the Oracle SQL Developer interface. In the top panel, there is a code editor window containing the SQL script to drop the 'note\_fixe' trigger. Below the code editor is a toolbar with various icons. The bottom panel displays the 'Script Output' tab, which shows the message 'Trigger NOTE\_FIXE dropped.' indicating the successful execution of the command.

```
Script Output X | Query Result X
| Task completed in 0.043 s
Trigger NOTE_FIXE dropped.
```

## Rezolvare:

--Varianta 4

```
create or replace trigger note_fixe
before update of evaluare on recenzie
for each row
when (new.evaluare <> old.evaluare)
call procedura
```

The screenshot shows the Oracle SQL Developer interface. In the top panel, there is a code editor window containing the SQL script to create a trigger named 'note\_fixe'. The trigger is defined with a 'before update of evaluare on recenzie' constraint, 'for each row' loop, and a 'when (new.evaluare <> old.evaluare)' condition, followed by a 'call procedura' statement. Below the code editor is a toolbar with various icons. The bottom panel displays the 'Script Output' tab, which shows the message 'Trigger NOTE\_FIXE compiled' indicating the successful compilation of the trigger.

```
--VARIANTA 4
create or replace trigger note_fixe
before update of evaluare on recenzie
for each row
when (new.evaluare <> old.evaluare)
call procedura
```

```
Script Output X | Query Result X
| Task completed in 0.039 seconds
Trigger NOTE_FIXE compiled
```

## Declansare trigger:

update RECENZIE

set Evaluare = Evaluare + 3

where id\_recenzie = 38;

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a project named "docker\_project". Below it, the main window has tabs for "Worksheet" and "Query Builder", with "Worksheet" currently selected. The code area contains the following SQL script:

```
--VARIANTA 4
create or replace trigger note_fixe
  before update of evaluare on recenzie
  for each row
  when (new.evaluare <> old.evaluare)
  call procedura
/
update RECENZIE
set Evaluare = Evaluare + 3
where id_recenzie = 38;
drop trigger note_fixe;
```

Below the code area, there are two tabs: "Script Output" and "Query Result". The "Script Output" tab shows the execution results:

```
Error starting at line : 16 in command -
update RECENZIE
set Evaluare = Evaluare + 3
where id_recenzie = 38
Error report -
ORA-20000: Evaluarea nu poate fi modificata!
ORA-06512: at "PROIECT.PROCEDURA", line 4
ORA-06512: at "PROIECT.NOTE_FIXE", line 1
ORA-04088: error during execution of trigger 'PROIECT.NOTE_FIXE'
```

## Stergere trigger:

drop trigger note\_fixe;

```
for each row
when (new.evaluare <>
call procedura
/
drop trigger note_fixe;
```

```
--Cerinta 12
anasta_table_modificare_at
```

```
Script Output x | Query Result x
x x x x | Task completed in 0
```

Trigger NOTE\_FIXE dropped.

## Cerinta 12:

Definiti un trigger de tip LDD. Declansati trigger-ul.

### **Enuntul problemei:**

Definiti un trigger care sa memoreze utilizatorul, numele bazei de date, actiunea efectuata, numele obiectului, tipul obiectului si data la care a avut loc modificarea, intr-un tabel nou creat numit ‘MODIFICARI\_EFFECTUATE’ de fiecare data cand utilizatorul foloseste o comanda LDD.

