

Gestionarea Restaurantelor dintr-un Oraș

**Popescu Vasile-Alin
Grupa 244**

Proiect SGBD anul II semestrul 1

1. Prezența pe scurt baza de date (utilitatea ei).

Mai multe restaurante dintr-un oraș comercializează diverse preparate, pe care le pot livra la o locație stabilită de client.

Dacă același magazin are două sau mai multe locații, restaurantul va fi inserat de mai multe ori și ele pot avea aceleași preparate dar și preparate diferite.

Pentru fiecare preparat se știu: denumirea, ingredientele, prețul, numărul de calorii, greutatea și timpul de preparare.

Ingredientele pot fi locale sau importate din alte țări. Pentru fiecare ingredient se cunosc: denumirea, țara de origine, preț per sută de grame și data expirării;

Țările sunt caracterizate prin: nume, monedă și continentul pe care se află.

Un client poate plasa una sau mai multe comenzi.

O comandă poate conține unul sau mai multe produse (nu neapărat toate de la același restaurant).

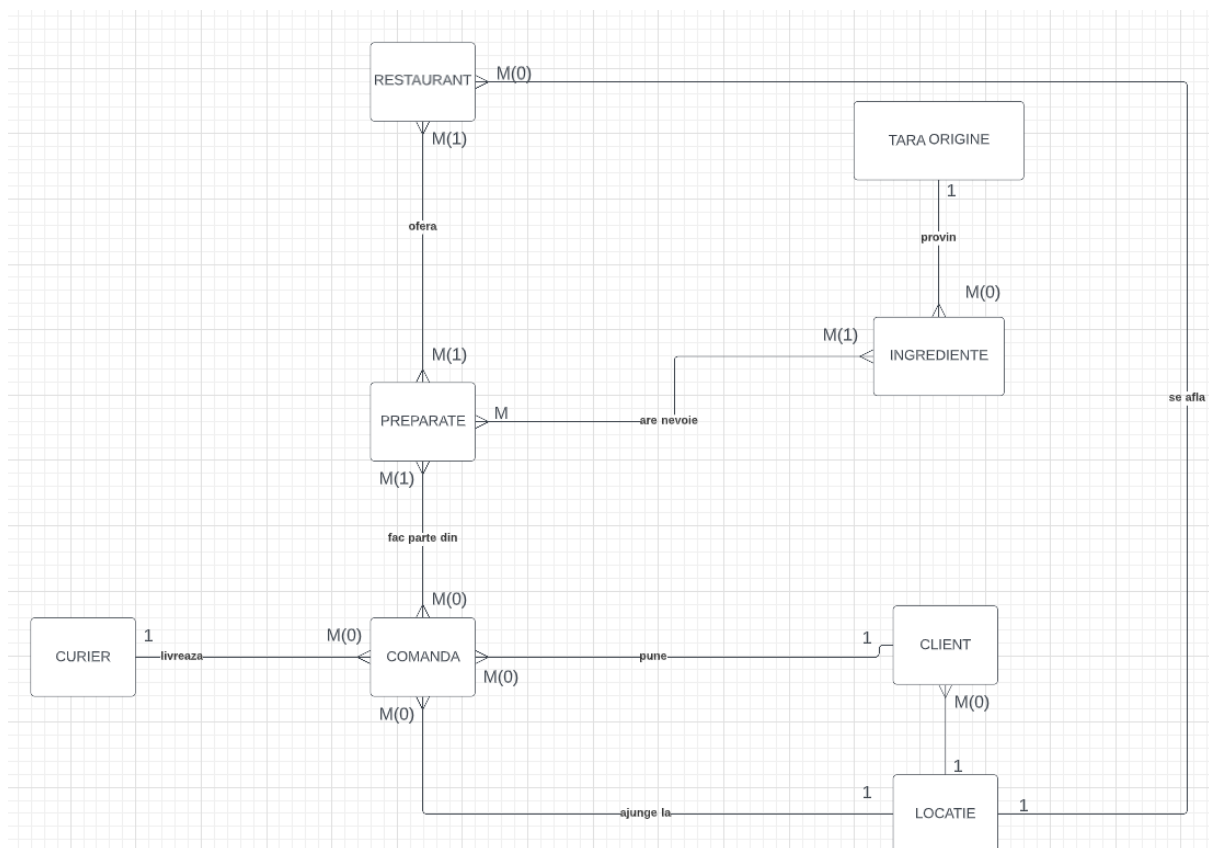
Livrarea se face la o locație aleasă de client, în ziua în care s-a efectuat comanda.

Clienții sunt caracterizați prin: nume, prenume, număr de telefon și adresă.

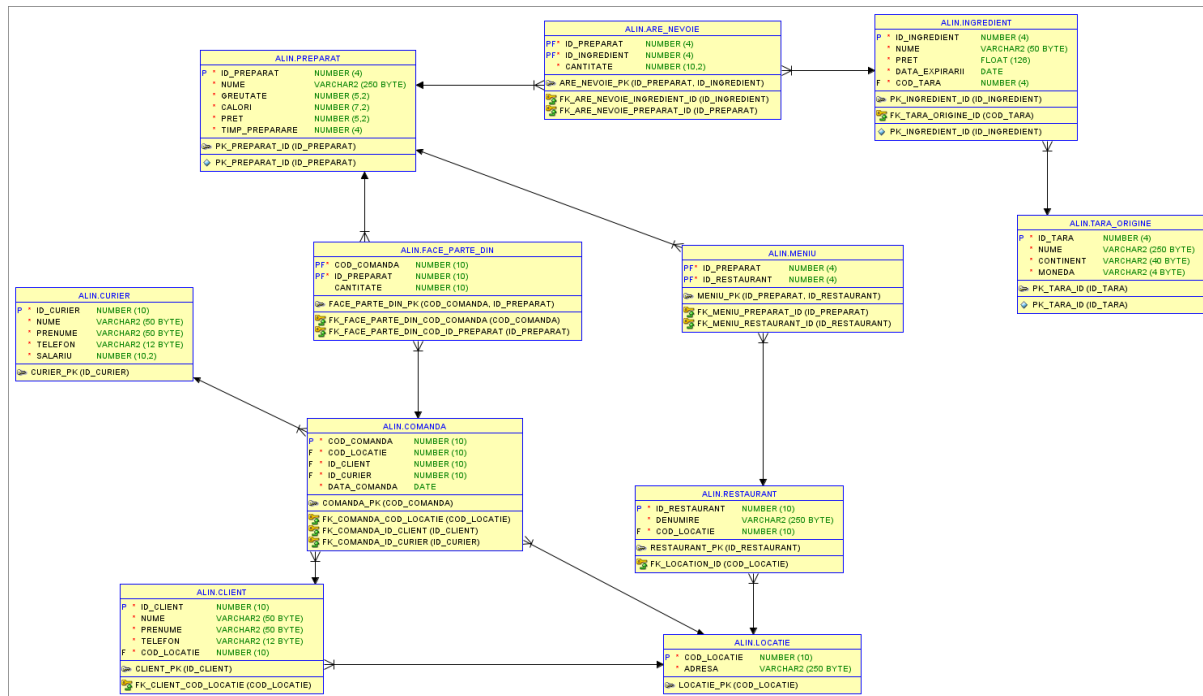
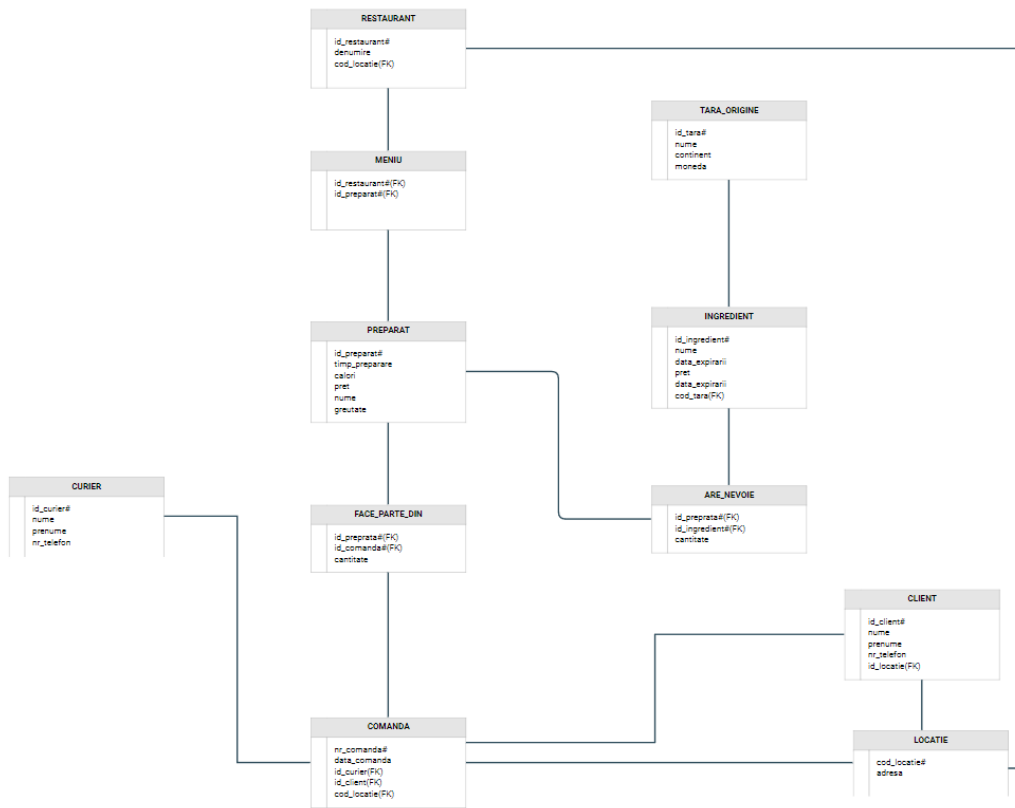
Comenzile sunt livrate de curieri, aceștia fiind caracterizați prin nume, prenume, număr de telefon și salariu.

