



# PROIECT BAZE DE DATE

Gestiunea unui depozit de lemn de foc

Profesor coordonator  
Prof. Univ. Dr. Vlad Diaconița

Gugiuman L C Irina  
Grupa 1052

# CUPRINS

I.	Descrierea temei .....	3
II.	Schema conceptuală .....	4
III.	Construirea tabelelor bazei de date.....	5
1.	Crearea tabeli <b>REGIUNI</b> conform machetei următoare: .....	5
2.	Crearea tabeli <b>CLIENTI</b> conform machetei următoare: .....	5
3.	Crearea tabeli <b>FURNIZORI</b> conform machetei următoare: .....	6
4.	Crearea tabeli <b>COMENZI</b> conform machetei următoare: .....	6
5.	Crearea tabeli <b>SORTIMENTE_LEMN</b> conform machetei următoare: .....	7
6.	Crearea tabeli <b>TRANSPORTURI</b> conform machetei următoare: .....	7
IV.	Operații de actualizare a datelor .....	9
	Inserare date în tabela <b>REGIUNI</b> : .....	9
	Inserare date în tabela <b>CLIENTI</b> : .....	10
	Inserare date în tabela <b>FURNIZORI</b> : .....	11
	Inserare date în tabela <b>COMENZI</b> : .....	12
	Inserare date în tabela <b>SORTIMENTE_LEMN</b> : .....	12
	Inserare date în tabela <b>TRANSPORTURI</b> : .....	13
V.	Exemple de interogări.....	14

