PROIECT SGBD

Gestiunea unui depozit de lemne

Profesor coordonator
Conf. Univ. Dr. Corbea Alexandra-Maria-Ioana

Gugiuman L C Irina

CUPRINS

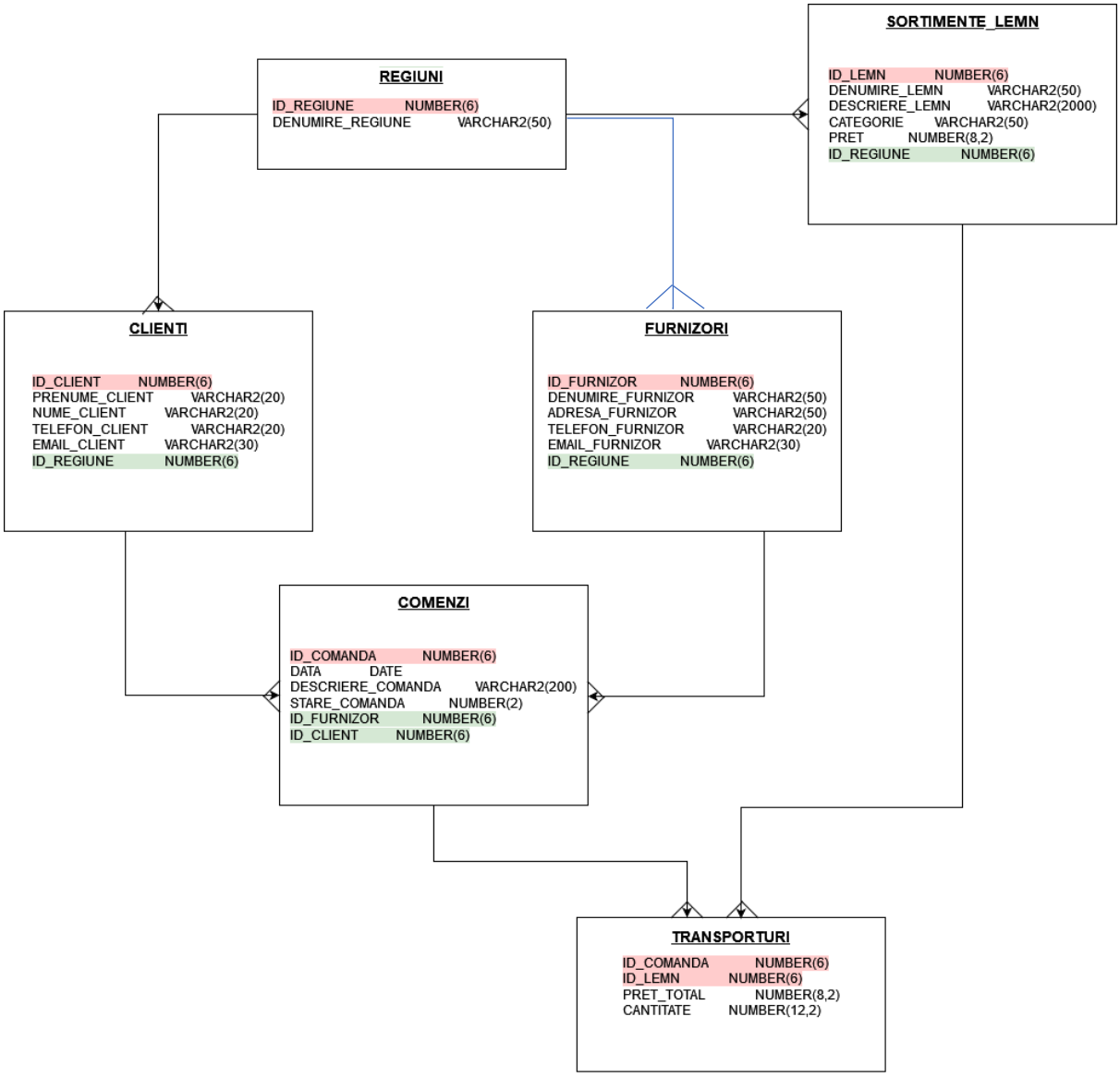
I. Descrierea temei

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 prelucrează 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 CLIENȚI_BD, evidența furnizorilor de la care se aprovizionează depozitul, prin tabela FURNIZORI_BD, evidența sortimentelor de lemn pe care le distribuie depozitul, prin tabela SORTIMENTE_LEMN_BD, evidența comenzilor plasate de către clienți, prin tabela COMENZI_BD, evidența livrărilor comenzilor către clienți, prin tabela TRANSPORTURI_BD, iar pentru a se indexa mai ușor județul de proveniență al clienților, furnizorilor și al sortimentelor de lemn, s-a creat și tabela REGIUNI_BD.