La o anumită locație se pot afla mai multe restaurante/clienți sau se pot livra mai multe comenzi.

2. Realizați diagrama entitate-relație (ERD).



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



4 și 5. Implementați în Oracle diagrama conceptuală realizată și adăugarea informațiilor.

```
CREATE TABLE PREPARAT(  
  id_preparat number(4) CONSTRAINT PK_PREPARAT_ID PRIMARY KEY,  
  nume VARCHAR2(250) NOT NULL,  
  greutate number(5,2) DEFAULT 0 NOT NULL,  
  calorii number(7,2) DEFAULT 0 NOT NULL,  
  pret number(5,2) DEFAULT 0 NOT NULL,  
  timp_preparare number(4) DEFAULT 0 NOT NULL  
);
```

```
INSERT INTO preparat  
VALUES (1,'Sarmalute in foi de vita',200, 2000, 5.2,120);
```

```
INSERT INTO preparat  
VALUES (2,'Cartofi copti',250, 2500, 7,30);
```

```
INSERT INTO preparat  
VALUES (3,'Supa de legume',150, 1000, 3 ,60);
```

```
INSERT INTO preparat  
VALUES (4,'Friptura',200, 3000, 12,25);
```

```
INSERT INTO preparat  
VALUES (5,'Clatite',100, 2000, 9.90,15);
```

	ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1	Sarmalute in foi de vita	200	2000	5.2	120
2	2	Cartofi copti	250	2500	7	30
3	3	Supa de legume	150	1000	3	60
4	4	Friptura	200	3000	12	25
5	5	Clatite	100	2000	9.9	15

```
CREATE TABLE TARA_ORIGINE(  
  id_tara number(4) CONSTRAINT PK_TARA_ID PRIMARY KEY ,  
  nume VARCHAR2(250) NOT NULL,  
  continent VARCHAR2(40) NOT NULL,  
  moneda VARCHAR(4) NOT NULL  
);
```

```
INSERT INTO tara_origine  
VALUES (1,'Romania','Europa', 'RON');
```

```
INSERT INTO tara_origine  
VALUES (2,'Italia','Europa', 'EUR');
```

```
INSERT INTO tara_origine
VALUES (3,'UK','Europa', 'GBP');
```

```
INSERT INTO tara_origine
VALUES (4,'Germania','Europa', 'EUR');
```

```
INSERT INTO tara_origine
VALUES (5,'USA','America de Nord', 'USD');
```

ID_TARA	NUME	CONTINENT	MONEDA
1	1 Romania	Europa	RON
2	2 Italia	Europa	EUR
3	3 UK	Europa	GBP
4	4 Germania	Europa	EUR
5	5 USA	America de Nord	USD

```
CREATE TABLE INGREDIENT(
  id_ingredient number(4) CONSTRAINT PK_INGREDIENT_ID PRIMARY KEY ,
  nume VARCHAR2(50) NOT NULL,
  pret float DEFAULT 0 NOT NULL,
  data_expirarii date NOT NULL,
  cod_tara number(4) NOT NULL
);
```

```
--adaugam cheia externa
--legam tara de origine de ingredient
alter table INGREDIENT
add constraint FK_TARA_ORIGINE_ID
foreign key (cod_tara) references TARA_ORIGINE(id_tara);
```

```
INSERT INTO INGREDIENT
VALUES (1,'Sare', 0.5,'10/may/24',5);
```

```
INSERT INTO INGREDIENT
VALUES (2,'Otet', 1,'1/may/23',4);
```

```
INSERT INTO INGREDIENT
VALUES (3,'Faina', 5,'10/JUN/23',5);
```

```
INSERT INTO INGREDIENT
VALUES (4,'Carne', 1,'29/JAN/24',5);
```

```
INSERT INTO INGREDIENT
VALUES (5,'Orez', 2,'10/JUN/23',1);
```

```

INSERT INTO INGREDIENT
VALUES (6,'Vanilie', 3,'10/JUN/23',5);
INSERT INTO INGREDIENT
VALUES (7,'Vanilie', 3,sysdate+30,5);

```

ID_INGREDIENT	NUME	PRET	DATA_EXPIRARII	COD_TARA
1	1 Sare	0.5	10-MAY-24	5
2	2 Otet	1	01-MAY-23	4
3	3 Faina	5	10-JUN-23	5
4	4 Carne	1	29-JAN-24	5
5	5 Orez	2	10-JUN-23	1
6	6 Vanilie	3	10-JUN-23	5
7	7 Vanilie	3	11-FEB-23	5

```

--Tabel Asociativ Ingredient-Preparat
--ARE_NEVOIE
CREATE TABLE ARE_NEVOIE(
    id_preparat number(4) NOT NULL ,
    id_ingredient number(4) NOT NULL,
    cantitate number(10,2) NOT NULL,
    PRIMARY KEY(id_preparat,id_ingredient)
);
--adaugam cheia externa
--legam ingredientul de origine de tabelul asociativ
alter table ARE_NEVOIE
add constraint FK_ARE_NEVOIE_INGREDIENT_ID
foreign key (id_ingredient) references INGREDIENT(id_ingredient);

--adaugam cheia externa
--legam ingredientul de origine de tabelul asociativ
alter table ARE_NEVOIE
add constraint FK_ARE_NEVOIE_PREPARAT_ID
foreign key (id_preparat) references PREPARAT(id_preparat);

INSERT INTO are_nevoie VALUES(1,1,20);

INSERT INTO are_nevoie VALUES(2,4,5);
INSERT INTO are_nevoie VALUES(1,4,6);
INSERT INTO are_nevoie VALUES(3,2,10);
INSERT INTO are_nevoie VALUES(2,2,23);
INSERT INTO are_nevoie VALUES(3,5,100);
INSERT INTO are_nevoie VALUES(1,2,20);
INSERT INTO are_nevoie VALUES(2,1,55);
INSERT INTO are_nevoie VALUES(5,6,12);
INSERT INTO are_nevoie VALUES(5,5,10);

```

	ID_PREPARAT	ID_INGREDIENT	CANTITATE
1	1	1	20
2	2	4	5
3	1	4	6
4	3	2	10
5	2	2	23
6	3	5	100
7	1	2	20
8	2	1	55
9	5	6	12
10	5	5	10

	NUME	NUME_1
1	Sarmalute in foi de vita	Sare
2	Cartofi copti	Sare
3	Sarmalute in foi de vita	Otet
4	Cartofi copti	Otet
5	Supa de legume	Otet
6	Sarmalute in foi de vita	Carne
7	Cartofi copti	Carne
8	Supa de legume	Orez
9	Clatite	Orez
10	Clatite	Vanilie

---LOCATIE--

```

CREATE TABLE LOCATIE(
    cod_locatie number(10) ,
    adresa VARCHAR2(250) NOT NULL,
    PRIMARY KEY(cod_locatie)
);
INSERT INTO locatie
VALUES(1,'Str. Valea cu Tei, Pitesti');
INSERT INTO locatie
VALUES(2,'Bulevardul I.C. Bratianu, Pitesti');
INSERT INTO locatie
VALUES(3,'Str. Dimitrie Butculescu, Pitesti');
INSERT INTO locatie
VALUES(4,'Str. Calea Craiovei, Pitesti');
INSERT INTO locatie
VALUES(5,'Str. Osiei, Pitesti');
INSERT INTO locatie
VALUES(6,'Str. Independetei, comuna Bradu');

```

	⚡ COD_LOCATIE	⚡ ADRESA
1		1 Str. Valea cu Tei, Pitesti
2		2 Bulevardul I.C. Bratianu, Pitesti
3		3 Str. Dimitrie Butculescu, Pitesti
4		4 Str. Calea Craiovei, Pitesti
5		5 Str. Osiei, Pitesti
6		6 Str. Independetei, comuna Bradu

--RESTAURANT--

```
CREATE TABLE RESTAURANT(
  id_restaurant number(10) ,
  denumire VARCHAR2(250) NOT NULL,
  cod_locatie number(10) NOT NULL,
  PRIMARY KEY(id_restaurant)
);
```

--adaugam cheia externa pentru locatie

alter table RESTAURANT

add constraint FK_LOCATION_ID

foreign key (cod_locatie) references LOCATIE(cod_locatie);

INSERT INTO restaurant

VALUES (1,'Pizza Luca Pitesti',2);

INSERT INTO restaurant

VALUES (2,'Atelier',1);

INSERT INTO restaurant

VALUES (3,'Papao',2);

INSERT INTO restaurant

VALUES (4,'Pizza Luca Pitesti2',3);

INSERT INTO restaurant

VALUES (5,'Mateo',1);

	⚡ ID_RESTAURANT	⚡ DENUMIRE	⚡ COD_LOCATIE
1		1 Pizza Luca Pitesti	2
2		2 Atelier	1
3		3 Papao	2
4		4 Pizza Luca Pitesti2	3
5		5 Mateo	1

--Tabel asociativ restaurant preparat.