### **Rezolvare:**

--Cod pentru a crearea tabelului:

```
create table modificari_efectuate(
    utilizator varchar2(100),
    baza_de_date varchar2(100),
    actiune varchar2(100),
    nume_object varchar2(100),
    tip_object varchar2(100),
    data_modificare date
);
```

```
Table MODIFICARI_EFFECTUATE created.
```

```
create table modificari_efectuate(
    utilizator varchar2(100),
    baza_de_date varchar2(100),
    actiune varchar2(100),
    nume_object varchar2(100),
    tip_object varchar2(100),
    data_modificare date
);
```

--Trigger:

```
create or replace trigger Modificare
    after create or alter or drop on schema
begin
    insert into modificari_efectuate
        values(sys.login_user, sys.database_name, sys.sysevent,
        sys.dictionary_obj_name, sys.dictionary_obj_type, sysdate);
end;
/
```

```
create or replace trigger Modificare
  after create or alter or drop on schema
begin
  insert into modificari_efectuate
    values(sys.login_user, sys.database_name, sys.sysevent, sys.dictionary_obj_name, sys.dictionary_obj_type, sysdate);
end;
/
```

Script Output X Query Result X  
Trigger MODIFICARE compiled

## Declansare trigger:

--Cream un tabel nou:

```
create table test_trigger(
  id number,
  nume varchar2(50),
  nota number
);
```

Table TEST\_TRIGGER created.

```
create table test_trigger(
  id number,
  nume varchar2(50),
  nota number
);
```

--Stergem o coloana din tabel:

```
alter table test_trigger
drop column nota;
```

Table TEST\_TRIGGER altered.

```
alter table test_trigger
drop column nota;
```

--Adaugam o coloanal in tabel:

```
alter table test_trigger
add (prenume varchar(50));
```

Table TEST\_TRIGGER altered.

```
alter table test_trigger
add (prenume varchar(50));
```

--Stergem tabelul:

```
drop table test_trigger;
```

Table TEST\_TRIGGER dropped.

```
drop table test_trigger;
```

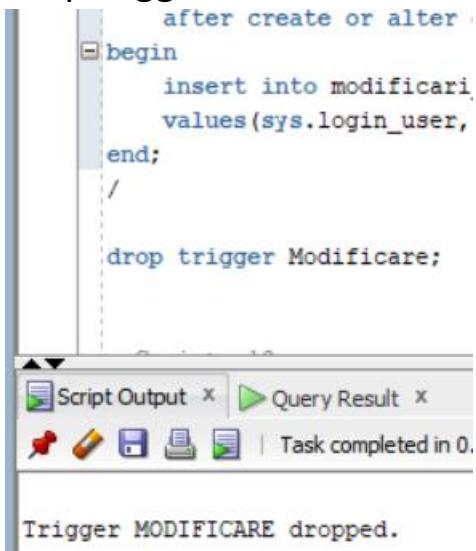
--Obervam ca in tabelul 'MODIFICARI\_EFFECTUATE' s-au memorat datele corespunzatoare de fiecare data cand utilizatorul foloseste o comanda LDD, datorita triggerului definit mai sus.

```
select *
from modificari_efectuate;
```

UTILIZATOR	BAZA_DE_DATE	ACTIUNE	NUME_OBIECT	TIP_OBIECT	DATA_MODIFICARE
1 PROIECT	ORCLPDB1	CREATE	TEST_TRIGGER	TABLE	06-JAN-24
2 PROIECT	ORCLPDB1	ALTER	TEST_TRIGGER	TABLE	06-JAN-24
3 PROIECT	ORCLPDB1	ALTER	TEST_TRIGGER	TABLE	06-JAN-24
4 PROIECT	ORCLPDB1	DROP	TEST_TRIGGER	TABLE	06-JAN-24

### Stergere trigger:

```
drop trigger Modificare;
```



```
after create or alter
begin
    insert into modificari
    values(sys.login_user,
end;
/

drop trigger Modificare;
```