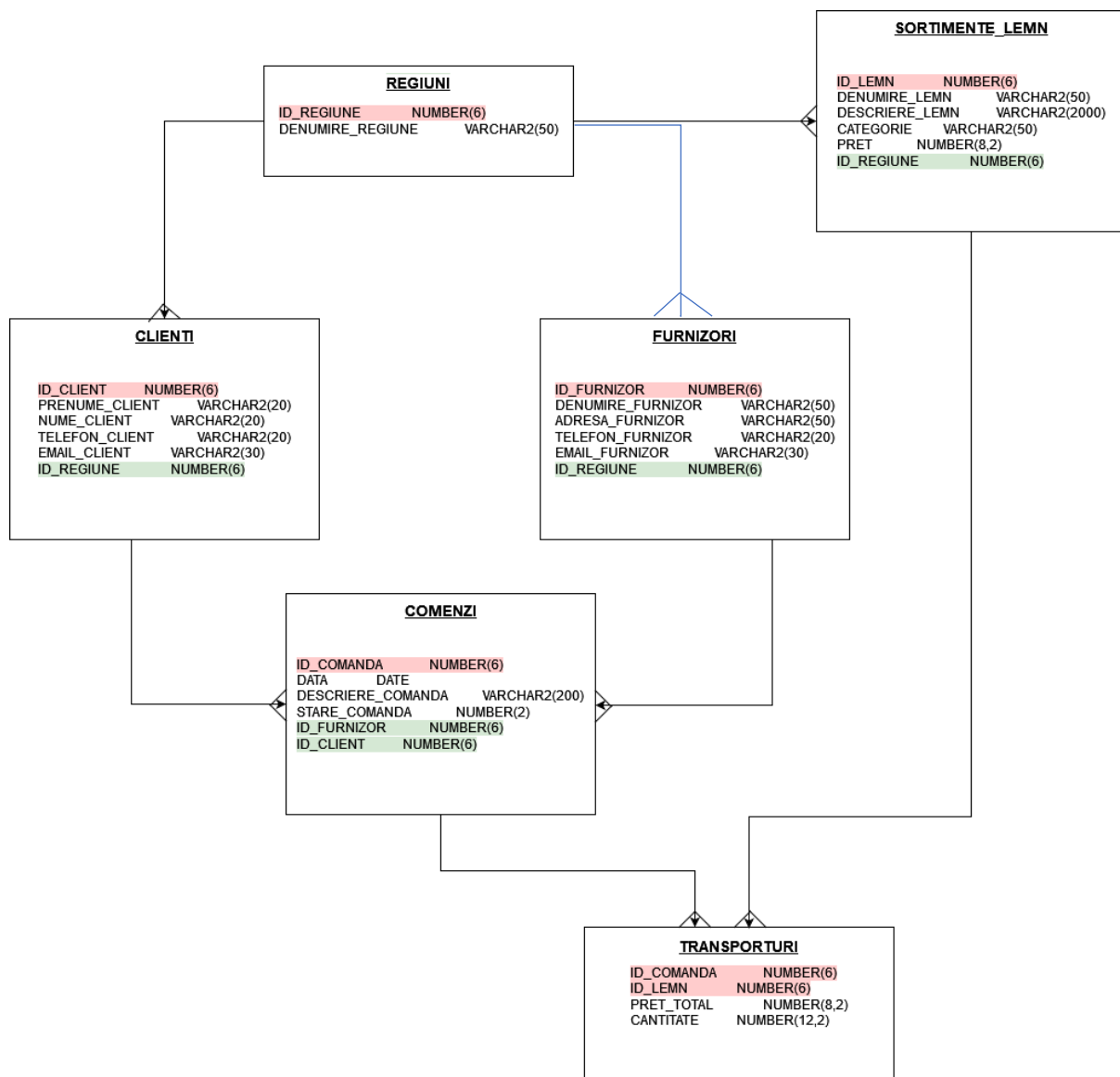
## I. Descrierea temei

În cadrul unui depozit pentru lemn de foc este necesară o evidență strictă a cantităților de lemn care intră (de la furnizori) și care ies (prin livrări către clienți), cu atât mai mult cu cât intervin și minime prelucrări, ce au impact asupra cantităților de produse și a prețurilor acestora. Pentru a asigura gestiunea unei astfel de firme este necesară o bază de date care să țină evidența activităților ce se desfășoară.

Activitatea firmei (depozit lemn de foc) se desfășoară astfel: furnizorii – firme ce recoltează lemn și îl transportă până la depozite primare, vând produsul „lemn de foc” către depozitul de lemn de foc. Este de menționat că dintr-un arbore recoltat, în funcție de rectitudinea trunchiului sau defecte interioare doar 30-40% reprezintă lemn de foc. Furnizorii livrează produsul pe bază de comenzi, iar transportul se poate asigura direct de către furnizori sau pot fi contractați prestatori servicii de transport. Produsul „lemn foc” este stocat în depozit, unde se prelucurează prin tăiere la dimensiuni solicitate de piață sau de client. Prelucrarea adaugă costuri la prețul de achiziție. După tăiere se realizează pachetizarea la diferite dimensiuni, rezultând sortimente de produse care se livrează clienților.

Pentru acces rapid la informații legate de toate aceste activități s-a realizat o bază de date în care s-a pus accent pe: evidența clienților care solicită serviciile depozitului, prin tabela CLIEȚI, evidența furnizorilor de la care se aprovizionează depozitul, prin tabela FURNIZORI, evidența sortimentelor de lemn pe care le distribuie depozitul, prin tabela SORTIMENTE\_LEMN, evidența comenzilor plasate de către clienți, prin tabela COMENZI, evidența livrărilor comenzilor către clienți, prin tabela TRANSPORTURI, iar pentru a se indexa mai ușor județul de proveniență al clienților, furnizorilor și al sortimentelor de lemn, s-a creat și tabela REGIUNI.

## II.Schema conceptuală



### III. Construirea tabelelor bazei de date

#### 1. Crearea tabelii **REGIUNI** conform machetei următoare:

Nume coloană	ID_regiune	Denumire_regiune
Tipul restricției	Primary key	Not Null
Tip dată	Number	Varchar2
Lungime	6	50

```
CREATE TABLE REGIUNI
(ID_REGIUNE NUMBER(6) CONSTRAINT PK_REGIUNI PRIMARY KEY,
DENUMIRE_REGIUNE VARCHAR2(30) NOT NULL
);
```

#### 2. Crearea tabelii **CLIENTI** conform machetei următoare:

Nume coloană	ID_client	Prenume_client	Nume_client	Telefon_client	Email_client	ID_regiune
Tipul restricției	Primary key	Not Null	Not Null			Foreign key - referă tabela <b>REGIUNI</b> , coloana ID_regiune
Tip dată	Number	Varchar2	Varchar2	Varchar2	Varchar2	Number
Lungime	6	20	20	20	30	6

```
CREATE TABLE CLIENTI
(ID_CLIENT NUMBER(6) CONSTRAINT PK_CLIENTI PRIMARY KEY,
PRENUME_CLIENT VARCHAR2(20) NOT NULL,
NUME_CLIENT VARCHAR2(20) NOT NULL,
TELEFON VARCHAR2(20),
EMAIL_CLIENT VARCHAR2(30),
ID_REGIUNE NUMBER(6)
);

ALTER TABLE CLIENTI
ADD CONSTRAINT FK_CLIENTI FOREIGN KEY (ID_REGIUNE)
REFERENCES REGIUNI(ID_REGIUNE);
```