--Meniu

CREATE TABLE MENIU(

id_preparat number(4) NOT NULL ,

id_restaurant number(4) NOT NULL,


```
PRIMARY KEY(id_preparat,id_restaurant)
);
```

```
alter table MENIU
add constraint FK_MENIU_PREPARAT_ID
foreign key (id_preparat) references PREPARAT(id_preparat);
```

```
alter table MENIU
add constraint FK_MENIU_RESTAURANT_ID
foreign key (id_restaurant) references RESTAURANT(id_restaurant);
```

```
insert into MENIU VALUES(3, 1);
insert into MENIU VALUES(2, 1);
insert into MENIU VALUES(3, 4);
insert into MENIU VALUES(2, 4);
insert into MENIU VALUES(1, 5);
insert into MENIU VALUES(1, 2);
insert into MENIU VALUES( 2, 5);
insert into MENIU VALUES(5, 5);
insert into MENIU VALUES( 2, 3);
insert into MENIU VALUES( 5, 3);
```

	ID_PREPARAT	ID_RESTAURANT
1	1	2
2	1	5
3	2	1
4	2	3
5	2	4
6	2	5
7	3	1
8	3	4
9	5	3
10	5	5

	NUME	DENUMIRE
1	Cartofi copti	Pizza Luca Pitesti
2	Supa de legume	Pizza Luca Pitesti
3	Sarmalute in foi de vita	Atelier
4	Cartofi copti	Papao
5	Clatite	Papao
6	Cartofi copti	Pizza Luca Pitesti2
7	Supa de legume	Pizza Luca Pitesti2
8	Sarmalute in foi de vita	Mateo
9	Cartofi copti	Mateo
10	Clatite	Mateo

--CURIER--

```
CREATE TABLE CURIER(  
    id_curier number(10) ,  
    nume VARCHAR2(50) NOT NULL,  
    prenume VARCHAR2(50) NOT NULL,  
    telefon VARCHAR2(12) NOT NULL,  
    salariu number(10,2) NOT NULL,  
    PRIMARY KEY(id_curier)  
);  
INSERT INTO CURIER  
VALUES (1,'Dragomirescu','Claudia','0772722321',1500);  
INSERT INTO CURIER  
VALUES (2,'Popescu','Marcus','0736969697',2000);  
INSERT INTO CURIER  
VALUES (3,'Ion','Catalin','0736661111',2300.25);  
INSERT INTO CURIER  
VALUES (4,'Popa','Cosmina','0736916765',555.60);  
INSERT INTO CURIER  
VALUES (5,'Costescu','Marian','0736916789',1000);  
select * from CURIER;
```

ID_C...	NUME	PRENUME	TELEFON	SALARIU
1	1 Dragomirescu	Claudia	0772722321	1500
2	2 Popescu	Marcus	0736969697	2000
3	3 Ion	Catalin	0736661111	2300.25
4	4 Popa	Cosmina	0736916765	555.6
5	5 Costescu	Marian	0736916789	1000

--CLIENT--

```
CREATE TABLE CLIENT(  
    id_client number(10) ,  
    nume VARCHAR2(50) CONSTRAINT last_name_client_nn NOT NULL,  
    prenume VARCHAR2(50) CONSTRAINT first_name_client_nn NOT NULL,  
    telefon VARCHAR2(12) CONSTRAINT telefon_client_nn NOT NULL,  
    cod_locatie number(10) NOT NULL,  
    PRIMARY KEY(id_client)  
);
```

```
alter table CLIENT  
add constraint FK_CLIENT_COD_LOCATIE  
foreign key (cod_locatie) references LOCATIE(cod_locatie);
```

```
INSERT INTO CLIENT  
VALUES (1,'Dragomirescu','Claudia','0772744621',1);  
INSERT INTO CLIENT  
VALUES (2,'Popescu','Catalin','0734469697',5);  
INSERT INTO CLIENT
```

```
VALUES (3,'Ion','Marcus','07366615611',4);
INSERT INTO CLIENT
VALUES (4,'Popa','Marian','0736917965',3);
INSERT INTO CLIENT
VALUES (5,'Costescu','Cosmina','0736914759',2);
```

ID_CLIENT	NUME	PRENUME	TELEFON	COD_LOCATIE
1	1 Dragomirescu	Claudia	0772744621	1
2	2 Popescu	Catalin	0734469697	5
3	3 Ion	Marcus	07366615611	4
4	4 Popa	Marian	0736917965	3
5	5 Costescu	Cosmina	0736914759	2

---COMANDA---

```
CREATE TABLE COMANDA(
    cod_comanda number(10) ,
    cod_locatie number(10) NOT NULL,
    id_client number(10) NOT NULL,
    id_curier number(10) NOT NULL,
    data_comanda date NOT NULL,
    PRIMARY KEY(cod_comanda)
);
```

```
alter table COMANDA
add constraint FK_COMANDA_COD_LOCATIE
foreign key (cod_locatie) references LOCATIE(cod_locatie);
```

```
alter table COMANDA
add constraint FK_COMANDA_id_client
foreign key (id_client) references CLIENT(id_client);
```

```
alter table COMANDA
add constraint FK_COMANDA_id_curier
foreign key (id_curier) references CURIER(id_curier);
```

```
INSERT INTO COMANDA
VALUES(1,1,2,1,sysdate-33);
INSERT INTO COMANDA
VALUES(2,2,1,1,sysdate-13);
INSERT INTO COMANDA
VALUES(3,4,2,5,sysdate-39);
INSERT INTO COMANDA
VALUES(4,1,3,1,sysdate-40);
INSERT INTO COMANDA
VALUES(5,5,4,1,sysdate-60);
INSERT INTO COMANDA
VALUES(6,6,4,1,'20-JUN-22');
```

	⚡ COD_COMANDA	⚡ COD_LOCATIE	⚡ ID_CLIENT	⚡ ID_CURIER	⚡ DATA_COMANDA
1	2	2	1	1	30-DEC-22
2	1	1	2	1	10-DEC-22
3	3	4	2	5	04-DEC-22
4	4	1	3	1	03-DEC-22
5	5	5	4	1	13-NOV-22
6	6	6	4	1	20-JUN-22

-- tabela asociativa Preparat-Comanda

```
CREATE TABLE FACE_PARTE_DIN(
  cod_comanda number(10) NOT NULL,
  id_preparat number(10) NOT NULL,
  cantitate number(10) default 1,
  PRIMARY KEY(cod_comanda,id_preparat)
);
```

```
alter table FACE_PARTE_DIN
add constraint FK_FACE_PARTE_DIN_COD_ID_PREPARAT
foreign key (id_preparat) references PREPARAT(id_preparat);
```

```
alter table FACE_PARTE_DIN
add constraint FK_FACE_PARTE_DIN_cod_comanda
foreign key (cod_comanda) references COMANDA(cod_comanda);
```

```
INSERT INTO FACE_PARTE_DIN
VALUES(4,5,1);
INSERT INTO FACE_PARTE_DIN
VALUES(1,2,1);
INSERT INTO FACE_PARTE_DIN
VALUES(2,1,3);
INSERT INTO FACE_PARTE_DIN
VALUES(3,3,1);
INSERT INTO FACE_PARTE_DIN
VALUES(3,2,4);
INSERT INTO FACE_PARTE_DIN
VALUES(3,5,2);
INSERT INTO FACE_PARTE_DIN
VALUES(1,5,1);
INSERT INTO FACE_PARTE_DIN
VALUES(5,5,1);
INSERT INTO FACE_PARTE_DIN
VALUES(4,1,1);
INSERT INTO FACE_PARTE_DIN
VALUES(4,4,3);
```

	⚡ COD_COMANDA	⚡ ID_PREPARAT	⚡ CANTITATE
1	4	5	1
2	1	2	1
3	2	1	3
4	3	3	1
5	3	2	4
6	3	5	2
7	1	5	1
8	5	5	1
9	4	1	1
10	4	4	3

	⚡ COD_COMANDA	⚡ NUME
1	1	Cartofi copti
2	1	Clatite
3	2	Sarmalute in foi de vita
4	3	Cartofi copti
5	3	Supa de legume
6	3	Clatite
7	4	Sarmalute in foi de vita
8	4	Friptura
9	4	Clatite
10	5	Clatite

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

-- pentru fiecare client sa se efiaseze numele, prenumele si o lista cu toate locatiile la care a dorit livrat comenzi cat si data livrarii

--(precizam ca o comanda este livrata in ziua in care a fost plasata)

--colectii folosite:vector si tablou indexat

create or replace procedure ex_6

is

```
type v_client IS RECORD(id client.id_client%type,
                        nume client.nume%type,
                        prenume client.prenume%type
                        );
```

```
type v_adresa IS RECORD(adresa locatie.adresa%type,
                        data_livrare date
                        );
```

TYPE v_clienti IS TABLE OF v_client INDEX BY PLS_INTEGER;