### Cerinta 13:

Definiti un pachet care sa contine toate obiectele definite in cadrul proiectului.

```
create or replace package pachet as
procedure AdaugaProdus
(v_id_produs produs.id_produs%type,
 v_numecategorie categorie_produs.numecategorie%type,
 v_numefurnizor furnizor.numefurnizor%type,
 v_numeprodus produs.numeprodus%type,
 v_pret produs.pret%type);
```

```
procedure AfiseazaProduseDupaCategorie;
```

```
function ComenziAngajat
    (p_numelangajat angajat.numeangajat%type,
     p_prenume_angajat angajat.prenumeangajat%type)
return number;
```

```
procedure DateComenzi
    (nume client.numeclient%type);
end pachet;
/
```

```
create or replace package body pachet as
```

```
procedure AdaugaProdus
```

```
(v_id_produs produs.id_produs%type,
 v_numecategorie categorie_produs.numecategorie%type,
 v_numefurnizor furnizor.numefurnizor%type,
 v_numeprodus produs.numeprodus%type,
 v_pret produs.pret%type)
```

```
is
```

```
type tablou_indexat is table of varchar2(255) index by pls_integer;
```

```
type tablou_imbricat is table of varchar2(255);
```

```
type vector is varray(50) of varchar2(255);
```

```
tablou_categprod tablou_indexat;
```

```
tablou_furnizori tablou_imbricat := tablou_imbricat();
```

```
vector_produse vector := vector();
```

```
v_categprod_exista boolean := false;
```

```
v_furnizor_exista boolean := false;
```

```
v_produs_exista boolean := false;
```

```
v_id_produs_exista number;
```

```
begin
```

```
select NumeCategorie
```

```
bulk collect into tablou_categprod
```

```
from categorie_produs;
```

```
for i in tablou_categprod.first..tablou_categprod.last loop
    if tablou_categprod(i) = v_numecategorie then
        v_categprod_exista := true;
        exit;
    end if;
end loop;
```

```
select NumeFurnizor
bulk collect into tablou_furnizori
from furnizor;
```

```
for i in tablou_furnizori.first..tablou_furnizori.last loop
    if tablou_furnizori(i) = v_numefurnizor then
        v_furnizor_exista := true;
        exit;
    end if;
end loop;
```

```
select NumeProdus
bulk collect into vector_produse
from produs;
```

```
for i in vector_produse.first..vector_produse.last loop
    if vector_produse(i) = v_numeprodus then
        v_produs_exista := true;
        exit;
    end if;
end loop;
```

```
select count(*)
into v_id_produs_exista
from produs
where id_produs = v_id_produs;
```

```
if v_categprod_exista = false then
    dbms_output.put_line('Categoria de produs nu exista.');
end if;

if v_furnizor_exista = false then
    dbms_output.put_line('Furnizorul nu exista.');
end if;

if v_produs_exista = true then
    dbms_output.put_line('Produsul deja exista.');
end if;

if v_id_produs_exista > 0 then
    dbms_output.put_line('ID-ul produsului nu este unic.');
end if;

if v_categprod_exista = false or v_furnizor_exista = false or v_produs_exista =
true or v_id_produs_exista > 0 then
    return;
end if;

insert into produs
values
(v_id_produs,
(select id_categprod from categorie_produs where numecategorie =
v_numecategorie),
(select id_furnizor from furnizor where numefurnizor = v_numefurnizor),
v_numeprodus,
v_pret);

dbms_output.put_line('Produsul a fost adaugat cu succes!');
exception
when others then
```

```

    raise_application_error(-20002,'A aparut o eroare!');
end AdaugaProdus;

procedure AfiseazaProduseDupaCategorie is
    cursor CursorCategorii is
        select id_categprod id_categorie, numecategorie nume_categorie
        from CATEGORIE_PRODUS;

    cursor CursorProduse (v_id_categprod categorie_produs.id_categprod%type)
is
    select numeprodus nume_produs, id_produs idprodus
    from PRODUS
    where id_categprod = v_id_categprod;

    var_nume_produs produs.numeprodus%type;
    var_id_produs produs.id_produs%type;
begin
    for i in CursorCategorii loop
        dbms_output.put_line('Categorie: ' || i.nume_categorie);
        open CursorProduse(i.id_categorie);
        loop
            fetch CursorProduse into var_nume_produs, var_id_produs;
            exit when CursorProduse%notfound;
            dbms_output.put_line(' Produs: ' || var_nume_produs || '(id = ' ||
var_id_produs || ')');
        end loop;
        dbms_output.put_line('');
        close CursorProduse;
    end loop;
end AfiseazaProduseDupaCategorie;

function ComenziAngajat(
    p_nume_angajat angajat.numeangajat%type,
    p_prenume_angajat angajat.prenumeangajat%type