### 3. Crearea tabelii **FURNIZORI** conform machetei următoare:

Nume coloană	ID_furnizor	Denumire_furnizor	Adresa_furnizor	Telefon_furnizor	Email_furnizor	ID_regiune
Tipul restricției	Primary key	Not Null	Not Null			Foreign key - referă tabela <b>REGIUNI</b> , coloana ID_regiune
Tip dată	Number	Varchar2	Varchar2	Varchar2	Varchar2	Number
Lungime	6	50	50	20	30	6

```

CREATE TABLE FURNIZORI
(ID_FURNIZOR NUMBER(6) CONSTRAINT PK_FURNIZOR PRIMARY KEY,
DENUMIRE_FURNIZOR VARCHAR2(50) NOT NULL,
ADRESA_FURNIZOR VARCHAR2(50) NOT NULL,
TELEFON_FURNIZOR VARCHAR2(20),
EMAIL_FURNIZOR VARCHAR2(30),
ID_REGIUNE NUMBER(6),
CONSTRAINT FK_FURNIZORI FOREIGN KEY (ID_REGIUNE) REFERENCES
REGIUNI(ID_REGIUNE)
);

```

### 4. Crearea tabelii **COMENZI** conform machetei următoare:

Nume coloană	ID_comanda	Data	Descriere_comanda	Stare_comanda	ID_furnizor	ID_CLIENT
Tipul restricției	Primary key				Foreign key - referă tabela <b>FURNIZORI</b> , coloana ID_FURNIZOR	Foreign key - referă tabela <b>CLIENTI</b> , coloana ID_client
Tip dată	Number	Date	Varchar2	Number	Number	Number
Lungime	6	-	200	2	6	6

```

CREATE TABLE COMENZI
(ID_COMANDA NUMBER(6) CONSTRAINT PK_COMENZI PRIMARY KEY,
DATA DATE DEFAULT SYSDATE,
DESCRIERE_COMANDA VARCHAR2(200),
STARE_COMANDA NUMBER(2),
ID_FURNIZOR NUMBER(6),
ID_CLIENT NUMBER(6),

```

```

        CONSTRAINT FK_FURNIZOR_COMENZI FOREIGN KEY (ID_FURNIZOR)
REFERENCES FURNIZORI(ID_FURNIZOR),
        CONSTRAINT FK_CLIENT_COMENZI FOREIGN KEY (ID_CLIENT) REFERENCES
CLIENTI(ID_CLIENT)
);

```

#### 5. Crearea tabelii **SORTIMENTE\_LEMN** conform machetei următoare:

Nume coloană	ID_lemn	Denumire_lemn	Descriere_lemn	Categorie	Pret	ID_regiune
Tipul restricției	Primary key	Not Null			Check pret > 0	Foreign key - referă tabela <b>REGIUNI</b> , coloana ID_regiune
Tip dată	Number	Varchar2	Varchar2	Varchar2	Number	Number
Lungime	6	50	2000	50	8,2	6

```

CREATE TABLE SORTIMENTE_LEMN
(ID_LEMN NUMBER(6) CONSTRAINT PK_LEMN PRIMARY KEY,
DENUMIRE_LEMN VARCHAR2(50) NOT NULL,
DESCRIERE_LEMN VARCHAR2(2000),
CATEGORIE VARCHAR2(50),
PRET NUMBER(8,2),
ID_REGIUNE NUMBER(6),
CONSTRAINT FK_LEMN FOREIGN KEY (ID_REGIUNE) REFERENCES
REGIUNI(ID_REGIUNE)
);

```

```

ALTER TABLE SORTIMENTE_LEMN ADD CONSTRAINT LE MN_PRET_MIN
CHECK(PRET>0);

```

#### 6. Crearea tabelii **TRANSPORTURI** conform machetei următoare:

Nume coloană	ID_comanda	ID_lemn	Pret_total	Cantitate
Tipul restricției	Primary key  Foreign key <b>COMENZI</b> , coloana ID_comanda	Primary key  Foreign key <b>SORTIMENTE_LEMN</b> , coloana ID_lemn	Check pret_total>0	Check cantitate>0
Tip dată	Number	Number	Number	Number
Lungime	6	6	8,2	12,2

```
CREATE TABLE TRANSPORTURI
(ID_COMANDA NUMBER(6) NOT NULL,
ID_LEMN NUMBER(6) NOT NULL,
PRET_TOTAL NUMBER(8,2),
CANTITATE NUMBER(12)
);
```

```
ALTER TABLE TRANSPORTURI ADD CONSTRAINT PK_TRANSPORTURI
PRIMARY KEY (ID_COMANDA, ID_LEMN);
```

```
ALTER TABLE TRANSPORTURI ADD CONSTRAINT FK_TRANS_LEMN FOREIGN
KEY (ID_LEMN)
REFERENCES SORTIMENTE_LEMN(ID_LEMN);
```

```
ALTER TABLE TRANSPORTURI ADD CONSTRAINT FK_TANS_COMANDA
FOREIGN KEY (ID_COMANDA)
REFERENCES COMENZI(ID_COMANDA) ON DELETE CASCADE;
```

```
ALTER TABLE TRANSPORTURI ADD CONSTRAINT CK_PRET
CHECK(PRET_TOTAL>0);
```

```
ALTER TABLE TRANSPORTURI ADD CONSTRAINT CK_CANTITATE
CHECK(CANTITATE>0);
```

```
ALTER TABLE TRANSPORTURI MODIFY CANTITATE NUMBER(12,2);
```