TYPE v_adrese IS varray(200) OF v_adresa;

clienti v_clienti :=v_clienti();

adrese v_adrese :=v_adrese();

sem number(1):=0;

begin

```
select id_client, nume, prenume bulk collect into clienti
from client;
```

for i in clienti.first..clienti.last loop

```
DBMS_OUTPUT.PUT_LINE( clienti(i).nume || ' ' || clienti(i).prenume);
```

```
select adresa,data_comanda bulk collect into adrese
from comanda join locatie using(cod_locatie)
```

```
where id_client=clienti(i).id;
```

```
if adrese.count()>0 then
```

```
for j in adrese.first..adrese.last loop
```

```
for j in adrese.first..adrese.last loop
```

```
DBMS_OUTPUT.PUT_LINE(' ' || adrese(j).adresa || ' la data de:' ||
```

```
adrese(j).data_livrare);
```

```
end loop;
```

else

```
DBMS_OUTPUT.PUT_LINE(' acest client nu a a facut nicio comanda.');
```

```

        end if;
        DBMS_OUTPUT.PUT_LINE(' ');
    end loop;

```

```

end ex_6;
/

```

```

begin
    ex_6;
end;
/

```

The screenshot displays the Oracle SQL Developer environment. The main window is titled 'Project.sql' and contains the following PL/SQL code:

```

        DBMS_OUTPUT.PUT_LINE(' acest client nu a a facut ni
    end if;
    DBMS_OUTPUT.PUT_LINE(' ');
end loop;

end ex_6;
/

begin
    ex_6;
end;
/

```

The 'Dbms Output' pane on the right shows the output of the procedure execution:

```

Dragomirescu Claudia
Bulevardul I.C. Bratianu, Pitesti la data de:30-DEC-22

Popescu Catalin
Str. Valea cu Tei, Pitesti la data de:10-DEC-22
Str. Calea Craiovei, Pitesti la data de:04-DEC-22

Ion Marcus
Str. Valea cu Tei, Pitesti la data de:03-DEC-22

Popa Marian
Str. Osiei, Pitesti la data de:13-NOV-22
Str. Independetei, comuna Bradu la data de:20-JUN-22

Costescu Cosmina
acest client nu a a facut nicio comanda.

```

The 'Script Output' pane at the bottom shows the following messages:

```

?Procedure EX_6 compiled

?PL/SQL procedure successfully completed.

```

7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul

--Generati toate meniurile stocate in baza de date, numele restaurantului si locatia sa vor fi specificat o singura data la inceput, iar pentru fiecare preparat se vor afisa si ingredientele.

--Cursoare folosite:ref-cursor si ciclu cursor cu subcerei

create or replace procedure cerinta_7

is

TYPE refcursor IS REF CURSOR;

cursor preparat_c(id_rest meniu.id_restaurant%type) is

select nume, greutate, calori,pret, cursor (select nume from ingredient join
are_nevoie using(id_ingredient) where id_preparat=p.id_preparat)
from preparat p join meniu m on(p.id_preparat=m.id_preparat)
where id_restaurant=id_rest;

ingr refcursor;

nume preparat.nume%type;

greutate preparat.greutate%type;

calori preparat.calori%type;

pret preparat.pret%type;

nume_ingr ingredient.nume%type;

begin

for rest in (select denumire,id_restaurant,adresa from restaurant join locatie
using(cod_locatie)) loop

DBMS_OUTPUT.PUT_LINE ('Restaurantul : ' || rest.denumire || ', aflat la adresa : ' ||
rest.adresa);

DBMS_OUTPUT.PUT_LINE ("");

open preparat_c(rest.id_restaurant);

loop

FETCH preparat_c INTO nume, greutate, calori, pret, ingr;

exit when preparat_c %notfound;

DBMS_OUTPUT.PUT_LINE (nume || ' pret: ' || pret || ' calori ' || calori || '
greutate ' || greutate);

loop

FETCH ingr INTO nume_ingr;

exit when ingr %notfound;

DBMS_OUTPUT.PUT (nume_ingr || ' ');

end loop;

DBMS_OUTPUT.PUT_line (' ');

DBMS_OUTPUT.PUT_line (' ');

end loop;

/

Worksheet	Query Builder	Project X
	<pre> -- Create procedure preparat_pret&type; nume_ingr ingredient.num&type; begin for rest in (select denumire,id_restaurant,adresa from restaurant join locatie using(cod_locatie)) loop DBMS_OUTPUT.PUT_LINE ('Restaurantul : ' rest.denumire ', afiat la adresa : ' rest.adresa); DBMS_OUTPUT.PUT_LINE (''); open preparat_c(rest.id_restaurant); loop FETCH preparat_c INTO nume, greutate, calorii, pret, ingr; exit when preparat_c %notfound; DBMS_OUTPUT.PUT_LINE (nume ' pret: ' pret ' calorii ' calorii ' greutate ' ' '); loop; FETCH ingr INTO nume_ingr; exit when ingr %notfound; DBMS_OUTPUT.PUT (nume_ingr ' '); end loop; DBMS_OUTPUT.PUT_line (' '); DBMS_OUTPUT.PUT_line (' '); end loop; close preparat_c; DBMS_OUTPUT.PUT_LINE ('-----'); DBMS_OUTPUT.PUT_LINE(''); end loop; </pre>	<pre> Restaurantul : Mateo, afiat la adresa : Str. Valea cu Tei, Pitesti Sarmalute in foi de vita pret: 5.3 calorii 2000 greutate 200 Sare Otet Carne Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Clatite pret: 9.9 calorii 2000 greutate 100 Orez Vanilie ----- Restaurantul : Atelier, afiat la adresa : Str. Valea cu Tei, Pitesti Sarmalute in foi de vita pret: 5.3 calorii 2000 greutate 200 Sare Otet Carne ----- Restaurantul : Pizza Luca Pitesti, afiat la adresa : Bulevardul I.C. Bratianu, Pitesti Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Supa de legume pret: 3 calorii 1000 greutate 150 Otet Orez ----- Restaurantul : Papao, afiat la adresa : Bulevardul I.C. Bratianu, Pitesti Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Clatite pret: 9.9 calorii 2000 greutate 100 Orez Vanilie ----- Restaurantul : Pizza Luca Pitesti2, afiat la adresa : Str. Dimitrie Butculescu, Pitesti Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Supa de legume pret: 3 calorii 1000 greutate 150 Otet Orez </pre>
	<p>Script Output x Query Result x</p> <p>Task completed in 0.055 seconds</p> <p>PL/SQL procedure successfully completed.</p> <p>Procedure CERINTA_7 compiled</p> <p>PL/SQL procedure successfully completed.</p> <p>Messages - Log</p>	

8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate

--Sa se determine id-ul si valoarea celei mai scumpe comenzi livrate de un curier,raportul dintre valoarea determinata si valoarea totala a comenzilor, cat si adresa la care a fost livrata comanda ,

--functia are ca parametru doar prenumele unui curier

--tabele folosite:curier, comanda, locatie, preparat, face_parte_din

--inseram date pentru evidentierea erorilor

```
insert into curier values (6,'test','Marian','0000',0);
```

```
insert into curier values (7,'test','test2','0000',0);
```

```
insert into curier values (8,'test','test3','0000',0);
```

```
INSERT INTO COMANDA VALUES(11,6,4,7,'20-JUN-22');
```

```
INSERT INTO COMANDA VALUES(12,6,4,7,'20-JUN-22');
```

```
INSERT INTO FACE_PARTE_DIN VALUES(11,2,1);
```

```
INSERT INTO FACE_PARTE_DIN VALUES(12,2,1);
```

```
create or replace function cerinta_8(prenume_c curier.prenume%type) return varchar2
is
```

```
    id number(10);
```

```
    comanda number(10):=-1;
```

```
    suma_comanda number(10,2):=0;
```

```
    adresa locatie.adresa%type;
```

```
    suma_totala number(10,2):=0;
```

```
    counter number(10):=0;
```

```
    zero_comenzi exception;
```

```
    prea_multe_comenzi exception;
```

```
begin
```

```
    select id_curier into id
```

```
    from curier
```

```
    where upper(prenume)=upper(prenume_c);
```

```
    for i in ( select id_curier, cod_comanda ,sum(cantitate * pret ) as suma, adresa
```

```
              from comanda join face_parte_din using(cod_comanda)
```

```
              join preparat using(id_preparat)
```

```
              join locatie using(cod_locatie)
```

```
              group by cod_comanda,id_curier,adresa) loop
```

```
        suma_totala:=suma_totala+i.suma;
```

```
        if i.id_curier=id and i.suma>suma_comanda then
```

```
            counter:=1;
```

```
            suma_comanda:=i.suma;
```

```

        comanda:=i.cod_comanda;
        adresa:=i.adresa;
    elsif i.id_curier=id and i.suma=suma_comanda then
        counter:=counter+1;
    end if;
end loop;

if counter=0 then
    raise zero_comenzi;
elsif counter>1 then
    raise prea_multe_comenzi;
end if;

return 'comanda: ' || comanda || ', cu valoarea: ' || suma_comanda || ', reprezinta :'||
round(suma_comanda*100/suma_totala,3) ||'% , livrata la adresa : ' || adresa ;

exception
    when too_many_rows then DBMS_OUTPUT.PUT_LINE('prea multi curieri cu
prenumele : ' ||prenume_c);return null;
    when no_data_found then DBMS_OUTPUT.PUT_LINE('nu exista curieri cu
prenumele : ' ||prenume_c);return null;
    when zero_comenzi then DBMS_OUTPUT.PUT_LINE('curierul cu prenumele : '
||prenume_c || ' nu a livrat comenzi') ;return null;
    when prea_multe_comenzi then DBMS_OUTPUT.PUT_LINE('curierul cu prenumele
: ' ||prenume_c || ' a livrat mai multe comenzi cu valoare maxima : '|| suma_comanda ) ;return
null;
end cerinta_8;
/

begin
    DBMS_OUTPUT.PUT_LINE('too many rows');
    DBMS_OUTPUT.PUT_LINE(cerinta_8('Marian'));

    DBMS_OUTPUT.PUT_LINE('no data found');
    DBMS_OUTPUT.PUT_LINE(cerinta_8('Test'));

    DBMS_OUTPUT.PUT_LINE('mai multe comenzi cu aceeaasi suma');
    DBMS_OUTPUT.PUT_LINE(cerinta_8('Test2'));

    DBMS_OUTPUT.PUT_LINE('curier care nu a livrat comenzi');
    DBMS_OUTPUT.PUT_LINE(cerinta_8('Test3'));

    DBMS_OUTPUT.PUT_LINE('date corecte');
    DBMS_OUTPUT.PUT_LINE(cerinta_8('Claudia'));