```

```
) return number is
    cursor CursorClienti (id_ang comanda.id_angajat%type) is
        select id_client id_client_
        from comanda
        where id_angajat = id_ang;

    var_id_client client.id_client%type;
    v_id_angajat angajat.id_angajat%type;
    nr_comenzi number;
    nume client.numeclient%type;
    prenume client.prenumeclient%type;
    angajat_unic number;
    id_rol_angajat number;
    comenzi_reduse number;

    --exceptii
    angajatul_nu_exista exception;
    angajatul_nu_are_comenzi exception;
    angajat_care_nu_e_responsabil_de_comenzi exception;
begin
    select count(*) into angajat_unic
    from angajat
    where NumeAngajat = p_nume_angajat and PrenumeAngajat =
p_prenume_angajat;

    if angajat_unic < 1 then
        raise angajatul_nu_exista;
    end if;

    select id_angajat into v_id_angajat
    from angajat
    where NumeAngajat = p_nume_angajat and PrenumeAngajat =
p_prenume_angajat;
```

```
select id_rol into id_rol_angajat
from angajat
where id_angajat = v_id_angajat;

if id_rol_angajat not in (1,3) then
    raise angajat_care_nu_e_responsabil_de_comenzi;
end if;

select count(*) into nr_comenzi
from comanda
where id_angajat = v_id_angajat;

if nr_comenzi = 0 then
    raise angajatul_nu_are_comenzi;
end if;

dbms_output.put_line('Numele clientilor care au plasat comenzile:');
open CursorClienti(v_id_angajat);
loop
    fetch CursorClienti into var_id_client;
    exit when CursorClienti%notfound;
    select numeclient, prenumeclient into nume, prenume
    from client
    where id_client = var_id_client;
    dbms_output.put_line(' ' || nume || ' ' || prenume);
end loop;
close CursorClienti;

select count (*) into comenzi_reduse
from comanda c
left join reducere r on c.id_reducere = r.id_reducere
left join angajat a on a.id_angajat = c.id_angajat
where a.numeangajat = p_nume_angajat and a.prenumeangajat =
p_prenume_angajat and r.codreducere is not null;
```

```

dbms_output.put_line('Numar comenzi reduse:');
dbms_output.put_line(comenzi_reduse);

dbms_output.put_line('Numar comenzi:');
return nr_comenzi;
exception
when angajatul_nu_exista then
  dbms_output.put_line('Acest angajat nu există');
  return -1;
when angajatul_nu_are_comenzi then
  dbms_output.put_line('Angajatului nu îl corespunde nicio comandă');
  return -1;
when angajat_care_nu_e_responsabil_de_comenzi then
  dbms_output.put_line('Angajatul nu poate să se ocupe de comenzi
deoarece are un rol care nu îl permite să facă acest lucru');
  return -1;
when others then
  dbms_output.put_line('Alta eroare: ' || SQLERRM);
  return -1;
end;

procedure DateComenzi
  (nume client.numeclient%type)
is
type tabel_info_rec is record(
  numar_produse number,
  nume_produs varchar2(255),
  pret number,
  nume_furnizor varchar2(255)
);
type tabel_info is table of tabel_info_rec index by pls_integer;
type tabel_verificare is table of client%rowtype index by pls_integer;