## IV.Operații de actualizare a datelor

### Inserare date în tabela REGIUNI:

```
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (1,'Bihor');

INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (2,'Bistrita-
Nasaud');

INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (3,'Cluj');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (4,'Maramures');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (5,'Satu Mare');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (6,'Salaj');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (7,'Alba');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (8,'Brasov');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (9,'Covasna');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (10,'Harghita');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (11,'Mures');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (12,'Sibiu');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (13,'Bacau');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (14,'Botosani');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (15,'Iasi');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (16,'Neamt');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (17,'Suceava');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (18,'Vaslui');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (19,'Braila');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (20,'Buzau');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES
(21,'Constanta');

INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (22,'Galati');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (23,'Tulcea');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (24,'Vrancea');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (25,'Arges');
```

```

INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (26,'Calarasi');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES
(27,'Dambovita');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (28,'Giurgiu');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (29,'Ialomita');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (30,'Prahova');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES
(31,'Teleorman');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (32,'Ilfov');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (33,'Municipiul
Bucuresti');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (34,'Dolj');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (35,'Gorj');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES
(36,'Mehedinti');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (37,'Olt');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (38,'Valcea');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (39,'Arad');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (40,'Caras -
Severin');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES
(41,'Hunedoara');
INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (42,'Timis');

```

#### Inserare date în tabela **CLIENTI**:

```

INSERT INTO CLIENTI (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, TELEFON,
EMAIL_CLIENT, ID_REGIUNE)
VALUES (1, 'Irina', 'Gugiuman', '1234567890',
'gugiumanirina20@stud.ase.ro',33);
INSERT INTO CLIENTI (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)
VALUES(2,'Cosmin','Popescu',32);
INSERT INTO CLIENTI (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)

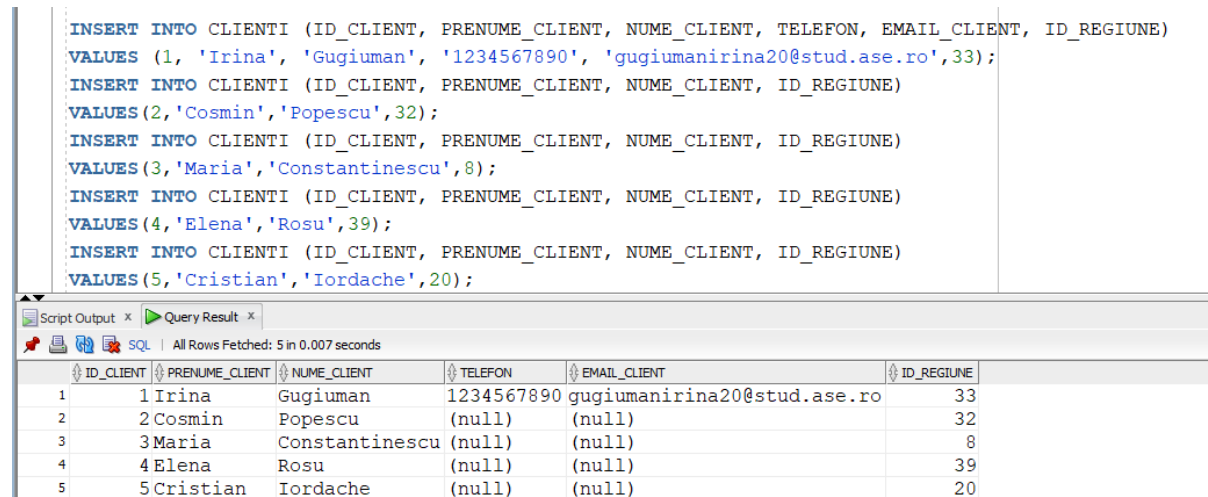
```

```
VALUES(3,'Maria','Constantinescu',8);

INSERT INTO CLIENTI (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)
VALUES(4,'Elena','Rosu',39);

INSERT INTO CLIENTI (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)
VALUES(5,'Cristian','Iordache',20);

UPDATE CLIENTI SET TELEFON='4568521563' WHERE ID_CLIENT=2;
```



The screenshot shows a SQL query execution window with a 'Query Result' tab. The query executed is an INSERT statement for the CLIENTI table. The results are displayed in a table with 6 columns: ID\_CLIENT, PRENUME\_CLIENT, NUME\_CLIENT, TELEFON, EMAIL\_CLIENT, and ID\_REGIUNE. The results show 5 rows of data inserted.

