

Proiect SGBD -MAGAZIN ONLINE-

Realizat de OANCEA ELENA-ANTONIA

GRUPA 242

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

În acest proiect voi crea baza de date a unui magazin online. Magazinul online va conține :

- Produse : - produsele pe care le pot comanda clientii
- Comenzi : - comenzile realizate de clienti
- Retururi : - retururile realizate
- Furnizori : - date despre furnizori
- Clienti : - date despre clinti
- Departamente: - retinem numele departamentelor
- Angajati: – date despre angajati
- Posturi : - posturile pe care le pot ocupa angajatii si salariul minim pentru fiecare
- Istoric : - retinem evenimentele ALTER, CREATE, DROP

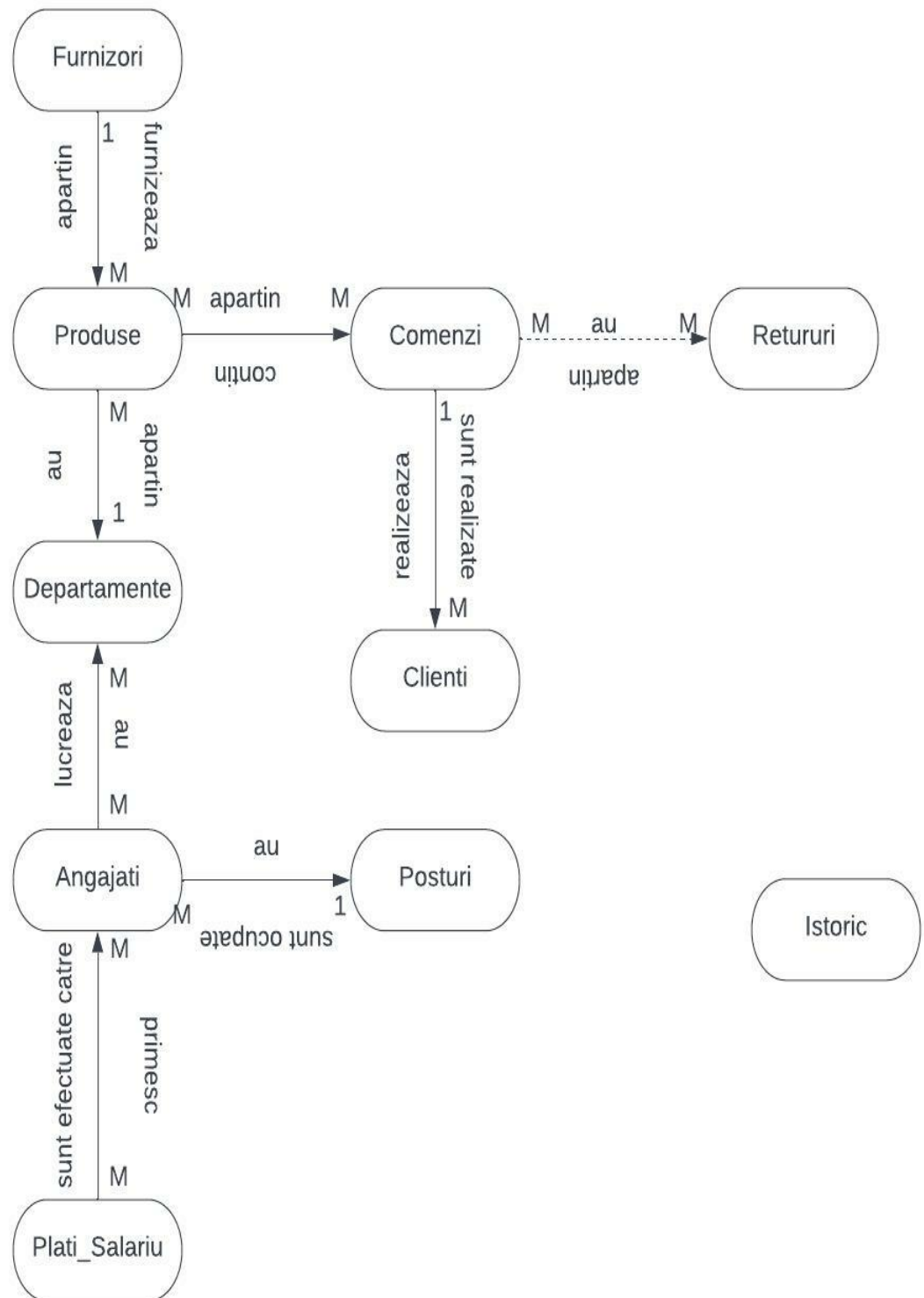
Fiecare produs apartine unui departament.

Comenzile sunt realizate de clienti contin cel putin un produs.

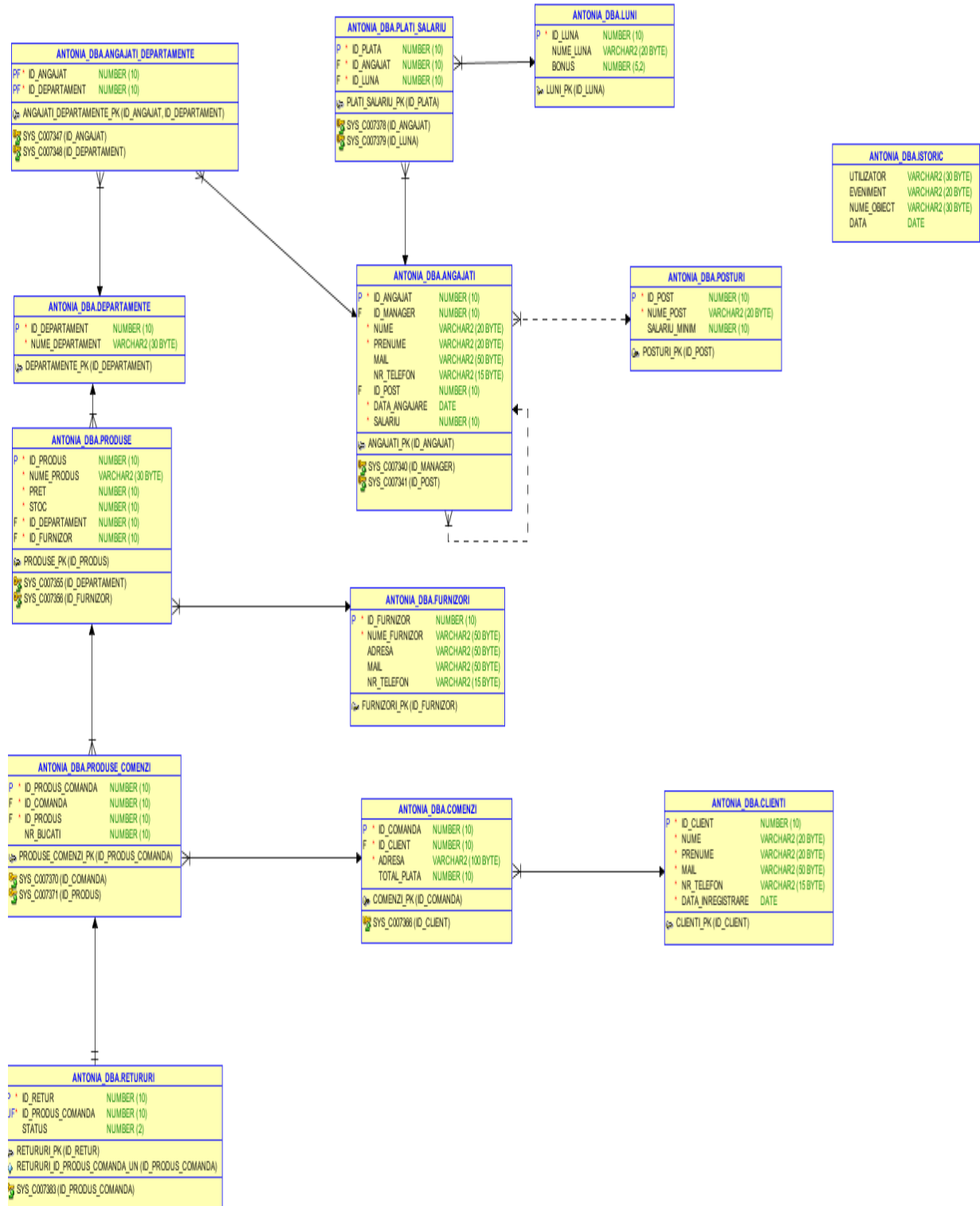
Retururile sunt realizate de clienti asupra unei comenzi existente.

Angajati lucreaza la cel putin un departament.

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 tablele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

+

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

```
CREATE SEQUENCE secv
```

```
START WITH 100
```

```
INCREMENT BY 1
```

ISTORIC

```
CREATE TABLE istoric(
```

```
    utilizator VARCHAR2(30),
```

```
    eveniment VARCHAR2(20),
```

```
    nume_obiect VARCHAR2(30),
```

```
    data DATE
```

```
);
```

```
--ADAUGAM EVENIMENTELE DE TIP CREATE, DROP SI ALTER IN TABELUL ISTORIC
```

```
CREATE OR REPLACE TRIGGER istoric_LDD
```

```
AFTER CREATE OR DROP OR ALTER ON SCHEMA
```

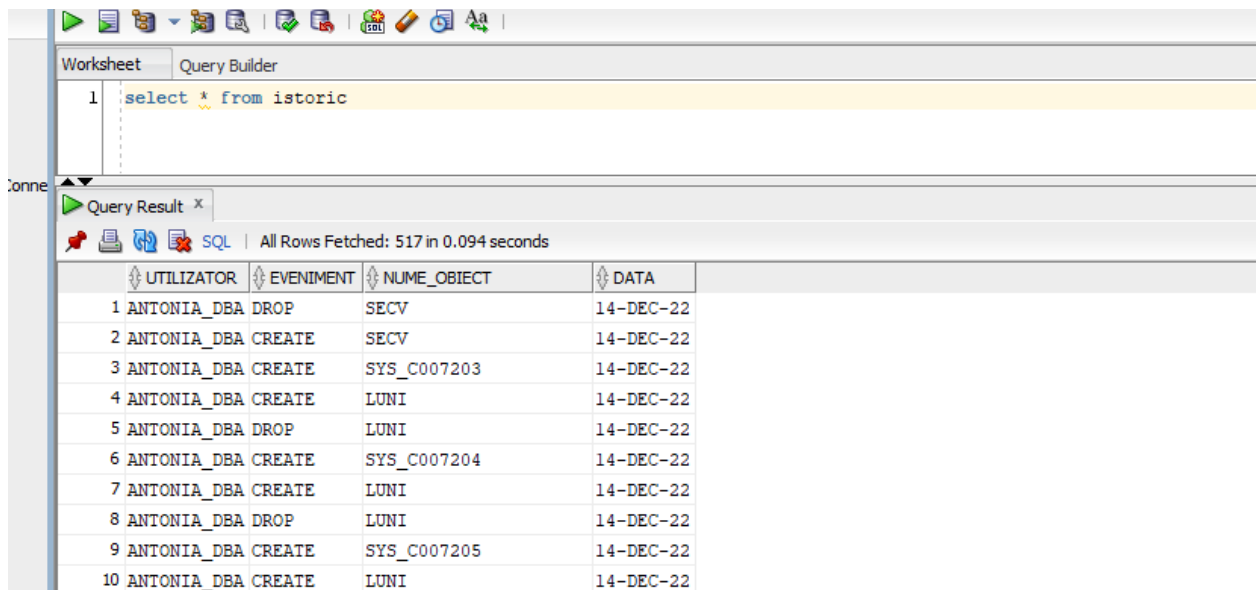
```
BEGIN
```

```
INSERT INTO istoric
```

```
VALUES (SYS.LOGIN_USER, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME, SYSDATE);
```

```
END;
```

```
/
```



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons. Below it, a 'Worksheet' tab is active, displaying a SQL query: `select * from istoric`. Below the query, a 'Query Result' tab is active, showing a table with 10 rows of data. The table has four columns: UTILIZATOR, EVENIMENT, NUME_OBJECT, and DATA. The data is as follows:

