

Proiect SGBD-Baza de date parfumerie online

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

În cadrul acestui proiect, voi prezenta modul de funcționare al unui site web de tipul e-commerce, unde se comercializează parfumuri.

Site-ul este accesat de clienți, care pot plasa comenzi de produse, care sunt cumparate de către magazin de la furnizori. Aceste comenzi au o anumită modalitate de plată, și sunt livrate de la depozitul de produse către client. Comenzile sunt procesate de către angajați, care lucrează în cadrul unui anumit departament.

Atributele asociate acestor entități:

-pentru clienți, se cunoaște numele, prenumele, numărul de telefon, email-ul, țara, localitatea și adresa;

-pentru livrări, se cunoaște id-ul de livrare, numărul livrării, data estimată de livrare, firma de curierat care va efectua transportul și statusul;

-pentru comenzi știu id-ul comenzii, id-ul clientului, data la care s-a plasat comanda, id-ul produsului, cantitatea, id-ul angajatului, id-ul plății și al livrării;

-pentru plăți, am asociat atributele: modalitatea plății, numărul cardului și numele detinatorului acestuia (în caz de plată prin card);

-pentru produse știu id-ul produsului, id-ul depozitului în care se află, numele parfumului, id-ul furnizorului, cantitatea care se află în stoc, descrierea parfumului, și dacă are sau nu discount;

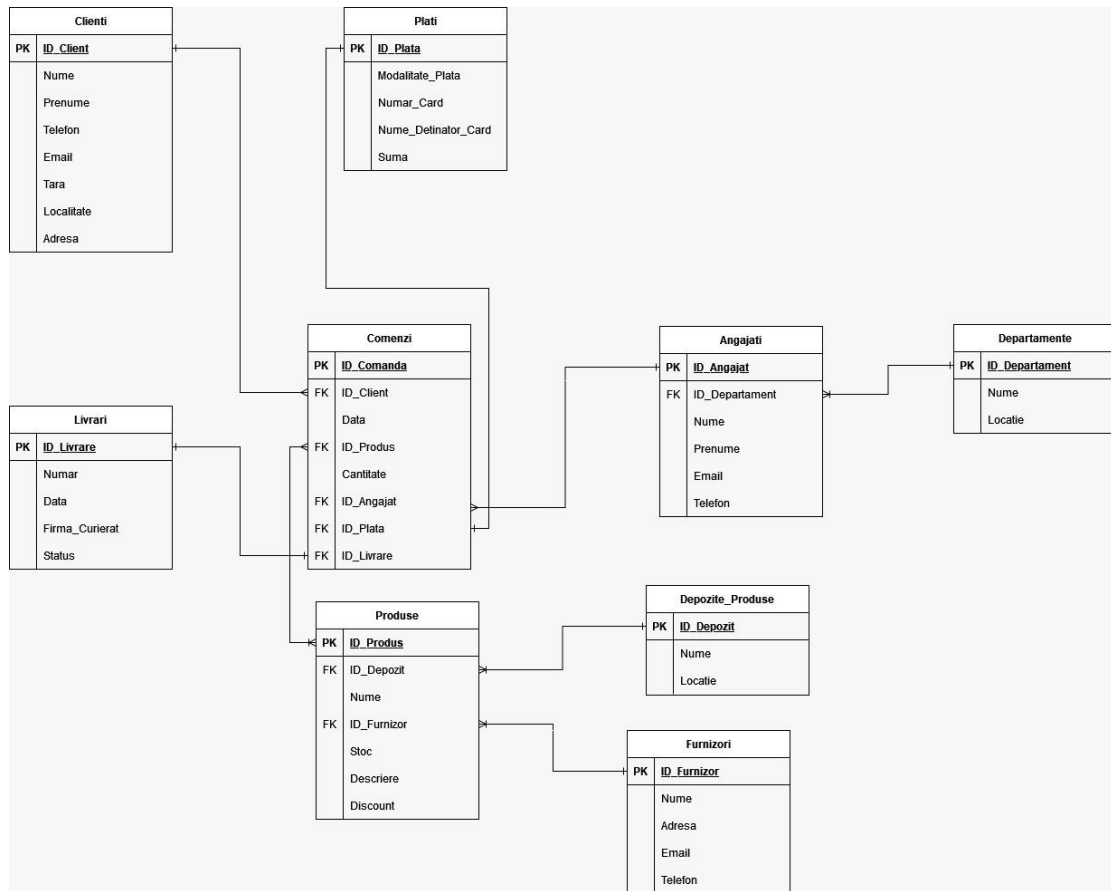
-pentru angajați cunoaștem id-ul angajatului, id-ul departamentului în care lucrează, numele și prenumele angajatului, email-ul și numărul de telefon al acestuia;

-pentru depozitele de produse cunoaștem id-ul acestora, numele și locația lor;

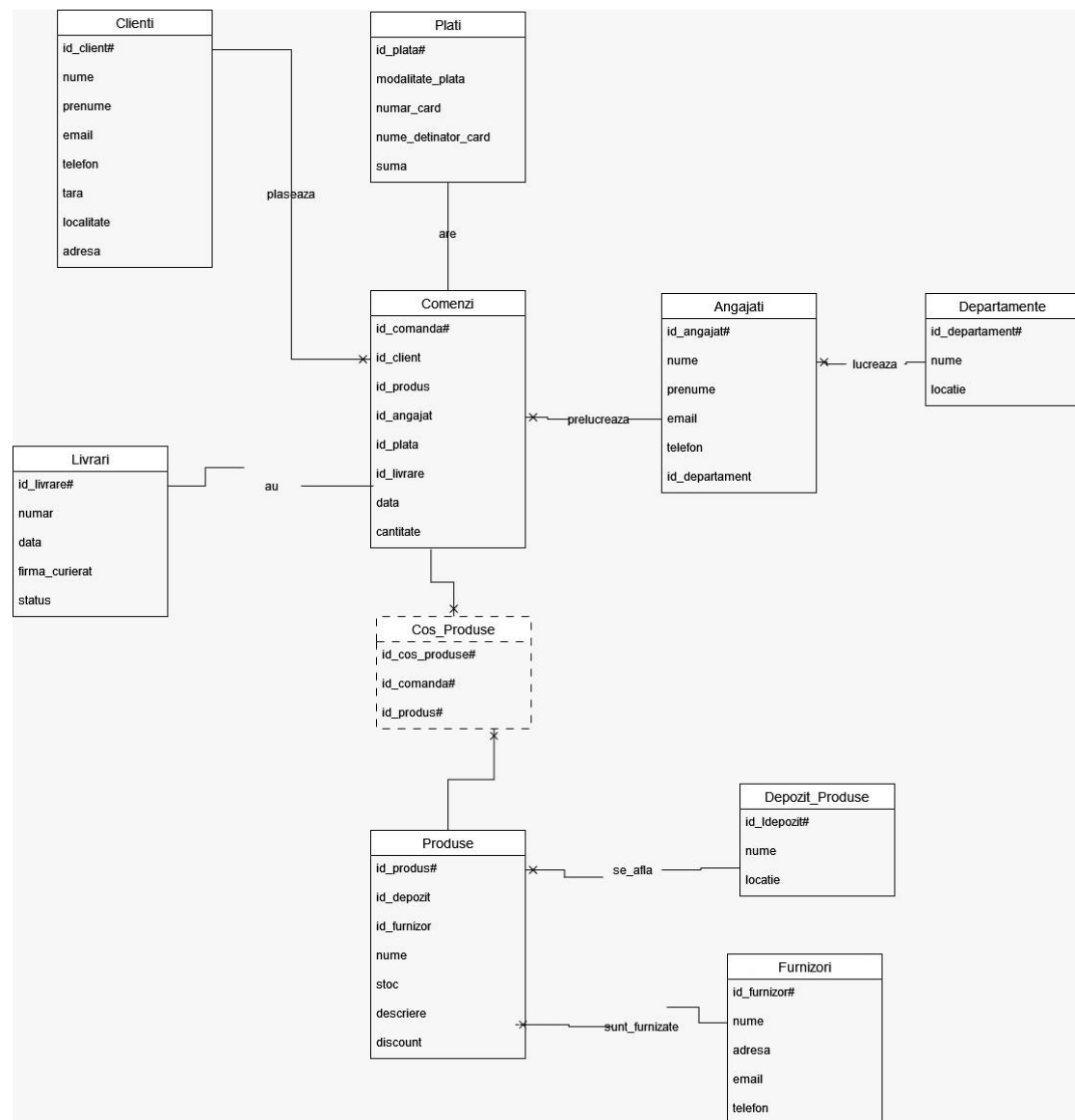
-pentru furnizor cunoaștem id-ul, numele, adresa, email-ul și nr de telefon al acestora;

-pentru departamente avem id-ul departamentului, numele și locația acestora.

2. Realizați diagrama entitate-relație(ERD).



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



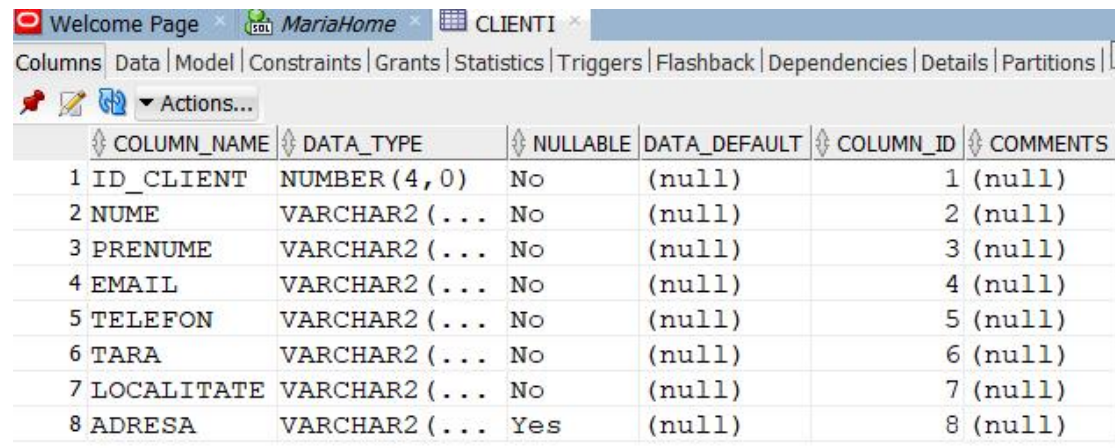
4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

--creare tabele

CREATE TABLE CLIENTI

```
(
    id_client NUMBER(4) NOT NULL,
    nume VARCHAR2(30) NOT NULL,
    prenume VARCHAR2(30) NOT NULL,
    email VARCHAR2(40) NOT NULL,
    telefon VARCHAR2(10) NOT NULL,
    tara VARCHAR2(25) NOT NULL,
    localitate VARCHAR2(30) NOT NULL,
    adresa VARCHAR2(100),

    CONSTRAINT pk_client PRIMARY KEY(id_client),
    UNIQUE(telefon)
);
```



The screenshot shows the MariaDB GUI with the 'CLIENTI' table selected. The 'Columns' tab is active, displaying the following table structure:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_CLIENT	NUMBER (4, 0)	No	(null)	1	(null)
2	NUME	VARCHAR2 (...	No	(null)	2	(null)
3	PRENUME	VARCHAR2 (...	No	(null)	3	(null)
4	EMAIL	VARCHAR2 (...	No	(null)	4	(null)
5	TELEFON	VARCHAR2 (...	No	(null)	5	(null)
6	TARA	VARCHAR2 (...	No	(null)	6	(null)
7	LOCALITATE	VARCHAR2 (...	No	(null)	7	(null)
8	ADRESA	VARCHAR2 (...	Yes	(null)	8	(null)

CREATE TABLE PLATI

```
(
    id_plata NUMBER(4) NOT NULL,
    modalitate_plata VARCHAR2(10) CHECK (modalitate_plata='ramburs' OR
modalitate_plata='card'),
    numar_card VARCHAR2(20),
    nume_detinator_card VARCHAR2(40),
    suma NUMBER(6) NOT NULL,
    CONSTRAINT pk_plata PRIMARY KEY(id_plata)
);
```

Welcome Page x MariaHome x PLATI x						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_PLATA	NUMBER (4, 0)	No	(null)	1	(null)
2	MODALIT...	VARCHAR2 (...	Yes	(null)	2	(null)
3	NUMAR_CARD	VARCHAR2 (...	Yes	(null)	3	(null)
4	NUME_DE...	VARCHAR2 (...	Yes	(null)	4	(null)
5	SUMA	NUMBER (6, 0)	No	(null)	5	(null)

```
CREATE TABLE LIVRARI
(
    id_livrare NUMBER(4) NOT NULL,
    numar VARCHAR2(10) NOT NULL,
    data DATE NOT NULL,
    firma_curierat VARCHAR2(40),
    status VARCHAR(30) CHECK(status='in depozit' OR status='in curs de
livrare' OR status='livrata'),
    CONSTRAINT pk_livrare PRIMARY KEY(id_livrare)
);
```

Welcome Page x MariaHome x LIVRARI x						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_LIVRARE	NUMBER (4, 0)	No	(null)	1	(null)
2	NUMAR	VARCHAR2 (...	No	(null)	2	(null)
3	DATA	DATE	No	(null)	3	(null)
4	FIRMA_C...	VARCHAR2 (...	Yes	(null)	4	(null)
5	STATUS	VARCHAR2 (...	Yes	(null)	5	(null)

```
CREATE TABLE DEPOZITE_PRODUSE
(
    id_depozit NUMBER(4) NOT NULL,
    nume VARCHAR2(30) NOT NULL,
    locatie VARCHAR2(30),
    CONSTRAINT pk_depozit PRIMARY KEY(id_depozit)
);
```