```

end;

/

The screenshot displays the SQL Developer interface with a PL/SQL procedure named `CERINTA_8` in the Query Builder. The procedure logic includes a `raise` statement for too many rows, a `return` statement for command details, and an `exception` block for various data states. The `begin` block contains three `DBMS_OUTPUT.PUT_LINE` calls. The Demo Output window shows the execution results, including error messages and the final command details.

```
raise prea_multe_comenzi;
end if;

return 'comanda: ' || comanda || ', cu valoarea: ' || suma_comanda || ', reprezinta: ' || round((suma_comanda*100/suma_totala),3) || '% , livrata la adresa: ' || adresa;

exception
when too_many_rows then DBMS_OUTPUT.PUT_LINE('prea multi curieri cu prenumele: ' || prenume_c || '); return null;
when no_data_found then DBMS_OUTPUT.PUT_LINE('nu exista curieri cu prenumele: ' || prenume_c || '); return null;
when zero_comenzi then DBMS_OUTPUT.PUT_LINE('curierul cu prenumele: ' || prenume_c || ' nu a livrat comenzi'); return null;
when prea_multe_comenzi then DBMS_OUTPUT.PUT_LINE('curierul cu prenumele: ' || prenume_c || ' a livrat mai multe comenzi cu valoare maxima: ' || suma_maxima); return null;
end cerinta_8;

begin
DBMS_OUTPUT.PUT_LINE('too many rows');
DBMS_OUTPUT.PUT_LINE(cerinta_8('Marian'));
DBMS_OUTPUT.PUT_LINE('no data found');
DBMS_OUTPUT.PUT_LINE(cerinta_8('Test1'));
end;
```

Script Output: Task completed in 0.054 seconds

Function CERINTA_8 compiled

PL/SQL procedure successfully completed.

Demo Output:

```
too many rows
prea multi curieri cu prenumele :Marian
no data found
nu exista curieri cu prenumele :Test
mai multe comenzi cu aceeaasi suma
curierul cu prenumele :Test2 a livrat mai multe comenzi cu valoare maxima : 7
curier care nu a livrat comenzi
curierul cu prenumele :Test3 nu a livrat comenzi
date corecte
comanda: 4, cu valoarea: 51.1, reprezinta :32.28% , livrata la adresa : Str. Valea cu Tei, Pitesti
```

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

--Sa se afiseze adresa, numele și nr de telefon al tuturor clienților care au comandat în ultimele 30 de zile un produs care conținea un anumit ingredient dintr-o anumita tara.Se vor primi ca parametri numele tarii si denumirea ingredientului.

--tabele folosite: tara_origine, ingredient, preparat, comanda, client, locatie, face_parte_din
--vom face insert la noi date pentru evidentiarea erorilor.

```
INSERT INTO INGREDIENT  
VALUES (11,'ORZ', 3,sysdate+70,5);
```

```
create or replace procedure cerinta_9 (tara tara_origine.ume%type,ingr  
ingredient.ume%type)
```

```
is  
id_ingr number(10);  
id_tara number(10);  
counter number(10):=0;
```

```
INGREDIENT_NEFOLOSIT exception;  
PREPARAT_NELIVRAT exception;
```

```
begin
```

```
select id_ingredient , id_tara into id_ingr,id_tara  
from tara_origine t join ingredient i on(i.cod_tara=t.id_tara)  
where upper(t.ume)=upper(tara) and upper(i.ume)=upper(ingr) ;
```

```
select count(*) into counter  
from are_nevoie where id_ingredient = id_ingr;
```

```
if counter=0 then  
raise INGREDIENT_NEFOLOSIT;  
end if;
```

```
counter:=0;  
for i in (select distinct c.ume,c.prenume,telefon,cl.adresa  
from preparat join face_parte_din using(id_preparat)  
join comanda com using (cod_comanda)  
join client c using (id_client)  
join locatie cl on (c.cod_locatie=cl.cod_locatie)  
where com.data_comanda>sysdate-30 and id_preparat in (select
```

```

id_preparat from are_nevoie where id_ingredient = id_ingr )) loop
    counter:=counter+1;
    DBMS_OUTPUT.PUT_LINE ('Client:' || i.numel|| ' || i.prenume || ' cu nr de telefon : ' ||
i.telefon || ' cu adresa: ' ||i.adresa);
end loop;

if counter=0 then
    raise PREPARAT_NELIVRAT;
end if;
exception
    when no_data_found then DBMS_OUTPUT.PUT_LINE ('Nu a fost gasit in baza de
date ingredientul: ' || ingr || ' din tara:' || tara);
    when too_many_rows then DBMS_OUTPUT.PUT_LINE ('In baza de date exista de
mai multe ori ingredientul : ' || ingr || ' din tara:' || tara);
    when INGREDIENT_NEFOLOSIT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' ||
ingr || ' din tara:' || tara || ' nu a fost folosit in niciun preparat');
    when PREPARAT_NELIVRAT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' || ingr
|| ' din tara:' || tara || ' nu a fost inclus in niciun preparat livrat in ultimele 30 de zile');
end cerinta_9;
/

begin
    DBMS_OUTPUT.PUT_LINE('eroarea de no_data_found');
    cerinta_9('usa','muschi');

    DBMS_OUTPUT.PUT_LINE('too_many_rows');
    cerinta_9('usa','vanilie');

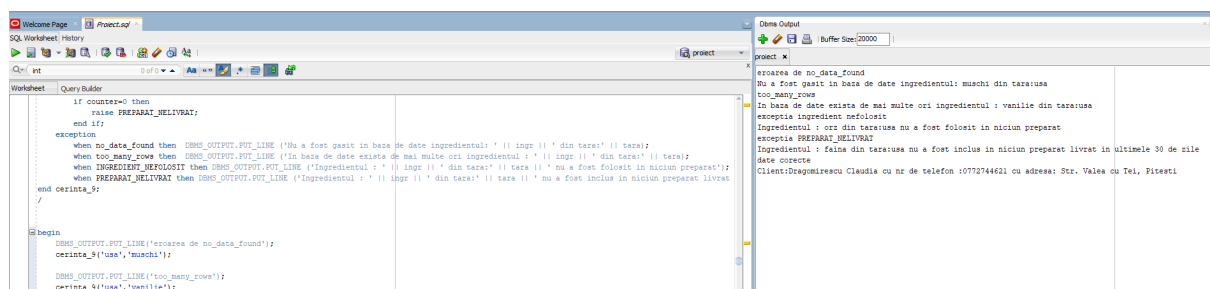
    DBMS_OUTPUT.PUT_LINE('exceptia ingredient nefolosit');
    cerinta_9('usa','orz');

    DBMS_OUTPUT.PUT_LINE('exceptia PREPARAT_NELIVRAT');
    cerinta_9('usa','sare');

    DBMS_OUTPUT.PUT_LINE('date corecte');
    cerinta_9('romania','orez');

end;
/

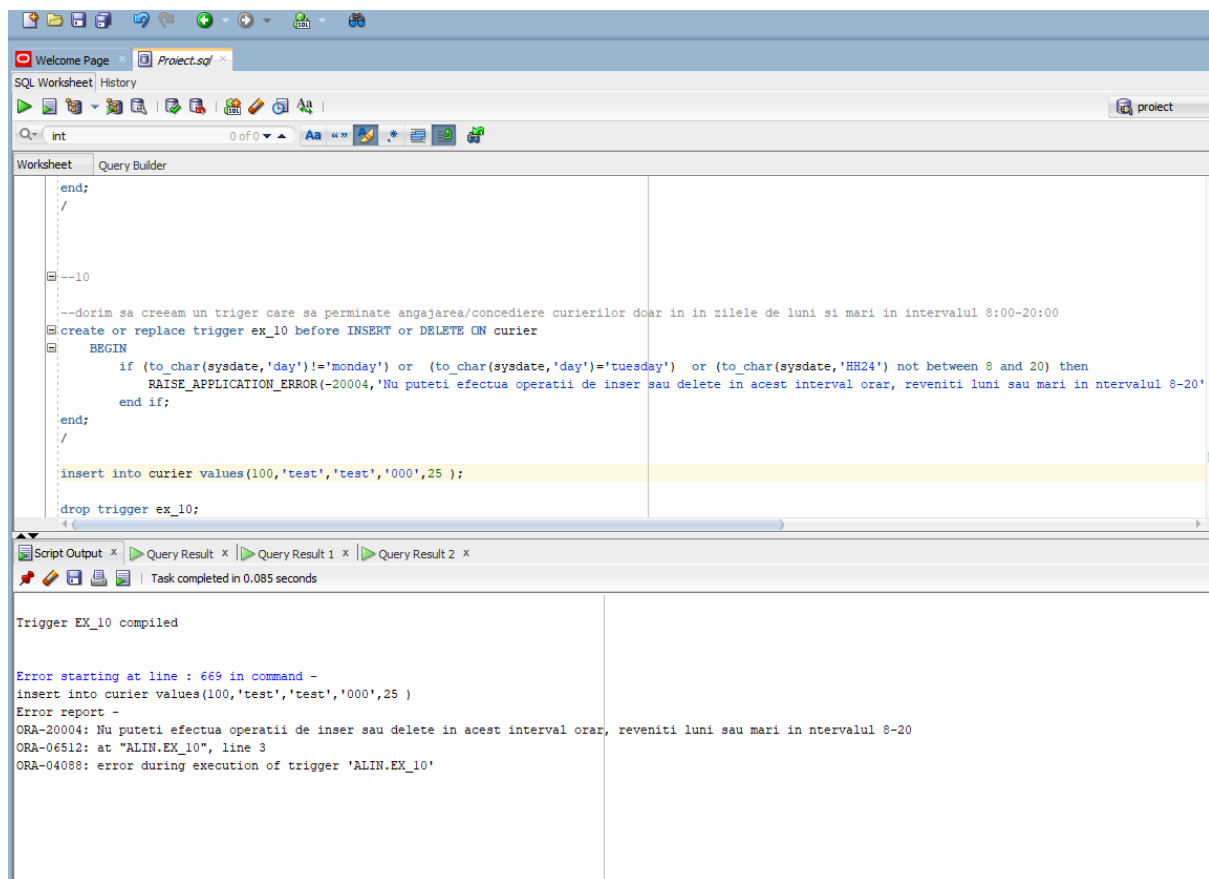
```