II.Schema conceptuală



III. Interacțiunea cu serverul Oracle prin intermediul comenzilor SQL (LDD și LMD)

Crearea tabelelor bazei de date

```
CREATE TABLE REGIUNI_BD
(ID_REGIUNE NUMBER(6) CONSTRAINT PK_REGIUNI PRIMARY KEY,
 DENUMIRE_REGIUNE VARCHAR2(30) NOT NULL
);
CREATE TABLE CLIENTI_BD
(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_BD
ADD CONSTRAINT FK_CLIENTI FOREIGN KEY (ID_REGIUNE)
REFERENCES REGIUNI_BD(ID_REGIUNE);
CREATE TABLE FURNIZORI_BD
(ID_FURNIZOR NUMBER(6) CONSTRAINT PK_FURNIZOR_BD 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_BD FOREIGN KEY (ID_REGIUNE) REFERENCES
REGIUNI_BD(ID_REGIUNE)
);
CREATE TABLE COMENZI_BD
(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_BD(ID_FURNIZOR),
 CONSTRAINT FK_CLIENT_COMENZI FOREIGN KEY (ID_CLIENT) REFERENCES
CLIENTI_BD(ID_CLIENT)
);
CREATE TABLE SORTIMENTE_LEMN_BD
(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_BD(ID_REGIUNE)
);
ALTER TABLE SORTIMENTE_LEMN_BD ADD CONSTRAINT LEMN_PRET_MIN
CHECK(PRET>0);
CREATE TABLE TRANSPORTURI_BD
(ID_COMANDA NUMBER(6) NOT NULL,
 ID_LEMN NUMBER(6) NOT NULL,
 PRET_TOTAL NUMBER(8,2),
 CANTITATE NUMBER(12)
);
ALTER TABLE TRANSPORTURI_BD ADD CONSTRAINT PK_TRANSPORTURI PRIMARY KEY
(ID_COMANDA, ID_LEMN);
ALTER TABLE TRANSPORTURI_BD ADD CONSTRAINT FK_TRANS_LEMN FOREIGN KEY (ID_LEMN)
REFERENCES SORTIMENTE_LEMN_BD(ID_LEMN);
```

```
ALTER TABLE TRANSPORTURI_BD ADD CONSTRAINT FK_TANS_COMANDA FOREIGN KEY
(ID_COMANDA)

REFERENCES COMENZI_BD(ID_COMANDA) ON DELETE CASCADE;

ALTER TABLE TRANSPORTURI_BD ADD CONSTRAINT CK_PRET

CHECK(PRET_TOTAL>0);

ALTER TABLE TRANSPORTURI_BD ADD CONSTRAINT CK_CANTITATE

CHECK(CANTITATE>0);

ALTER TABLE TRANSPORTURI_BD MODIFY CANTITATE NUMBER(12,2);

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

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

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (3,'Cluj');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (4,'Maramures');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (5,'Satu Mare');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (6,'Salaj');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (7,'Alba');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (8,'Brasov');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (9,'Covasna');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (10,'Harghita');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (11,'Mures');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (12,'Sibiu');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (13,'Bacau');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (14,'Botosani');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (15,'Iasi');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (16,'Neamt');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (17,'Suceava');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (18,'Vaslui');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (19,'Braila');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (20,'Buzau');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (21,'Constanta');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (22,'Galati');
```

```
INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (23,'Tulcea');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (24,'Vrancea');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (25,'Arges');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (26,'Calarasi');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (27,'Dambovita');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (28,'Giurgiu');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (29,'Ialomita');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (30,'Prahova');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (31,'Teleorman');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (32,'Ilfov');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (33,'Municipiul
Bucuresti');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (34,'Dolj');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (35,'Gorj');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (36,'Mehedinti');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (37,'Olt');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (38,'Valcea');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (39,'Arad');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (40,'Caras-Severin');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (41,'Hunedoara');

INSERT INTO REGIUNI_BD (ID_REGIUNE, DENUMIRE_REGIUNE) VALUES (42,'Timis');

INSERT INTO CLIENTI_BD (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, TELEFON,
EMAIL_CLIENT, ID_REGIUNE)

VALUES (1, 'Irina', 'Gugiuman', '1234567890', 'gugiumanirina20@stud.ase.ro',33);

INSERT INTO CLIENTI_BD (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)

VALUES(2,'Cosmin','Popescu',32);

INSERT INTO CLIENTI_BD (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)

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

INSERT INTO CLIENTI_BD (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)

VALUES(4,'Elena','Rosu',39);

INSERT INTO CLIENTI_BD (ID_CLIENT, PRENUME_CLIENT, NUME_CLIENT, ID_REGIUNE)
```

```
VALUES(5,'Cristian','Iordache',20);

SELECT * FROM CLIENTI_BD;

UPDATE CLIENTI_BD SET TELEFON='4568521563' WHERE ID_CLIENT=2;

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

VALUES (1,'Lemn Prod','Domnesti',25);

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

VALUES (2,'Silva Prod','Bicaz',16);

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

VALUES (3,'Valdo Forest','Dumitresti',24);

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

VALUES (4,'Diana Forest','Arpas',12);

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

VALUES (5,'Forest Grup','Fetesti',29);

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

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

DELETE FROM FURNIZORI_BD WHERE ID_FURNIZOR=6;

INSERT INTO COMENZI_BD (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)

VALUES (1,TO_DATE('30.11.2021','DD.MM.YYYY'),1,2);

INSERT INTO COMENZI_BD (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)

VALUES (2,TO_DATE('07.10.2021','DD.MM.YYYY'),1,3);

INSERT INTO COMENZI_BD (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)

VALUES (3,TO_DATE('12.12.2021','DD.MM.YYYY'),2,2);

INSERT INTO COMENZI_BD (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)

VALUES (4,TO_DATE('27.09.2021','DD.MM.YYYY'),2,4);

INSERT INTO COMENZI_BD (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)

VALUES (5,TO_DATE('04.01.2022','DD.MM.YYYY'),3,4);
```

```
INSERT INTO COMENZI_BD (ID_COMANDA, DATA, ID_FURNIZOR, ID_CLIENT)
VALUES (6,TO_DATE('07.01.2022','DD.MM.YYYY'),5,3);
UPDATE COMENZI_BD SET STARE_COMANDA=0 WHERE ID_COMANDA IN(1,2,3,4,5,6);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (1,'Carpen',500,12);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (2,'Fag',500,16);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (3,'Plop',300,29);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (4,'Salcie',300,29);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (5,'Stejar',500,25);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (6,'Frasin',500,13);
INSERT INTO SORTIMENTE_LEMN_BD (ID_LEMN, DENUMIRE_LEMN, PRET, ID_REGIUNE)
VALUES (7,'Salcam',700,25);
UPDATE SORTIMENTE_LEMN_BD SET CATEGORIE = 'lemn de foc' WHERE ID_LEMN IN
(1,2,3,4,5,6,7);
INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (1,3,1500,50);
INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (1,2,12500,25);
INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (2,1,5000,10);
INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (2,7,3850,5.5);
INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
VALUES (3,5,5000,10);
INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)
```

```
VALUES (4,4,7500,25);

INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)

VALUES (5,2,7500,15);

INSERT INTO TRANSPORTURI_BD (ID_COMANDA, ID_LEMN, PRET_TOTAL, CANTITATE)

VALUES (6,3,7650,25.5);
```