Welcome Page x MariaHome x DEPOZITE_PRODUSE x						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_DEPOZIT	NUMBER (4, 0)	No	(null)	1	(null)
2	NUME	VARCHAR2 (...	No	(null)	2	(null)
3	LOCATIE	VARCHAR2 (...	Yes	(null)	3	(null)

```
CREATE TABLE FURNIZORI
(
    id_furnizor NUMBER(4) NOT NULL,
    nume VARCHAR2(30) NOT NULL,
    adresa VARCHAR2(70) NOT NULL,
    email VARCHAR2(40) NOT NULL,
```

```

        telefon VARCHAR2(10) NOT NULL,

        CONSTRAINT pk_furnizor PRIMARY KEY(id_furnizor),
        UNIQUE(telefon)
    );

```

The screenshot shows the MariaHome database interface with the 'FURNIZORI' table selected. The 'Columns' tab is active, displaying a table with 6 columns: COLUMN_NAME, DATA_TYPE, NULLABLE, DATA_DEFAULT, COLUMN_ID, and COMMENTS. The table has 5 rows of data.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_FURN...	NUMBER (4, 0)	No	(null)	1	(null)
2	NUME	VARCHAR2 (...	No	(null)	2	(null)
3	ADRESA	VARCHAR2 (...	No	(null)	3	(null)
4	EMAIL	VARCHAR2 (...	No	(null)	4	(null)
5	TELEFON	VARCHAR2 (...	No	(null)	5	(null)

```

CREATE TABLE DEPARTAMENTE
(
    id_departament NUMBER(4) NOT NULL,
    nume VARCHAR2(30) NOT NULL,
    locatie VARCHAR2(70) NOT NULL,
    CONSTRAINT pk_departament PRIMARY KEY(id_departament)
);

```

The screenshot shows the MariaHome database interface with the 'DEPARTAMENTE' table selected. The 'Columns' tab is active, displaying a table with 6 columns: COLUMN_NAME, DATA_TYPE, NULLABLE, DATA_DEFAULT, COLUMN_ID, and COMMENTS. The table has 3 rows of data.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_DEPA...	NUMBER (4, 0)	No	(null)	1	(null)
2	NUME	VARCHAR2 (...	No	(null)	2	(null)
3	LOCATIE	VARCHAR2 (...	No	(null)	3	(null)

```

CREATE TABLE ANGAJATI
(
    id_angajat NUMBER(4) NOT NULL,
    id_departament NUMBER(4) NOT NULL,
    nume VARCHAR2(30) NOT NULL,
    prenume VARCHAR2(30) NOT NULL,
    salariu NUMBER(6) NOT NULL,
    email VARCHAR2(40) NOT NULL,
    telefon VARCHAR2(10) NOT NULL,

    CONSTRAINT pk_angajat PRIMARY KEY(id_angajat),
    CONSTRAINT fk_angajat FOREIGN KEY(id_departament) REFERENCES
DEPARTAMENTE(id_departament),
    UNIQUE(telefon)
);

```


Welcome Page * MariaHome * ANGAJATI *						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ANGAJAT	NUMBER (4, 0)	No	(null)	1	(null)
2	ID_DEPA...	NUMBER (4, 0)	No	(null)	2	(null)
3	NUME	VARCHAR2 (...	No	(null)	3	(null)
4	PRENUME	VARCHAR2 (...	No	(null)	4	(null)
5	SALARIU	NUMBER (6, 0)	No	(null)	5	(null)
6	EMAIL	VARCHAR2 (...	No	(null)	6	(null)
7	TELEFON	VARCHAR2 (...	No	(null)	7	(null)

```

CREATE TABLE PRODUSE
(
    id_produș NUMBER(4) NOT NULL,
    id_depozit NUMBER(4) NOT NULL,
    id_furnizor NUMBER(4) NOT NULL,
    nume VARCHAR2(30) NOT NULL,
    descriere VARCHAR2(200),
    stoc NUMBER(4),
    pret NUMBER(4),
    discount NUMBER(4,2) NOT NULL,

    CONSTRAINT pk_produș PRIMARY KEY(id_produș),
    CONSTRAINT fk1_angajat FOREIGN KEY(id_depozit) REFERENCES
DEPOZITE_PRODUSE(id_depozit),
    CONSTRAINT fk2_angajat FOREIGN KEY(id_furnizor) REFERENCES
FURNIZORI(id_furnizor)
);

```

Welcome Page * MariaHome * PRODUSE *						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions						
Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_PRODUS	NUMBER (4, 0)	No	(null)	1	(null)
2	ID_DEPOZIT	NUMBER (4, 0)	No	(null)	2	(null)
3	ID_FURN...	NUMBER (4, 0)	No	(null)	3	(null)
4	NUME	VARCHAR2 (...	No	(null)	4	(null)
5	DESCRIERE	VARCHAR2 (...	Yes	(null)	5	(null)
6	STOC	NUMBER (4, 0)	Yes	(null)	6	(null)
7	PRET	NUMBER (4, 0)	Yes	(null)	7	(null)
8	DISCOUNT	NUMBER (4, 2)	No	(null)	8	(null)

```

CREATE TABLE COMENZI
(
    id_comanda NUMBER(4) NOT NULL,
    id_client NUMBER(4) NOT NULL,

    id_angajat NUMBER(4) NOT NULL,
    id_plata NUMBER(4) NOT NULL,
    id_livrare NUMBER(4) NOT NULL,
    data DATE NOT NULL,

```

```

cantitate NUMBER(4),

CONSTRAINT fk1_comanda FOREIGN KEY(id_client) REFERENCES
CLIENTI(id_client),

CONSTRAINT fk2_comanda FOREIGN KEY(id_angajat) REFERENCES
ANGAJATI(id_angajat),
CONSTRAINT fk3_comanda FOREIGN KEY(id_plata) REFERENCES
PLATI(id_plata),
CONSTRAINT fk4_comanda FOREIGN KEY(id_livrare) REFERENCES
LIVRARI(id_livrare),
CONSTRAINT pk_comanda PRIMARY KEY(id_comanda)
);

```

The screenshot shows the MariaHome database interface with the 'COMENZI' table selected. The table structure is as follows:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_COMANDA	NUMBER (4, 0)	No	(null)	1	(null)
2	ID_CLIENT	NUMBER (4, 0)	No	(null)	2	(null)
3	ID_ANGAJAT	NUMBER (4, 0)	No	(null)	3	(null)
4	ID_PLATA	NUMBER (4, 0)	No	(null)	4	(null)
5	ID_LIVRARE	NUMBER (4, 0)	No	(null)	5	(null)
6	DATA	DATE	No	(null)	6	(null)
7	CANTITATE	NUMBER (4, 0)	Yes	(null)	7	(null)

```

CREATE TABLE COSURI_PRODUSE
(
    id_cos_produce NUMBER(4) NOT NULL,
    id_comanda NUMBER(4) NOT NULL,
    id_produs NUMBER(4) NOT NULL,

    CONSTRAINT fk1_cos_produce FOREIGN KEY(id_comanda)
REFERENCES COMENZI(id_comanda),
    CONSTRAINT fk2_cos_produce FOREIGN KEY(id_produs) REFERENCES
PRODUSE(id_produs),
    CONSTRAINT pk_cos_produce PRIMARY KEY(id_comanda, id_produs,
id_cos_produce)
);

```

The screenshot shows the MariaHome database interface with the 'COSURI_PRODUSE' table selected. The table structure is as follows:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_COS_...	NUMBER (4, 0)	No	(null)	1	(null)
2	ID_COMANDA	NUMBER (4, 0)	No	(null)	2	(null)
3	ID_PRODUS	NUMBER (4, 0)	No	(null)	3	(null)

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

--inserari

INSERT INTO CLIENTI

VALUES(1, 'Popescu', 'Dragos', 'pop@mail.com', '0752725387', 'Romania', 'Craiova', 'Str. N. Titulescu nr. 3, 303');

INSERT INTO CLIENTI

VALUES(2, 'Comanescu', 'Bogdan', 'bog@mail.com', '0752725380', 'Romania', 'Vaslui', 'Str. Zambilelor nr. 7');

INSERT INTO CLIENTI

VALUES(3, 'Dragu', 'Viorel', 'vio@mail.com', '0752725381', 'Belgia', 'Bruxelles', 'Str. Willem nr. 4');

INSERT INTO CLIENTI

VALUES(4, 'Maxilu', 'Veronica', 'vera@mail.com', '0752725382', 'Spania', 'Zaragoza', 'Str. Zanahoria nr. 2, 20');

INSERT INTO CLIENTI

VALUES(5, 'Adam', 'Anda', 'andadam@mail.com', '0752725383', 'Ungaria', 'Budapesta', 'Str. Kovacs nr. 4');

INSERT INTO CLIENTI

VALUES(6, 'Rosu', 'Daiana', 'rdai@mail.com', '0876592919', 'Canada', 'Montreal', 'Str. Baker nr 5');

INSERT INTO CLIENTI

VALUES(7, 'Popescu', 'Mircea', 'popmir@mail.com', '0876592920', 'Romania', 'Constanta', 'Str. Drumului nr 34');

INSERT INTO CLIENTI

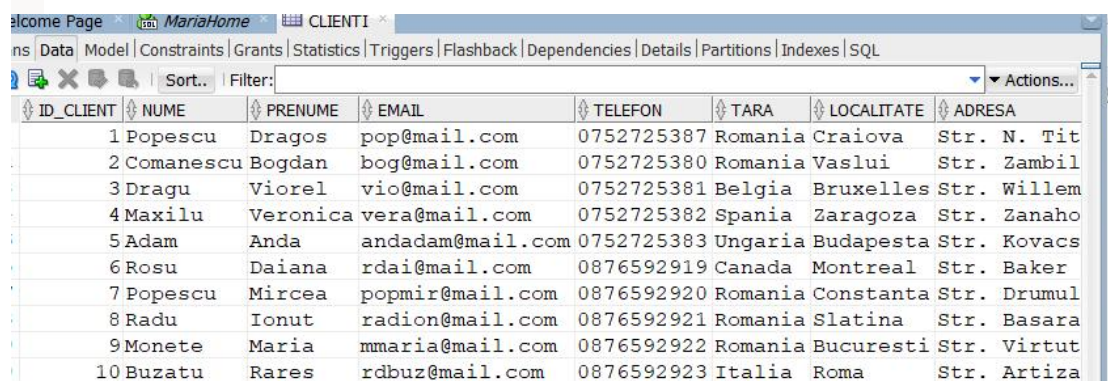
VALUES(8, 'Radu', 'Ionut', 'radion@mail.com', '0876592921', 'Romania', 'Slatina', 'Str. Basarab nr 19');

INSERT INTO CLIENTI

VALUES(9, 'Monete', 'Maria', 'mmaria@mail.com', '0876592922', 'Romania', 'Bucuresti', 'Str. Virtutii nr 40');

INSERT INTO CLIENTI

VALUES(10, 'Buzatu', 'Rares', 'rdbuz@mail.com', '0876592923', 'Italia', 'Roma', 'Str. Artiza nr 8');



The screenshot shows a database management interface with a table named 'CLIENTI'. The table has 8 columns: ID_CLIENT, NUME, PRENUME, EMAIL, TELEFON, TARA, LOCALITATE, and ADRESA. There are 10 rows of data, corresponding to the SQL insert statements provided above. The interface includes a toolbar with icons for various database actions and a tabbed window titled 'CLIENTI'.