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

--dorim sa creem un trigger care sa permiata angajarea/concediere curierilor doar in zilele de luni si mari in intervalul 8:00-20:00

```
create or replace trigger ex_10 before INSERT or DELETE ON curier
BEGIN
    if (to_char(sysdate,'day')!='monday') or (to_char(sysdate,'day')='tuesday') or
    (to_char(sysdate,'HH24') not between 8 and 20) then
        RAISE_APPLICATION_ERROR(-20004,'Nu puteti efectua operatii de inser sau
delete in acest interval orar, reveniti luni sau mari in intervalul 8-20');
    end if;
end;
/
```



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

--Pretul unui preparat se va modifica in functie de pretul ingredientelor. Vom crea un trigger pentru tabela asociativa dintre preparat
-- si ingredient astfel in cat pretul preparatului sa se modifice in functie de pret_ingredient * cantitate/100;
-- un ingredient poate fi adaugat/sters pt un preparat sau i se poate modifica cantitatea
--iar cantitatea noua trebuie sa fie mai mare ca 0;

```
create or replace trigger ex_11 before DELETE OR UPDATE OR INSERT ON are_nevoie
FOR EACH ROW
DECLARE
    p_old number(10,2):=0;
    p_new number(10,2):=0;
BEGIN
    if deleting then
        select pret into p_old from ingredient where id_ingredient = :old.id_ingredient;
        p_old:= p_old * :old.cantitate /100 ;--calculez valoarea pe care trebuie sa o scad.

        update preparat set
            pret=pret-p_old
        where id_preparat= :old.id_preparat;

    elsif updating then
        if :new.cantitate<=0 then
            RAISE_APPLICATION_ERROR(-20002,'Cantitatea nu poate fi un nr negativ sau
0');
        elsif :old.id_preparat = :new.id_preparat and :old.id_ingredient=:new.id_ingredient
then

            select pret into p_old from ingredient where id_ingredient = :old.id_ingredient;
            p_old:= p_old * :old.cantitate /100 ;--calculez valoarea pe care trebuie sa o scad.

            select pret into p_new from ingredient where id_ingredient = :new.id_ingredient;
            p_new:= p_new * :new.cantitate /100 ;--calculez valoarea pe care trebuie sa o
adun.

            update preparat set
                pret=pret + p_new - p_old
            where id_preparat= :new.id_preparat;

        else
            RAISE_APPLICATION_ERROR(-20001,'Nu poti modifica id_preparat sau
id_ingredient!');
        end if;
    else
        if :new.cantitate<=0 then
            RAISE_APPLICATION_ERROR(-20002,'Cantitatea nu poate fi un nr negativ sau
```



```

0');
    end if;
    select pret into p_new from ingredient where id_ingredient = :new.id_ingredient;
    p_new:= p_new * :new.cantitate /100 ;--calculez valoarea pe care trebuie sa o adun.

    update preparat set
        pret=pret + p_new
    where id_preparat= :new.id_preparat;
    end if;
exception
    when no_data_found then RAISE_APPLICATION_ERROR(-20010,'nu exista
ingredient cu acest id');
end;
/

```

--Teste:

--date înainte modificărilor

```
select * from preparat where id_preparat=1;
```

ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1 Sarmalute in foi de vita	200	2000	5.2	120

```
delete from are_nevoie where id_preparat=1 and id_ingredient=1;
```

```
select * from preparat where id_preparat=1;
```

--date după stergerea unui ingredient

ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1 Sarmalute in foi de vita	200	2000	5.1	120

```
insert into are_nevoie values(1,1,20);
```

```
select * from preparat where id_preparat=1;
```

--inseram ingredientul

ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1 Sarmalute in foi de vita	200	2000	5.2	120

```
update are_nevoie set
```

```
    cantitate=40
```

```
where id_preparat=1 and id_ingredient=1;
```

```
select * from preparat where id_preparat=1;
```

ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1 Sarmalute in foi de vita	200	2000	5.3	120

--evidentiam faptul ca nu putem modifica cantitate in 0 sau sa modificăm

--id_ingredient/id_preparat

```
update are_nevoie set
```

```
    id_ingredient=3
```

```
where id_preparat=1 and id_ingredient=1;
```

update are_nevoie set
cantitate=0
where id_preparat=1 and id_ingredient=1;

The screenshot shows the SQL Developer interface with a trigger script in the 'Query Builder' tab. The script is for a trigger named 'ALIN.EX_11'. It contains logic to update the 'are_nevoie' table based on changes to the 'preparat' table. The script includes comments in Romanian. Below the script, the 'Script Output' tab shows the execution results, including a message 'Trigger EX_11 compiled' and two error messages. The first error occurs at line 739, stating 'ORA-20001: Nu poti modifica id_preparat sau id_ingredient!' and 'ORA-06512: at "ALIN.EX_11", line 29'. The second error occurs at line 743, stating 'ORA-20002: Cantitatea nu poate fi un nr negativ sau 0' and 'ORA-06512: at "ALIN.EX_11", line 15'.

```

p_new:= p_new * :new.cantitate /100 ;--calculez valoarea pe care trebuie sa o adun.

update preparat set
pret=pret + p_new - p_old
where id_preparat= :new.id_preparat;

else
RAISE_APPLICATION_ERROR(-20001,'Nu poti modifica id_preparat sau id_ingredient!');
end if;
else
select pret into p_new from ingredient where id_ingredient = :new.id_ingredient;
p_new:= p_new * :new.cantitate /100 ;--calculez valoarea pe care trebuie sa o adun.

update preparat set
pret=pret + p_new
where id_preparat= :new.id_preparat;
end if;
end;

```

Trigger EX_11 compiled

Error starting at line : 739 in command -
update are_nevoie set
id_ingredient=3
where id_preparat=1 and id_ingredient=1
Error report -
ORA-20001: Nu poti modifica id_preparat sau id_ingredient!
ORA-06512: at "ALIN.EX_11", line 29
ORA-04088: error during execution of trigger 'ALIN.EX_11'

Error starting at line : 743 in command -
update are_nevoie set
cantitate=0
where id_preparat=1 and id_ingredient=1
Error report -
ORA-20002: Cantitatea nu poate fi un nr negativ sau 0
ORA-06512: at "ALIN.EX_11", line 15
ORA-04088: error during execution of trigger 'ALIN.EX_11'

-- daca introducem date incorecte:
insert into are_nevoie values(1,100,20);
select * from preparat where id_preparat=1;

ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1 Sarmalute in foi de vita	200	2000	5.3	120

```

Error starting at line : 754 in command -
insert into are_nevoie values(1,100,20)
Error report -
ORA-20010: nu exista ingredient cu acest id
ORA-06512: at "ALIN.EX_11", line 43
ORA-04088: error during execution of trigger 'ALIN.EX_11'

```

insert into are_nevoie values(1,3,0);
select * from preparat where id_preparat=1;

ID_PREPARAT	NUME	GREUTATE	CALORI	PRET	TIMP_PREPARARE
1	1 Sarmalute in foi de vita	200	2000	5.3	120

Error starting at line : 757 in command -
insert into are_nevoie values(1,3,0)
Error report -
ORA-20002: Cantitatea nu poate fi un nr negativ sau 0
ORA-06512: at "ALIN.EX_11", line 33
ORA-04088: error during execution of trigger 'ALIN.EX_11'

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

-dorim sa memoram intr o tabela toate operatiile de create alter sau drop facute pe baza de date.