	UTILIZATOR	EVENIMENT	NUME_OBJECT	DATA
1	ANTONIA_DBA	DROP	SECV	14-DEC-22
2	ANTONIA_DBA	CREATE	SECV	14-DEC-22
3	ANTONIA_DBA	CREATE	SYS_C007203	14-DEC-22
4	ANTONIA_DBA	CREATE	LUNI	14-DEC-22
5	ANTONIA_DBA	DROP	LUNI	14-DEC-22
6	ANTONIA_DBA	CREATE	SYS_C007204	14-DEC-22
7	ANTONIA_DBA	CREATE	LUNI	14-DEC-22
8	ANTONIA_DBA	DROP	LUNI	14-DEC-22
9	ANTONIA_DBA	CREATE	SYS_C007205	14-DEC-22
10	ANTONIA_DBA	CREATE	LUNI	14-DEC-22

LUNI

create table luni(

id_luna number(10) NOT NULL ,

nume_luna varchar2(20)NOT NULL,

bonus numeric(5,2) DEFAULT 0,

CONSTRAINT luni_pk PRIMARY KEY (id_luna)

);

insert into luni (id_luna,nume_luna,bonus) values (1, 'Ianuarie', 0.1);

insert into luni (id_luna,nume_luna) values (2, 'Februarie');

insert into luni (id_luna,nume_luna,bonus) values (3, 'Martie', 0.05);

insert into luni (id_luna,nume_luna,bonus) values (4, 'Aprilie', 0.02);

insert into luni (id_luna,nume_luna) values (5, 'Mai');

insert into luni (id_luna,nume_luna,bonus) values (6, 'Iunie', 0.1);

insert into luni (id_luna,nume_luna) values (7, 'Iulie');

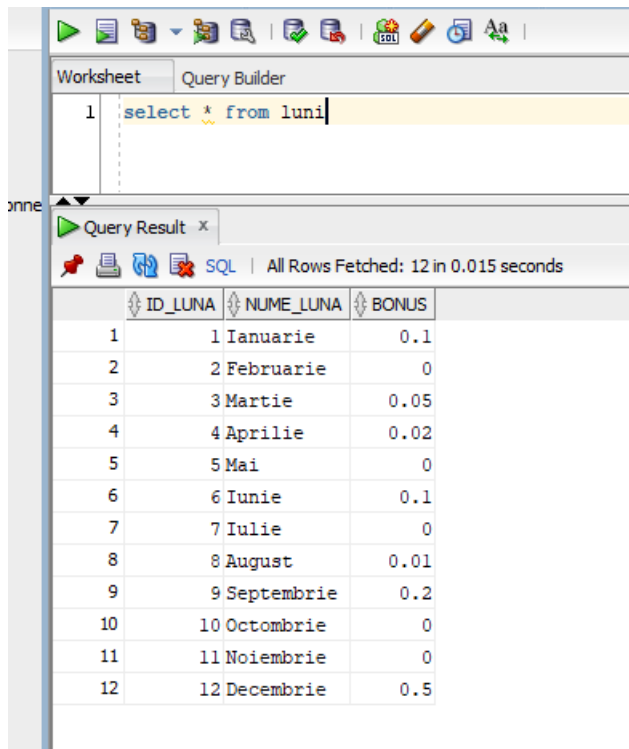
insert into luni (id_luna,nume_luna,bonus) values (8, 'August', 0.01);

insert into luni (id_luna,nume_luna,bonus) values (9, 'Septembrie', 0.2);

insert into luni (id_luna,nume_luna) values (10, 'Octombrie');

insert into luni (id_luna,nume_luna) values (11, 'Noiembrie');

insert into luni (id_luna,nume_luna,bonus) values (12, 'Decembrie', 0.5);



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons. Below it, a 'Worksheet' tab is active, displaying a query: `select * from luni`. Below the query, a 'Query Result' tab is active, showing a table with 12 rows and 3 columns: `ID_LUNA`, `NUME_LUNA`, and `BONUS`. The status bar indicates 'All Rows Fetched: 12 in 0.015 seconds'.

	ID_LUNA	NUME_LUNA	BONUS
1	1	Ianuarie	0.1
2	2	Februarie	0
3	3	Martie	0.05
4	4	Aprilie	0.02
5	5	Mai	0
6	6	Iunie	0.1
7	7	Iulie	0
8	8	August	0.01
9	9	Septembrie	0.2
10	10	Octombrie	0
11	11	Noiembrie	0
12	12	Decembrie	0.5

POSTURI

create table furnizori(

id_furnizor number(10) NOT NULL,

nume_furnizor varchar2(50) NOT NULL,

adresa varchar2(50),

mail varchar2(50),

nr_telefon varchar2(15),

CONSTRAINT furnizori_pk PRIMARY KEY (id_furnizor)

);

insert into posturi (id_post, nume_post, salariu_minim) values (1,'CONTABIL',2000);

insert into posturi (id_post, nume_post, salariu_minim) values (2,'AMBALARE COLETE',2000);

insert into posturi (id_post, nume_post, salariu_minim) values (3,'CURIER',2500);

insert into posturi (id_post, nume_post) values (4,'DEPOZIT');

insert into posturi (id_post, nume_post, salariu_minim) values (5,'MANAGER',3000);

Worksheet

Query Builder

1

select * from posturi

Query Result x

SQL | All Rows Fetched: 5 in 0.015 seconds

ID_POST	NUME_POST	SALARIU_MINIM
1	1 CONTABIL	2000
2	2 AMBALARE COLETE	2000
3	3 CURIER	2500
4	4 DEPOZIT	1500
5	5 MANAGER	3000

FURNIZORI

create table furnizori(

id_furnizor number(10) NOT NULL,

nume_furnizor varchar2(50) NOT NULL,

adresa varchar2(50),

mail varchar2(50),

nr_telefon varchar2(15),

CONSTRAINT furnizori_pk PRIMARY KEY (id_furnizor)

);

insert into furnizori (id_furnizor,nume_furnizor) values (secd.nextval,'furnizor1');

insert into furnizori (id_furnizor,nume_furnizor) values (secd.nextval,'furnizor2');

insert into furnizori (id_furnizor,nume_furnizor) values (secd.nextval,'furnizor3');

insert into furnizori (id_furnizor,nume_furnizor) values (secd.nextval,'furnizor4');

insert into furnizori (id_furnizor,nume_furnizor) values (secd.nextval,'furnizor5');

The screenshot shows a database query tool interface. At the top, there's a 'Worksheet' tab and a 'Query Builder' tab. Below the 'Query Builder' tab, a query is entered: `select * from furnizori`. Below the query, there's a 'Query Result' tab showing the results of the query. The results are displayed in a table with 5 rows and 5 columns: ID_FURNIZOR, NUME_FURNIZOR, ADRESA, MAIL, and NR_TELEFON. The data is as follows:

ID_FURNIZOR	NUME_FURNIZOR	ADRESA	MAIL	NR_TELEFON
1	147 furnizor1	(null)	(null)	(null)
2	148 furnizor2	(null)	(null)	(null)
3	149 furnizor3	(null)	(null)	(null)
4	150 furnizor4	(null)	(null)	(null)
5	151 furnizor5	(null)	(null)	(null)

ANGAJATI

create table angajati(

id_angajat number(10) NOT NULL,

id_manager number(10),

nume varchar2(20) NOT NULL,

prenume varchar2(20) NOT NULL,

mail varchar2(50),

nr_telefon varchar2(15),

id_post number(10),

data_angajare date NOT NULL,

salariu number(10) NOT NULL,

CONSTRAINT angajati_pk PRIMARY KEY (id_angajat),

CONSTRAINT angajati_fk1 FOREIGN KEY (id_manager) REFERENCES angajati(id_angajat),

CONSTRAINT angajati_fk2 FOREIGN KEY (id_post) REFERENCES posturi(id_post)

);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values
(secd.nextval,null,'Popa','Daniel',3,TO_DATE('2015-07-21','YYYY-MM-DD'),2700);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values
(secd.nextval,null,'Andrei','Marian',1,TO_DATE('2017-06-02','YYYY-MM-DD'),2100);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values
(secd.nextval,null,'Popescu','Carina',2,TO_DATE('2018-01-09','YYYY-MM-DD'),2500);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values
(secd.nextval,null,'Ionescu','Sorin',3,TO_DATE('2019-02-14','YYYY-MM-DD'),2700);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,286,'Popescu','Diana',4,TO_DATE('2020-03-27','YYYY-MM-DD'),1600);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,287,'Popa','Ion',3,TO_DATE('2021-07-24','YYYY-MM-DD'),2700);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,288,'David','Marius',1,TO_DATE('2022-04-19','YYYY-MM-DD'),2700);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,289,'Popovici','Maria',2,TO_DATE('2016-04-07','YYYY-MM-DD'),2200);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,null,'Simionescu','Sorin',3,TO_DATE('2017-05-05','YYYY-MM-DD'),2600);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,288,'Oancea','Andreea',4,TO_DATE('2018-08-12','YYYY-MM-DD'),1800);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,null,'Coman','Ana',2,TO_DATE('2019-06-23','YYYY-MM-DD'),2100);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,null,'Danescu','Florin',1,TO_DATE('2020-09-07','YYYY-MM-DD'),2000);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,288,'Florescu','Alexandru',4,TO_DATE('2021-12-13','YYYY-MM-DD'),2000);

insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secv.nextval,null,'Banica','Dan',3,TO_DATE('2015-06-20','YYYY-MM-DD'),2500);

Worksheet





Query Builder

1

select * from angajati

ne

Query Result x

    SQL | All Rows Fetched: 14 in 0.015 seconds

	ID_ANGAJAT	ID_MANAGER	NUME	PRENUME	MAIL	NR_TELEFON	ID_POST	DATA_ANGAJARE	SALARIU
1	286	(null)	Popa	Daniel	(null)	(null)	3 21-JUL-15		2700
2	287	(null)	Andrei	Marian	(null)	(null)	1 02-JUN-17		2100
3	288	(null)	Popescu	Carina	(null)	(null)	2 09-JAN-18		2500
4	289	(null)	Ionescu	Sorin	(null)	(null)	3 14-FEB-19		2700
5	290	286	Popescu	Diana	(null)	(null)	4 27-MAR-20		1600
6	291	287	Popa	Ion	(null)	(null)	3 24-JUL-21		2700
7	292	288	David	Marius	(null)	(null)	1 19-APR-22		2700
8	293	289	Popovici	Maria	(null)	(null)	2 07-APR-16		2200
9	294	(null)	Simionescu	Sorin	(null)	(null)	3 05-MAY-17		2600
10	295	288	Oancea	Andreea	(null)	(null)	4 12-AUG-18		1800
11	296	(null)	Coman	Ana	(null)	(null)	2 23-JUN-19		2100
12	297	(null)	Danescu	Florin	(null)	(null)	1 07-SEP-20		2000
13	298	288	Florescu	Alexandru	(null)	(null)	4 13-DEC-21		2000
14	299	(null)	Banica	Dan	(null)	(null)	3 20-JUN-15		2500

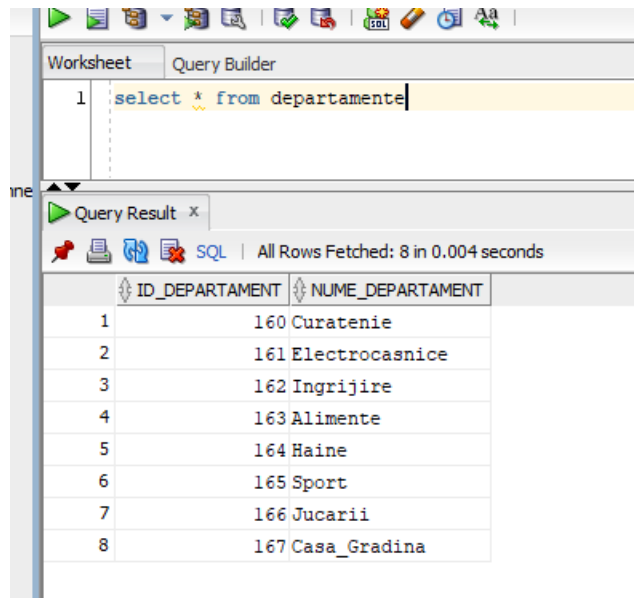
DEPARTAMENTE

```

create table departamente(
    id_departament number(10) NOT NULL,
    nume_departament varchar2(30) NOT NULL,
    CONSTRAINT departamente_pk PRIMARY KEY (id_departament)
);

insert into departamente (id_departament, nume_departament) values (secd.nextval,'Curatenie');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Electrocasnice');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Ingrijire');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Alimente');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Haine');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Sport');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Jucarii');
insert into departamente (id_departament, nume_departament) values (secd.nextval,'Casa_Gradina');