ID_CLIENT	NUME	PRENUME	EMAIL	TELEFON	TARA	LOCALITATE	ADRESA
1	Popescu	Dragos	pop@mail.com	0752725387	Romania	Craiova	Str. N. Tit
2	Comanescu	Bogdan	bog@mail.com	0752725380	Romania	Vaslui	Str. Zambil
3	Dragu	Viorel	vio@mail.com	0752725381	Belgia	Bruxelles	Str. Willem
4	Maxilu	Veronica	vera@mail.com	0752725382	Spania	Zaragoza	Str. Zanho
5	Adam	Anda	andadam@mail.com	0752725383	Ungaria	Budapesta	Str. Kovacs
6	Rosu	Daiana	rdai@mail.com	0876592919	Canada	Montreal	Str. Baker
7	Popescu	Mircea	popmir@mail.com	0876592920	Romania	Constanta	Str. Drumul
8	Radu	Ionut	radion@mail.com	0876592921	Romania	Slatina	Str. Basara
9	Monete	Maria	mmaria@mail.com	0876592922	Romania	Bucuresti	Str. Virtut
10	Buzatu	Rares	rdbuz@mail.com	0876592923	Italia	Roma	Str. Artiza

INSERT INTO PLATI

```

VALUES(1,'ramburs',NULL,NULL,300);
INSERT INTO PLATI
VALUES(2,'card','4140 4980','Matei Basarab',1500);
INSERT INTO PLATI
VALUES(3,'card','4140 4000 3','Rares Buzatu',50);
INSERT INTO PLATI
VALUES(4,'ramburs',NULL,NULL,278);
INSERT INTO PLATI
VALUES(5,'card','4140 4220 2222','Maria Monete',2370);
INSERT INTO PLATI
VALUES(6,'ramburs',NULL,NULL,300);
INSERT INTO PLATI
VALUES(7,'card','4140 5678','Popescu Mircea',1230);
INSERT INTO PLATI
VALUES(8,'card','4140 3345','Radu Ionut',500);
INSERT INTO PLATI
VALUES(9,'ramburs',NULL,NULL,278);
INSERT INTO PLATI
VALUES(10,'card','4140 4859','Buzatu Rares',2300);
INSERT INTO PLATI
VALUES(11,'ramburs',NULL,NULL,450);

```

Welcome Page MariaHome PLATI					
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions					
Sort.. Filter:					
	ID_PLATA	MODALITATE_PLATA	NUMAR_CARD	NUME_DETINATOR_CARD	SUMA
1	1	ramburs	(null)	(null)	300
2	2	card	4140 4980	Matei Basarab	1500
3	3	card	4140 4000 3	Rares Buzatu	50
4	4	ramburs	(null)	(null)	278
5	5	card	4140 4220 2222	Maria Monete	2370
6	6	ramburs	(null)	(null)	300
7	7	card	4140 5678	Popescu Mircea	1230
8	8	card	4140 3345	Radu Ionut	500
9	9	ramburs	(null)	(null)	278
10	10	card	4140 4859	Buzatu Rares	2300
11	11	ramburs	(null)	(null)	450
12	12	ramburs	(null)	(null)	2000

```

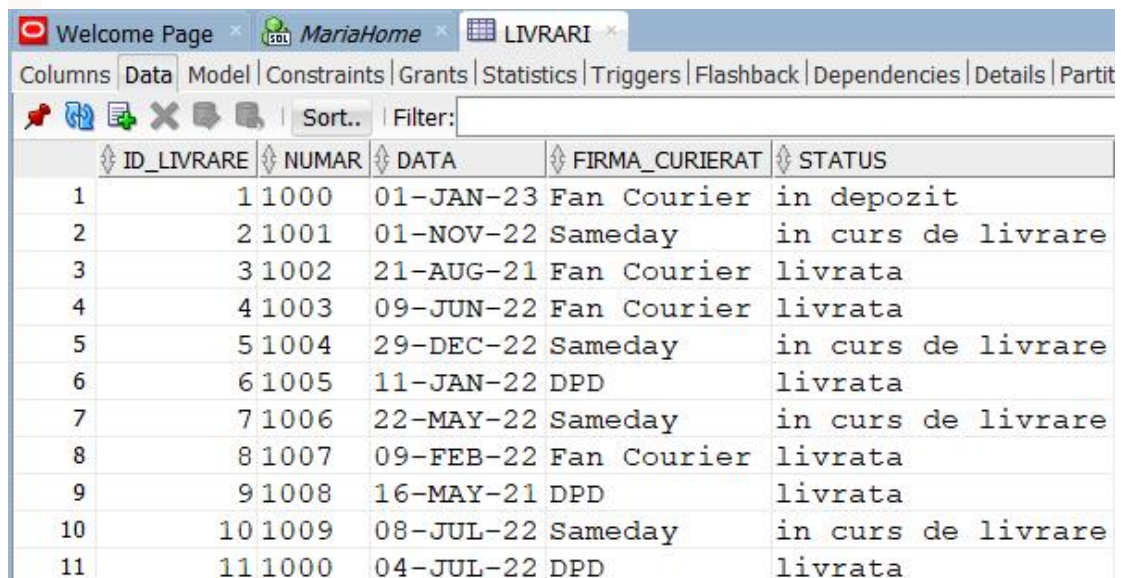
INSERT INTO LIVRARI
VALUES(1,'1000',to_date('01-01-2023','dd-mm-yyyy'),'Fan Courier','in depozit');
INSERT INTO LIVRARI
VALUES(2,'1001',to_date('01-11-2022','dd-mm-yyyy'),'Sameday','in curs de
livrare');
INSERT INTO LIVRARI
VALUES(3,'1002',to_date('21-08-2021','dd-mm-yyyy'),'Fan Courier','livrata');
INSERT INTO LIVRARI
VALUES(4,'1003',to_date('09-06-2022','dd-mm-yyyy'),'Fan Courier','livrata');
INSERT INTO LIVRARI

```

```

VALUES(5,'1004',to_date('29-12-2022','dd-mm-yyyy'),'Sameday','in curs de
livrare');
INSERT INTO LIVRARI
VALUES(6,'1005',to_date('11-01-2022','dd-mm-yyyy'),'DPD','livrata');
INSERT INTO LIVRARI
VALUES(7,'1006',to_date('22-05-2022','dd-mm-yyyy'),'Sameday','in curs de
livrare');
INSERT INTO LIVRARI
VALUES(8,'1007',to_date('09-02-2022','dd-mm-yyyy'),'Fan Courier','livrata');
INSERT INTO LIVRARI
VALUES(9,'1008',to_date('16-05-2021','dd-mm-yyyy'),'DPD','livrata');
INSERT INTO LIVRARI
VALUES(10,'1009',to_date('08-07-2022','dd-mm-yyyy'),'Sameday','in curs de
livrare');
INSERT INTO LIVRARI
VALUES(11,1000,to_date('04-07-2022','dd-mm-yyyy'),'DPD','livrata');

```



The screenshot shows a database management tool interface with the 'LIVRARI' table selected. The table has five columns: ID_LIVRARE, NUMAR, DATA, FIRMA_CURIERAT, and STATUS. The data is displayed in a table with 11 rows, numbered 1 to 11. The first row (ID 1) shows '1 1000' for ID_LIVRARE and NUMAR, '01-JAN-23' for DATA, 'Fan Courier' for FIRMA_CURIERAT, and 'in depozit' for STATUS. The subsequent rows (IDs 2-11) show various dates and courier companies, with statuses either 'in curs de livrare' or 'livrata'.

ID_LIVRARE	NUMAR	DATA	FIRMA_CURIERAT	STATUS
1	1 1000	01-JAN-23	Fan Courier	in depozit
2	2 1001	01-NOV-22	Sameday	in curs de livrare
3	3 1002	21-AUG-21	Fan Courier	livrata
4	4 1003	09-JUN-22	Fan Courier	livrata
5	5 1004	29-DEC-22	Sameday	in curs de livrare
6	6 1005	11-JAN-22	DPD	livrata
7	7 1006	22-MAY-22	Sameday	in curs de livrare
8	8 1007	09-FEB-22	Fan Courier	livrata
9	9 1008	16-MAY-21	DPD	livrata
10	10 1009	08-JUL-22	Sameday	in curs de livrare
11	11 1000	04-JUL-22	DPD	livrata

```

INSERT INTO DEPOZITE_PRODUSE
VALUES(1,'Depozit Bucuresti','Bucuresti');
INSERT INTO DEPOZITE_PRODUSE
VALUES(2,'Depozit Pitesti','Pitesti');
INSERT INTO DEPOZITE_PRODUSE
VALUES(3,'Depozit Timisoara','Timisoara');
INSERT INTO DEPOZITE_PRODUSE
VALUES(4,'Depozit Cluj','Cluj');
INSERT INTO DEPOZITE_PRODUSE
VALUES(5,'Depozit Iasi','Iasi');

```

DEPOZITE_PRODUSE			
ID_DEPOZIT	NUME	LOCATIE	
1	1 Depozit Bucuresti	Bucuresti	
2	2 Depozit Pitesti	Pitesti	
3	3 Depozit Timisoara	Timisoara	
4	4 Depozit Cluj	Cluj	
5	5 Depozit Iasi	Iasi	

INSERT INTO FURNIZORI

VALUES(1,'Arabian Scent','Bucuresti Sector 6 Str Uverturii nr 21','arabscent@mail.com','0786543244');

INSERT INTO FURNIZORI

VALUES(2,'French essence','Timisoara Bd Mihail Kogalniceanu nr 158','frenchessence@mail.com','0786543245');

INSERT INTO FURNIZORI

VALUES(3,'Parfumuri Romania','Bucuresti Sector 3 Str Dreptatii nr 100','ropar@mail.com','0786543246');

INSERT INTO FURNIZORI

VALUES(4,'Esente tari','Craiova Str Libertatii nr 39','esta@mail.com','0786543247');

INSERT INTO FURNIZORI

VALUES(5,'Parfumas','Sibiu Str Pacii nr 40','parfumas3@mail.com','0786543248');

FURNIZORI			
ID_FURNIZOR	NUME	ADRESA	EMAIL
1	1 Arabian Scent	Bucuresti Sector 6 Str Uverturii nr 21	arabscent@mail
2	2 French essence	Timisoara Bd Mihail Kogalniceanu nr 158	frenchessence@
3	3 Parfumuri Romania	Bucuresti Sector 3 Str Dreptatii nr 100	ropar@mail.com
4	4 Esente tari	Craiova Str Libertatii nr 39	esta@mail.com
5	5 Parfumas	Sibiu Str Pacii nr 40	parfumas3@mail

INSERT INTO DEPARTAMENTE

VALUES(1,'Serviciu clienti','Bucuresti');

INSERT INTO DEPARTAMENTE

VALUES(2,'Call-center','Bucuresti');

INSERT INTO DEPARTAMENTE

VALUES(3,'Logistica','Bucuresti');

INSERT INTO DEPARTAMENTE

VALUES(4,'Sesizari','Pitesti');

INSERT INTO DEPARTAMENTE

VALUES(5,'Livrari','Ilfov');

Welcome Page			
MariaHome			
DEPARTAMENTE			
Columns	Data	Model	Constraints
Grants	Statistics	Triggers	Flashback
Sort.. Filter:			
ID_DEPARTAMENT	NUME	LOCATIE	
1	1 Serviciu clienti	Bucuresti	
2	2 Call-center	Bucuresti	
3	3 Logistica	Bucuresti	
4	4 Sesizari	Pitesti	
5	5 Livrari	Ilfov	

INSERT INTO ANGAJATI

VALUES(1,1,'Pop','Alexandru',1200,'popalex@gmail.com','0777777777');

INSERT INTO ANGAJATI

VALUES(2,1,'Lulea','Ana',2300,'lola23@gmail.com','0777777760');

INSERT INTO ANGAJATI

VALUES(3,1,'Monea','Rares',4670,'morares@gmail.com','0777777761');

INSERT INTO ANGAJATI

VALUES(4,2,'Ilea','Bianca',5000,'bibixoxo@gmail.com','0777777762');

INSERT INTO ANGAJATI

VALUES(5,2,'Luzi','Ioana',9000,'luzilz@gmail.com','0777777763');

INSERT INTO ANGAJATI

VALUES(6,2,'Popa','Sandu',4000,'popasan@gmail.com','0777777764');

INSERT INTO ANGAJATI

VALUES(7,3,'Aluza','Enola',3500,'enola@gmail.com','0777777765');

INSERT INTO ANGAJATI

VALUES(8,3,'Mutacu','Darius',1500,'mudarius@gmail.com','0777777766');

INSERT INTO ANGAJATI

VALUES(9,3,'Radoi','Alexandru',6000,'radoialex@gmail.com','0777777767');

INSERT INTO ANGAJATI

VALUES(10,4,'Selini','Paula',4500,'paula@gmail.com','0777777768');

INSERT INTO ANGAJATI

VALUES(11,4,'Filip','Vlad',8000,'filvald@gmail.com','0777777769');

INSERT INTO ANGAJATI

VALUES(12,4,'Ieftiimie','Raluca',3900,'ralucya@gmail.com','0777777888');

INSERT INTO ANGAJATI

VALUES(13,5,'Destiper','Mircea',7000,'destiperola@gmail.com','0777779999');

INSERT INTO ANGAJATI

VALUES(14,5,'Puscasu','Mihai',2500,'mihpusc@gmail.com','0777777000');

INSERT INTO ANGAJATI

VALUES(15,5,'Anton','Andreea',5700,'andreeant@gmail.com','0777777222');

Welcome Page MariaHome ANGAJATI						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL						
Sort.. Filter: Actions..						
ID_ANGAJAT	ID_DEPARTAMENT	NUME	PRENUME	SALARIU	EMAIL	TELEFON
1	1	1 Pop	Alexandru	1200	popalex@gmail.com	0777777777
2	2	1 Lulea	Ana	2300	lola23@gmail.com	0777777760
3	3	1 Monea	Rares	4670	morares@gmail.com	0777777761
4	4	2 Ilea	Bianca	5000	bibixoxo@gmail.com	0777777762
5	5	2 Luzi	Ioana	9000	luzilz@gmail.com	0777777763
6	6	2 Popa	Sandu	4000	popasan@gmail.com	0777777764
7	7	3 Aluza	Enola	3500	enola@gmail.com	0777777765
8	8	3 Mutacu	Darius	1500	mudarius@gmail.com	0777777766
9	9	3 Radoi	Alexandru	6000	radoialex@gmail.com	0777777767
10	10	4 Selini	Paula	4500	paula@gmail.com	0777777768
11	11	4 Filip	Vlad	8000	filvald@gmail.com	0777777769
12	12	4 Ieftiimie	Raluca	3900	ralucya@gmail.com	0777777888
13	13	5 Destiper	Mircea	7000	destiperola@gmail.com	0777779999
14	14	5 Puscasu	Mihai	2500	mihpusc@gmail.com	0777777000
15	15	5 Anton	Andreea	5700	andreeant@gmail.com	0777777222
16	16	2 Anton	Gabriela	8000	gabant@email.com	0778965411

INSERT INTO PRODUSE

VALUES(1, 1, 1,'Versace-Bright-Crystal', 'Note varf rodie, acorduri de gheata, yuzu;Note inima lotus, magnolie, bujor;
Note baza cihlimbar, lemn, mosc', 56,3000, 0.10);