ID_CLIENT	PRENUME_CLIENT	NUME_CLIENT	TELEFON	EMAIL_CLIENT	ID_REGIUNE
1	Irina	Gugiuman	1234567890	gugiumanirina20@stud.ase.ro	33
2	Cosmin	Popescu	(null)	(null)	32
3	Maria	Constantinescu	(null)	(null)	8
4	Elena	Rosu	(null)	(null)	39
5	Cristian	Iordache	(null)	(null)	20

### Inserare date în tabela FURNIZORI:

```
INSERT INTO FURNIZORI
(ID_FURNIZOR,DENUMIRE_FURNIZOR,ADRESA_FURNIZOR,ID_REGIUNE)
VALUES (1,'Lemn Prod','Domnesti',25);

INSERT INTO FURNIZORI
(ID_FURNIZOR,DENUMIRE_FURNIZOR,ADRESA_FURNIZOR,ID_REGIUNE)
VALUES (2,'Silva Prod','Bicaz',16);

INSERT INTO FURNIZORI
(ID_FURNIZOR,DENUMIRE_FURNIZOR,ADRESA_FURNIZOR,ID_REGIUNE)
VALUES (3,'Valdo Forest','Dumitresti',24);

INSERT INTO FURNIZORI
(ID_FURNIZOR,DENUMIRE_FURNIZOR,ADRESA_FURNIZOR,ID_REGIUNE)
VALUES (4,'Diana Forest','Arpas',12);

INSERT INTO FURNIZORI
(ID_FURNIZOR,DENUMIRE_FURNIZOR,ADRESA_FURNIZOR,ID_REGIUNE)
VALUES (5,'Forest Grup','Fetesti',29);
```

```

INSERT INTO FURNIZORI
(ID_FURNIZOR,DENUMIRE_FURNIZOR,ADRESA_FURNIZOR,ID_REGIUNE)

VALUES (6,'Foredist','Bran',8);

DELETE FROM FURNIZORI WHERE ID_FURNIZOR=6;

```

#### Inserare date în tabela COMENZI:

```

INSERT INTO COMENZI (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (1,TO_DATE('30.11.2021','DD.MM.YYYY'),1,2);

INSERT INTO COMENZI (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (2,TO_DATE('07.10.2021','DD.MM.YYYY'),1,3);

INSERT INTO COMENZI (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (3,TO_DATE('12.12.2021','DD.MM.YYYY'),2,2);

INSERT INTO COMENZI (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (4,TO_DATE('27.09.2021','DD.MM.YYYY'),2,4);

INSERT INTO COMENZI (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (5,TO_DATE('04.01.2022','DD.MM.YYYY'),3,4);

INSERT INTO COMENZI (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (6,TO_DATE('07.01.2022','DD.MM.YYYY'),5,3);

UPDATE COMENZI SET STARE_COMANDA=0 WHERE ID_COMANDA IN(1,2,3,4,5,6);

```

#### Inserare date în tabela SORTIMENTE\_LEMN:

```

INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (1,'Carpen',500,12);

INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (2,'Fag',500,16);

INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (3,'Plop',300,29);

INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (4,'Salcie',300,29);

INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (5,'Stejar',500,25);

```

```
INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (6, 'Frasin', 500, 13);
```

```
INSERT INTO SORTIMENTE_LEMN (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (7, 'Salcam', 700, 25);
```

```
UPDATE SORTIMENTE_LEMN SET CATEGORIE = 'lemn de foc' WHERE ID_LEMN IN
(1,2,3,4,5,6,7);
```

#### **Inserare date în tabela TRANSPORTURI:**

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (1, 3, 1500, 50);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (1, 2, 12500, 25);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (2, 1, 5000, 10);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (2, 7, 3850, 5.5);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (3, 5, 5000, 10);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (4, 4, 7500, 25);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (5, 2, 7500, 15);
```

```
INSERT INTO TRANSPORTURI (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (6, 3, 7650, 25.5);
```

## V. Exemple de interogări

1. Sa se afișeze pentru fiecare sortiment de lemn categoria de preț din care face parte