- Sa se creeze tabela Transporturi_Anulate in cadrul unui bloc PL/SQL

```
BEGIN

    EXECUTE IMMEDIATE 'CREATE TABLE TRANSPORTURI_ANULATE AS SELECT * FROM
TRANSPORTURI_BD WHERE ID_COMANDA IN (3,4)';

END;

/
```

- Sa se adauge in tabela transporturi_anulate a noua coloana(data_anulare), care contine data la care transportul solicitat a fost anulat.

```
begin

    execute immediate 'ALTER TABLE TRANSPORTURI_ANULATE ADD DATA_ANULARE DATE';

end;

/
```

- Sa se adauge clientul Toader Mircea. Datele personale se vor citi de la tastatura, iar id-ul de client se va calcula prin incrementarea cu o unitate a ultimului client inregistrat.

```
DECLARE

    v_id CLIENTI_BD.ID_CLIENT%TYPE;

BEGIN

    SELECT MAX(ID_CLIENT) INTO V_ID FROM CLIENTI_BD;

    INSERT INTO CLIENTI_BD
VALUES(V_ID+1,'&PRENUME','&NUME','&TELEFON','&EMAIL',&REGIUNE);

    COMMIT;

END;

/
```

- Sa se adauge furnizorul CherestComp pentru care adresa si regiunea sunt citite de la tastatura.

```
DECLARE
    V_ID FURNIZORI.ID_FURNIZOR%TYPE;
BEGIN
    SELECT MAX(ID_FuRNIZOR)INTO V_ID FROM FURNIZORI_BD;
    INSERT INTO FURNIZORI_BD
VALUES(V_ID+1,'CherestComp','&adresa',null,null,'&regiune');
END;
/
```

- Sa se adauge Platan in tabela sortimente_lemn_bd cu categoria lemn de lucru, pretul 450 si regiunea 14

```
DECLARE
    V_ID SORTIMENTE_LEMN_BD.ID_LEMN%TYPE;
BEGIN
    SELECT MAX(ID_LEMN)INTO V_ID FROM SORTIMENTE_LEMN_BD;
    INSERT INTO SORTIMENTE_LEMN_BD VALUES(V_ID+1,'Platan',null,'lemn de
lucru',450,14);
    COMMIT;
END;
/
```

- Sa se adauge datele la care au fost anulate transporturile din tabela transporturi_anulate.

```
BEGIN
    UPDATE TRANSPORTURI_ANULATE SET data_anulare='03.04.2022' WHERE ID_COMANDA=3;
    UPDATE TRANSPORTURI_ANULATE SET DATA_ANULARE='15.02.2022' WHERE ID_COMANDA=4;
END;
/
```

- Sa se modifice categoria in lemn de lucru in tabela sortimente_lemn_bd unde denumire_lemn este Stejar

```
BEGIN
```

```
    UPDATE SORTIMENTE_LEMN_BD SET CATEGORIE='lemn de lucru' WHERE
denumire_lemn='Stejar';

    COMMIT;

END;

/
```

Sa se șteargă din tabela trasporturi_bd transporturile anulate.

```
BEGIN

    DELETE FROM TRANSPORTURI_BD WHERE ID_COMANDA IN (SELECT ID_COMANDA FROM
TRANSPORTURI_ANULATE);

END;

/
```

IV. Structuri de control

- Sa se modifice prețul tipului de lemn cu id-ul citit de la tastatura cu 30% daca este sub 300, cu 20% daca este intre 300 si 400, respectiv cu 10% daca este peste 400.

```
DECLARE

    V_PRET SORTIMENTE_LEMN_BD.PRET%TYPE;

BEGIN

    SELECT PRET INTO V_PRET FROM SORTIMENTE_LEMN_BD WHERE ID_LEMN=&ID;

    DBMS_OUTPUT.PUT_LINE('Pret initial:'||v_pret);

    CASE

        WHEN V_PRET<300 THEN

            V_PRET:=V_PRET*1.3;

        WHEN V_PRET BETWEEN 300 AND 400 then

            V_PRET:=V_PRET*1.2;

        ELSE

            V_PRET:=V_PRET*1.1;

    END CASE;

    DBMS_OUTPUT.PUT_LINE('Pret modificat:'||v_pret);

END;

/
```