INSERT INTO PRODUSE

VALUES(2, 2, 1,'Versace-Crystal-Noir', 'Note varf coacaze negre, Fig negru, violeta;Note inima tuberoza, frangipani, iasomie;
Note baza lemn de santal, kasmeran, vanilie, heliotrop', 25,2000, 0);

INSERT INTO PRODUSE

VALUES(3, 1, 2,'Versace-Yellow-Diamond', 'Note varf extras din portocala amara, para, bergamota, lamaie;Note inima iasomie, osmanthusl
Note baza cihlimbar, benzoin', 38,1750, 0.05);

INSERT INTO PRODUSE

VALUES(4, 2, 3,'Dior-Jadore', 'Compozitie extras din portocala amara, Jasmine (arab), magnolie;Grupe de parfumuri florale, acvatice', 16, 230,0.05);

INSERT INTO PRODUSE

VALUES(5, 3, 3,'Dior-Miss-Dior', 'Note varf lacramioare, bujor, iris;Note inima trandafir;Note baza mosc, vanilie, benzoin, tonka, lemn de santal;
Grupe de parfumuri florale', 12,1230, 0);

INSERT INTO PRODUSE

VALUES(6, 2, 3,'Dior-Poison-Girl', 'Note varf portocala amara;Note inima trandafiri din Grasse;
Note baza tonka;Grupe de parfumuri orientale', 4, 1100,0.10);

INSERT INTO PRODUSE

VALUES(7, 3, 4,'CH-Good-Girl', 'Note varf migdale;Note inima iasomie Sambac, tuberoza;Note baza tonka, cacao;
Grupe de parfumuri orientale, florale', 123, 500, 0);

INSERT INTO PRODUSE

VALUES(8, 4, 4,'CH-212-VIP', 'Note varf bergamota, piper;Note inima gardenie, geranium;Note baza vata de zahar, mosc, lemn de santal, vanilie;
Grupe de parfumuri orientale, florale', 56, 2099,0);

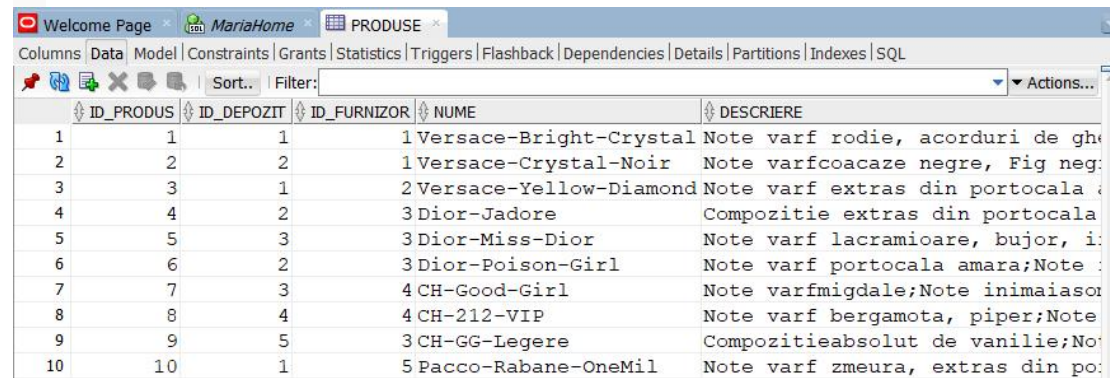
INSERT INTO PRODUSE

VALUES(9, 5, 3,'CH-GG-Legere', 'Compozitie absolut de vanilie;Note varf cires, migdale amare;Note inima trandafir, crin;Note baza vetiver, vanilie Bourbon, absolut de vanilie;

```

        Grupe de parfumuri    'florale, lemnoase', 47, 3400, 0.05);
INSERT INTO PRODUSE
VALUES(10, 1, 5, 'Pacco-Rabane-OneMil', 'Note varf zmeura, extras din portocala
amara;Note inima    gardenie, Jasmine (arab),
        floare de portocal african;Note baza    patchouli, miere alba, cihlimbar;', 87,
1299,0.20);

```



ID_PRODUS	ID_DEPOZIT	ID_FURNIZOR	NUME	DESCRIERE
1	1	1	1Versace-Bright-Crystal	Note varf rodie, acorduri de gheata
2	2	2	1Versace-Crystal-Noir	Note varfcoacaze negre, Fig negr
3	3	1	2Versace-Yellow-Diamond	Note varf extras din portocala
4	4	2	3Dior-Jadore	Compozitie extras din portocala
5	5	3	3Dior-Miss-Dior	Note varf lacramioare, bujor, i
6	6	2	3Dior-Poison-Girl	Note varf portocala amara;Note
7	7	3	4CH-Good-Girl	Note varfmigdale;Note inimaia
8	8	4	4CH-212-VIP	Note varf bergamota, piper;Note
9	9	5	3CH-GG-Legere	Compozitieabsolut de vanilie;No
10	10	1	5Pacco-Rabane-OneMil	Note varf zmeura, extras din po

```

INSERT INTO COMENZI
VALUES(1,1,5,1,1,to_date('12-12-2022', 'dd-mm-yyyy'),2);
INSERT INTO COMENZI
VALUES(2,2,5,2,2,to_date('29-07-2022', 'dd-mm-yyyy'),5);
INSERT INTO COMENZI
VALUES(3,3,5,3,3,to_date('04-08-2022', 'dd-mm-yyyy'),1);
INSERT INTO COMENZI
VALUES(4,4,4,4,4,to_date('23-09-2022', 'dd-mm-yyyy'),4);
INSERT INTO COMENZI
VALUES(5,5,4,5,5,to_date('14-11-2022', 'dd-mm-yyyy'),1);
INSERT INTO COMENZI
VALUES(6,6,3,6,6,to_date('12-12-2022', 'dd-mm-yyyy'),2);
INSERT INTO COMENZI
VALUES(7,7,5,7,7,to_date('29-07-2022', 'dd-mm-yyyy'),5);
INSERT INTO COMENZI
VALUES(8,8,1,8,8,to_date('04-08-2022', 'dd-mm-yyyy'),1);
INSERT INTO COMENZI
VALUES(9,9,5,9,9,to_date('23-09-2022', 'dd-mm-yyyy'),4);
INSERT INTO COMENZI
VALUES(10,10,4,10,10,to_date('14-11-2022', 'dd-mm-yyyy'),1);
INSERT INTO COMENZI
VALUES(11,2,3,11,11,to_date('02-07-2022', 'dd-mm-yyyy'),2);

```

Welcome PageMariaHomeCOMENZI

ColumnsDataModelConstraintsGrantsStatisticsTriggersFlashbackDependenciesDetailsPartitionsIndex

```

INSERT INTO COSURI_PRODUSE
VALUES(1,1,2);
INSERT INTO COSURI_PRODUSE
VALUES(2,2,3);
INSERT INTO COSURI_PRODUSE
VALUES(3,3,5);
INSERT INTO COSURI_PRODUSE
VALUES(4,4,4);
INSERT INTO COSURI_PRODUSE
VALUES(5,5,1);
INSERT INTO COSURI_PRODUSE
VALUES(6,6,8);
INSERT INTO COSURI_PRODUSE
VALUES(7,7,6);
INSERT INTO COSURI_PRODUSE
VALUES(8,8,7);
INSERT INTO COSURI_PRODUSE
VALUES(9,9,10);
INSERT INTO COSURI_PRODUSE
VALUES(10,10,9);

```

Welcome Page x MariaHome x COSURI_PRODUSE x				
Columns Data Model Constraints Grants Statistics Triggers Flashb				
Sort.. Filter:				
	ID_COS_PRODUSE	ID_COMANDA	ID_PRODUS	
1	1	1	2	
2	2	2	3	
3	3	3	5	
4	4	4	4	
5	5	5	1	
6	6	6	8	
7	7	7	6	
8	8	8	7	
9	9	9	10	
10	10	10	9	

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.

--6 Afisati toti angajatii(si salariile lor) care

--lucreaza intr-un departament care se afla intr-o locatie introdusa de la tastatura

```
CREATE OR REPLACE PROCEDURE proc_ex6(v_oras in
departamente.locatie%TYPE) IS
```

--tablou indexat

```
TYPE tab_ind IS TABLE OF angajati.nume%TYPE INDEX BY PLS_INTEGER;
```

--var care vor stoca numele si prenumele

```
v_nume tab_ind;
```

```
v_prenume tab_ind;
```

--vector

```
TYPE vector IS VARRAY(30) OF angajati.salariu%TYPE;
```

--var care va stoca salariul angajatului

```
v_sal vector;
```

BEGIN

--introducem in vector salariile angajatilor din acea locatie

```
SELECT salariu
```

```
BULK COLLECT INTO v_sal
```

```
FROM angajati
```

```
JOIN departamente on departamente.id_departament=angajati.id_departament
```

```
WHERE UPPER(departamente.locatie)=upper(v_oras)
```

```
ORDER BY salariu DESC;
```

--gasim numele si prenumele angajatilor din dep respectiv

```
FOR i IN v_sal.first..v_sal.last LOOP
```

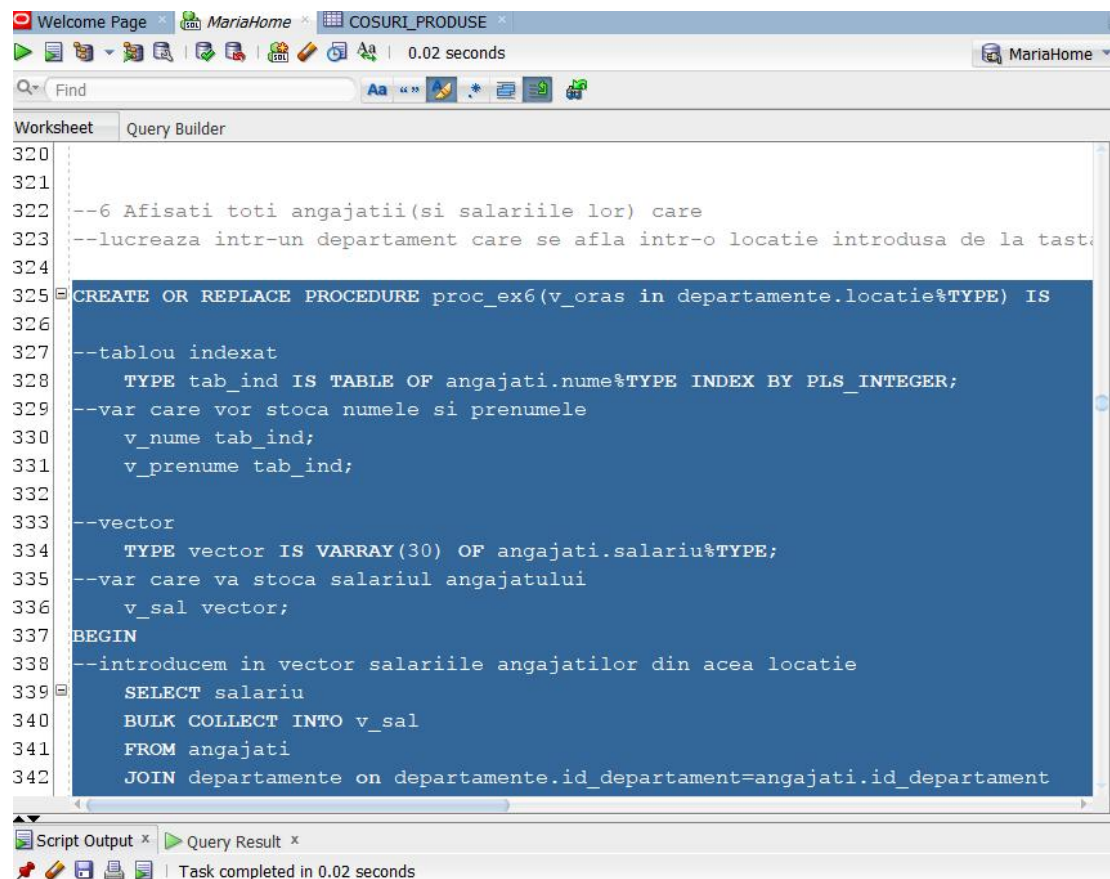
```

        SELECT angajati.numa
        INTO v_nume(i)
        FROM ANGAJATI
        JOIN DEPARTAMENTE ON
departamente.id_departament=angajati.id_departament
        WHERE upper(departamente.locatie)=upper(v_oras)
        AND angajati.salariu=v_sal(i);
    END LOOP;

    --gasim numele si prenumele angajatilor din dep respectiv
    FOR i IN v_sal.first..v_sal.last LOOP
        SELECT angajati.prenume
        INTO v_prenume(i)
        FROM ANGAJATI
        JOIN DEPARTAMENTE ON
departamente.id_departament=angajati.id_departament
        WHERE upper(departamente.locatie)=upper(v_oras)
        AND angajati.salariu=v_sal(i);
    END LOOP;

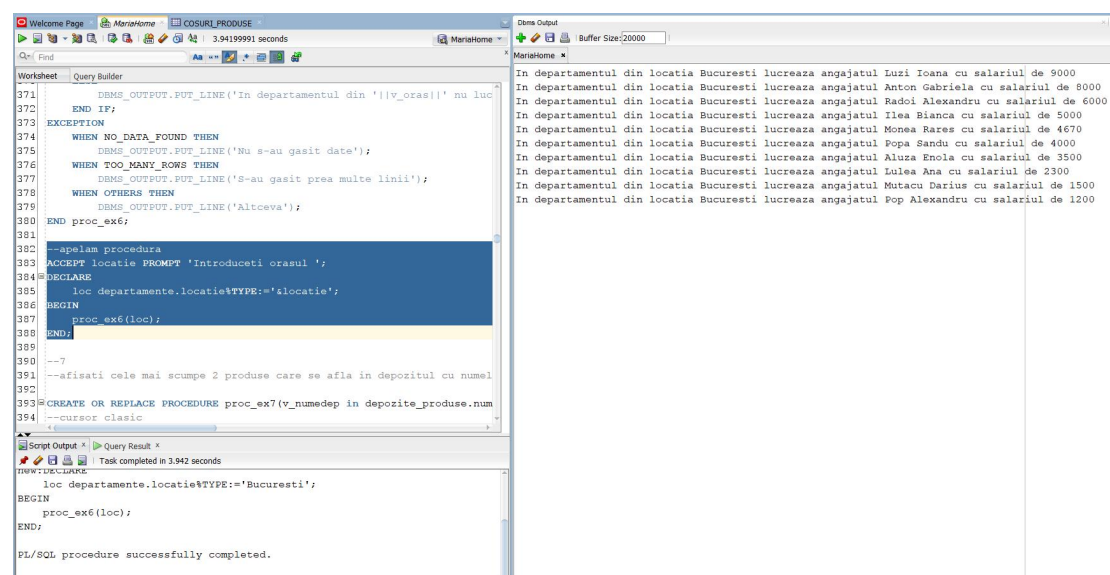
    --parcurgem tabelelele si vectorul
    IF v_nume.count>0 THEN
        FOR i in v_sal.first..v_sal.last LOOP
            DBMS_OUTPUT.PUT_LINE('In departamentul din locatia '||v_oras ||
lucreaza angajatul '||v_nume(i)||' '||v_prenume(i)||' cu salariul de '||v_sal(i));
        END LOOP;
    ELSE
        DBMS_OUTPUT.PUT_LINE('In departamentul din '||v_oras||' nu lucreaza
niciun angajat');
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('S-au gasit prea multe linii');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Altceva');
END proc_ex6;

--apelam procedura
ACCEPT locatie PROMPT 'Introduceti orasul ';
DECLARE
    loc departamente.locatie%TYPE:='&locatie';
BEGIN
    proc_ex6(loc);
END;
```

```
320
321
322 --6 Afisati toti angajatii (si salariile lor) care
323 --lucreaza intr-un departament care se afla intr-o locatie introdusa de la tastatura
324
325 CREATE OR REPLACE PROCEDURE proc_ex6(v_oras in departamente.locatie%TYPE) IS
326
327 --tablou indexat
328     TYPE tab_ind IS TABLE OF angajati.nume%TYPE INDEX BY PLS_INTEGER;
329 --var care vor stoca numele si prenumele
330     v_nume tab_ind;
331     v_prename tab_ind;
332
333 --vector
334     TYPE vector IS VARRAY(30) OF angajati.salariu%TYPE;
335 --var care va stoca salariul angajatului
336     v_sal vector;
337 BEGIN
338 --introducem in vector salariile angajatilor din acea locatie
339     SELECT salariu
340     BULK COLLECT INTO v_sal
341     FROM angajati
342     JOIN departamente on departamente.id_departament=angajati.id_departament
```