```

```
tabel_date tabel_info;
tabel_client tabel_verificare;

no_data_found1 exception;
no_data_found2 exception;
too_many_rows exception;

begin
    select *
    bulk collect into tabel_client
    from client
    where numeclient = nume;

    if sql%notfound then
        raise no_data_found1;
    end if;

    if tabel_client.count >=2 then
        raise too_many_rows;
    end if;

    select dc.numarproduse, p.numeprodus, p.pret, f.numefurnizor
    bulk collect into tabel_date
    from client cl
    join comanda co on cl.id_client = co.id_client
    join detalii_comanda dc on co.id_comanda = dc.id_comanda
    join produs p on dc.id_produs = p.id_produs
    join furnizor f on p.id_furnizor = f.id_furnizor
    where cl.numeclient = nume;

    if sql%notfound then
        raise no_data_found2;
    end if;
```

```
for i in tabel_date.first..tabel_date.last loop
    dbms_output.put_line('Pret pentru numarul de articole comandate: ' || 
tabel_date(i).numar_produse * tabel_date(i).pret);
    dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
    dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
    dbms_output.new_line;
end loop;

exception
    when no_data_found1 then
        dbms_output.put_line('Nu exista clientul cu acel nume.');
    when no_data_found2 then
        dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are
atribuita vreo comanda.');
    when too_many_rows then
        dbms_output.put_line('Exista mai multi clienti cu acest nume.');
    when others then
        dbms_output.put_line('Codul de eroare: ' || SQLCODE);
        dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
end pachet;
/
```

```

if sql%notfound then
    raise no_data_found2;
end if;

for i in tabel_date.first..tabel_date.last loop
    dbms_output.put_line('Pret pentru numarul de articole comandate: ' || tabel_date(i).numar_produse * tabel_date(i).pret);
    dbms_output.put_line('Nume produs: ' || tabel_date(i).nume_produs);
    dbms_output.put_line('Nume furnizor: ' || tabel_date(i).nume_furnizor);
    dbms_output.new_line;
end loop;

exception
    when no_data_found1 then
        dbms_output.put_line('Nu exista clientul cu acel nume.');
    when no_data_found2 then
        dbms_output.put_line('Clientul a ramas in baza de date chiar daca nu are atribuita vreo comanda.');
    when too_many_rows then
        dbms_output.put_line('Exista mai multi clienti cu acest nume.');
    when others then
        dbms_output.put_line('Codul de eroare: ' || SQLCODE);
        dbms_output.put_line('Mesajul erorii: ' || SQLERRM);
end DateComenzi;
end pachet;
/

```

Script Output X | Query Result X

| Task completed in 0.07 seconds

Package PACHET compiled

Package Body PACHET compiled

## Apelarea procedurilor si a functiei definite in pachet:

declare

    v\_rezultat\_functie number;

begin

    dbms\_output.put\_line('Cerinta 6');

    pachet.AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);

    dbms\_output.put\_line(null);

    dbms\_output.put\_line('Cerinta 7');

    pachet.AfiseazaProduseDupaCategorie;

    dbms\_output.put\_line(null);

    dbms\_output.put\_line('Cerinta 8');

    v\_rezultat\_functie := pachet.ComenziAngajat('Ionescu','Anastasia');

    dbms\_output.put\_line(v\_rezultat\_functie);

    dbms\_output.put\_line(null);

```

dbms_output.put_line('Cerinta 9');
pachet.DateComenzi('Nica');
dbms_output.put_line(null);
end;
/
declare
    v_rezultat_functie number;
begin
    dbms_output.put_line('Cerinta 6');
    pachet.AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
    dbms_output.put_line(null);

    dbms_output.put_line('Cerinta 7');
    pachet.AfiseazaProduseDupaCategorie;
    dbms_output.put_line(null);

    dbms_output.put_line('Cerinta 8');
    v_rezultat_functie := pachet.ComenziAngajat('Ionescu','Anastasia');
    dbms_output.put_line(v_rezultat_functie);
    dbms_output.put_line(null);

    dbms_output.put_line('Cerinta 9');
    pachet.DateComenzi('Nica');
    dbms_output.put_line(null);
end;
/

```

The screenshot shows the Oracle SQL Developer interface with a PL/SQL procedure in the Worksheet tab and its execution results in the Dbms Output tab.