```



The screenshot shows a database query tool interface. The top pane, labeled 'Query Builder', contains the SQL query: `select * from departamente`. The bottom pane, labeled 'Query Result', displays the results of the query. It shows a table with two columns: 'ID_DEPARTAMENT' and 'NUME_DEPARTAMENT'. The results are as follows:

ID_DEPARTAMENT	NUME_DEPARTAMENT
160	Curatenie
161	Electrocasnice
162	Ingrijire
163	Alimente
164	Haine
165	Sport
166	Jucarii
167	Casa_Gradina

ANGAJATI_DEPARTAMENTE

```

create table angajati_departamente(
    id_angajat number(10) not null,
    id_departament number(10) not null,
    CONSTRAINT ang_dep_pk PRIMARY KEY(id_angajat, id_departament),
    CONSTRAINT ang_dep_fk1 FOREIGN KEY (id_angajat) REFERENCES angajati(id_angajat),

```

```
CONSTRAINT ang_dep_fk2 FOREIGN KEY (id_department) REFERENCES  
departamente(id_department)
```

```
);
```

```
insert into angajati_departamente (id_angajat,id_department) values (286,160);
```

```
insert into angajati_departamente (id_angajat,id_department) values (287,161);
```

```
insert into angajati_departamente (id_angajat,id_department) values (288,162);
```

```
insert into angajati_departamente (id_angajat,id_department) values (289,163);
```

```
insert into angajati_departamente (id_angajat,id_department) values (290,164);
```

```
insert into angajati_departamente (id_angajat,id_department) values (291,165);
```

```
insert into angajati_departamente (id_angajat,id_department) values (292,166);
```

```
insert into angajati_departamente (id_angajat,id_department) values (294,167);
```

```
insert into angajati_departamente (id_angajat,id_department) values (296,160);
```

```
insert into angajati_departamente (id_angajat,id_department) values (297,161);
```

```
insert into angajati_departamente (id_angajat,id_department) values (286,162);
```

```
insert into angajati_departamente (id_angajat,id_department) values (298,163);
```

```
insert into angajati_departamente (id_angajat,id_department) values (299,163);
```

```
insert into angajati_departamente (id_angajat,id_department) values (286,164);
```

```
insert into angajati_departamente (id_angajat,id_department) values (287,164);
```

```
insert into angajati_departamente (id_angajat,id_department) values (288,164);
```

```
insert into angajati_departamente (id_angajat,id_department) values (289,165);
```

```
insert into angajati_departamente (id_angajat,id_department) values (286,165);
```

```
insert into angajati_departamente (id_angajat,id_department) values (287,166);
```

```
insert into angajati_departamente (id_angajat,id_department) values (291,167);
```

Worksheet		Query Builder
1		<code>select * from angajati_departamente</code>
Query Result x		
All Rows Fetched: 20 in 0.004 seconds		
	ID_ANGAJAT	ID_DEPARTAMENT
1	286	160
2	286	162
3	286	164
4	286	165
5	287	161
6	287	164
7	287	166
8	288	162
9	288	164
10	289	163
11	289	165
12	290	164
13	291	165
14	291	167
15	292	166
16	294	167
17	296	160
18	297	161
19	298	163
20	299	163

PRODUSE

create table produse(

id_produs number(10) NOT NULL,

nume_produs varchar2(30) NOT NULL,

pret number(10) NOT NULL,

stoc number(10) NOT NULL,

id_departament NOT NULL,

id_furnizor NOT NULL,

CONSTRAINT produse_pk PRIMARY KEY(id_produs),

CONSTRAINT produse_fk1 FOREIGN KEY (id_furnizor) REFERENCES furnizori(id_furnizor),

CONSTRAINT produse_fk2 FOREIGN KEY (id_departament) REFERENCES departamente(id_departament)

);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Scaun',100,4,167,147);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Masa',200,2,167,151);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Detergent vase',10,5,160,148);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Apa',8,20,163,149);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Zahar',5,10,163,150);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Faina',5,12,163,147);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Minge',50,7,165,148);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Paleta de tenis',80,3,165,149);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Fileu',40,6,165,150);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Pantaloni scurti',100,2,164,151);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Pantaloni',120,4,164,147);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Tricou',70,4,164,148);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Hanorac',90,4,164,149);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Robot de bucatarie',300,3,161,150);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Puzzle',100,2,166,147);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Monopoly',200,1,166,147);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Tren de jucarie',100,3,166,149);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Papusa',100,4,166,149);

insert into produse (id_produs,nume_produs,pret,stoc,id_departament,id_furnizor) values
(secv.nextval,'Dulap',500,2,167,151);

Worksheet

Query Builder

1

select * from produse

Query Result x

SQL | All Rows Fetched: 19 in 0.005 seconds

	ID_PRODUS	NUME_PRODUS	PRET	STOC	ID_DEPARTAMENT	ID_FURNIZOR
1	168	Scaun	100	2	167	147
2	190	Masa	200	2	167	151
3	191	Detergent vase	10	3	160	148
4	192	Apa	8	16	163	149
5	193	Zahar	5	6	163	150
6	194	Faina	5	11	163	147
7	195	Minge	50	7	165	148
8	196	Paleta de tenis	80	2	165	149
9	197	Fileu	40	5	165	150
10	198	Pantaloni scurti	100	0	164	151
11	199	Pantaloni	120	2	164	147
12	200	Tricou	70	3	164	148
13	201	Hanorac	90	4	164	149
14	202	Robot de bucatarie	300	4	161	150
15	203	Puzzle	100	0	166	147
16	204	Monopoly	200	0	166	147
17	205	Tren de jucarie	100	3	166	149
18	206	Papusa	100	4	166	149
19	207	Dulap	500	2	167	151

CLIENTI

create table clienti(

id_client number(10) NOT NULL,

nume varchar2(20) NOT NULL,

prenume varchar2(20) NOT NULL,

mail varchar2(50) NOT NULL,

nr_telefon varchar2(15) NOT NULL,

data_inregistrare date NOT NULL,

CONSTRAINT clienti_pk PRIMARY KEY(id_client)

);

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Popescu','Ion','popescu.ion@yahoo.com','0722334455',TO_DATE('2019-06-21','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Popescu','Andrei','popescu.andrei@yahoo.com','0722334456',TO_DATE('2020-07-01','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Ionescu','Mihai','ionescu.mihai@yahoo.com','0712334455',TO_DATE('2022-05-01','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Savu','Ion','savu.ion@yahoo.com','0722344455',TO_DATE('2022-05-20','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Georgescu','Mihai','georgescu.mihai@yahoo.com','0772334455',TO_DATE('2019-07-24','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Ionescu','Daniela','ionescudaniela@yahoo.com','0752334455',TO_DATE('2021-07-22','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Marinescu','Maria','maria_marinescu@yahoo.com','0792334455',TO_DATE('2022-08-27','YYYY-MM-DD'));
```

```
insert into clienti (id_client,nume,prenume,mail,nr_telefon,data_inregistrare) values
(secv.nextval,'Coman','Sorin','coman_sorin@yahoo.com','0762334455',TO_DATE('2020-08-04','YYYY-MM-DD'));
```





Worksheet

Query Builder

1

select * from clienti

Query Result x

    SQL | All Rows Fetched: 8 in 0.016 seconds

	ID_CLIENT	NUME	PRENUME	MAIL	NR_TELEFON	DATA_INREGISTRARE
1	152	Popescu	Ion	popescu.ion@yahoo.com	0722334455	21-JUN-19
2	153	Popescu	Andrei	popescu.andrei@yahoo.com	0722334456	01-JUL-20
3	154	Ionescu	Mihai	ionescu.mihai@yahoo.com	0712334455	01-MAY-22
4	155	Savu	Ion	savu.ion@yahoo.com	0722344455	20-MAY-22
5	156	Georgescu	Mihai	georgescu.mihai@yahoo.com	0772334455	24-JUL-19
6	157	Ionescu	Daniela	ionescudaniela@yahoo.com	0752334455	22-JUL-21
7	158	Marinescu	Maria	maria_marinescu@yahoo.com	0792334455	27-AUG-22
8	159	Coman	Sorin	coman_sorin@yahoo.com	0762334455	04-AUG-20

COMENZI

```
create table comenzi(
```



```

id_comanda number(10) NOT NULL,
id_client number(10) NOT NULL,
adresa varchar2(100) NOT NULL,
total_plata number(10) DEFAULT 0,
CONSTRAINT comenzi_pk PRIMARY KEY(id_comanda),
CONSTRAINT comenzi_fk1 FOREIGN KEY (id_client) REFERENCES clienti(id_client)
);

insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,152,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,152,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,152,'Brasov');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,152,'Pitesti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,154,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,154,'Sibiu');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,153,'Sibiu');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,156,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,156,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,153,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,154,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,157,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,159,'Brasov');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,159,'Campulung');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,158,'Targoviste');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,153,'Brasov');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,155,'Cluj');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,155,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,155,'Timisoara');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,155,'Sibiu');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,154,'Bucuresti');
insert into comenzi (id_comanda,id_client,adresa) values (sekv.nextval,153,'Brasov');

```

Worksheet

Query Builder

1

select * from comenzi

Query Result x

SQL | All Rows Fetched: 22 in 0.016 seconds

	ID_COMANDA	ID_CLIENT	ADRESA	TOTAL_PLATA
1	208	152	Bucuresti	1200
2	209	152	Bucuresti	600
3	210	152	Brasov	270
4	211	152	Pitesti	200
5	212	154	Bucuresti	130
6	213	154	Sibiu	5
7	214	153	Sibiu	8
8	215	156	Bucuresti	8
9	216	156	Bucuresti	20
10	217	153	Bucuresti	80
11	218	154	Bucuresti	40
12	219	157	Bucuresti	90
13	220	159	Brasov	120
14	221	159	Campulung	8
15	222	158	Targoviste	5
16	223	153	Brasov	16
17	224	155	Cluj	100
18	225	155	Bucuresti	18
19	226	155	Timisoara	8
20	227	155	Sibiu	0
21	228	154	Bucuresti	0
22	229	153	Brasov	0

PRODUSE_COMENZI

```

create table produse_comenzi(
    id_produș_comanda number(10) NOT NULL,
    id_comanda number(10) NOT NULL,
    id_produș number(10) NOT NULL,
    nr_bucati number(10) DEFAULT 1,
    CONSTRAINT prod_com_pk PRIMARY KEY(id_produș_comanda),
    CONSTRAINT prod_com_fk1 FOREIGN KEY (id_comanda) REFERENCES comenzi(id_comanda),
    CONSTRAINT prod_com_fk2 FOREIGN KEY (id_produș) REFERENCES produse(id_produș)
);

insert into produse_comenzi (id_produș_comanda,id_comanda,id_produș,nr_bucati) values
(secv.nextval,209,202,1);

```

```

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,209,203,2);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,208,204,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,209,205,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,210,206,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,211,168,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,212,191,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,213,194,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,214,192,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,215,192,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,216,193,4);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,217,196,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,218,197,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,219,201,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,210,200,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,210,198,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,211,198,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,212,199,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,220,199,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,221,192,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,222,193,1);

```

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,223,192,2);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,224,168,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,225,191,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,225,192,1);

insert into produse_comenzi (id_produs_comanda,id_comanda,id_produs,nr_bucati) values
(secv.nextval,226,192,1);

Worksheet

Query Builder

1

select * from produse_comenzi

Query Result x

SQL | All Rows Fetched: 27 in 0.006 seconds

	ID_PRODUS_COMANDA	ID_COMANDA	ID_PRODUS	NR_BUCATI
1	304	208	207	2
2	306	209	202	1
3	307	209	203	2
4	308	208	204	1
5	309	209	205	1
6	310	210	206	1
7	311	211	168	1
8	312	212	191	1
9	313	213	194	1
10	314	214	192	1
11	315	215	192	1
12	316	216	193	4
13	317	217	196	1
14	318	218	197	1
15	319	219	201	1
16	320	210	200	1
17	321	210	198	1
18	322	211	198	1
19	323	212	199	1
20	324	220	199	1
21	325	221	192	1
22	326	222	193	1
23	327	223	192	2
24	328	224	168	1
25	329	225	191	1
26	330	225	192	1
27	331	226	192	1

[RETURURI](#)