```
SELECT ID_LEMN, DENUMIRE_LEMN, PRET,  
CASE WHEN PRET BETWEEN 200 AND 300 THEN 'PRET MIC'  
      WHEN PRET BETWEEN 301 AND 600 THEN 'PRET MEDIU'  
      WHEN PRET >600 THEN 'PRET MARE'  
      ELSE 'PRET ERONAT/LIPSA' END PRET_CASE  
FROM SORTIMENTE_LEMN ORDER BY PRET;
```

2. Sa se modifice starea comenzii în 1 pentru comenzile din luna septembrie

```
UPDATE COMENZI SET STARE_COMANDA=1 WHERE TO_CHAR(DATA, 'MM')='09';
```

3. Sa se modifice starea comenzii în 2 pentru comenzile din lunile octombrie si noiembrie

```
UPDATE COMENZI SET STARE_COMANDA=2 WHERE TO_CHAR(DATA, 'MM')='10' OR  
TO_CHAR(DATA, 'MM')='11';
```

4. Sa se modifice starea comenzii în 3 pentru comenzile din luna decembrie

```
UPDATE COMENZI SET STARE_COMANDA=3 WHERE TO_CHAR(DATA, 'MM')='12';
```

5. Sa se modifice starea comenzii în 4 pentru comenzile din anul curent

```
UPDATE COMENZI SET STARE_COMANDA=4 WHERE EXTRACT(YEAR FROM  
DATA)=EXTRACT(YEAR FROM SYSDATE);
```

6. Sa se afișeze starea decodificată a comenzilor

```
SELECT ID_COMANDA, DATA, STARE_COMANDA,  
DECODE(STARE_COMANDA,  
0, 'PLASATA', 1, 'TERMINATA', 2, 'LIVRATA', 3, 'ACHITATA', 4, 'IN CURS DE  
ACHITARE') DECODE_STARE  
FROM COMENZI;
```

7. Sa se afișeze furnizorii și regiunea din care fac parte

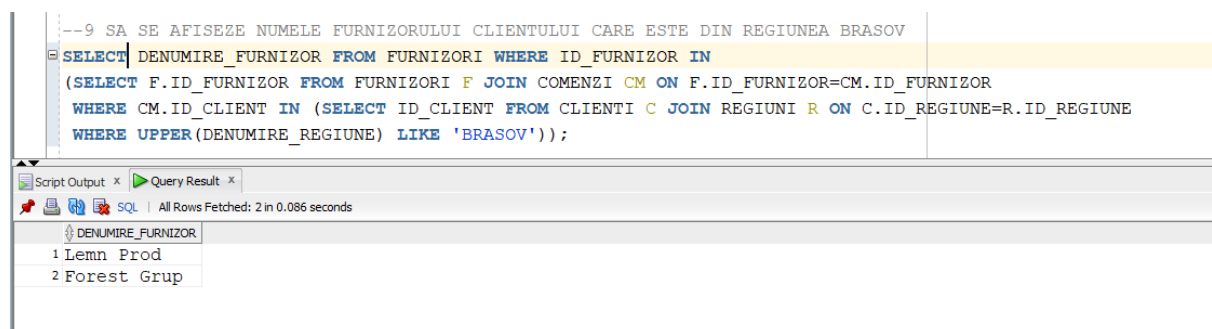
```
SELECT ID_FURNIZOR, DENUMIRE_FURNIZOR, DENUMIRE_REGIUNE  
FROM FURNIZORI F JOIN REGIUNI R  
ON F.ID_REGIUNE=R.ID_REGIUNE;
```

8. Sa se afișeze valoarea totala a comenzilor > 7000

```
SELECT ID_COMANDA, SUM(PRET_TOTAL) VAL FROM TRANSPORTURI  
GROUP BY ID_COMANDA  
HAVING SUM(PRET_TOTAL)>7000;
```

9. Sa se afișeze numele furnizorului clientului care este din regiunea Brașov

```
SELECT DENUMIRE_FURNIZOR FROM FURNIZORI WHERE ID_FURNIZOR IN  
(SELECT F.ID_FURNIZOR FROM FURNIZORI F JOIN COMENZI CM ON  
F.ID_FURNIZOR=CM.ID_FURNIZOR  
WHERE CM.ID_CLIENT IN (SELECT ID_CLIENT FROM CLIENTI C JOIN REGIUNI R ON  
C.ID_REGIUNE=R.ID_REGIUNE  
WHERE UPPER(DENUMIRE_REGIUNE) LIKE 'BRASOV'));
```



(captura 1 din 4)

10. Sa se adauge nr de telefon 0268526044 clientului din regiunea Brașov