- Pentru transportul cu id-ul de comanda 2 si sortimentul de lemn cu id-ul 1 sa se mareasca cantitatea cu 15 unitati daca acesta este sub 20, altfel sa se maresca cu 5 unitati

```
DECLARE
    V_CANT TRANSPORTURI_BD.CANTITATE%TYPE;
BEGIN
    SELECT CANTITATE INTO V_CANT FROM TRANSPORTURI_BD WHERE ID_COMANDA=2 AND
ID_LEMN=1;
    DBMS_OUTPUT.PUT_LINE('Cantitatea initiala: '||v_cant);
    IF V_CANT<20 THEN
        V_CANT:=V_CANT+15;
    ELSE
        V_CANT:=V_CANT+5;
    END IF;
    DBMS_OUTPUT.PUT_LINE('Cantitatea finala: '||v_cant);
END;
/
```

- Sa se maresca pretul sortimentului Frasin calculat astfel: daca pretul este sub 300, se majoreaza cu 30%, intre 300 si 500, cu 20%, peste 500, cu 10%

```
DECLARE
    V_PRET SORTIMENTE_LEMN_BD.PRET%TYPE;
BEGIN
    SELECT PRET INTO V_PRET FROM SORTIMENTE_LEMN_BD WHERE DENUMIRE_LEMN='Frasin';
    DBMS_OUTPUT.PUT_LINE('Pretul inaite de majorare: '||v_pret);
    IF V_PRET<300 THEN
        V_PRET:=1.3*V_PRET;
    ELSIF V_PRET BETWEEN 300 AND 500 THEN
        V_PRET:=1.2*V_PRET;
    ELSE
        V_PRET:=1.1*V_PRET;
```

```
    END IF;
    DBMS_OUTPUT.PUT_LINE('Pretul dupa majorare: '||v_pret);
END;
/
```

- Sa se afiseze denumire sortimentelor de lemn atata timp cat pretul este mai mic ca 600

```
DECLARE
    V_PRET SORTIMENTE_LEMN_BD.PRET%TYPE;
    I NUMBER:=1;
    V_DEN SORTIMENTE_LEMN_BD.DENUMIRE_LEMN%TYPE;
BEGIN
    LOOP
        SELECT PRET INTO V_PRET FROM SORTIMENTE_LEMN_BD WHERE ID_LEMN=I;
        SELECT DENUMIRE_LEMN INTO V_DEN FROM SORTIMENTE_LEMN_BD WHERE ID_LEMN=I;
        DBMS_OUTPUT.PUT_LINE('Sortimentul '||v_den||' are pretul '||v_pret);
        i:=i+1;
        EXIT WHEN V_PRET>500 OR I>8;
    END LOOP;
END;
/
```

- Sa se afiseze regiunile din intervalul 10-25

```
DECLARE
    V_DEN REGIUNI_BD.DENUMIRE_REGIUNE%TYPE;
    I NUMBER:=10;
BEGIN
    WHILE I<=25 LOOP
        SELECT DENUMIRE_REGIUNE INTO V_DEN FROM REGIUNI_BD WHERE ID_REGIUNE=I;
        DBMS_OUTPUT.PUT_LINE(I||' - '||V_DEN);
        I:=I+1;
    END LOOP;
END;
```

```
/
```

- Sa se afiseze denumirea si REGIUNEA sortimentelor de lemn din regiunile in intervalul 12-20 dar sa se intrerupa afisarea cand pretul este mai mic ca 500

```
DECLARE
     V_REG REGIUNI_BD.DENUMIRE_REGIUNE%TYPE;
     V_DEN SORTIMENTE_LEMN_BD.DENUMIRE_LEMN%TYPE;
     V_PRET SORTIMENTE_LEMN_BD.PRET%TYPE;
BEGIN
    FOR I IN 12..20 LOOP
        SELECT DENUMIRE_REGIUNE INTO V_REG FROM REGIUNI_BD WHERE ID_REGIUNE=I;
        SELECT DENUMIRE_LEMN INTO V_DEN FROM SORTIMENTE_LEMN_BD WHERE
ID_REGIUNE=I;
        SELECT PRET INTO V_PRET FROM SORTIMENTE_LEMN_BD WHERE ID_REGIUNE=I;
        DBMS_OUTPUT.PUT_LINE(V_DEN||': specific regiunii '||v_reg);
        EXIT WHEN v_pret < 500;
    END LOOP;
END;
/
```

V. Tratarea excepțiilor

- Sa se afiseze prenumele si numarul de telefon al clientilor din regiunile in intervalul 30-35 si sa se trateze exceptia care apare

```
DECLARE
  V_REC CLIENTI_BD%ROWTYPE;
BEGIN
  FOR I IN 32..35 LOOP
    SELECT * INTO V_REC FROM CLIENTI_BD WHERE ID_REGIUNE=I;
    DBMS_OUTPUT.PUT_LINE(v_rec.id_regiune||': '||V_REC.PRENUME_CLIENT||' -
'||V_REC.TELEFON);
  END LOOP;
EXCEPTION
```

```
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista clienti din aceasta regiune');
END;
/
```

- Sa se afiseze pretul total pentru comanda cu id-ul 2 si sa se trateze eventualele exceptii

```
DECLARE
  V_PRET NUMBER;
BEGIN
  SELECT PRET_TOTAL INTO V_PRET FROM transporturi_bd WHERE ID_COMANDA=2;
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Exista mai multe transporturi pentru comanda cu id-ul 2');
END;
/
```

- Sa se mărească prețul tipurilor de lemn cu 20% daca prețul curent este <400

```
DECLARE
  CURSOR C_LEMN IS SELECT DENUMIRE_LEMN, PRET FROM SORTIMENTE_LEMN_BD;
  REC_LEMN C_LEMN%ROWTYPE;
BEGIN
  OPEN C_LEMN;
  LOOP
    FETCH C_LEMN INTO REC_LEMN;
    EXIT WHEN C_LEMN%NOTFOUND;
    IF REC_LEMN.PRET<400 THEN
      REC_LEMN.PRET:=1.2*REC_LEMN.PRET;
    END IF;
    DBMS_OUTPUT.PUT_LINE(REC_LEMN.DENUMIRE_LEMN||' -> '||REC_LEMN.PRET);
```

```
  END LOOP;
EXCEPTION
  WHEN CURSOR_ALREADY_OPEN THEN
    DBMS_OUTPUT.PUT_LINE('Cursorul este deja deschis');
END;
/
```