--se vor memora intr-o tabela de istoric

```
create table istoric(
    id number(10),
    data_efectuare_operatie date,
    utilizator varchar(200),
    operatie varchar(30),
    tip_obiect varchar(200),
    nume_obiect varchar(200),
    PRIMARY KEY(id)
);
/
create or replace trigger istoric_operati AFTER CREATE OR ALTER OR DROP ON
SCHEMA
DECLARE
BEGIN
    INSERT INTO istoric VALUES( (select count(*)+1 from istoric), --id generat automat
                                sysdate,
                                user,
                                SYS.SYSEVENT,
                                SYS.DICTIONARY_OBJ_TYPE,
                                SYS.DICTIONARY_OBJ_NAME);
END;
/
```

– teste care arată funcționare trigger-ului

```
create table teste (id number(10));
alter table teste add nume varchar(20);
drop table teste;
```

```
select * from istoric;
```

```
drop trigger istoric_operati;
```

```

/
create or replace trigger istoric_operati AFTER CREATE OR ALTER OR DROP ON SCHEMA
DECLARE
BEGIN
    INSERT INTO istoric VALUES( (select count(*)+1 from istoric), --id generat automat
                                sysdate,
                                user,
                                SYS.SYSEVENT,
                                SYS.DICTIONARY_OBJ_TYPE,
                                SYS.DICTIONARY_OBJ_NAME);

END;

/

create table teste (id number(10));
alter table teste add nume varchar(20);
drop table teste;

select * from istoric;

drop trigger istoric_operati;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 3 in 0.012 seconds

ID	DATA_EFACTUARE_OPERATIE	UTILIZATOR	OPERATIE	TIP_OBIECT	NUME_OBIECT
1	1 12-JAN-23	ALIN	CREATE	TABLE	TESTE
2	2 12-JAN-23	ALIN	ALTER	TABLE	TESTE
3	3 12-JAN-23	ALIN	DROP	TABLE	TESTE

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

```
CREATE OR REPLACE PACKAGE ex_13 AS
    procedure ex_6 ;
    procedure cerinta_7 ;
    function cerinta_8(prenume_c curier.prenume%type) return varchar2 ;
    procedure cerinta_9 (tara tara_origine.ume%type, ingr ingredient.ume%type);

end ex_13;
/

create or replace package body ex_13 as
    procedure ex_6
    is
        type v_client IS RECORD(id client.id_client%type,
                                nume client.ume%type,
                                prenume client.prenume%type
                                );

        type v_adresa IS RECORD(adresa locatie.adresa%type,
                                data_livrare date
                                );
        TYPE v_clienti IS TABLE OF v_client INDEX BY PLS_INTEGER;
        TYPE v_adrese IS varray(200) OF v_adresa;

        clienti v_clienti :=v_clienti();
        adrese v_adrese :=v_adrese();

        sem number(1):=0;
    begin

        select id_client, nume, prenume bulk collect into clienti
        from client;

        for i in clienti.first..clienti.last loop
            DBMS_OUTPUT.PUT_LINE( clienti(i).nume || ' ' || clienti(i).prenume);

            select adresa,data_comanda bulk collect into adrese
            from comanda join locatie using(cod_locatie)
            where id_client=clienti(i).id;
            if adrese.count(>)>0 then
                for j in adrese.first..adrese.last loop
                    DBMS_OUTPUT.PUT_LINE(' ' || adrese(j).adresa || ' la data de:' ||
adrese(j).data_livrare);
                end loop;
            end if;
        end loop;
    end;
```

```

        else
            DBMS_OUTPUT.PUT_LINE(' acest client nu a a facut nicio comanda. ');
        end if;
        DBMS_OUTPUT.PUT_LINE(' ');
    end loop;

end ex_6;

procedure cerinta_7
is
    TYPE refcursor IS REF CURSOR;
    cursor preparat_c( id_rest meniu.id_restaurant%type) is
        select nume, greutate, calori,pret, cursor (select nume from ingredient join
are_nevoie using(id_ingredient) where id_preparat=p.id_preparat )
        from preparat p join meniu m on(p.id_preparat=m.id_preparat)
        where id_restaurant=id_rest;

    ingr refcursor;
    nume preparat.nume%type;
    greutate preparat.greutate%type;
    calori preparat.calori%type;
    pret preparat.pret%type;
    nume_ingr ingredient.nume%type;

begin

    for rest in (select denumire,id_restaurant,adresa from restaurant join locatie
using(cod_locatie)) loop
        DBMS_OUTPUT.PUT_LINE ('Restaurantul : ' || rest.denumire || ', aflat la adresa : ' ||
rest.adresa );
        DBMS_OUTPUT.PUT_LINE ("");

        open preparat_c(rest.id_restaurant);

        loop
            FETCH preparat_c INTO nume, greutate, calori, pret, ingr;
            exit when preparat_c %notfound;
            DBMS_OUTPUT.PUT_LINE ( nume || ' pret: ' || pret || ' calori ' || calori || '
greutate ' || greutate);

            loop
                FETCH ingr INTO nume_ingr;
                exit when ingr %notfound;
                DBMS_OUTPUT.PUT ( nume_ingr || ' ');
            end loop;
            DBMS_OUTPUT.PUT_line ( ' ');
            DBMS_OUTPUT.PUT_line ( ' ');
        end loop;
    end for;
end cerinta_7;

```

```

        end loop;

    close preparat_c;

    DBMS_OUTPUT.PUT_LINE ('-----');
    DBMS_OUTPUT.PUT_LINE("");
end loop;

end cerinta_7;

function cerinta_8(prenume_c curier.prenume%type) return varchar2
is
    id number(10);
    comanda number(10):=-1;
    suma_comanda number(10,2):=0;
    adresa locatie.adresa%type;
    suma_totala number(10,2):=0;
    counter number(10):=0;
    zero_comenzi exception;
    prea_multe_comenzi exception;
begin
    select id_curier into id
    from curier
    where upper(prenume)=upper(prenume_c);

    for i in ( select id_curier, cod_comanda ,sum(cantitate * pret ) as suma, adresa
              from comanda join face_parte_din using(cod_comanda)
              join preparat using(id_preparat)
              join locatie using(cod_locatie)
              group by cod_comanda,id_curier,adresa) loop
        suma_totala:=suma_totala+i.suma;
        if i.id_curier=id and i.suma>suma_comanda then
            counter:=1;
            suma_comanda:=i.suma;
            comanda:=i.cod_comanda;
            adresa:=i.adresa;
        elsif i.id_curier=id and i.suma=suma_comanda then
            counter:=counter+1;
        end if;
    end loop;

    if counter=0 then
        raise zero_comenzi;
    elsif counter>1 then
        raise prea_multe_comenzi;
    end if;

    return 'comanda: ' || comanda ||', cu valoarea: ' || suma_comanda || ', reprezinta :'||

```



```
round(suma_comanda*100/suma_totala,3) ||'% , livrata la adresa : ' || adresa ;
```

```
exception
```

```
    when too_many_rows then DBMS_OUTPUT.PUT_LINE('prea multi curieri cu  
prenumele : ' || prenume_c) ;return null;
```

```
    when no_data_found then DBMS_OUTPUT.PUT_LINE('nu exista curieri cu  
prenumele : ' || prenume_c) ;return null;
```

```
    when zero_comenzi then DBMS_OUTPUT.PUT_LINE('curierul cu prenumele : '  
|| prenume_c || ' nu a livrat comenzi') ;return null;
```

```
    when prea_multe_comenzi then DBMS_OUTPUT.PUT_LINE('curierul cu prenumele  
: ' || prenume_c || ' a livrat mai multe comenzi cu valoare maxima : ' || suma_comanda ) ;return  
null;
```

```
end cerinta_8;
```

```
procedure cerinta_9 (tara tara_origine.nume%type, ingr ingredient.nume%type)
```

```
is
```

```
    id_ingr number(10);
```

```
    id_tara number(10);
```

```
    counter number(10):=0;
```

```
    INGREDIENT_NEFOLOSIT exception;
```

```
    PREPARAT_NELIVRAT exception;
```

```
begin
```

```
    select id_ingredient , id_tara into id_ingr, id_tara  
    from tara_origine t join ingredient i on (i.cod_tara=t.id_tara)  
    where upper(t.nume)=upper(tara) and upper(i.nume)=upper(ingr) ;
```

```
    select count(*) into counter  
    from are_nevoie where id_ingredient = id_ingr;
```

```
    if counter=0 then  
        raise INGREDIENT_NEFOLOSIT;  
    end if;
```

```
    counter:=0;
```

```
    for i in (select distinct c.nume, c.prenume, telefon, cl.adresa  
        from preparat join face_parte_din using(id_preparat)  
        join comanda com using (cod_comanda)  
        join client c using (id_client)  
        join locatie cl on (c.cod_locatie=cl.cod_locatie)  
        where com.data_comanda>sysdate-30 and id_preparat in (select  
id_preparat from are_nevoie where id_ingredient = id_ingr )) loop  
        counter:=counter+1;  
        DBMS_OUTPUT.PUT_LINE ('Client: ' || i.nume || ' ' || i.prenume || ' cu nr de telefon : '  
|| i.telefon || ' cu adresa: ' || i.adresa);  
    end loop;
```

```

        if counter=0 then
            raise PREPARAT_NELIVRAT;
        end if;
    exception
        when no_data_found then DBMS_OUTPUT.PUT_LINE ('Nu a fost gasit in baza de
date ingredientul: ' || ingr || ' din tara:' || tara);
        when too_many_rows then DBMS_OUTPUT.PUT_LINE ('In baza de date exista de
mai multe ori ingredientul : ' || ingr || ' din tara:' || tara);
        when INGREDIENT_NEFOLOSIT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : '
|| ingr || ' din tara:' || tara || ' nu a fost folosit in niciun preparat');
        when PREPARAT_NELIVRAT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' ||
ingr || ' din tara:' || tara || ' nu a fost inclus in niciun preparat livrat in ultimele 30 de zile');
    end cerinta_9;
end ex_13;
/
begin
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE('exercitiu 6');
    ex_13.ex_6;