```
UPDATE CLIENTI SET TELEFON='0268526044' WHERE ID_CLIENT IN  
(SELECT ID_CLIENT FROM CLIENTI C JOIN REGIUNI R ON  
C.ID_REGIUNE=R.ID_REGIUNE  
WHERE UPPER(DENUMIRE_REGIUNE) LIKE 'BRASOV');
```

11. Sa se afişeze regiunile care nu au a doua litera 'a'

```
SELECT * FROM REGIUNI WHERE SUBSTR(UPPER(DENUMIRE_REGIUNE),2,1)!='A';
```

12. Sa se afişeze clienţii si numerele de telefon ale acestora

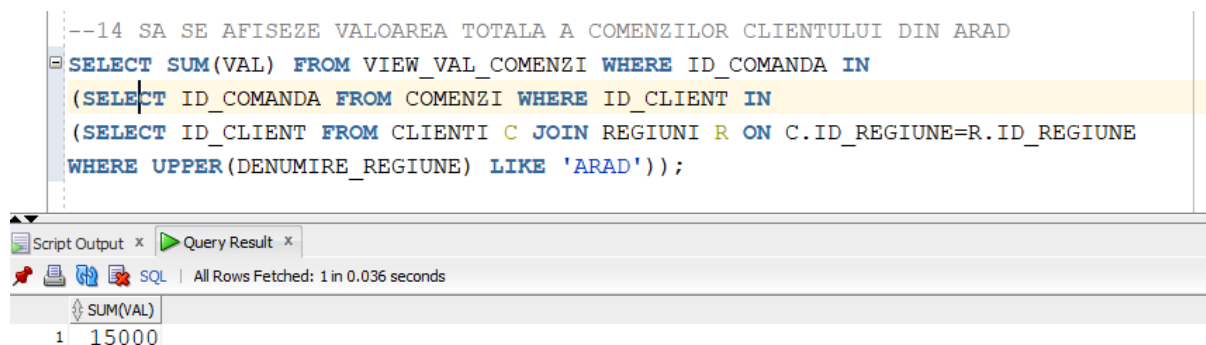
```
SELECT NUME_CLIENT, PRENUME_CLIENT, NVL(TELEFON, '-') FROM CLIENTI;
```

13. Sa se creeze o tabela virtuala care sa conţină valoarea totala a fiecărei comenzi

```
CREATE OR REPLACE VIEW VIEW_VAL_COMENZI AS  
SELECT ID_COMANDA, SUM(PRET_TOTAL) VAL FROM TRANSPORTURI  
GROUP BY ID_COMANDA;
```

14. Sa se afişeze valoarea totala a comenzilor clientului din Arad

```
SELECT SUM(VAL) FROM VIEW_VAL_COMENZI WHERE ID_COMANDA IN  
(SELECT ID_COMANDA FROM COMENZI WHERE ID_CLIENT IN  
(SELECT ID_CLIENT FROM CLIENTI C JOIN REGIUNI R ON  
C.ID_REGIUNE=R.ID_REGIUNE  
WHERE UPPER(DENUMIRE_REGIUNE) LIKE 'ARAD'));
```



(captura 2 din 4)

15. Sa se creeze o tabela virtuala care sa conţină valoarea totala a fiecărei comenzi si cantitatea corespunzătoare

```
CREATE OR REPLACE VIEW TABEL_COMENZI AS
```



```
SELECT ID_COMANDA, SUM(PRET_TOTAL) VALOARE, SUM(CANTITATE) CANTITATE FROM
TRANSPORTURI

GROUP BY ID_COMANDA;
```


16. Sa se afişeze comenzile care au cantitatea comandata cuprinsa intre 10 si 100 fără comenzile care au valoarea cuprinsa intre 7000 si 7500

```
SELECT * FROM TABEL_COMENZI WHERE CANTITATE BETWEEN 10 AND 100

MINUS

SELECT * FROM TABEL_COMENZI WHERE VALOARE BETWEEN 7000 AND 7500;
```

```
--16 SA SE AFISEZE COMENZILE CARE AU CANTITATEA COMANDATA CUPRINSA INTRE 10 SI 100
--FARA COMENZILE CARE AU VALOAREA CUPRINSA INTRE 7000 SI 7500
SELECT * FROM TABEL_COMENZI WHERE CANTITATE BETWEEN 10 AND 100
MINUS
SELECT * FROM TABEL_COMENZI WHERE VALOARE BETWEEN 7000 AND 7500;
```



ID_COMANDA	VALOARE	CANTITATE
1	1	14000
2	8850	15.5
3	5000	10
4	6	7650

(captura 3 din 4)

17. Sa se afişeze denumirea sortimentului de lemn, valoarea totala comandata pentru sortimentele comandate de cel puţin 2 ori si care au valoarea totala diferita de 20000