```

create table retururi(
    id_retur number(10) NOT NULL,
    id_produș_comanda UNIQUE NOT NULL,
    status number(2) DEFAULT 1,
    CONSTRAINT retururi_pk PRIMARY KEY(id_retur),
    CONSTRAINT retururi_fk1 FOREIGN KEY (id_produș_comanda) REFERENCES
produse_comenzi(id_produș_comanda)
);

insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,306,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,310,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,309,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,319,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,320,0);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,325,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,331,0);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,330,0);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,327,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,326,1);
insert into retururi (id_retur,id_produș_comanda,status) VALUES (secv.nextval,324,0);

```

Worksheet Query Builder

1 select * from retururi

Query Result x

SQL | All Rows Fetched: 12 in 0.017 seconds

	ID_RETUR	ID_PRODUS_COMANDA	STATUS
1	332	304	1
2	334	306	1
3	335	310	1
4	336	309	1
5	337	319	1
6	338	320	0
7	339	325	1
8	340	331	0
9	341	330	0
10	342	327	1
11	343	326	1
12	344	324	0

PLATI SALARIU

create table plati_salariu(

id_plata number(10) NOT NULL,

id_angajat number(10) NOT NULL,

id_luna number(10) NOT NULL,

CONSTRAINT plati_sal_pk PRIMARY KEY(id_plata),

CONSTRAINT plati_sal_fk1 FOREIGN KEY (id_angajat) REFERENCES angajati(id_angajat),

CONSTRAINT plati_sal_fk2 FOREIGN KEY (id_luna) REFERENCES luni(id_luna)

);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,286,1);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,287,2);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,288,3);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,289,4);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,290,5);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,291,6);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,292,7);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,293,8);

insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,294,9);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,295,10);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,296,11);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,297,12);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,298,1);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,299,2);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,286,3);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,287,4);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,288,5);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,289,6);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,286,7);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,287,8);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,288,9);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,289,10);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,290,11);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,291,12);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,292,1);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,293,2);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,294,3);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,295,4);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,296,5);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,297,6);
insert into plati_salariu (id_plata,id_angajat,id_luna) values (secv.nextval,298,7);

Worksheet

Query Builder

1

select * from plati_salariu

1e

Query Result x

SQL | All Rows Fetched: 31 in 0.016 seconds

	ID_PLATA	ID_ANGAJAT	ID_LUNA
1	346	286	1
2	347	287	2
3	348	288	3
4	349	289	4
5	350	290	5
6	351	291	6
7	352	292	7
8	353	293	8
9	354	294	9
10	355	295	10
11	356	296	11
12	357	297	12
13	358	298	1
14	359	299	2
15	360	286	3
16	361	287	4
17	362	288	5
18	363	289	6
19	364	286	7
20	365	287	8
21	366	288	9
22	367	289	10
23	368	290	11
24	369	291	12
25	370	292	1
26	371	293	2
27	372	294	3
28	373	295	4
29	374	296	5
30	375	297	6
31	376	298	7

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 un manager dat ca parametru afisati pentru angajatii pe care ii are in subordine salariile pe care
 --le primesc acestia in fiecare luna a anului. Daca id-ul nu corespunde unui manager afisati un mesaj
 semnificativ.

CREATE OR REPLACE PROCEDURE ANG_SAL (man angajati.id_manager%type) AS

TYPE vector IS VARRAY(13) OF NUMBER;


```

ang vector := vector();

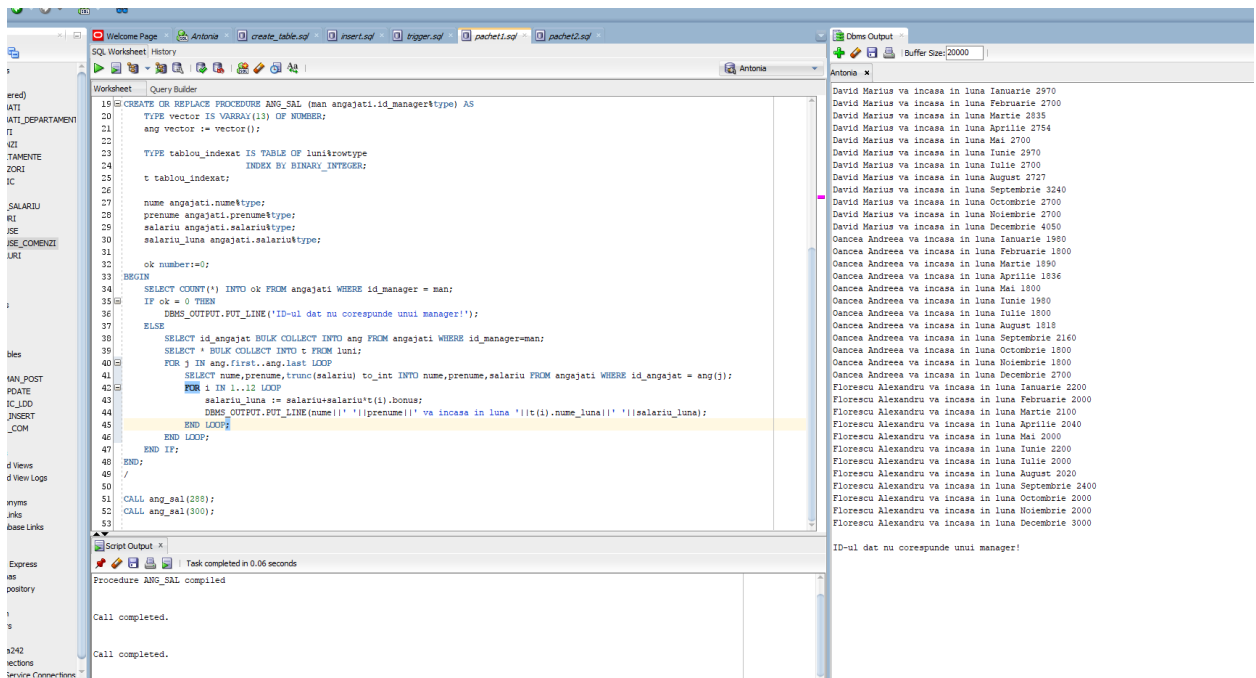
TYPE tablou_indexat IS TABLE OF luni%rowtype
        INDEX BY BINARY_INTEGER;

t tablou_indexat;

nume angajati.nume%type;
prenume angajati.prenume%type;
salariu angajati.salariu%type;
salariu_luna angajati.salariu%type;

ok number:=0;
BEGIN
    SELECT COUNT(*) INTO ok FROM angajati WHERE id_manager = man;
    IF ok = 0 THEN
        DBMS_OUTPUT.PUT_LINE('ID-ul dat nu corespunde unui manager!');
    ELSE
        SELECT id_angajat BULK COLLECT INTO ang FROM angajati WHERE id_manager=man;
        SELECT * BULK COLLECT INTO t FROM luni;
        FOR j IN ang.first..ang.last LOOP
            SELECT nume,prenume,trunc(salariu) to_int INTO nume,prenume,salariu FROM angajati WHERE
id_angajat = ang(j);
            FOR i IN 1..12 LOOP
                salariu_luna := salariu+salariu*t(i).bonus;
                DBMS_OUTPUT.PUT_LINE(nume||' '||prenume||' va incasa in luna '||t(i).nume_luna||'
'||salariu_luna);
            END LOOP;
        END LOOP;
    END IF;
END;
/

```



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.

--Pentru fiecare departament afisati produsele care au pretul mai mare sau egal decat un parametru.

CREATE OR REPLACE PROCEDURE PROD_DEP(par number) AS

CURSOR c_prod (parametru NUMBER) IS

SELECT id_produc, nume_produc, pret, id_furnizor, id_departament

FROM produse WHERE pret >= parametru;

produc produse.id_produc%type;

nume produse.nume_produc%type;

pret produse.pret%type;

furnizor produse.id_furnizor%type;

departament produse.id_departament%type;

dep departamente.id_departament%type;

BEGIN

FOR i IN (SELECT id_departament FROM departamente) LOOP --ciclu cursor

OPEN c_prod(par);

DBMS_OUTPUT.PUT_LINE('DEPARTAMENTUL '||i.id_departament);

LOOP

```

FETCH c_prod INTO produs,nume,pret,furnizor,departament;

EXIT WHEN c_prod%NOTFOUND;

IF departament = i.id_departament THEN

    DBMS_OUTPUT.PUT_LINE(produs||' : ||nume||',pret = ||pret);

END IF;

END LOOP;

CLOSE c_prod;

END LOOP;

END;

```

The screenshot shows the SQL Developer interface. The main window displays a PL/SQL script with line numbers 267 to 301. The script defines a function FROD_DEP that iterates through departments and prints product details. The output window shows the results of the function, listing departments and their associated products with prices.

```

267 --Pentru fiecare departament afisati produsele care au pretul mai mare sau egal decat un parametru.
268
269 CREATE OR REPLACE PROCEDURE FROD_DEP (par number) AS
270 CURSOR c_prod (parametru NUMBER) IS
271 SELECT id_produs, nume_produs, pret, id_furnizor, id_departament
272 FROM produse WHERE pret >= parametru;
273
274 produs produse.id_produs%type;
275 nume produse.nume_produs%type;
276 pret produse.pret%type;
277 furnizor produse.id_furnizor%type;
278 departament produse.id_departament%type;
279 dep departamente.id_departament%type;
280
281 BEGIN
282 FOR i IN (SELECT id_departament FROM departamente) LOOP
283 OPEN c_prod(par);
284 DBMS_OUTPUT.PUT_LINE('DEPARTAMENTUL '||i.id_departament);
285 LOOP
286 FETCH c_prod INTO produs,nume,pret,furnizor,departament;
287 EXIT WHEN c_prod%NOTFOUND;
288 IF departament = i.id_departament THEN
289 DBMS_OUTPUT.PUT_LINE(produs||' : ||nume||',pret = ||pret);
290 END IF;
291 END LOOP;
292 CLOSE c_prod;
293 END LOOP;
294 END;
295
296 CALL prod_dep(100);
297 CALL prod_dep(0);
298
299
300
301

```

The output window shows the results of the function, listing departments and their associated products with prices.

```

DEPARTAMENTUL 160
DEPARTAMENTUL 161
202 : Robot de bucatarie,pret = 300
DEPARTAMENTUL 162
DEPARTAMENTUL 163
DEPARTAMENTUL 164
198 : Pantaloni scurti,pret = 100
199 : Pantaloni,pret = 120
DEPARTAMENTUL 165
DEPARTAMENTUL 166
203 : Puzzle,pret = 100
204 : Monopoly,pret = 200
205 : Tren de jucarie,pret = 100
206 : Papusa,pret = 100
DEPARTAMENTUL 167
168 : Scaun,pret = 100
385 : Lampa,pret = 100
190 : Masa,pret = 200
207 : Dulap,pret = 500
DEPARTAMENTUL 160
191 : Detergent vase,pret = 10
DEPARTAMENTUL 161
202 : Robot de bucatarie,pret = 300
DEPARTAMENTUL 162
DEPARTAMENTUL 163
192 : Apa,pret = 8
193 : Zahar,pret = 70
194 : Faina,pret = 5
DEPARTAMENTUL 164
198 : Pantaloni scurti,pret = 100
199 : Pantaloni,pret = 120
200 : Tricou,pret = 70
201 : Hanorac,pret = 90
DEPARTAMENTUL 165
196 : Mingea,pret = 80
196 : Paleta de tenis,pret = 80
197 : Fiecu,pret = 40
DEPARTAMENTUL 166
203 : Puzzle,pret = 100
204 : Monopoly,pret = 200
205 : Tren de jucarie,pret = 100
206 : Papusa,pret = 100
DEPARTAMENTUL 167
168 : Scaun,pret = 100
385 : Lampa,pret = 100
190 : Masa,pret = 200
207 : Dulap,pret = 500

```

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.

-- Pentru un angajat dat afisati departamentul la care lucreaza.