- Sa se majoreze cu 40% pretul sortimentul de lemn cu denumirea Abanos

```
DECLARE
  EX_1 EXCEPTION;
  PRAGMA EXCEPTION_INIT(EX_1,-20001);
BEGIN
  UPDATE SORTIMENTE_LEMN_BD SET PRET=PRET*1.4 WHERE DENUMIRE_LEMN='Abanos';
  IF SQL%NOTFOUND THEN
    RAISE_APPLICATION_ERROR(-20001, 'Sortiment inexistent!');
  END IF;
EXCEPTION
  WHEN EX_1 THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista acest sotiment de lemn');
END;
/
```

- Sa se modifice starea comenzii in 5 pentru comanda cu iD-ul de furnizor 2 si sa se ridice o exceptie in cazul in care exista mai mult de o astfel de comnada

```
DECLARE
  ex_2 EXCEPTION;
BEGIN
  UPDATE COMENZI_BD SET STARE_COMANDA=5 WHERE ID_FURNIZOR=2;
  IF SQL%ROWCOUNT >1 THEN
    RAISE EX_2;
```

```
    END IF;
  EXCEPTION
    WHEN EX_2 THEN
      DBMS_OUTPUT.PUT_LINE('S-a actualizat mai mult de o camanda');
  END;
  /
```

VI. Gestionarea cursorilor

- Sa se modifice adresa furnizorului din regiunea 20 in Buzau, iar in cazul in care nu exista nici un furnizor in acesta regiune sa se afiseze un mesaj adecvat

```
BEGIN
  UPDATE FURNIZORI_BD SET adresa_furnizor='Buzau' WHERE ID_REGIUNE=20;
  IF SQL%NOTFOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu ezista furnizori in acesta regiune');
  END IF;
END;
/
```

- Sa se scada pretul sortimentelor de lemn cu 10% pentru care pretul actual este 500 si sa se afiseze numerul de randuri actualizate

```
DECLARE
  v_nr number;
BEGIN
  UPDATE SORTIMENTE_LEMN_BD SET PRET=0.9*PRET WHERE PRET=500;
  V_NR:=SQL%ROWCOUNT;
  DBMS_OUTPUT.PUT_LINE(V_NR||' RANDURI ACTUALIZATE');
END;
/
```

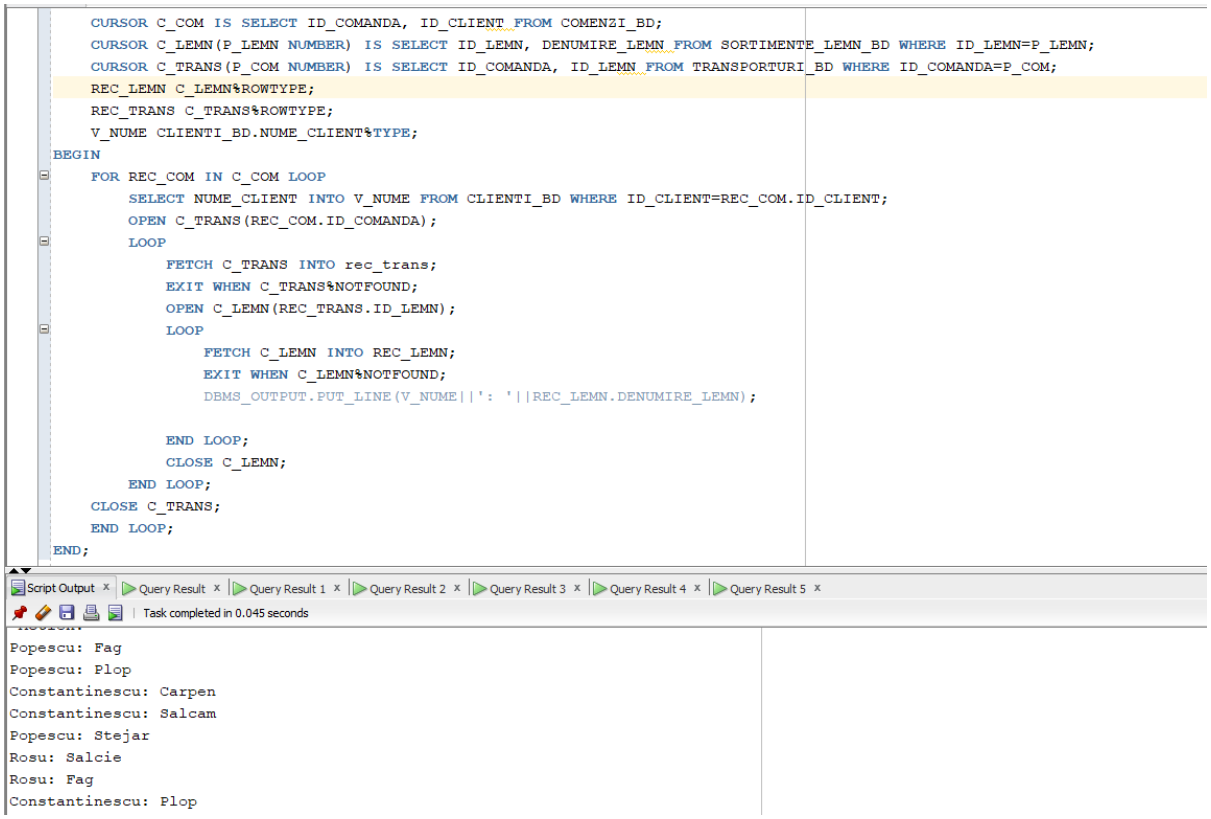
- Sa se modifice regiunea in 42 pentru client cu numarul de telefon 0745296813

```
BEGIN
   UPDATE CLIENTI_BD SET id_regiune=42 WHERE TELEFON='0745296813';
   IF SQL%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE('Nici un client nu are acest numar de telefon');
   END IF;
END;
/
```