Procedure PROC_EX6 compiled



```
371 DBMS_OUTPUT.PUT_LINE('In departamentul din '||v_oras||' nu luc
372 END IF;
373 EXCEPTION
374 WHEN NO_DATA_FOUND THEN
375     DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
376 WHEN TOO_MANY_ROWS THEN
377     DBMS_OUTPUT.PUT_LINE('S-au gasit prea multe linii');
378 WHEN OTHERS THEN
379     DBMS_OUTPUT.PUT_LINE('Altceva');
380 END proc_ex6;
381
382 --apelam procedura
383 ACCEPT locatie PROMPT 'Introduceti orasul ';
384 DECLARE
385     loc departamente.locatie%TYPE:='silocatie';
386 BEGIN
387     proc_ex6(loc);
388 END;
389
390 --7
391 --afisati cele mai scumpe 2 produse care se afla in depozitul cu numel
392
393 CREATE OR REPLACE PROCEDURE proc_ex7(v_numedep in depozite_produce.num
394 --cursor clasic
```

In departamentul din locatie Bucuresti lucreaza angajatul Luzi Ioana cu salariul de 9000
In departamentul din locatie Bucuresti lucreaza angajatul Anton Gabriela cu salariul de 8000
In departamentul din locatie Bucuresti lucreaza angajatul Radoi Alexandru cu salariul de 6000
In departamentul din locatie Bucuresti lucreaza angajatul Ilea Bianca cu salariul de 5000
In departamentul din locatie Bucuresti lucreaza angajatul Monea Rares cu salariul de 4670
In departamentul din locatie Bucuresti lucreaza angajatul Popa Sandu cu salariul de 4000
In departamentul din locatie Bucuresti lucreaza angajatul Aluza Enola cu salariul de 3500
In departamentul din locatie Bucuresti lucreaza angajatul Lulea Ana cu salariul de 2300
In departamentul din locatie Bucuresti lucreaza angajatul Mutacu Darius cu salariul de 1500
In departamentul din locatie Bucuresti lucreaza angajatul Pop Alexandru cu salariul de 1200

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.

--7

--afisati cele mai scumpe 2 produse care se afla in depozitul cu numele introdus de la tastatura

```
CREATE OR REPLACE PROCEDURE proc_ex7(v_numedep in
depozite_produce.nume%TYPE) IS
```

```
--cursor clasic
```

```
    CURSOR c1 IS
        SELECT id_depozit
        FROM depozite_produce
        WHERE INITCAP(nume)=INITCAP(v_numedep);
```

```
--cursor parametrizat
```

```
    CURSOR c2(v_id depozite_produce.id_depozit%type) IS
        SELECT nume, pret
        FROM produse
        WHERE id_depozit=v_id
        ORDER BY pret desc;
```

```
    c_pret produse.pret%TYPE;
    c_denumire produse.nume%TYPE;
    c_id depozite_produce.id_depozit%TYPE;
```

```
BEGIN
```

```
    OPEN c1;
    LOOP
        FETCH c1 INTO c_id;
        EXIT WHEN c1%NOTFOUND;
    END LOOP;
    --cautam id-ul depozitului cu numele dat
```

```
    OPEN c2(c_id);
    LOOP
        FETCH c2 INTO c_denumire, c_pret;
        EXIT WHEN c2%NOTFOUND OR c2%ROWCOUNT>2;
        DBMS_OUTPUT.PUT_LINE('In depozitul '||v_numedep||' se afla parfumul
'||c_denumire||' care costa '||c_pret);
    END LOOP;
    --gasim primele 2 produse in functie de pret
```

```
    IF c2%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista produse in acest depozit');
    END IF;
    CLOSE c2;
    DBMS_OUTPUT.NEW_LINE;
    CLOSE c1;
```

```
EXCEPTION
```

```

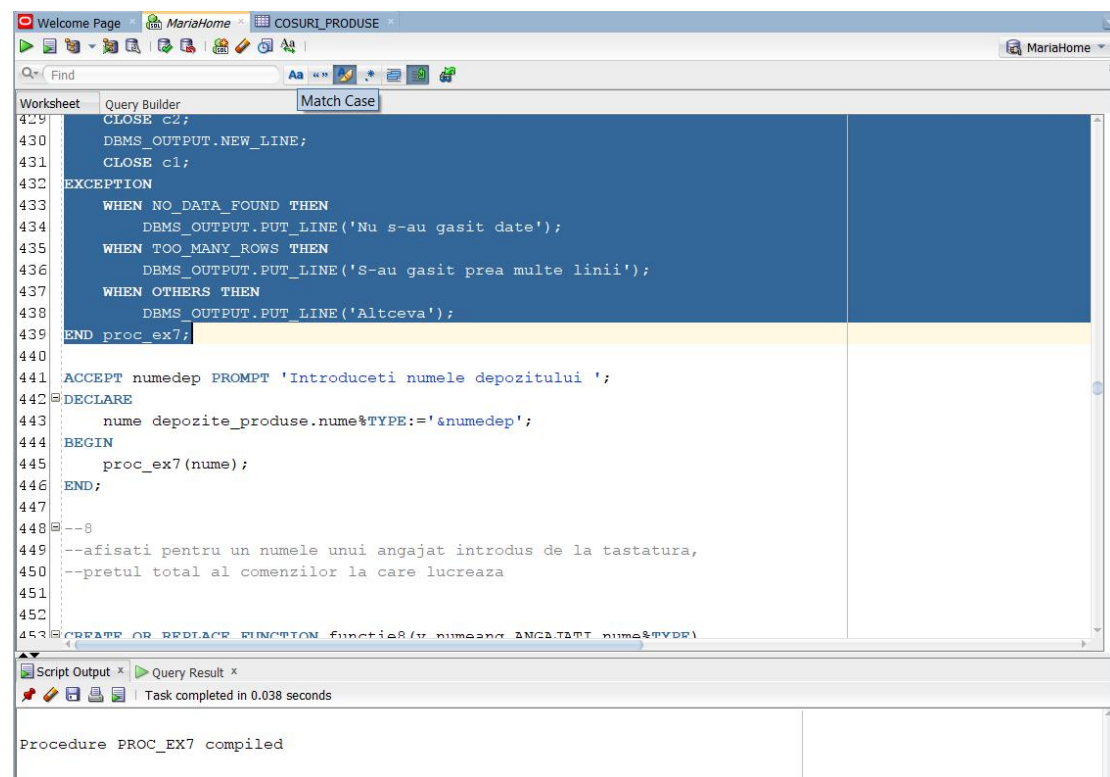
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('S-au gasit prea multe linii');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Altceva');
END proc_ex7;

```

```

ACCEPT numedep PROMPT 'Introduceti numele depozitului ';
DECLARE
    nume depozite_produce.nume%TYPE:='&numedep';
BEGIN
    proc_ex7(nume);
END;

```



```

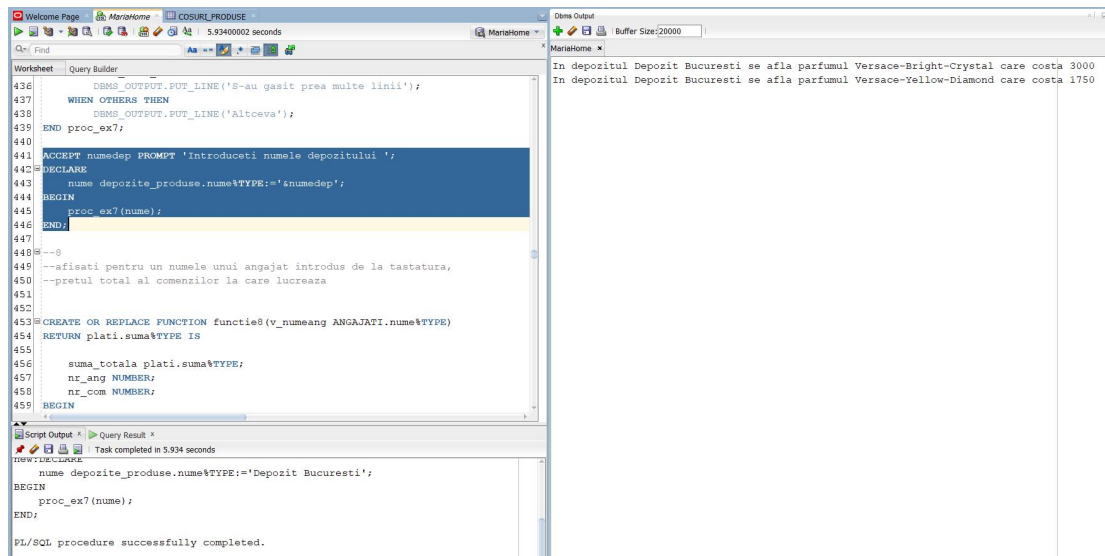
429 CLOSE c2;
430 DBMS_OUTPUT.NEW_LINE;
431 CLOSE c1;
432 EXCEPTION
433     WHEN NO_DATA_FOUND THEN
434         DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
435     WHEN TOO_MANY_ROWS THEN
436         DBMS_OUTPUT.PUT_LINE('S-au gasit prea multe linii');
437     WHEN OTHERS THEN
438         DBMS_OUTPUT.PUT_LINE('Altceva');
439 END proc_ex7;
440
441 ACCEPT numedep PROMPT 'Introduceti numele depozitului ';
442 DECLARE
443     nume depozite_produce.nume%TYPE:='&numedep';
444 BEGIN
445     proc_ex7(nume);
446 END;
447
448 --8
449 --afisati pentru un numele unui angajat introdus de la tastatura,
450 --pretul total al comenzilor la care lucreaza
451
452
453 CREATE OR REPLACE FUNCTION functie8(v_numeang IN VARCHAR2, nume%TYPE)