```
CREATE OR REPLACE FUNCTION ANG_DEP (nume_a angajati.nume%type,prenume_a angajati.prenume%type)
```

```
RETURN departamente.nume_departament%type IS
```

```
dep departamente.nume_departament%type;
```

```
ct number;
```

```
MAI_MULTE_DEPARTAMENTE EXCEPTION;
```

```
BEGIN
```

```

SELECT COUNT(*) INTO ct FROM angajati
WHERE nume = nume_a AND prenume = prenume_a;
IF ct = 0 THEN
    RAISE NO_DATA_FOUND;
END IF;

SELECT nume_department INTO dep FROM departamente
WHERE id_department = (
    SELECT max(id_department) FROM angajati_departmente
    WHERE id_angajat = (
        SELECT id_angajat FROM angajati
        WHERE upper(nume) = upper(nume_a) AND upper(prenume) =
upper(prenume_a)
    )
);

SELECT COUNT(*) INTO ct FROM angajati_departmente
WHERE id_angajat = (
    SELECT id_angajat FROM angajati
    WHERE upper(nume) = upper(nume_a) AND upper(prenume) =
upper(prenume_a)
);

IF ct > 1 THEN
    RAISE MAI_MULTE_DEPARTAMENTE;
ELSE
    RETURN dep;
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Nu exista angajatul');
    WHEN MAI_MULTE_DEPARTAMENTE THEN
        RAISE_APPLICATION_ERROR(-20002,'Angajatul lucreaza la mai multe departamente,printre
care si '||dep);
    WHEN OTHERS THEN

```

```

        RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');
END;

/

BEGIN

    DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','Daniel')); -- Angajatul lucreaza la mai multe
departamente,printre care si Sport
END;

/

BEGIN

    DBMS_OUTPUT.PUT_LINE(ang_dep('Popescu','Diana')); -- Haine
END;

/

BEGIN

    DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','')); -- Nu exista angajatul
END;

```

```

105
106 -- Pentru un angajat dat afisati departamentul la care lucreaza.
107 CREATE OR REPLACE FUNCTION ANG_DEP (nume_a angajati.numetype,prenume_a angajati.prenumetype)
108 RETURN departamente.numetype IS
109     dep departamente.numetype;
110     ct number;
111     MAI_MULTE_DEPARTAMENTE EXCEPTION;
112 BEGIN
113     SELECT nume_departament INTO dep FROM departamente
114     WHERE id_departament = (
115         SELECT max(id_departament) FROM angajati_departamente
116         WHERE id_angajat = (
117             SELECT id_angajat FROM angajati
118             WHERE upper(nume) = upper(nume_a) AND upper(prenume) = upper(prenume_a)
119         )
120     );
121     SELECT COUNT(*) INTO ct FROM angajati_departamente
122     WHERE id_angajat = (
123         SELECT id_angajat FROM angajati
124         WHERE upper(nume) = upper(nume_a) AND upper(prenume) = upper(prenume_a)
125     );
126 IF ct > 1 THEN
127     RAISE MAI_MULTE_DEPARTAMENTE;
128 ELSE
129     RETURN dep;
130 END IF;
131 EXCEPTION
132     WHEN NO_DATA_FOUND THEN
133         RAISE_APPLICATION_ERROR(-20001,'Nu exista angajatul');
134     WHEN MAI_MULTE_DEPARTAMENTE THEN
135         RAISE_APPLICATION_ERROR(-20002,'Angajatul lucreaza la mai multe departamente,printre care si '||dep);
136     WHEN OTHERS THEN
137         RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');
138 END;
139 /
140
141 BEGIN
142     DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','Daniel')); -- Angajatul lucreaza la mai multe departamente,printre care si Sport

```

Script Output x Query Result x

Task completed in 0.059 seconds

Function ANG_DEP compiled

SQL Worksheet History

Worksheet Query Builder

```

140 BEGIN
141     DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','Daniel')); -- Angajatul lucreaza la mai multe departamente,printre care si Sport
142 END;
143
144 BEGIN
145     DBMS_OUTPUT.PUT_LINE(ang_dep('Popescu','Diana')); -- Haine
146 END;
147 /
148 BEGIN
149     DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','')); -- Nu exista angajatul
150 END;
151 /
152
153
154
155

```

Script Output x Query Result x

Task completed in 0.059 seconds

Function ANG_DEP compiled

Error starting at line : 141 in command -

```

BEGIN
    DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','Daniel')); -- Angajatul lucreaza la mai multe departamente,printre care si Sport
END;
Error report -
ORA-20002: Angajatul lucreaza la mai multe departamente,printre care si Sport
ORA-06512: at "ANTONIA.DBA.ANG_DEP", line 29
ORA-06512: at line 2

```

PL/SQL procedure successfully completed.

Error starting at line : 149 in command -

```

BEGIN
    DBMS_OUTPUT.PUT_LINE(ang_dep('Popa','')); -- Nu exista angajatul
END;
Error report -
ORA-20001: Nu exista angajatul
ORA-06512: at "ANTONIA.DBA.ANG_DEP", line 27
ORA-06512: at line 2

```

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.

--Pentru un furnizor dat afisati un client care a cumparat unul din produsele ce apartin furnizorul respectiv.

```
CREATE OR REPLACE PROCEDURE FURN_CL (nume_f furnizori.nume_furnizor%type) IS
```

```
    ct number;
```

```
    cli clienti.nume%type;
```

```
    furnizor furnizori.id_furnizor%type;
```

```
    ACELASI_NUME EXCEPTION;
```

```
    NU_EXISTA_PRODUSE EXCEPTION;
```

```
    NU_EXISTA_COMENZI EXCEPTION;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO ct FROM furnizori
```

```
    WHERE nume_furnizor = nume_f;
```

```
    IF ct>1 THEN
```

```
        RAISE ACELASI_NUME;
```

```
    END IF;
```

```
    SELECT id_furnizor INTO furnizor FROM furnizori
```

```
    WHERE nume_furnizor = nume_f;
```

```
    SELECT COUNT(*) INTO ct FROM produse
```

```
    WHERE id_furnizor = furnizor;
```

```
    IF ct=0 THEN
```

```
        RAISE NU_EXISTA_PRODUSE;
```

```
    END IF;
```

```
    SELECT count(*) INTO ct FROM produse_comenzi
```

```
    WHERE id_produs in (
```

```
        SELECT id_produs FROM produse
```

```

        WHERE id_furnizor = furnizor
    );

IF ct=0 THEN
    RAISE NU_EXISTA_COMENZI;
END IF;

-- Pentru a arunca TOO_MANY_ROWS am cautat numele persoanei dupa id, apoi am cautat id-ul dupa
nume, fapt ce mi-a returnat mai multe id-uri,
-- apoi am introdus in cli numele angajatului cu id-ul dat , iar in cli pot insera doar unul.
SELECT nume INTO cli FROM clienti
WHERE id_client in (
    SELECT id_client FROM clienti
    where nume = (
        SELECT nume FROM clienti
        WHERE id_client in (
            SELECT max(id_client) FROM comenzi
            WHERE id_comanda in (
                SELECT id_comanda FROM produse_comenzi
                WHERE id_produc in (
                    SELECT id_produc FROM produse
                    WHERE id_furnizor = (
                        SELECT id_furnizor
FROM furnizori
                                WHERE
upper(nume_furnizor) = upper(nume_f)
                                )
                    )
                )
            )
        )
    )
    );

DBMS_OUTPUT.PUT_LINE('Un produs de la furnizorul: '||nume_f||' a fost achizitionat de '||cli);

```



```

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Nu exista furnizorul!');
    WHEN NU_EXISTA_PRODUSE THEN
        RAISE_APPLICATION_ERROR(-20002,'Nu exista produse ce apartin furnizorului dat ca
parametru!');
    WHEN NU_EXISTA_COMENZI THEN
        RAISE_APPLICATION_ERROR(-20003,'Nu exista comenzi ce contin produse ce apartin
furnizorului dat ca parametru!');
    WHEN ACELASI_NUME THEN
        RAISE_APPLICATION_ERROR(-20004,'Exista mai multi furnizori cu acest nume!');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20006,'Mai multi clienti cu acelasi nume!');

END;

/

BEGIN
    FURN_CL('furnizor1'); -- Mai multi clienti cu acelasi nume!
END;

/

BEGIN
    FURN_CL('furnizor'); -- Nu exista furnizorul!
END;

/

BEGIN
    FURN_CL('furnizor5'); -- Popescu
END;

/

BEGIN
    FURN_CL('furnizor4'); -- Exista mai multi furnizori cu acest nume!

```

```
END;
```

```
/
```

```
BEGIN
```

```
    FURN_CL('furnizor6'); --Nu exista produse ce apartin furnizorului dat ca parametru!
```

```
END;
```

```
/
```

```
BEGIN
```

```
    FURN_CL('furnizor7'); -- Nu exista comenzi ce contin produse ce apartin furnizorului dat ca parametru!
```

```
END;
```

```
/
```

Worksheet

Query Builder

```

364 --Pentru un furnizor dat arisati un client care a cumparat unul din produsele ce apartin furnizorului respectiv.
365 CREATE OR REPLACE PROCEDURE FURN_CL (nume_f furnizori.nume_furnizor%type) IS
366     ct number;
367     cli clienti.nume%type;
368     furnizor furnizori.id_furnizor%type;
369     ACELASI_NUME EXCEPTION;
370     NU_EXISTA_PRODUSE EXCEPTION;
371     NU_EXISTA_COMENZI EXCEPTION;
372 BEGIN
373     SELECT COUNT(*) INTO ct FROM furnizori
374     WHERE nume_furnizor = nume_f;
375
376     IF ct>1 THEN
377         RAISE ACELASI_NUME;
378     END IF;
379
380     SELECT id_furnizor INTO furnizor FROM furnizori
381     WHERE nume_furnizor = nume_f;
382
383     SELECT COUNT(*) INTO ct FROM produse
384     WHERE id_furnizor = furnizor;
385
386     IF ct=0 THEN
387         RAISE NU_EXISTA_PRODUSE;
388     END IF;
389
390     SELECT count(*) INTO ct FROM produse_comenzi
391     WHERE id_produs in (
392         SELECT id_produs FROM produse
393         WHERE id_furnizor = furnizor
394     );
395
396     IF ct=0 THEN
397         RAISE NU_EXISTA_COMENZI;
398     END IF;
399
400     -- Pentru a arunca TOO_MANY_ROWS am cautat numele persoanei dupa id, apoi am cautat id-ul dupa nume, fapt ce mi-a returnat mai multe id-uri,
401     -- apoi am introdus in cli numele angajatului cu id-ul dat , iar in cli pot insera doar unul.
402     SELECT nume INTO cli FROM clienti
403     WHERE id_client in (

```

Script Output x

Task completed in 0.094 seconds

Procedure FURN_CL compiled

```
Script Output x
Task completed in 0.107 seconds

Error starting at line : 439 in command -
BEGIN
    FURN_CL('furnizor1'); -- Mai multi clienti cu acelasi nume!
END;
Error report -
ORA-20006: Mai multi clienti cu acelasi nume!
ORA-06512: at "ANTONIA_DBA.FURN_CL", line 70
ORA-06512: at line 2

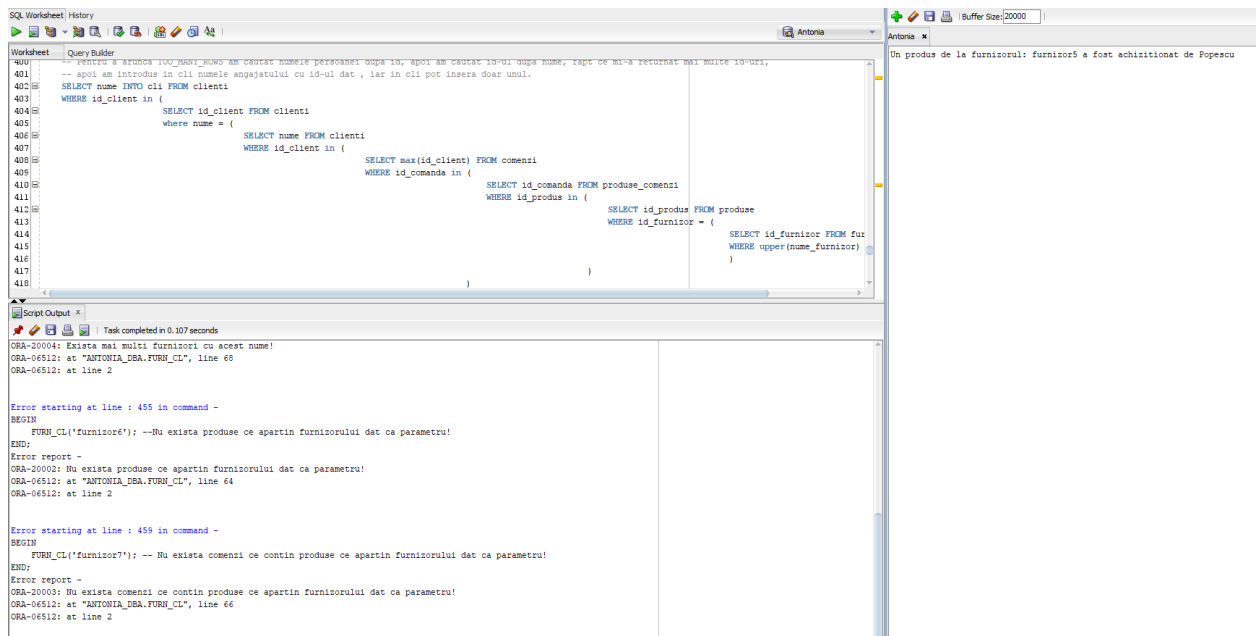
Error starting at line : 443 in command -
BEGIN
    FURN_CL('furnizor'); -- Nu exista furnizorul!
END;
Error report -
ORA-20001: Nu exista furnizorul!
ORA-06512: at "ANTONIA_DBA.FURN_CL", line 62
ORA-06512: at line 2

PL/SQL procedure successfully completed.