- Sa se afiseze numele clientului si denumirea sorimentului de lemn comandat pentru toate comenzile

```
DECLARE
   CURSOR C_COM IS SELECT ID_COMANDA, ID_CLIENT FROM COMENZI_BD;
   CURSOR C_LEMN(P_LEMN NUMBER) IS SELECT ID_LEMN, DENUMIRE_LEMN FROM
SORTIMENTE_LEMN_BD WHERE ID_LEMN=P_LEMN;
   CURSOR C_TRANS(P_COM NUMBER) IS SELECT ID_COMANDA, ID_LEMN FROM TRANSPORTURI_BD
WHERE ID_COMANDA=P_COM;
   REC_LEMN C_LEMN%ROWTYPE;
   REC_TRANS C_TRANS%ROWTYPE;
   V_NUME CLIENTI_BD.NUME_CLIENT%TYPE;
BEGIN
   FOR REC_COM IN C_COM LOOP
      SELECT NUME_CLIENT INTO V_NUME FROM CLIENTI_BD WHERE ID_CLIENT=REC_COM.ID_CLIENT;
      OPEN C_TRANS(REC_COM.ID_COMANDA);
      LOOP
         FETCH C_TRANS INTO rec_trans;
         EXIT WHEN C_TRANS%NOTFOUND;
         OPEN C_LEMN(REC_TRANS.ID_LEMN);
         LOOP
            FETCH C_LEMN INTO REC_LEMN;
            EXIT WHEN C_LEMN%NOTFOUND;
```

```
            DBMS_OUTPUT.PUT_LINE(V_NUME||': '||REC_LEMN.DENUMIRE_LEMN);


        END LOOP;

        CLOSE C_LEMN;

    END LOOP;

  CLOSE C_TRANS;

  END LOOP;

END;

/
```



VII. Funcții proceduri și includerea acestora în pachete

- Sa se creeze o functie care afiseza nr total de transporturi

```
CREATE OR REPLACE FUNCTION NR_TRANSPORTURI

RETURN NUMBER

IS

V_NR NUMBER;
```

```
BEGIN
   SELECT COUNT(*) INTO V_NR FROM TRANSPORTURI_BD;
   RETURN V_NR;
END;
/
DECLARE
   V NUMBER;
BEGIN
   V:=NR_TRANSPORTURI;
   DBMS_OUTPUT.PUT_LINE('Numar total de transporturi:'||v);
END;
/
```

- Sa se creeze o functie care returneaza numarul de regiuni care au furnizori de lemn

```
CREATE OR REPLACE FUNCTION NR_REGIUNI_FURNIZORI
RETURN NUMBER
IS
V_NR NUMBER;
BEGIN
   SELECT DISTINCT COUNT(*) INTO V_NR FROM FURNIZORI_BD;
   RETURN V_NR;
END;
/
DECLARE
   V NUMBER;
BEGIN
   V:=nr_regiuni_furnizori;
   DBMS_OUTPUT.PUT_LINE('Numar total de regiuni care au furnizori:'||v);
```

```
END;
/
```

- Sa se creeze o procedura care afiseaza pentru regiunea data furnizorul, clientul si sortimentul de lemn daca acestea exista

```
CREATE OR REPLACE PROCEDURE DATE_REGIUNE(P_ID REGIUNI_BD.ID_REGIUNE%TYPE)
IS
   V_FUR FURNIZORI_BD.DENUMIRE_FURNIZOR%TYPE:='';
   V_CLI CLIENTI_BD.NUME_CLIENT%TYPE:='';
   V_REG REGIUNI_BD.DENUMIRE_REGIUNE%TYPE;
BEGIN
   SELECT DENUMIRE_REGIUNE INTO V_REG FROM REGIUNI_BD WHERE ID_REGIUNE=P_ID;
   SELECT DENUMIRE_FURNIZOR INTO V_FUR FROM FURNIZORI_BD WHERE ID_REGIUNE=P_ID;
   SELECT NUME_CLIENT INTO V_CLI FROM CLIENTI_BD WHERE ID_REGIUNE=P_ID;
   DBMS_OUTPUT.PUT_LINE(V_REG||':'||V_FUR||'; '||V_CLI);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
   WHEN TOO_MANY_ROWS THEN
      DBMS_OUTPUT.PUT_LINE('PREA MULTE DATE');
END;
/


BEGIN
   DATE_REGIUNE(14);
END;
/
```