    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE('exercitiu 7');
    ex_13.cerinta_7;

    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE('exercitiu 8');

    DBMS_OUTPUT.PUT_LINE('too many rows');
    DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Marian'));
    DBMS_OUTPUT.PUT_LINE(' ');

    DBMS_OUTPUT.PUT_LINE('no data found');
    DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Test'));
    DBMS_OUTPUT.PUT_LINE(' ');

    DBMS_OUTPUT.PUT_LINE('mai multe comenzi cu aceeasi suma');
    DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Test2'));
    DBMS_OUTPUT.PUT_LINE(' ');

    DBMS_OUTPUT.PUT_LINE('curier care nu a livrat comenzi');
    DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Test3'));
    DBMS_OUTPUT.PUT_LINE(' ');

    DBMS_OUTPUT.PUT_LINE('date corecte');

```

```
DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Claudia'));
DBMS_OUTPUT.PUT_LINE(' ');
```

```
DBMS_OUTPUT.PUT_LINE(' ');
DBMS_OUTPUT.PUT_LINE(' ');
DBMS_OUTPUT.PUT_LINE('exercitiu 9');
DBMS_OUTPUT.PUT_LINE('eroarea de no_data_found');
ex_13.cerinta_9('usa','muschi');
DBMS_OUTPUT.PUT_LINE(' ');
```

```
DBMS_OUTPUT.PUT_LINE('too_many_rows');
ex_13.cerinta_9('usa','vanilie');
DBMS_OUTPUT.PUT_LINE(' ');
```

```
DBMS_OUTPUT.PUT_LINE('exceptia ingredient nefolosit');
ex_13.cerinta_9('usa','orz');
DBMS_OUTPUT.PUT_LINE(' ');
```

```
DBMS_OUTPUT.PUT_LINE('exceptia PREPARAT_NELIVRAT');
ex_13.cerinta_9('usa','faina');
DBMS_OUTPUT.PUT_LINE(' ');
```

```
DBMS_OUTPUT.PUT_LINE('date corecte');
ex_13.cerinta_9('USA','sare');
DBMS_OUTPUT.PUT_LINE(' ');
```

end;
/

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab on the left contains a PL/SQL script with line numbers 969 to 985. The script includes an exception block for 'no_data_found' and 'too_many_rows', followed by a call to 'ex_13.cerinta_9' with parameters 'usa' and 'muschi'. The 'Results' window on the right displays the output of the script, showing a list of client names and addresses, such as 'Dragomirescu Claudia' and 'Bulevardul I.C. Bratianu, Pitesti la data de:30-DEC-22'.

```
969      end if;
970      exception
971      when no_data_found then DBMS_OUTPUT.PUT_LINE ('Nu a fost gasit in baza de date ingredientul: ' ||
972      when too_many_rows then DBMS_OUTPUT.PUT_LINE ('In baza de date exista de mai multe ori ingredientul: ' ||
973      when INGREDIENT_NEFOSIT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' || ingr || ' din tara: ' || t
974      when PREPARAT_NELIVRAT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' || ingr || ' din tara: ' || t
975      end cerinta_9;
976      end ex_13;
977      /
978      begin
979      DBMS_OUTPUT.PUT_LINE(' ');
980      DBMS_OUTPUT.PUT_LINE(' ');
981      DBMS_OUTPUT.PUT_LINE('exercitiu 6');
982      ex_13.ex_6;
983
984      DBMS_OUTPUT.PUT_LINE(' ');
985      DBMS_OUTPUT.PUT_LINE(' ');
```

exercitiu 6
Dragomirescu Claudia
Bulevardul I.C. Bratianu, Pitesti la data de:30-DEC-22
Popescu Catalin
Str. Valea cu Tei, Pitesti la data de:10-DEC-22
Str. Calea Craiovei, Pitesti la data de:04-DEC-22
Ion Marcus
Str. Valea cu Tei, Pitesti la data de:03-DEC-22
Popa Marian
Str. Osiei, Pitesti la data de:13-NOV-22
Str. Independetei, comuna Bradu la data de:20-JUN-22
Str. Independetei, comuna Bradu la data de:20-JUN-22
Str. Independetei, comuna Bradu la data de:20-JUN-22
Costescu Cosmina
acest client nu a a facut nicio comanda.

<pre>end loop; if counter=0 then raise PREPARAT_NELIVRAT; end if; exception when no_data_found then DBMS_OUTPUT.PUT_LINE ('Nu a fost gasit in baza de date ingredientul: ' when too_many_rows then DBMS_OUTPUT.PUT_LINE ('In baza de date exista de mai multe ori ingredientul when INGREDIENT_NEFOLOSIT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' ingr ' din tara: ' tara when PREPARAT_NELIVRAT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' ingr ' din tara: ' tara end cerinta_9; end ex_13; / begin DBMS_OUTPUT.PUT_LINE(' '); DBMS_OUTPUT.PUT_LINE(' '); DBMS_OUTPUT.PUT_LINE('exercitiu 6'); ex_13.ex_6; DBMS_OUTPUT.PUT_LINE(' '); DBMS_OUTPUT.PUT_LINE(' '); DBMS_OUTPUT.PUT_LINE('exercitiu 7'); ex_13.cerinta_7; DBMS_OUTPUT.PUT_LINE(' '); DBMS_OUTPUT.PUT_LINE(' '); DBMS_OUTPUT.PUT_LINE('exercitiu 8'); DBMS_OUTPUT.PUT_LINE('too many rows'); DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Marian'));</pre>	<pre>exercitiu 7 Restaurantul : Mateo, aflat la adresa : Str. Valea cu Tei, Pitesti Sarmalute in foi de vita pret: 5.3 calorii 2000 greutate 200 Sare Otet Carne Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Clatite pret: 9.9 calorii 2000 greutate 100 Orez Vanilie ----- Restaurantul : Atelier, aflat la adresa : Str. Valea cu Tei, Pitesti Sarmalute in foi de vita pret: 5.3 calorii 2000 greutate 200 Sare Otet Carne ----- Restaurantul : Pizza Luca Pitesti, aflat la adresa : Bulevardul I.C. Bratianu, Pitesti Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Supa de legume pret: 3 calorii 1000 greutate 150 Otet Orez ----- Restaurantul : Papao, aflat la adresa : Bulevardul I.C. Bratianu, Pitesti Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Clatite pret: 9.9 calorii 2000 greutate 100 Orez Vanilie ----- Restaurantul : Pizza Luca Pitesti2, aflat la adresa : Str. Dimitrie Butculescu, Pitesti Cartofi copti pret: 7 calorii 2500 greutate 250 Sare Otet Carne Supa de legume pret: 3 calorii 1000 greutate 150 Otet Orez</pre>
<pre>14 when PREPARAT_NELIVRAT then DBMS_OUTPUT.PUT_LINE ('Ingredientul : ' ingr ' din tara: ' tara 15 end cerinta_9; 16 end ex_13; 17 / 18 begin 19 DBMS_OUTPUT.PUT_LINE(' '); 20 DBMS_OUTPUT.PUT_LINE(' '); 21 DBMS_OUTPUT.PUT_LINE('exercitiu 6'); 22 ex_13.ex_6; 23 24 DBMS_OUTPUT.PUT_LINE(' '); 25 DBMS_OUTPUT.PUT_LINE(' '); 26 DBMS_OUTPUT.PUT_LINE('exercitiu 7'); 27 ex_13.cerinta_7; 28 29 DBMS_OUTPUT.PUT_LINE(' '); 30 DBMS_OUTPUT.PUT_LINE(' '); 31 DBMS_OUTPUT.PUT_LINE('exercitiu 8'); 32 33 DBMS_OUTPUT.PUT_LINE('too many rows'); 34 DBMS_OUTPUT.PUT_LINE(ex_13.cerinta_8('Marian')); 35 DBMS_OUTPUT.PUT_LINE(' '); 36 DBMS_OUTPUT.PUT_LINE(' '); 37 DBMS_OUTPUT.PUT_LINE('no data found'); 38 end;</pre>	<pre>exercitiu 8 too many rows prea multi curieri cu prenumele :Marian no data found nu exista curieri cu prenumele :Test mai multe comenzi cu aceeaasi suma curierul cu prenumele :Test2 a livrat mai multe comenzi cu valoare maxima : 7 curier care nu a livrat comenzi curierul cu prenumele :Test3 nu a livrat comenzi date corecte comanda: 4, cu valoarea: \$1.1, reprezinta :32.28% , livrata la adresa : Str. Valea cu Tei, Pitesti exercitiu 9 eroarea de no_data_found Nu a fost gasit in baza de date ingredientul: muschi din tara:usa too_many_rows In baza de date exista de mai multe ori ingredientul : vanilie din tara:usa exceptia ingredient nefolosit Ingredientul : orez din tara:usa nu a fost folosit in niciun preparat exceptia PREPARAT_NELIVRAT Ingredientul : faina din tara:usa nu a fost inclus in niciun preparat livrat in ultimele 30 de zile date corecte Client:Dragomirescu Claudia cu nr de telefon :0772744621 cu adresa: Str. Valea cu Tei, Pitesti</pre>