Error starting at line : 451 in command -
BEGIN
    FURN_CL('furnizor4'); -- Exista mai multi furnizori cu acest nume!
END;
Error report -
ORA-20004: Exista mai multi furnizori cu acest nume!
ORA-06512: at "ANTONIA_DBA.FURN_CL", line 68
ORA-06512: at line 2

Error starting at line : 455 in command -
BEGIN
    FURN_CL('furnizor6'); --Nu exista produse ce apartin furnizorului dat ca parametru!
END;
Error report -
ORA-20002: Nu exista produse ce apartin furnizorului dat ca parametru!
ORA-06512: at "ANTONIA_DBA.FURN_CL", line 64
ORA-06512: at line 2

v/database/technology/express-edition/overview/index.html
```



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

--NU SE PERMITE MODIFICAREA TABELULUI DEPARTAMENTE

CREATE OR REPLACE TRIGGER DEP_UPDATE

BEFORE UPDATE ON DEPARTAMENTE

BEGIN

RAISE_APPLICATION_ERROR(-20005,'Nu poti modifica tabelul departamente!');

END;

/

--UPDATE departamente SET nume_departament='IT' WHERE id_departament = 160;

```
49
50 --NU SE PERMITE MODIFICARE TABELULUI DEPARTAMENTE
51
52 CREATE OR REPLACE TRIGGER DEP_UPDATE
53 BEFORE UPDATE ON DEPARTAMENTE
54 BEGIN
55     RAISE_APPLICATION_ERROR(-20005,'Nu poti modifica tabelul departamente!');
56 END;
57 /
58
59 UPDATE departamente SET nume_departament='IT' WHERE id_departament = 160;
60
61
62
```

Script Output x

Task completed in 0.027 seconds

Trigger DEP_UPDATE compiled

Error starting at line : 59 in command -
UPDATE departamente SET nume_departament='IT' WHERE id_departament = 160
Error report -
ORA-20005: Nu poti modifica tabelul departamente!
ORA-06512: at "ANTONIA_DBA.DEP_UPDATE", line 2
ORA-04088: error during execution of trigger 'ANTONIA_DBA.DEP_UPDATE'

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

```

9
10 -- NU PERMITE MODIFICAREA COLOANELOR id_produș, nume_produș, id_departament, id_furnizor
11 -- DIN TABELUL PRODUSE SI NU PERMITE STERGEREA UNUI PRODUS
12
13 CREATE OR REPLACE TRIGGER PROD_UPDATE_DELETE
14 BEFORE UPDATE OF id_produș, nume_produș, id_departament, id_furnizor OR DELETE ON produse
15 FOR EACH ROW
16 BEGIN
17     IF UPDATING THEN
18         RAISE_APPLICATION_ERROR(-20001, 'Modificarea nu poate fi realizata!');
19     ELSE
20         RAISE_APPLICATION_ERROR(-20002, 'Nu poti sterge un produs! Poti modifica stocul produsului!');
21     END IF;
22 END;
23 /
24
25 UPDATE produse SET id_furnizor=101 WHERE id_produș=122;
26 DELETE FROM produse WHERE ID_PRODUS=122;
27
28
29

```

Script Output x

Task completed in 0.034 seconds

Trigger PROD_UPDATE_DELETE compiled

Error starting at line : 25 in command -

UPDATE produse SET id_furnizor=101 WHERE id_produș=122

Error report -

ORA-20001: Modificarea nu poate fi realizata!

ORA-06512: at "ANTONIA_DBA.PROD_UPDATE_DELETE", line 3

ORA-04088: error during execution of trigger 'ANTONIA_DBA.PROD_UPDATE_DELETE'

Error starting at line : 26 in command -

DELETE FROM produse WHERE ID_PRODUS=122

Error report -

ORA-20002: Nu poti sterge un produs! Poti modifica stocul produsului!

ORA-06512: at "ANTONIA_DBA.PROD_UPDATE_DELETE", line 5

ORA-04088: error during execution of trigger 'ANTONIA_DBA.PROD_UPDATE_DELETE'

```

-- NU PERMITE MODIFICAREA COLOANELOR id_produș,nume_produș,id_departament,id_furnizor
-- DIN TABELUL PRODUSE SI NU PERMITE STERGEREA UNUI PRODUS

```

```

CREATE OR REPLACE TRIGGER PROD_UPDATE_DELETE

```

```

BEFORE UPDATE OF id_produș,nume_produș,id_departament,id_furnizor OR DELETE ON produse
FOR EACH ROW

```

```

DECLARE

```

```

    nr INT:=null;

```

```

BEGIN

```

```

IF UPDATING THEN

    RAISE_APPLICATION_ERROR(-20001,'Modificarea nu poate fi realizata!');

ELSE

    RAISE_APPLICATION_ERROR(-20002,'Nu poti sterge un produs!Poti modifica stocul
produsului!');

END IF;

END;

/

```

```

29
30
31 -- NU PERMITE ADAUGAREA UNUI PRODUS DEJA EXISTENT
32 CREATE OR REPLACE TRIGGER PROD_INSERT
33 BEFORE INSERT ON produse
34 FOR EACH ROW
35 DECLARE
36     nr int;
37 BEGIN
38     SELECT COUNT(*) INTO nr
39     FROM produse
40     WHERE nume_produc = :NEW.nume_produc;
41     IF nr <> 0 THEN
42         RAISE_APPLICATION_ERROR(-20002,'Produsul exista deja!Poti modifica stocul produsului!');
43     END IF;
44 END;
45 /
46
47 INSERT INTO produse VALUES (secd.nextval,'Scaun',120,3,167,147);
48
49

```

Script Output x

Task completed in 0.04 seconds

Trigger PROD_INSERT compiled

Error starting at line : 47 in command -

INSERT INTO produse VALUES (secd.nextval,'Scaun',120,3,167,147)

Error report -

ORA-20002: Produsul exista deja!Poti modifica stocul produsului!

ORA-06512: at "ANTONIA_DBA.PROD_INSERT", line 8

ORA-04088: error during execution of trigger 'ANTONIA_DBA.PROD_INSERT'

```

-- NU PERMITE ADAUGAREA UNUI PRODUS DEJA EXISTENT

CREATE OR REPLACE TRIGGER PROD_INSERT

BEFORE INSERT ON produse

```


FOR EACH ROW

DECLARE

nr int;

BEGIN

SELECT COUNT(*) INTO nr

FROM produse

WHERE nume_produs = :NEW.nume_produs;

IF nr <> 0 THEN

RAISE_APPLICATION_ERROR(-20002,'Produsul exista deja!Poti modifica stocul produsului!');

END IF;

END;

/

```
63
64 --DACA LA INSERAREA UNUI ANGAJAT ID_POST NU ESTE NULL VERIFICAM DACA SALARIUL ANGAJATULUI ESTE MAI MARE DECAT
65 --SALARIUL MINIM PENTRU POSTUL RESPECTIV
66 --UN ANGAJAT NOU NU POATE FI MANAGER
67 CREATE OR REPLACE TRIGGER ANG_MAN_POST
68 BEFORE INSERT ON angajati
69 FOR EACH ROW
70 DECLARE
71     salariu_post posturi.salariu_minim%type;
72 BEGIN
73     IF :NEW.id_post = 5 THEN
74         RAISE_APPLICATION_ERROR(-20007,'Un angajat nou nu poate fi manager!');
75     END IF;
76     IF :NEW.id_post >= 1 AND :NEW.id_post <= 5
77     THEN
78         SELECT salariu_minim INTO salariu_post FROM posturi WHERE id_post = :NEW.id_post;
79         IF salariu_post > :NEW.salariu THEN
80             RAISE_APPLICATION_ERROR(-20007,'Angajatul trebuie sa aiba salariul mai mare de '||salariu_post||');
81         END IF;
82     END IF;
83 END;
84 /
85 insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secdv.nextval,230,'Mihai','Ioana',4,TO_DATE('2016-08-17','YYYY-MM-DD'),1000);
86 insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secdv.nextval,230,'Mihai','Ioana',5,TO_DATE('2016-08-17','YYYY-MM-DD'),1000);
```

Script Output x

Task completed in 0.048 seconds

Trigger ANG_MAN_POST compiled

Error starting at line : 85 in command -

```
insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secdv.nextval,230,'Mihai','Ioana',4,TO_DATE('2016-08-17','YYYY-MM-DD'),1000)
```

Error report -

```
ORA-20007: Angajatul trebuie sa aiba salariul mai mare de 1500!
ORA-06512: at "ANTONIA_DBA.ANG_MAN_POST", line 11
ORA-04088: error during execution of trigger 'ANTONIA_DBA.ANG_MAN_POST'
```

Error starting at line : 86 in command -

```
insert into angajati (id_angajat, id_manager,nume,prenume,id_post,data_angajare,salariu) values (secdv.nextval,230,'Mihai','Ioana',5,TO_DATE('2016-08-17','YYYY-MM-DD'),1000)
```

Error report -

```
ORA-20007: Un angajat nou nu poate fi manager!
ORA-06512: at "ANTONIA_DBA.ANG_MAN_POST", line 5
ORA-04088: error during execution of trigger 'ANTONIA_DBA.ANG_MAN_POST'
```

--DACA LA INSERAREA UNUI ANGAJAT ID_POST NU ESTE NULL VERIFICAM DACA
SALARIUL ANGAJATULUI ESTE MAI MARE DECAT

--SALARIUL MINIM PENTRU POSTUL RESPECTIV

--UN ANGAJAT NOU NU POATE FI MANAGER

```

CREATE OR REPLACE TRIGGER ANG_MAN_POST
BEFORE INSERT ON angajati
FOR EACH ROW
DECLARE
    salariu_post posturi.salariu_minim%type;
BEGIN
    IF :NEW.id_post = 5 THEN
        RAISE_APPLICATION_ERROR(-20007,'Un angajat nou nu poate fi manager!');
    END IF;
    IF :NEW.id_post >=1 AND :NEW.id_post<=5
    THEN
        SELECT salariu_minim INTO salariu_post FROM posturi WHERE id_post = :NEW.id_post;
        IF salariu_post > :NEW.salariu THEN
            RAISE_APPLICATION_ERROR(-20007,'Angajatul trebuie sa aiba salariul mai mare de
            '||salariu_post||'!');
        END IF;
    END IF;
END;
/

```