- Creati o procedura care returneaza cea mai recenta comanda si valoarea ei

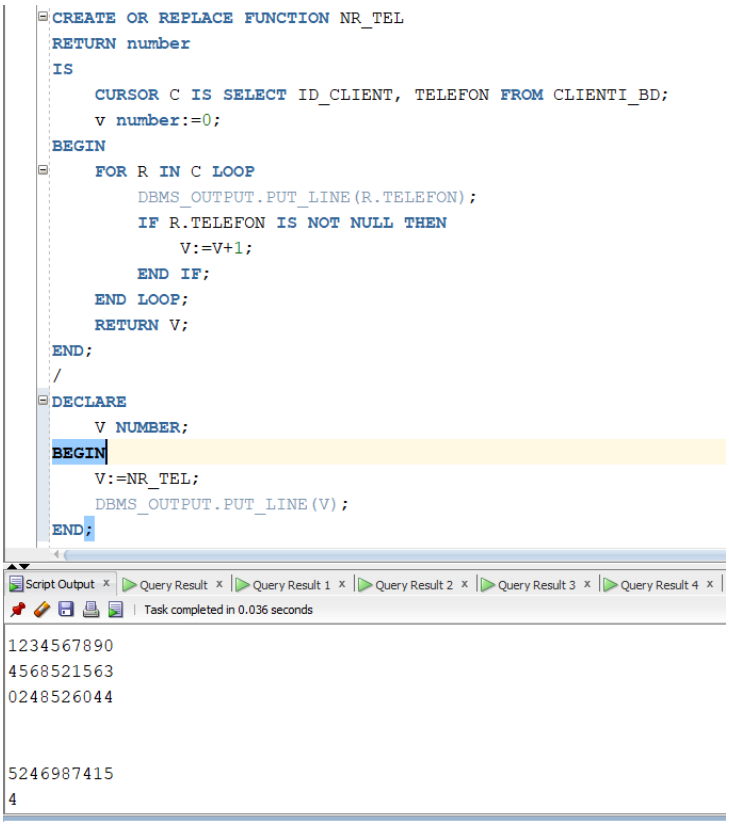
```
create or replace procedure VAL_COMANDA_RECENTA(P_DATA OUT COMENZI_BD.DATA%TYPE,P_VAL
OUT NUMBER)
```

```
IS
   V_ID NUMBER;
BEGIN
   SELECT MAX(DATA) INTO P_DATA FROM COMENZI_BD;
   SELECT ID_COMANDA INTO V_ID FROM COMENZI_BD WHERE DATA=P_DATA;
   SELECT SUM(PRET_TOTAL) INTO P_VAL FROM TRANSPORTURI_BD WHERE ID_COMANDA=V_ID;
END;
/
DECLARE
   V_DATA COMENZI_BD.DATA%TYPE;
   V_VAL NUMBER;
BEGIN
   VAL_COMANDA_RECENTA(V_DATA,V_VAL);
   DBMS_OUTPUT.PUT_LINE('DATA:'||V_DATA||'; VALOARE:'||V_VAL);
END;
/
```

- Sa se creeze o functie care afiseaza numarul de telefon al clientilor si cati au nr de telefon

```
CREATE OR REPLACE FUNCTION NR_TEL
RETURN number
IS
   CURSOR C IS SELECT ID_CLIENT, TELEFON FROM CLIENTI_BD;
   v number:=0;
BEGIN
   FOR R IN C LOOP
      DBMS_OUTPUT.PUT_LINE(R.TELEFON);
      IF R.TELEFON IS NOT NULL THEN
         V:=V+1;
```

```
      END IF;
   END LOOP;
   RETURN V;
END;
/
DECLARE
   V NUMBER;
BEGIN
   V:=NR_TEL;
   DBMS_OUTPUT.PUT_LINE(V);
END;
/
```



- Sa se creeze o procedura care calculeaza si returneaza pretul mediu al asortimentelor de lemn

```
CREATE OR REPLACE PROCEDURE PRET_MEDIU(P_PRET_MEDIU OUT NUMBER)
```

```
IS
BEGIN
   SELECT AVG(PRET) INTO P_PRET_MEDIU FROM SORTIMENTE_LEMN_BD;
END;
/
DECLARE
   V NUMBER;
BEGIN
   PRET_MEDIU(V);
   DBMS_OUTPUT.PUT_LINE('Pretul mediu: '||v);
END;
/
```

```
CREATE OR REPLACE PROCEDURE PRET_MEDIU(P_PRET_MEDIU OUT NUMBER)
IS
BEGIN
    SELECT AVG(PRET) INTO P_PRET_MEDIU FROM SORTIMENTE_LEMN_BD;
END;
/
DECLARE
    V NUMBER;
BEGIN
    PRET_MEDIU(V);
    DBMS_OUTPUT.PUT_LINE('Pretul mediu: '||v);
END;
```

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

Task completed in 0.049 seconds

```
*Action:
Pretul mediu: 443.75


PL/SQL procedure successfully completed.
```