**Worksheet Tab:**

```

--apelare
declare
    v_rezultat_functie number;
begin
    dbms_output.put_line('Cerinta 6');
    pachet.AdaugaProdus(71,'Ingrijire','Bloom Londra','Bujor',50);
    dbms_output.put_line(null);

    dbms_output.put_line('Cerinta 7');
    pachet.AfiseazaProduseDupaCategorie;
    dbms_output.put_line(null);

    dbms_output.put_line('Cerinta 8');
    v_rezultat_functie := pachet.ComenziAngajat('Ionescu','Anastasia');
    dbms_output.put_line(v_rezultat_functie);
    dbms_output.put_line(null);

    dbms_output.put_line('Cerinta 9');
    pachet.DateComenzi('Nica');
    dbms_output.put_line(null);
end;
/

```

**Dbms Output Tab:**

- Cerinta 6**
  - Categorie de produs nu exista.
  - Furnizorul nu exista.
  - Produsul deja exista.
  - ID-ul produsului nu este unic.
- Cerinta 7**
  - Categorie: Buchete
    - Produs: Buchet flori roz (id = 100)
    - Produs: Buchet flori galbene (id = 101)
    - Produs: Buchet trandafiri (id = 66)
    - Produs: Buchet de vara (id = 67)
    - Produs: Buchet de primavara (id = 68)
  - Categorie: Plante de interior
    - Produs: Dracena (id = 69)
    - Produs: Ficus lyrata (id = 70)
    - Produs: Trandafir japonez (id = 71)
  - Categorie: Plante de exterior
    - Produs: Trandafir englezesc (id = 72)
    - Produs: Cupressus sempervirens (id = 73)
    - Produs: Bujor (id = 74)
  - Categorie: Accesoriu
    - Produs: Ghiveci ceramic (id = 75)
    - Produs: Vaza de sticla (id = 76)
    - Produs: Felicitare (id = 77)
  - Categorie: La bucata
    - Produs: Trandafir roz (id = 78)
    - Produs: Crin alb (id = 79)
    - Produs: Frezie galbenă (id = 80)

PL/SQL procedure successfully completed.

```

Pachet.AfiseazaProduse( / , ANGAJAT , Denumirea , Numar , pret )
dbms_output.put_line(null);

dbms_output.put_line('Cerinta 7');
pachet.AfiseazaProduseDupaCategoria;
dbms_output.put_line(null);

dbms_output.put_line('Cerinta 8');
v_rezultat_functie := pachet.ComenziAngajat('Ionescu','Anastasia');
dbms_output.put_line(v_rezultat_functie);
dbms_output.put_line(null);

dbms_output.put_line('Cerinta 9');
pachet.DateComenzi('Nica');
dbms_output.put_line(null);

end;
/

```

Script Output X | Query Result X  
✖ ✚ ✖ ✖ | Task completed in 0.046 seconds

PL/SQL procedure successfully completed.

#### Cerinta 8

Numele clientilor care au plasat comenzi:

Feurdean Mihai  
Tintea Alexandru  
Dogareci Bianca

Numar comenzi reduse:

2

Numar comenzi:

3

#### Cerinta 9

Pret pentru numarul de articole comandate: 100

Nume produs: Buchet trandafiri

Nume furnizor: Fresco Verde

Pret pentru numarul de articole comandate: 150

Nume produs: Frezie galbenă

Nume furnizor: Happy Flower Holland

Pret pentru numarul de articole comandate: 25

Nume produs: Vaza de sticla

Nume furnizor: Natures Garden Supplies

Pret pentru numarul de articole comandate: 15

Nume produs: Felicitare

Nume furnizor: Bloom Nursery Co.