```

Script Output x Query Result x

Task completed in 0.038 seconds

Procedure PROC_EX7 compiled



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 SQL3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--afisati pentru un numele unui angajat introdus de la tastatura,
 --pretul total al comenzilor la care lucreaza

```
CREATE OR REPLACE FUNCTION functie8(v_numeang
ANGAJATI.nume%TYPE)
RETURN plati.suma%TYPE IS
```

```
    suma_totala plati.suma%TYPE;
    nr_ang NUMBER;
    nr_com NUMBER;
BEGIN
    --vedem cati angajati exista( cu numele de familie dat)
    SELECT COUNT(*)
    INTO nr_ang
    FROM angajati
    WHERE UPPER(nume)=UPPER(v_numeang);
```

```
    --aruncam exceptii in caz ca sunt 0/mai multi ang cu numele dat
    IF nr_ang=0
        THEN RAISE_APPLICATION_ERROR(-20000,'Nu exista niciun angajat cu
    acest nume de familie');
    ELSIF nr_ang>1
        THEN RAISE_APPLICATION_ERROR(-20001,'Exista mai multi angajati cu
    acest nume de familie');
    END IF;
```

```

--vedem la cate comenzi lucreaza angajatul
SELECT COUNT(*)
  INTO nr_com
  FROM comenzi
  JOIN angajati ON angajati.id_angajat=comenzi.id_angajat
  WHERE UPPER(angajati.ume)=UPPER(v_umeang);

--aruncam exceptie in caz ca angajatul nu lucreaza la nicio comanda
IF nr_com=0
  THEN RAISE_APPLICATION_ERROR(-20002,'Nu exista nicio comanda la
care se lucreaza');
END IF;

--calculm pretul tuturor comenzilor la care lucreaza angajatul
SELECT SUM(suma)
  INTO suma_totala
  FROM plati
  JOIN comenzi ON plati.id_plata=comenzi.id_plata
  JOIN angajati ON comenzi.id_angajat=angajati.id_angajat
  WHERE UPPER(angajati.ume)=UPPER(v_umeang);
--am facut join pe 3 tabele pt a obtine rezultatul

  RETURN suma_totala;

END functie8;

--apelam functia pt un angajat care nu exista
SELECT functie8('Popescu') FROM DUAL;

--apelam functia pentru un nume care se gaseste la 2 angajati
SELECT functie8('Anton') FROM DUAL;

--apelam functia pentru un angajat care nu lucreaza la nicio comanda
SELECT functie8('Popa') FROM DUAL;

--apelam functia pt un angajat care exista si unul singur, si lucreaza la cel putin o
comanda
SELECT functie8('Pop') FROM DUAL;

```


Welcome Page MariaHome COSURI_PRODUSE 0.037 seconds MariaHome

Find

Worksheet Query Builder

```
486 SELECT SUM(suma)
487 INTO suma_totala
488 FROM plati
489 JOIN comenzi ON plati.id_plata=comenzi.id_plata
490 JOIN angajati ON comenzi.id_angajat=angajati.id_angajat
491 WHERE UPPER(angajati.numa)=UPPER(v_numeang);
492 --am facut join pe 3 tabele pt a obtine rezultatul
493
494 RETURN suma_totala;
495
496 END functie8;
497
498 --apelam functia pt un angajat care nu exista
499 SELECT functie8('Popescu') FROM DUAL;
500
501 --apelam functia pentru un nume care se gaseste la 2 angajati
502 SELECT functie8('Anton') FROM DUAL;
503
504 --apelam functia pentru un angajat care nu lucreaza la nicio comanda
505 SELECT functie8('Lulea') FROM DUAL;
506
507 --apelam functia pt un angajat care exista si are unul singur, si lucreaza la ce
508 SELECT functie8('Pop') FROM DUAL;
509
510
```

Script Output x Query Result x

Task completed in 0.037 seconds

Function FUNCTIE8 compiled

```
496 END functie8;
497
498 --apelam functia pt un angajat care nu exista
499 SELECT functie8('Popescu') FROM DUAL;
500
501 --apelam functia pentru un nume care se gaseste la 2 angajati
502 SELECT functie8('Anton') FROM DUAL;
503
504 --apelam functia pentru un angajat care nu lucreaza la nicio comanda
505 SELECT functie8('Lulea') FROM DUAL;
506
507 --apelam functia pt un angajat care exista si are unul singur, si lucreaza la ce
508 SELECT functie8('Pop') FROM DUAL;
509
510
```

Script Output x Query Result x

SQL | Executing:SELECT functie8('Popescu') FROM DUAL in 0 seconds

ORA-20000: Nu exista niciun angajat cu acest nume de familie
ORA-06512: at "MARIAHOME.FUNCTIE8", line 16
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

```
495
496 END functie8;
497
498 --apelam functia pt un angajat care nu exista
499 SELECT functie8('Popescu') FROM DUAL;
500
501 --apelam functia pentru un nume care se gaseste la 2 angajati
502 SELECT functie8('Anton') FROM DUAL;
503
504 --apelam functia pentru un angajat care nu lucreaza la nicio comanda
505 SELECT functie8('Lulea') FROM DUAL;
506
507 --apelam functia pt un angajat care exista si unul singur, si lucreaza la ce
508 SELECT functie8('Pop') FROM DUAL;
509
510
```

Script Output x Query Result x Query Result 1 x

SQL | Executing:SELECT functie8('Anton') FROM DUAL in 0 seconds

ORA-20001: Exista mai multi angajati cu acest nume de familie
ORA-06512: at "MARIAHOME.FUNCTIE8", line 18

```
502 SELECT functie8('Anton') FROM DUAL;
503
504 --apelam functia pentru un angajat care nu lucreaza la nicio comanda
505 SELECT functie8('Popa') FROM DUAL;
506
507 --apelam functia pt un angajat care exista si unul singur, si lucreaza la ce
508 SELECT functie8('Pop') FROM DUAL;
509
510
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL | Executing:SELECT functie8('Popa') FROM DUAL in 0 seconds

ORA-20002: Nu exista nicio comanda la care se lucreaza
ORA-06512: at "MARIAHOME.FUNCTIE8", line 30

```

505 SELECT functie8('Popa') FROM DUAL;
506
507 --apelam functia pt un angajat care exista sie unul singur, si lucreaza la ce
508 SELECT functie8('Pop') FROM DUAL;
509
510
511 --9
512 --livrari si depozit produse
513 --se da numarul unei livrari. afisati pt aceasta id-ul comenzii care ii
514 --corespunde, numele produsului din comanda si numele depozitului in care se a
515
516 CREATE OR REPLACE PROCEDURE proc_ex9 (nr_liv livrari.numar%TYPE) IS
517     CURSOR c9 IS
518         SELECT c.id_comanda, p.numa, dp.numa
519         FROM livrari l
520         JOIN comenzi c ON l.id_livrare=c.id_livrare
521         JOIN cosuri_produce cp ON cp.id_comanda=c.id_comanda
522         JOIN produse p ON p.id_produc=cp.id_produc
523         JOIN depozite_produce dp ON p.id_depozit=dp.id_depozit
524         WHERE l.numar=nr_liv;
525     CURSOR c9_1 IS

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL | All Rows Fetched: 1 in 0.005 seconds

FUNCTIE8('POP')	
1	500

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.

```

--9
--livrari si depozit produse
--se da numarul unei livrari. afisati pt aceasta id-ul comenzii care ii
--corespunde, numele produsului din comanda si numele depozitului in care se afla
acest produs

```

```

CREATE OR REPLACE PROCEDURE proc_ex9 (nr_liv livrari.numar%TYPE) IS
    CURSOR c9 IS
        SELECT c.id_comanda, p.numa, dp.numa
        FROM livrari l
        JOIN comenzi c ON l.id_livrare=c.id_livrare
        JOIN cosuri_produce cp ON cp.id_comanda=c.id_comanda
        JOIN produse p ON p.id_produc=cp.id_produc
        JOIN depozite_produce dp ON p.id_depozit=dp.id_depozit
        WHERE l.numar=nr_liv;
--cursor cu join intre 5 tabele pt obtinerea datelor cerute de enunt
    CURSOR c9_1 IS
        SELECT COUNT(*)
        FROM livrari
        WHERE numar=nr_liv;

```

```

--cursor care determina cate livrari exista cu numarul dat

id_com comenzi.id_comanda%TYPE;
nume_prod produse.nume%TYPE;
nume_dep depozite_produse.nume%TYPE;
nr NUMBER;
--var care vor stoca informatiile cerute in enunt si nr de livrari cu nr dat
BEGIN
  OPEN c9_1;
  LOOP
    FETCH c9_1 INTO nr;
    EXIT WHEN c9_1%NOTFOUND;
  END LOOP;
  --daca exista mai multe comenzi cu nr dat
  IF nr>1
    THEN RAISE_APPLICATION_ERROR(-20004, 'Exista mai multe comenzi cu
acest numar');
  END IF;
  CLOSE c9_1;

  OPEN c9;
  LOOP
    FETCH c9 INTO id_com, nume_prod, nume_dep;
    EXIT WHEN c9%NOTFOUND;
  END LOOP;
  --daca nu exista o livrare cu acel nr
  IF c9%ROWCOUNT = 0
    THEN RAISE_APPLICATION_ERROR(-20003, 'Nu exista nicio livrare cu
acest numar');
  END IF;
  --daca exista o singura livrare cu acel nr
  IF c9%ROWCOUNT = 1
    THEN DBMS_OUTPUT.PUT_LINE('Livrarii cu numarul '||nr_liv||' ii
corespunde comanda cu id-ul '||id_com||' care contine produsul '||
nume_prod||' aflat in depozitul cu numele '||nume_dep);
  END IF;

  CLOSE c9;

END proc_ex9;

EXECUTE proc_ex9(1002);

EXECUTE proc_ex9(1010);

EXECUTE proc_ex9(1000);

```

```

--ACCEPT nrliv PROMPT 'Introduceti numarul livrarii';
--DECLARE
--  numar_liv livrari.numar%TYPE:='&nrliv';
--BEGIN
--  proc_ex9(numar_liv);
--END;

```

The screenshot shows a database query editor window with a 'Worksheet' tab and a 'Query Builder' tab. The main area displays SQL code with line numbers 555 through 579. The code includes an IF statement, a procedure call, and a series of EXECUTE statements. Below the code, there is a status bar indicating 'Task completed in 0.035 seconds'. At the bottom, there are two messages: 'Function FUNCTIE8 compiled' and 'Procedure PROC_EX9 compiled'.

```

555
556 IF c9%ROWCOUNT =1
557     THEN DBMS_OUTPUT.PUT_LINE('Livrarii cu numarul '||nr_liv||' ii coresp
558     nume_prod||' aflat in depozitul cu numele '||nume_dep);
559 END IF;
560
561 CLOSE c9;
562
563
564
565 END proc_ex9;
566
567 EXECUTE proc_ex9(1002);
568
569 EXECUTE proc_ex9(1010);
570
571 EXECUTE proc_ex9(1000);
572
573 --ACCEPT nrliv PROMPT 'Introduceti numarul livrarii';
574 --DECLARE
575 --  numar_liv livrari.numar%TYPE:='&nrliv';
576 --BEGIN
577 --  proc_ex9(numar_liv);
578 --END;
579

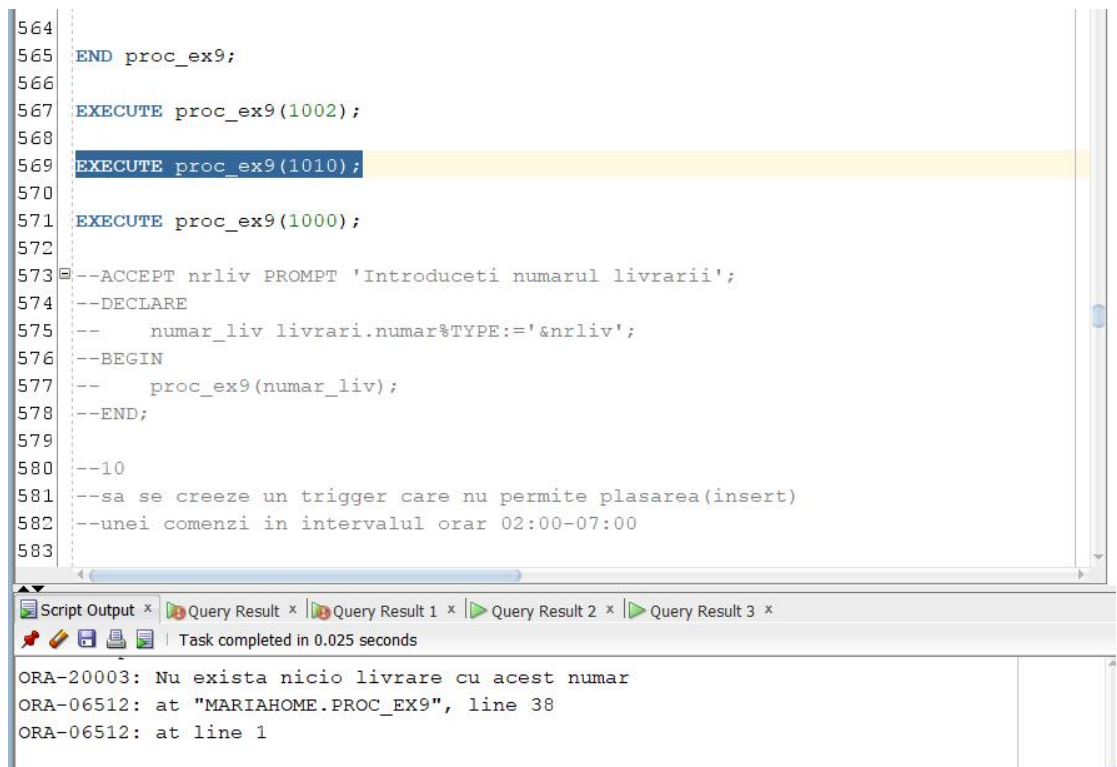
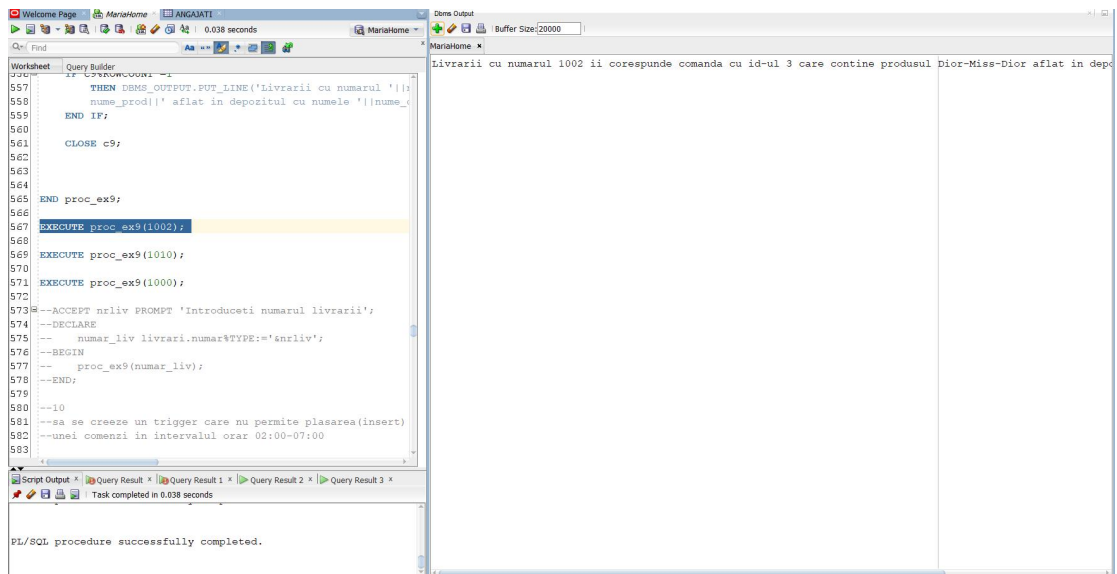
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.035 seconds

Function FUNCTIE8 compiled

Procedure PROC_EX9 compiled