```
SELECT L.DENUMIRE_LEMN, SUM(T.PRET_TOTAL) VALOARE FROM SORTIMENTE_LEMN L
JOIN TRANSPORTURI T

ON L.ID_LEMN=T.ID_LEMN GROUP BY L.DENUMIRE_LEMN HAVING
COUNT(T.ID_COMANDA)>=2

INTERSECT

SELECT L.DENUMIRE_LEMN, SUM(T.PRET_TOTAL) VALOARE FROM SORTIMENTE_LEMN L
JOIN TRANSPORTURI T

ON L.ID_LEMN=T.ID_LEMN GROUP BY L.DENUMIRE_LEMN HAVING SUM(T.PRET_TOTAL)
!= 20000;
```

18. Sa se afişeze ce preţ ar avea sortimentele de lemne care provin din Ialomiţa si Argeş daca preţul curent s-ar majora cu 10% respectiv 15%

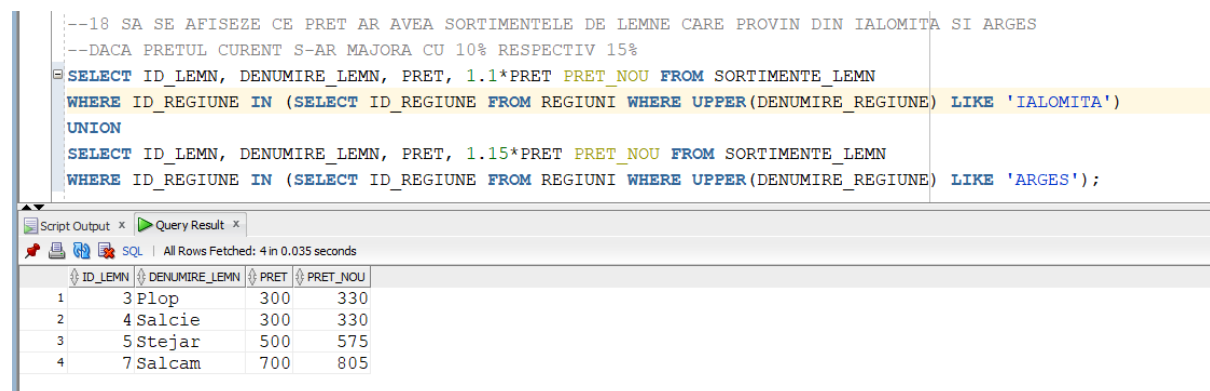
```
SELECT ID_LEMN, DENUMIRE_LEMN, PRET, 1.1*PRET PRET_NOU FROM  
SORTIMENTE_LEMN
```

```
WHERE ID_REGIUNE IN (SELECT ID_REGIUNE FROM REGIUNI WHERE  
UPPER(DENUMIRE_REGIUNE) LIKE 'IALOMITA')
```

```
UNION
```

```
SELECT ID_LEMN, DENUMIRE_LEMN, PRET, 1.15*PRET PRET_NOU FROM  
SORTIMENTE_LEMN
```

```
WHERE ID_REGIUNE IN (SELECT ID_REGIUNE FROM REGIUNI WHERE  
UPPER(DENUMIRE_REGIUNE) LIKE 'ARGES');
```



The screenshot shows a SQL query editor with a script window and a query results window. The script window contains the following SQL query:

```
--18 SA SE AFISEZE CE PRET AR AVEA SORTIMENTELE DE LEMNE CARE PROVIN DIN IALOMITA SI ARGES  
--DACA PRETUL CURENT S-AR MAJORA CU 10% RESPECTIV 15%  
SELECT ID_LEMN, DENUMIRE_LEMN, PRET, 1.1*PRET PRET_NOU FROM SORTIMENTE_LEMN  
WHERE ID_REGIUNE IN (SELECT ID_REGIUNE FROM REGIUNI WHERE UPPER(DENUMIRE_REGIUNE) LIKE 'IALOMITA')  
UNION  
SELECT ID_LEMN, DENUMIRE_LEMN, PRET, 1.15*PRET PRET_NOU FROM SORTIMENTE_LEMN  
WHERE ID_REGIUNE IN (SELECT ID_REGIUNE FROM REGIUNI WHERE UPPER(DENUMIRE_REGIUNE) LIKE 'ARGES');
```

The query results window shows the following data:

ID_LEMN	DENUMIRE_LEMN	PRET	PRET_NOU
1	3 Plop	300	330
2	4 Salcie	300	330
3	5 Stejar	500	575
4	7 Salcam	700	805

(captura 4 din 4)