```

89
90 CREATE OR REPLACE TRIGGER TOTAL_COM
91 AFTER INSERT ON produse_comenzi
92 FOR EACH ROW
93 DECLARE
94     pret_p produse.pret%type;
95     stoc_p produse.stoc%type;
96 BEGIN
97     select stoc into stoc_p from produse where id_produs=:NEW.id_produs;
98     IF stoc_p >= :NEW.nr_bucati THEN
99         SELECT pret INTO pret_p FROM produse WHERE id_produs=:NEW.id_produs;
100         UPDATE comenzi SET total_plata =total_plata + pret_p * :NEW.nr_bucati WHERE id_comanda=:NEW.id_comanda;
101         UPDATE PRODUSE SET stoc = stoc - :NEW.nr_bucati WHERE id_produs=:NEW.id_produs;
102     ELSE
103         RAISE_APPLICATION_ERROR (-20008,'Nu exista suficiente produse in stoc');
104     END IF;
105 END;
106
107 insert into produse_comenzi values (secv.nextval,208,203,3);
108 insert into produse_comenzi values (secv.nextval,208,207,2);
109
110 select * from produse_comenzi;
111 select * from comenzi;

```

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

SQL | All Rows Fetched: 22 in 0.002 seconds

ID_COMANDA	ID_CLIENT	ADRESA	TOTAL_PLATA
1	208	152 Bucuresti	1000
2	209	152 Bucuresti	0
3	210	152 Brasov	0
4	211	152 Pitesti	0
5	212	154 Bucuresti	0
6	213	154 Sibiu	0
7	214	153 Sibiu	0
8	215	156 Bucuresti	0

--LA INSERAREA UNUI PRODUS IN COMENZI SA SE REACTUALIZEZE PRETUL TOTAL DIN COMENZI SI STOCUL DIN PRODUSE IN CAZUL IN CARE EXISTA SUFICIENTE PRODUSE IN STOC

CREATE OR REPLACE TRIGGER TOTAL_COM

AFTER INSERT ON produse_comenzi

FOR EACH ROW

DECLARE

pret_p produse.pret%type;

stoc_p produse.stoc%type;

BEGIN

select stoc into stoc_p from produse where id_produs=:NEW.id_produs;

IF stoc_p >= :NEW.nr_bucati THEN

SELECT pret INTO pret_p FROM produse WHERE id_produs=:NEW.id_produs;

UPDATE comenzi SET total_plata =total_plata + pret_p * :NEW.nr_bucati WHERE id_comanda=:NEW.id_comanda;

UPDATE PRODUSE SET stoc = stoc - :NEW.nr_bucati WHERE id_produs=:NEW.id_produs;

```

ELSE

    RAISE_APPLICATION_ERROR (-20008,'Nu exista suficiente produse in stoc');

END IF;

END;

```

```

114
115 --DUPA INSERAREA UNUI RETUR REACTUALIZAM STOCUL PRODUSULUI RETURNAT DACA SE POATE ADAUGA IN STOC
116 CREATE OR REPLACE TRIGGER RET_STOC
117 AFTER INSERT ON retururi
118 FOR EACH ROW
119 DECLARE
120     nr_prod produse_comenzi.nr_bucati%type;
121     prod produse_comenzi.id_produs%type;
122 BEGIN
123     IF :NEW.status = 1 THEN
124         SELECT nr_bucati INTO nr_prod FROM produse_comenzi WHERE id_produs_comanda = :NEW.id_produs_comanda;
125         SELECT id_produs INTO prod FROM produse_comenzi WHERE id_produs_comanda = :NEW.id_produs_comanda;
126         UPDATE produse SET stoc = stoc + nr_prod WHERE id_produs = prod;
127     END IF;
128 END;
129
130 INSERT INTO RETURURI VALUES (secv.nextval,304,1); --s-au adaugat 2 dulapuri in stoc
131
132 select * from retururi; select * from produse;
133

```

Script Output x | Query Result 4 x | Query Result 5 x

SQL | All Rows Fetched: 19 in 0.001 seconds

ID_PRODUS	NUME_PRODUS	PRET	STOC	ID_DEPARTAMENT	ID_FURNIZOR
4	192 Apa	8	13	163	149
5	193 Zahar	5	5	163	150
6	194 Faina	5	11	163	147
7	195 Minge	50	7	165	148
8	196 Paleta de tenis	80	2	165	149
9	197 Fileu	40	5	165	150
10	198 Pantaloni scurti	100	0	164	151
11	199 Pantaloni	120	2	164	147
12	200 Tricou	70	3	164	148
13	201 Hanorac	90	3	164	149
14	202 Robot de bucatarie	300	2	161	150
15	203 Puzzle	100	0	166	147
16	204 Monopoly	200	0	166	147
17	205 Tren de jucarie	100	2	166	149
18	206 Papusa	100	3	166	149
19	207 Dulap	500	2	167	151

SQL Developer\trigger.sql

```

--DUPA INSERAREA UNUI RETUR REACTUALIZAM STOCUL PRODUSULUI RETURNAT DACA
SE POATE ADAUGA IN STOC

```

```

CREATE OR REPLACE TRIGGER RET_STOC

AFTER INSERT ON retururi

FOR EACH ROW

```

```

DECLARE

    nr_prod produse_comenzi.nr_bucati%type;

    prod produse_comenzi.id_produș%type;

BEGIN

    IF :NEW.status = 1 THEN

        SELECT nr_bucati INTO nr_prod FROM produse_comenzi WHERE id_produș_comanda =
:NEW.id_produș_comanda;

        SELECT id_produș INTO prod FROM produse_comenzi WHERE id_produș_comanda =
:NEW.id_produș_comanda;

        UPDATE produse SET stoc = stoc + nr_prod WHERE id_produș = prod;

    END IF;

END;

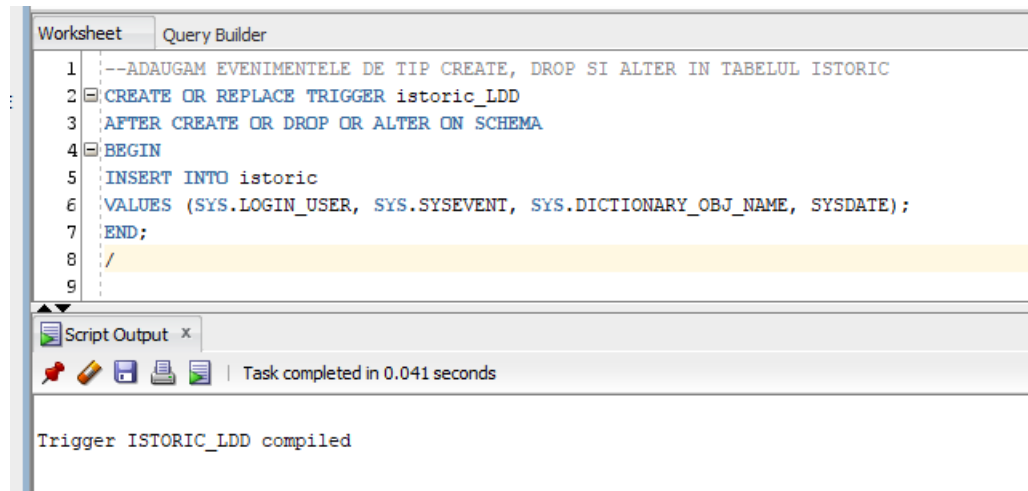
```

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

```

--ADAUGAM EVENIMENTELE DE TIP CREATE, DROP SI ALTER IN TABELUL ISTORIC
CREATE OR REPLACE TRIGGER istoric_LDD
AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
INSERT INTO istoric
VALUES (SYS.LOGIN_USER, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME, SYSDATE);
END;
/

```



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

```

CREATE OR REPLACE PACKAGE pachet1

AS

    PROCEDURE ANG_SAL (man angajati.id_manager%type); --ex6

    PROCEDURE PROD_DEP(par number); --ex7

```

```
FUNCTION ANG_DEP (nume_a angajati.nume%type,prenume_a angajati.prenume%type) RETURN  
departamente.nume_departament%type; --ex8
```

```
PROCEDURE FURN_CL (nume_f furnizori.nume_furnizor%type); --ex9
```

```
END pachet1;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY pachet1 AS
```

```
--Pentru un manager dat ca parametru afisati pentru angajatii pe care ii are in subordine salariile pe care
```

```
--le primesc acestia in fiecare luna a anului. Daca id-ul nu corespunde unui manager afisati un mesaj  
semnificativ.
```

```
PROCEDURE ANG_SAL (man angajati.id_manager%type) AS
```

```
TYPE vector IS VARRAY(13) OF NUMBER;
```

```
ang vector := vector();
```

```
TYPE tablou_indexat IS TABLE OF luni%rowtype
```

```
INDEX BY BINARY_INTEGER;
```

```
t tablou_indexat;
```

```
nume angajati.nume%type;
```

```
prenume angajati.prenume%type;
```

```
salariu angajati.salariu%type;
```

```
salariu_luna angajati.salariu%type;
```

```
ok number:=0;
```

```
BEGIN
```

```
SELECT COUNT(*) INTO ok FROM angajati WHERE id_manager = man;
```

```
IF ok = 0 THEN
```

```
DBMS_OUTPUT.PUT_LINE('ID-ul dat nu corespunde unui manager!');
```

```
ELSE
```

```
SELECT id_angajat BULK COLLECT INTO ang FROM angajati WHERE id_manager=man;
```

```
SELECT * BULK COLLECT INTO t FROM luni;
```

```

        FOR j IN ang.first..ang.last LOOP
            SELECT nume,prenume,trunc(salariu) to _int INTO nume,prenume,salariu FROM angajati
            WHERE id_angajat = ang(j);

            FOR i IN 1..12 LOOP

                salariu_luna := salariu+salariu*t(i).bonus;

                DBMS_OUTPUT.PUT_LINE(nume||' '||prenume||' va incasa in luna '||t(i).nume_luna||'
                '||salariu_luna);

            END LOOP;

        END LOOP;

    END IF;

END ANG_SAL;

```

--Pentru fiecare departament afisati produsele care au pretul mai mare sau egal decat un parametru.

```

PROCEDURE PROD_DEP(par number) AS

    CURSOR c_prod (parametru NUMBER) IS

        SELECT id_produș, nume_produș, pret, id_furnizor, id_departament

        FROM produse WHERE pret >= parametru;

    produș produse.id_produș%type;

    nume produse.nume_produș%type;

    pret produse.pret%type;

    furnizor produse.id_furnizor%type;

    departament produse.id_departament%type;

    dep departamente.id_departament%type;

BEGIN

    FOR i IN (SELECT id_departament FROM departamente)LOOP

        OPEN c_prod(par);

        DBMS_OUTPUT.PUT_LINE('DEPARTAMENTUL '||i.id_departament);

        LOOP

            FETCH c_prod INTO produș,nume,pret,furnizor,departament;

            EXIT WHEN c_prod%NOTFOUND;

```



```

        IF departament = i.id_departament THEN
            DBMS_OUTPUT.PUT_LINE(produs||' : '||nume||',pret = '||pret);
        END IF;
    END LOOP;

    CLOSE c_prod;
END LOOP;
END PROD_DEP;

```

-- Pentru un angajat dat afisati departamentul la care lucreaza.

```

FUNCTION ANG_DEP (nume_a angajati.nume%type,prenume_a angajati.prenume%type)
RETURN departamente.nume_departament%type IS
    dep departamente.nume_departament%type;
    ct number;

    MAI_MULTE_DEPARTAMENTE EXCEPTION;
BEGIN
    SELECT nume_departament INTO dep FROM departamente
    WHERE id_departament = (
        SELECT max(id_departament) FROM angajati_departamente
        WHERE id_angajat = (
            SELECT id_angajat FROM angajati
            WHERE upper(nume) = upper(nume_a) AND upper(prenume) =
upper(prenume_a)
        )
    );

    SELECT COUNT(*) INTO ct FROM angajati_departamente
    WHERE id_angajat = (
        SELECT id_angajat FROM angajati
        WHERE upper(nume) = upper(nume_a) AND upper(prenume) =
upper(prenume_a)
    );

    IF ct > 1 THEN

```

```

        RAISE MAI_MULTE_DEPARTAMENTE;
ELSE
    RETURN dep;
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Nu exista angajatul');
    WHEN MAI_MULTE_DEPARTAMENTE THEN
        RAISE_APPLICATION_ERROR(-20002,'Angajatul lucreaza la mai multe departamente,printre
care si '||dep);
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');
END ANG_DEP;

```

--Pentru un furnizor dat afisati un client care a cumparat unul din produsele ce apartin furnizorul respectiv.