```

569 EXECUTE proc_ex9(1010);
570
571 EXECUTE proc_ex9(1000);
572
573 --ACCEPT nrliv PROMPT 'Introduceti numarul livrarii';
574 --DECLARE
575 --    numar_liv livrari.numar%TYPE:='&nrliv';
576 --BEGIN
577 --    proc_ex9(numar_liv);
578 --END;
579
580 --10
581 --sa se creeze un trigger care nu permite plasarea(insert)
582 --unei comenzi in intervalul orar 02:00-07:00
583

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.037 seconds

ORA-20004: Exista mai multe comenzi cu acest numar
ORA-06512: at "MARIAHOME.PROC_EX9", line 26
ORA-06512: at line 1

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

--sa se creeze un trigger care nu permite plasarea(insert)
--unei comenzi in intervalul orar 02:00-07:00

```

CREATE OR REPLACE TRIGGER t_ex10
BEFORE INSERT OR DELETE OR UPDATE ON comenzi
BEGIN
--daca ora este intre 2 si 7, se declanseaza exceptia
IF(to_char(SYSDATE,'HH24')BETWEEN 2 AND 7)THEN
    IF INSERTING THEN
        RAISE_APPLICATION_ERROR(-20005,'Plasarea unei comenzi se
realizeaza in
        intervalul orar 07:00-23:59 si 00:00-02:00');
    END IF;
END IF;
END t_ex10;

INSERT INTO plati VALUES(12,'ramburs',NULL,NULL,2000);
INSERT INTO livrari VALUES(12,'1011',to_date('11-02-2023','dd-mm-
yyyy'),'Sameday','in depozit');
--declansare
INSERT INTO comenzi VALUES(12,9,2,12,12,to_date('11-02-2023','dd-mm-
yyyy'),1);

```

```

583
584 CREATE OR REPLACE TRIGGER t_ex10
585 BEFORE INSERT OR DELETE OR UPDATE ON comenzi
586 BEGIN
587 --daca ora este intre 2 si 7, se declanseaza exceptia
588 IF(to_char(SYSDATE,'HH24')BETWEEN 2 AND 7)THEN
589 IF INSERTING THEN
590 RAISE_APPLICATION_ERROR(-20005,'Plasarea unei comenzi se realizeaza in
591 intervalul orar 07:00-23:59 si 00:00-02:00');
592 END IF;
593 END IF;
594 END t_ex10;
595
596 INSERT INTO plati VALUES(12,'ramburs',NULL,NULL,2000);
597 INSERT INTO livrari VALUES(12,'1011',to_date('11-02-2023','dd-mm-yyyy'),'Sameday'
598 --declansare
599 INSERT INTO comenzi VALUES(12,9,2,12,12,to_date('11-02-2023','dd-mm-yyyy'),1);
600
601 --11
602 --creati un trigger care nu permite modificarea statusului
603 --unei livrare care are statusul 'livrata'
604 CREATE OR REPLACE TRIGGER t_ex11
605 BEFORE INSERT OR UPDATE OF status on livrari

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.029 seconds

Trigger T_EX10 compiled

--declansare la 3:32 am

```

598 --declansare
599 INSERT INTO comenzi VALUES(12,9,2,12,12,to_date('11-02-2023','dd-mm-yyyy'),1);
600
601 --11
602 --creati un trigger care nu permite modificarea statusului
603 --unei livrare care are statusul 'livrata'
604 CREATE OR REPLACE TRIGGER t_ex11
605 BEFORE INSERT OR UPDATE OF status on livrari

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.038 seconds

Error Report

ORA-20005: Plasarea unei comenzi se realizeaza in intervalul orar 07:00-23:59 si 00:00-02:00

ORA-06512: at "MARIAHOME.T_EX10", line 5

ORA-04088: error during execution of trigger 'MARIAHOME.T_EX10'

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

```

--creati un trigger care nu permite modificarea statusului
--unei livrare care are statusul 'livrata'
CREATE OR REPLACE TRIGGER t_ex11
BEFORE INSERT OR UPDATE OF status on livrari
FOR EACH ROW
BEGIN
--daca statusul nou e diferit de cel vechi, iar cel nou=status, declanseaza eroare
IF :OLD.status='livrata' AND :NEW.status<>:OLD.status
THEN RAISE_APPLICATION_ERROR(-20006,'Nu puteti modifica statusul
unei livrari deja efectuate');
END IF;
END t_ex11;
--declansare
UPDATE livrari

```

```
SET status='in curs de livrare'
WHERE id_livrare=3;
```

```

601 --11
602 --creati un trigger care nu permite modificarea statusului
603 --unei livrare care are statusul 'livrata'
604 CREATE OR REPLACE TRIGGER t_ex11
605     BEFORE INSERT OR UPDATE OF status on livrari
606     FOR EACH ROW
607 BEGIN
608     --daca statusul nou e diferit de cel vechi, iar cel nou=status, declanseaza eroare
609     IF :OLD.status='livrata' AND :NEW.status<>:OLD.status
610         THEN RAISE_APPLICATION_ERROR(-20006,'Nu puteti modifica statusul unei livrari deja e
611     END IF;
612 END t_ex11;
613 --declansare
614 UPDATE livrari
615 SET status='in curs de livrare'
616 WHERE id_livrare=3;
617
618 --12
619 --creati un trigger care nu permite
620 --stergerea unui tabel
621 CREATE OR REPLACE TRIGGER t_ex12

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.046 seconds

Trigger T_EX11 compiled

```

612 END t_ex11;
613 --declansare
614 UPDATE livrari
615 SET status='in curs de livrare'
616 WHERE id_livrare=3;
617
618 --12
619 --creati un trigger care nu permite
620 --stergerea unui tabel
621 CREATE OR REPLACE TRIGGER t_ex12

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.041 seconds

WHERE id_livrare=3

Error report -

ORA-20006: Nu puteti modifica statusul unei livrari deja efectuate

ORA-06512: at "MARIAHOME.T_EX11", line 4

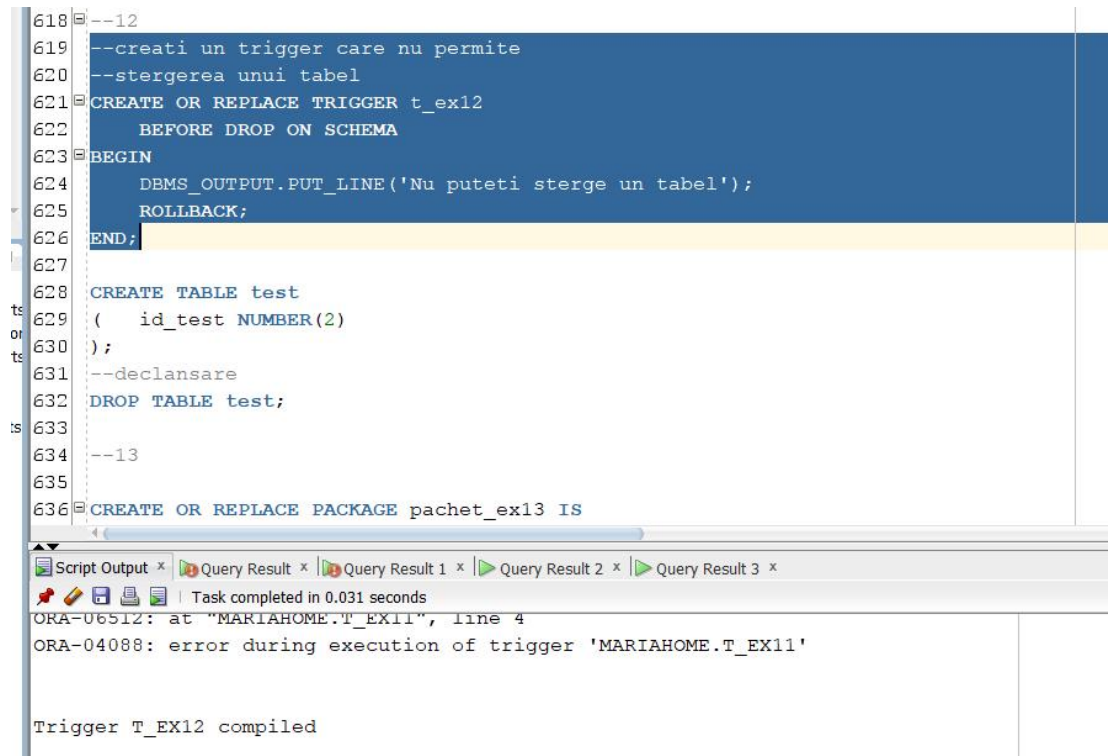
ORA-04088: error during execution of trigger 'MARIAHOME.T_EX11'

	ID_LIVRARE	NUMAR	DATA	FIRMA_CURIERAT	STATUS
1		11000	01-JAN-23	Fan Courier	in depozit
2		21001	01-NOV-22	Sameday	in curs de livrare
3		31002	21-AUG-21	Fan Courier	livrata

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

```
--creati un trigger care nu permite
--stergerea unui tabel
CREATE OR REPLACE TRIGGER t_ex12
  BEFORE DROP ON SCHEMA
BEGIN
  DBMS_OUTPUT.PUT_LINE('Nu puteti sterge un tabel');
  ROLLBACK;
END t_ex12;
```

```
CREATE TABLE test
( id_test NUMBER(2)
);
--declansare
DROP TABLE test;
```



```
618 --12
619 --creati un trigger care nu permite
620 --stergerea unui tabel
621 CREATE OR REPLACE TRIGGER t_ex12
622   BEFORE DROP ON SCHEMA
623 BEGIN
624   DBMS_OUTPUT.PUT_LINE('Nu puteti sterge un tabel');
625   ROLLBACK;
626 END;
627
628 CREATE TABLE test
629 ( id_test NUMBER(2)
630 );
631 --declansare
632 DROP TABLE test;
633
634 --13
635
636 CREATE OR REPLACE PACKAGE pachet_ex13 IS
```

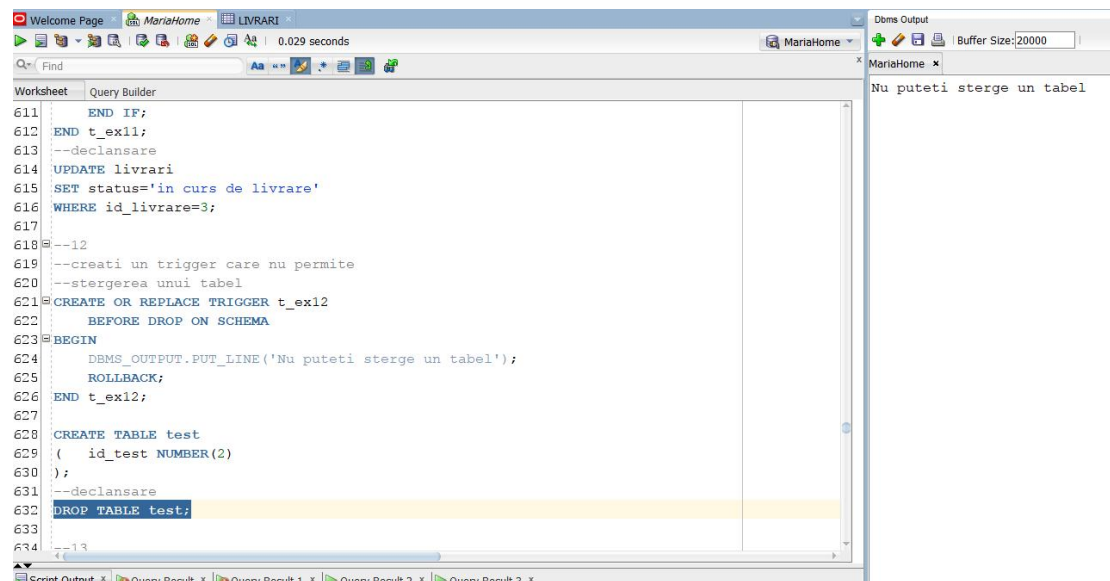
Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.031 seconds

ORA-06512: at "MARIAHOME.T_EX11", line 4

ORA-04088: error during execution of trigger 'MARIAHOME.T_EX11'

Trigger T_EX12 compiled



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

```

CREATE OR REPLACE PACKAGE pachet_ex13 IS
    FUNCTION functie8(v_numeang ANGAJATI.nume%TYPE) RETURN
    plati.suma%TYPE;
    PROCEDURE proc_ex6(v_oras in departamente.locatie%TYPE);
    PROCEDURE proc_ex7(v_numedep in depozite_produce.nume%TYPE);
    PROCEDURE proc_ex9(nr_liv livrari.numar%TYPE);
END pachet_ex13;

```

```

CREATE OR REPLACE PACKAGE BODY pachet_ex13 IS

```

```

--functie ex8
FUNCTION functie8(v_numeang ANGAJATI.nume%TYPE)
RETURN plati.suma%TYPE IS

```

```

    suma_totala plati.suma%TYPE;
    nr_ang NUMBER;
    nr_com NUMBER;
BEGIN

```

```

    SELECT COUNT(*)
    INTO nr_ang
    FROM angajati
    WHERE UPPER(nume)=UPPER(v_numeang);

```

```

    IF nr_ang=0
        THEN RAISE_APPLICATION_ERROR(-20000,'Nu exista niciun angajat cu
        acest nume de familie');
    ELSIF nr_ang>1
        THEN RAISE_APPLICATION_ERROR(-20001,'Exista mai multi angajati cu
        acest nume de familie');
    END IF;

```

```

SELECT COUNT(*)
INTO nr_com
FROM comenzi
JOIN angajati ON angajati.id_angajat=comenzi.id_angajat
WHERE UPPER(angajati.ume)=UPPER(v_numeang);

IF nr_com=0
    THEN RAISE_APPLICATION_ERROR(-20002,'Nu exista nicio comanda la
care se lucreaza');
END IF;

SELECT SUM(suma)
INTO suma_totala
FROM plati
JOIN comenzi ON plati.id_plata=comenzi.id_plata
JOIN angajati ON comenzi.id_angajat=angajati.id_angajat
WHERE UPPER(angajati.ume)=UPPER(v_numeang);

RETURN suma_totala;

END functie8;

--procedura ex 6
PROCEDURE proc_ex6(v_oras in departamente.locatie%TYPE) IS

    TYPE tab_ind IS TABLE OF angajati.ume%TYPE INDEX BY PLS_INTEGER;
    v_ume tab_ind;
    v_prenume tab_ind;
    TYPE vector IS VARRAY(30) OF angajati.salariu%TYPE;
    v_sal vector;
BEGIN

    SELECT salariu
    BULK COLLECT INTO v_sal
    FROM angajati
    JOIN departamente on departamente.id_departament=angajati.id_departament
    WHERE UPPER(departamente.locatie)=upper(v_oras)
    ORDER BY salariu DESC;

    FOR i IN v_sal.first..v_sal.last LOOP
        SELECT angajati.ume
        INTO v_ume(i)
        FROM ANGAJATI
        JOIN DEPARTAMENTE ON
departamente.id_departament=angajati.id_departament
        WHERE upper(departamente.locatie)=upper(v_oras)
        AND angajati.salariu=v_sal(i);
    END LOOP;

```

```

FOR i IN v_sal.first..v_sal.last LOOP
    SELECT angajati.prenume
    INTO v_prenume(i)
    FROM ANGAJATI
    JOIN DEPARTAMENTE ON
departamente.id_departament=angajati.id_departament
    WHERE upper(departamente.locatie)=upper(v_oras)
    AND angajati.salariu=v_sal(i);
END LOOP;

IF v_nume.count>0 THEN
    FOR i in v_sal.first..v_sal.last LOOP
        DBMS_OUTPUT.PUT_LINE('In departamentul din locatia '||v_oras ||
lucreaza angajatul '||v_nume(i)||' '||v_prenume(i)||' cu salariul de '||v_sal(i));
    END LOOP;
ELSE
    DBMS_OUTPUT.PUT_LINE('In departamentul din '||v_oras||' nu lucreaza
niciun angajat');
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('S-au gasit prea multe linii');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Altceva');
END proc_ex6;

```

--procedura ex 7

```

PROCEDURE proc_ex7(v_numedep in depozite_produce.nume%TYPE) IS

```

```

    CURSOR c1 IS
        SELECT id_depozit
        FROM depozite_produce
        WHERE nume=v_numedep;

```

```

    CURSOR c2(v_id depozite_produce.id_depozit%type) IS
        SELECT nume, pret
        FROM produse
        WHERE id_depozit=v_id
        ORDER BY pret desc;

```

```

    c_pret produse.pret%TYPE;
    c_denumire produse.nume%TYPE;
    c_id depozite_produce.id_depozit%TYPE;

```

```

BEGIN
    OPEN c1;
    LOOP
        FETCH c1 INTO c_id;

```

```

EXIT WHEN c1%NOTFOUND;
END LOOP;

OPEN c2(c_id);
LOOP
FETCH c2 INTO c_denumire, c_pret;
EXIT WHEN c2%NOTFOUND OR c2%ROWCOUNT>2;
DBMS_OUTPUT.PUT_LINE('In depozitul '||v_numedep||' se afla parfumul
'||c_denumire||' care costa '||c_pret);
END LOOP;

IF c2%ROWCOUNT = 0 THEN
DBMS_OUTPUT.PUT_LINE('Nu exista produse in acest depozit');
END IF;
CLOSE c2;
DBMS_OUTPUT.NEW_LINE;
CLOSE c1;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE('S-au gasit prea multe linii');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Altceva');
END proc_ex7;

--procedura ex 9
PROCEDURE proc_ex9 (nr_liv livrari.numar%TYPE) IS
CURSOR c9 IS
SELECT c.id_comanda, p.nume, dp.nume
FROM livrari l
JOIN comenzi c ON l.id_livrare=c.id_livrare
JOIN cosuri_produce cp ON cp.id_comanda=c.id_comanda
JOIN produse p ON p.id_produc=cp.id_produc
JOIN depozite_produce dp ON p.id_depozit=dp.id_depozit
WHERE l.numar=nr_liv;
CURSOR c9_1 IS
SELECT COUNT(*)
FROM livrari
WHERE numar=nr_liv;
id_com comenzi.id_comanda%TYPE;
nume_prod produse.nume%TYPE;
nume_dep depozite_produce.nume%TYPE;
nr NUMBER;
BEGIN
OPEN c9_1;
LOOP
FETCH c9_1 INTO nr;
EXIT WHEN c9_1%NOTFOUND;
END LOOP;

```

```

        IF nr>1
            THEN RAISE_APPLICATION_ERROR(-20004, 'Exista mai multe comenzi cu
acest numar');
        END IF;
        CLOSE c9_1;

        OPEN c9;
        LOOP
            FETCH c9 INTO id_com, nume_prod, nume_dep;
            EXIT WHEN c9%NOTFOUND;
        END LOOP;

        IF c9%ROWCOUNT = 0
            THEN RAISE_APPLICATION_ERROR(-20003, 'Nu exista nicio livrare cu
acest numar');
        END IF;

        IF c9%ROWCOUNT =1
            THEN DBMS_OUTPUT.PUT_LINE('Livrarii cu numarul '||nr_liv||' ii
corespunde comanda cu id-ul '||id_com||' care contine produsul '||
            nume_prod||' aflat in depozitul cu numele '||nume_dep);
        END IF;

        CLOSE c9;

    END proc_ex9;

END pachet_ex13;

```

```

635
636 CREATE OR REPLACE PACKAGE pachet_ex13 IS
637     FUNCTION funcție8(v_numeang ANGAJATI.nume%TYPE) RETURN plati.suma%TYPE;
638     PROCEDURE proc_ex6(v_oras in departamente.locatie%TYPE);
639     PROCEDURE proc_ex7(v_numedep in depozite_produce.nume%TYPE);
640     PROCEDURE proc_ex9 (nr_liv livrari.numar%TYPE);
641 END pachet_ex13;
642
643 CREATE OR REPLACE PACKAGE BODY pachet_ex13 IS
644
645     --functie ex8
646     FUNCTION funcție8(v_numeang ANGAJATI.nume%TYPE)
647     RETURN plati.suma%TYPE IS
648
649         suma_totala plati.suma%TYPE;
650         nr and NIMRRER:

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

Task completed in 0.032 seconds

Package PACHET_EX13 compiled


```
Worksheet Query Builder
812 CLOSE c9_1;
813
814 OPEN c9;
815 LOOP
816   FETCH c9 INTO id_com, nume_prod, nume_dep;
817   EXIT WHEN c9%NOTFOUND;
818   END LOOP;
819
820   IF c9%ROWCOUNT = 0
821     THEN RAISE_APPLICATION_ERROR(-20003, 'Nu exista nicio livrare cu acest numar');
822   END IF;
823
824   IF c9%ROWCOUNT = 1
825     THEN DEMS_OUTPUT.PUT_LINE('Livrarii cu numarul '||nr_liv||' ii corespunde comanda cu
826     nume_prod||' aflat in depozitul cu numele '||nume_dep);
827   END IF;
828
829   CLOSE c9;
830
831 END proc_ex9;
832
833 END pachet_ex13;
834
835 SELECT pachet_ex13.functie8('Monea') FROM DUAL;
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x
Task completed in 0.032 seconds

Package Body PACHET_EX13 compiled

SELECT pachet_ex13.functie8('Monea') FROM DUAL;

EXECUTE pachet_ex13.proc_ex6('Ilfov');

EXECUTE pachet_ex13.proc_ex7('Depozit Cluj');

EXECUTE pachet_ex13.proc_ex9(1002);

```
834
835 SELECT pachet_ex13.functie8('Monea') FROM DUAL;
836
837 EXECUTE pachet_ex13.proc_ex6('Ilfov');
838
839 EXECUTE pachet_ex13.proc_ex7('Depozit Cluj');
840
841 EXECUTE pachet_ex13.proc_ex9(1002);
842
843
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3
SQL All Rows Fetched: 1 in 0.007 seconds

PACHET_EX13.FUNCTIE8('MONEA')	
1	750

```

821     THEN RAISE_APPLICATION_ERROR(-20003, 'Nu exista nicio livrare');
822     END IF;
823
824     IF c9%ROWCOUNT = 1
825     THEN DBMS_OUTPUT.PUT_LINE('Livrarii cu numarul '||nr_liv||' :
826     nume_prod||' aflat in depozitul cu numele '||nume_dep);
827     END IF;
828
829     CLOSE c9;
830
831 END proc_ex9;
832
833 END pachet_ex13;
834
835 SELECT pachet_ex13.functie8('Monea') FROM DUAL;
836
837 EXECUTE pachet_ex13.proc_ex6('Ilfov');
838
839 EXECUTE pachet_ex13.proc_ex7('Depozit Cluj');
840
841 EXECUTE pachet_ex13.proc_ex9(1002);
842
843

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x

Task completed in 0.033 seconds

Package Body PACHET_EX13 compiled

PL/SQL procedure successfully completed.

In departamentul din locatia Ilfov lucreaza angajatul Destiper Mircea cu salariul de 7000
In departamentul din locatia Ilfov lucreaza angajatul Anton Andreea cu salariul de 5700
In departamentul din locatia Ilfov lucreaza angajatul Puscasu Mihai cu salariul de 2500

```

824     IF c9%ROWCOUNT = 1
825     THEN DBMS_OUTPUT.PUT_LINE('Livrarii cu numarul '||nr_liv||' :
826     nume_prod||' aflat in depozitul cu numele '||nume_dep);
827     END IF;
828
829     CLOSE c9;
830
831 END proc_ex9;
832
833 END pachet_ex13;
834
835 SELECT pachet_ex13.functie8('Monea') FROM DUAL;
836
837 EXECUTE pachet_ex13.proc_ex6('Ilfov');
838
839 EXECUTE pachet_ex13.proc_ex7('Depozit Cluj');
840
841 EXECUTE pachet_ex13.proc_ex9(1002);
842
843

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x

Task completed in 0.027 seconds

Package Body PACHET_EX13 compiled

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

In departamentul din locatia Ilfov lucreaza angajatul Puscasu Mihai cu sa:
In depozitul Depozit Cluj se afla parfumul CH-212-VIP care costa 2099

```

827     END IF;
828
829     CLOSE c9;
830
831 END proc_ex9;
832
833 END pachet_ex13;
834
835 SELECT pachet_ex13.functie8('Monea')
836
837 EXECUTE pachet_ex13.proc_ex6('Ilfov')
838
839 EXECUTE pachet_ex13.proc_ex7('Depozit
840
841 EXECUTE pachet_ex13.proc_ex9(1002);
842
843

```

Script Output x Query Result x Query... x x x x x

Task completed in 0.041 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Livrarii cu numarul 1002 ii corespunde comanda cu id-ul 3 care contine produsul Dior-Miss-Dior aflat in depozitul cu numele Depoz