```

PROCEDURE FURN_CL (nume_f furnizori.nume_furnizor%type) IS
    ct number;
    cli clienti.nume%type;
    furnizor furnizori.id_furnizor%type;
    ACELASI_NUME EXCEPTION;
    NU_EXISTA_PRODUSE EXCEPTION;
    NU_EXISTA_COMENZI EXCEPTION;
BEGIN
    SELECT COUNT(*) INTO ct FROM furnizori
    WHERE nume_furnizor = nume_f;

    IF ct>1 THEN
        RAISE ACELASI_NUME;
    END IF;

    SELECT id_furnizor INTO furnizor FROM furnizori

```

```
WHERE nume_furnizor = nume_f;
```

```
SELECT COUNT(*) INTO ct FROM produse  
WHERE id_furnizor = furnizor;
```

```
IF ct=0 THEN  
    RAISE NU_EXISTA_PRODUSE;  
END IF;
```

```
SELECT count(*) INTO ct FROM produse_comenzi  
WHERE id_produs in (  
    SELECT id_produs FROM produse  
    WHERE id_furnizor = furnizor  
);
```

```
IF ct=0 THEN  
    RAISE NU_EXISTA_COMENZI;  
END IF;
```

-- Pentru a arunca TOO_MANY_ROWS am cautat numele persoanei dupa id, apoi am cautat id-ul
dupa nume, fapt ce mi-a returnat mai multe id-uri,

-- apoi am introdus in cli numele angajatului cu id-ul dat , iar in cli pot insera doar unul.

```
SELECT nume INTO cli FROM clienti  
WHERE id_client in (  
    SELECT id_client FROM clienti  
    where nume = (  
        SELECT nume FROM clienti  
        WHERE id_client in (  
            SELECT max(id_client) FROM comenzi  
            WHERE id_comanda in (  
                SELECT id_comanda FROM produse_comenzi  
                WHERE id_produs in (  

```

```

                                SELECT id_produs FROM
produse
                                WHERE id_furnizor = (
                                SELECT
id_furnizor FROM furnizori
                                WHERE
upper(ume_furnizor) = upper(ume_f)
                                )
                                )
                                )
                                )
                                )
                                );
DBMS_OUTPUT.PUT_LINE('Un produs de la furnizorul: '||ume_f||' a fost achizitionat de '||cli);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Nu exista furnizorul!');
    WHEN NU_EXISTA_PRODUSE THEN
        RAISE_APPLICATION_ERROR(-20002,'Nu exista produse ce apartin furnizorului dat ca
parametru!');
    WHEN NU_EXISTA_COMENZI THEN
        RAISE_APPLICATION_ERROR(-20003,'Nu exista comenzi ce contin produse ce apartin
furnizorului dat ca parametru!');
    WHEN ACELASI_NUME THEN
        RAISE_APPLICATION_ERROR(-20004,'Exista mai multi furnizori cu acest nume!');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20006,'Mai multi clienti cu acelasi nume!');

END FURN_CL;

END pachet1;
/

```

```
--testam pachetul
```

```
CALL pachet1.ang_sal(288);
```

```
CALL pachet1.prod_dep(100);
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE(pachet1.ang_dep('Popescu','Diana')); -- Haine
```

```
END;
```

```
/
```

```
BEGIN
```

```
    pachet1.FURN_CL('furnizor5'); -- Popescu
```

```
END;
```

```
/
```



```

        pret_prod produse.pret%type);

PROCEDURE UPD_MAN(ang angajati.id_angajat%type);
PROCEDURE ALL_MAN;
PROCEDURE COM_PR (comanda comenzi.id_comanda%type);

FUNCTION RET_NR(comanda comenzi.id_comanda%type) RETURN NUMBER;
FUNCTION RET_PROD RETURN NUMBER;
FUNCTION GET_NUME_PRODUS(pr produse.id_produs%type) RETURN
produse.ume_produs%type;

END pachet2;

/

CREATE OR REPLACE PACKAGE BODY pachet2 AS
-- setati postul MANAGER pentru angajati
PROCEDURE UPD_MAN(ang angajati.id_angajat%type) AS
BEGIN
    UPDATE angajati SET id_post = 5 WHERE id_angajat = ang;
END UPD_MAN;

-- pentru toti angajati care apar ca si id_manager pentru un angajat setati postul MANAGER
PROCEDURE ALL_MAN AS
    emp tablou_indexat;
BEGIN
    SELECT * BULK COLLECT INTO emp FROM angajati;

    FOR i IN 1..emp.last LOOP
        IF emp(i).id_manager IS NOT NULL THEN

```

```

        UPD_MAN(emp(i).id_manager);
    END IF;
END LOOP;
END ALL_MAN;

-- returnati numarul de produse comandate pentru o comanda data ca parametru
FUNCTION RET_NR(comanda comenzi.id_comanda%TYPE ) RETURN NUMBER AS
    ct NUMBER:=0;
BEGIN
    SELECT SUM(nr_bucati) INTO ct FROM produse_comenzi
    WHERE id_comanda = comanda;

    return ct;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20020,'Nu exista comanda cu numarul '||comanda||'!');
END RET_NR;

-- returnati produsul pentru care au fost comandate cele mai multe exemplare
-- daca sunt mai multe afisati-l pe oricare dintre ele
FUNCTION RET_PROD RETURN NUMBER AS
    pr PRODUS;
    id_pr produse.id_produs%type;
    maxim NUMBER;
BEGIN

    SELECT * INTO id_pr,maxim
    FROM (SELECT id_produs,SUM(nr_bucati) FROM produse_comenzi
        GROUP BY id_produs
        ORDER BY SUM(nr_bucati) DESC
    )
    WHERE ROWNUM=1;

```



```

        RETURN id_pr;
    END RET_PROD;

-- returnati numele unui produs in functie de id-ul sau
    FUNCTION GET_NUME_PRODUS(pr produse.id_produs%type)
    RETURN produse.ume_produs%type IS
        ume produse.ume_produs%type;
    BEGIN
        SELECT ume_produs INTO ume FROM produse
        WHERE id_produs = pr;
        RETURN ume;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Nu exista produsul cu codul '||pr);
    END;

--pentru o comanda data prin id-ul sau afisati produsele , iar daca nu contine produse afisati un mesaj
sugestiv
    PROCEDURE COM_PR (comanda comenzi.id_comanda%type) AS
        CURSOR c IS SELECT * FROM produse_comenzi
            WHERE id_comanda = comanda;
        ct NUMBER:=0;
        ok NUMBER:=0;

    BEGIN
        SELECT COUNT(*) INTO ok FROM comenzi
        WHERE id_comanda = comanda;
        IF ok=0 THEN
            RAISE NO_DATA_FOUND;
        END IF;
        ct:=ret_nr(comanda);

```

```

IF ct=0 THEN
    DBMS_OUTPUT.PUT_LINE('Comanda cu numarul '||comanda||' nu contine produse momentan!');
ELSE IF ct=1 THEN
    DBMS_OUTPUT.PUT_LINE('Comanda cu numarul '||comanda||' contine un produs!');
ELSE
    DBMS_OUTPUT.PUT_LINE('Comanda cu numarul '||comanda||' contine '|| ct||' produse!');
END IF;
END IF;
FOR i in c LOOP
    DBMS_OUTPUT.PUT_LINE(GET_NUME_PRODUS(i.id_produs)||' x'||i.nr_bucati);
END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Nu exista comanda cu numarul '||comanda);
END COM_PR;

END pachet2;

/

--testam pachetul
BEGIN
    pachet2.all_man;
END;

/

CALL pachet2.upd_man(286);

BEGIN
    DBMS_OUTPUT.PUT_LINE(pachet2.ret_prod||' : '||pachet2.get_nume_produs(pachet2.ret_prod)||' este
    produsul pentru care s-au vandut cele mai multe exemplare'); --192 : Apa este produsul pentru care s-au
    vandut cele mai multe exemplare
END;

```

/

BEGIN

DBMS_OUTPUT.PUT_LINE('Comanda cu numarul 208 are '||pachet2.ret_nr(208) ||' produse');--
Comanda cu numarul 208 are 3 produse

END;

/

BEGIN

DBMS_OUTPUT.PUT_LINE('Produsul cu id-ul 190 este '||pachet2.get_nume_produs(190));--Produsul
cu id-ul 190 este Masa

END;

/

BEGIN

pachet2.com_pr(250); --ERROR : Nu exista comanda cu numarul 250

END;

/

BEGIN

pachet2.com_pr(210);

-- Comanda cu numarul 210 contine 3 produse!

-- Papusa x1

-- Tricou x1

-- Pantaloni scurti x1

END;

/

SQL Worksheet | History

Worksheet Query Builder

```
1 CREATE OR REPLACE PACKAGE pachet2
2 AS
3     TYPE tablou_indexat IS TABLE OF angajati%rowtype INDEX BY BINARY_INTEGER;
4     TYPE produs IS RECORD (id_prod produse.id_produs%type,
5                             nume_prod produse.nume_produs%type,
6                             stoc_prod produse.stoc%type,
7                             pret_prod produse.pret%type);
8
9
10
11     PROCEDURE UPD_MAN(ang angajati.id_angajat%type);
12     PROCEDURE ALL_MAN;
13     PROCEDURE COM_PR (comanda comenzi.id_comanda%type);
14
15     FUNCTION RET_NR(comanda comenzi.id_comanda%type) RETURN NUMBER;
16     FUNCTION RET_PROD RETURN NUMBER;
17     FUNCTION GET_NUME_PRODUS(pr produse.id_produs%type) RETURN produse.nume_produs%type;
18
19 END pachet2;
20 /
21
22 CREATE OR REPLACE PACKAGE BODY pachet2 AS
23     -- setati postul MANAGER pentru angajati
24     PROCEDURE UPD_MAN(ang angajati.id_angajat%type) AS
25     BEGIN
26         UPDATE angajati SET id_post = 5 WHERE id_angajat = ang;
27     END UPD_MAN;
28
29
30     -- pentru toti angajati care apar ca si id_manager pentru un angajat setati postul MANAGER
31     PROCEDURE ALL_MAN AS
32         emp tablou_indexat;
33     BEGIN
34         SELECT * BULK COLLECT INTO emp FROM angajati;
35
36         FOR i IN 1..emp.last LOOP
37             IF emp(i).id_manager IS NOT NULL THEN
38                 UPD_MAN(emp(i).id_manager);
39             END IF;
40         END LOOP;
41     END ALL_MAN;
42 END pachet2;
```

Script Output x

Task completed in 0.029 seconds

Package PACHET2 compiled

Package Body PACHET2 compiled

create_table.sql

insert.sql

trigger.sql

pacet1.sql

pacet2.sql

Welcome Page

Antonia

SQL Worksheet: History

Worksheet

Query Builder

```

123      pacet2.all_man;
124    END;
125  /
126  CALL pacet2.upd_man(206);
127
128  BEGIN
129    DBMS_OUTPUT.PUT_LINE(pacet2.ret_prod||' : '||pacet2.get_nume_produs(pacet2.ret_prod)||' este produsul pentru care s-au vandut cele mai multe ex
130  END;
131  /
132
133  BEGIN
134    DBMS_OUTPUT.PUT_LINE('Comanda cu numarul 208 are '||pacet2.ret_nr(208)||' produse');--Comanda cu numarul 208 are 3 produse
135  END;
136  /
137
138  BEGIN
139    DBMS_OUTPUT.PUT_LINE('Produsul cu id-ul 190 este '||pacet2.get_nume_produs(190)||'--Produsul cu id-ul 190 este Masa
140  END;
141  /
142
143  BEGIN
144    pacet2.com_pr(250); --ERROR : Nu exista comanda cu numarul 250
145  END;

```

Antonia

Buffer Size:20000

```

192 : Apa este produsul pentru care s-au vandut cele mai multe exemplare
Comanda cu numarul 208 are 3 produse
Produsul cu id-ul 190 este Masa
Comanda cu numarul 210 contine 3 produse!
Papusa xl
Irisou xl
Pantalonii scurti xl

```

Script Output x

Task completed in 0.029 seconds

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Error starting at line : 143 in command -
BEGIN
  pacet2.com_pr(250); --ERROR : Nu exista comanda cu numarul 250
END;
Error report -
ORA-20001: Nu exista comanda cu numarul 250
ORA-06512: at "ANTONIA_DBA.PACET2", line 93
ORA-06512: at line 2

PL/SQL procedure successfully completed.

```