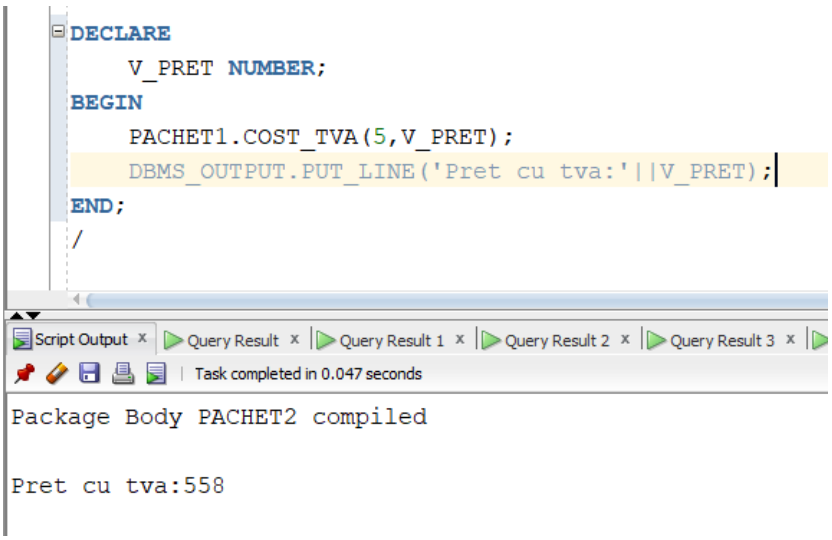
```
CREATE OR REPLACE PACKAGE PACHET1 IS
   PROCENT_TVA NUMBER:=0.24;
   PROCENT_TRANSPORT NUMBER:=0.30;
   FUNCTION GET_PROCENT_TVA RETURN NUMBER;
   FUNCTION GET_PROCENT_TRANSPORT RETURN NUMBER;
```

```
   FUNCTION NR_TRANSPORTURI RETURN NUMBER;
   PROCEDURE VAL_COMANDA_RECENTA(P_DATA OUT COMENZI_BD.DATA%TYPE,P_VAL OUT NUMBER);
   PROCEDURE COST_TVA(P_ID_LEMN IN SORTIMENTE_LEMN_BD.ID_LEMN%TYPE,P_COST OUT
NUMBER);
END PACHET1;
/
CREATE OR REPLACE PACKAGE BODY PACHET1 IS

   FUNCTION GET_PROCENT_TVA RETURN NUMBER IS
   BEGIN
      RETURN PROCENT_TVA;
   END;

   FUNCTION GET_PROCENT_TRANSPORT RETURN NUMBER IS
   BEGIN
      RETURN GET_PROCENT_TRANSPORT;
   END;

   FUNCTION NR_TRANSPORTURI
   RETURN NUMBER
   IS
   V_NR NUMBER;
   BEGIN
      SELECT COUNT(*) INTO V_NR FROM TRANSPORTURI_BD;
      RETURN V_NR;
   END;

   procedure VAL_COMANDA_RECENTA(P_DATA OUT COMENZI_BD.DATA%TYPE,P_VAL OUT NUMBER)
   IS
```

```
    V_ID NUMBER;
  BEGIN
    SELECT MAX(DATA) INTO P_DATA FROM COMENZI_BD;
    SELECT ID_COMANDA INTO V_ID FROM COMENZI_BD WHERE DATA=P_DATA;
    SELECT SUM(PRET_TOTAL) INTO P_VAL FROM TRANSPORTURI_BD WHERE ID_COMANDA=V_ID;
  END;

  PROCEDURE COST_TVA(P_ID_LEMN IN SORTIMENTE_LEMN_BD.ID_LEMN%TYPE,P_COST OUT
NUMBER)
  IS
    V_PRET SORTIMENTE_LEMN_BD.PRET%TYPE;
  BEGIN
    SELECT PRET INTO V_PRET FROM SORTIMENTE_LEMN_BD WHERE ID_LEMN=P_ID_LEMN;
    P_COST:=V_PRET+V_PRET*PROCENT_TVA;
  END;

END PACHET1;
/

CREATE OR REPLACE PACKAGE PACHET2 IS
  FUNCTION NR_REGIUNI_FURNIZORI RETURN NUMBER;
  FUNCTION NR_TEL RETURN number;
  PROCEDURE DATE_REGIUNE(P_ID REGIUNI_BD.ID_REGIUNE%TYPE);
  PROCEDURE PRET_MEDIU(P_PRET_MEDIU OUT NUMBER);
END PACHET2;
/
CREATE OR REPLACE PACKAGE BODY PACHET2 IS
  FUNCTION NR_REGIUNI_FURNIZORI
  RETURN NUMBER
```

```
IS
V_NR NUMBER;
BEGIN
  SELECT DISTINCT COUNT(*) INTO V_NR FROM FURNIZORI_BD;
  RETURN V_NR;
END;

FUNCTION NR_TEL
RETURN number
IS
   CURSOR C IS SELECT ID_CLIENT, TELEFON FROM CLIENTI_BD;
  v number:=0;
BEGIN
  FOR R IN C LOOP
      DBMS_OUTPUT.PUT_LINE(R.TELEFON);
      IF R.TELEFON IS NOT NULL THEN
         V:=V+1;
      END IF;
   END LOOP;
   RETURN V;
END;

PROCEDURE DATE_REGIUNE(P_ID REGIUNI_BD.ID_REGIUNE%TYPE)
IS
   V_FUR FURNIZORI_BD.DENUMIRE_FURNIZOR%TYPE:='';
   V_CLI CLIENTI_BD.NUME_CLIENT%TYPE:='';
   V_REG REGIUNI_BD.DENUMIRE_REGIUNE%TYPE;
BEGIN
```

```
    SELECT DENUMIRE_REGIUNE INTO V_REG FROM REGIUNI_BD WHERE ID_REGIUNE=P_ID;
    SELECT DENUMIRE_FURNIZOR INTO V_FUR FROM FURNIZORI_BD WHERE ID_REGIUNE=P_ID;
    SELECT NUME_CLIENT INTO V_CLI FROM CLIENTI_BD WHERE ID_REGIUNE=P_ID;
    DBMS_OUTPUT.PUT_LINE(V_REG||':'||V_FUR||'; '||V_CLI);
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Nu s-au gasit date');
    WHEN TOO_MANY_ROWS THEN
      DBMS_OUTPUT.PUT_LINE('PREA MULTE DATE');
  END;

  PROCEDURE PRET_MEDIU(P_PRET_MEDIU OUT NUMBER)
  IS
  BEGIN
    SELECT AVG(PRET) INTO P_PRET_MEDIU FROM SORTIMENTE_LEMN_BD;
  END;

END PACHET2;
/

DECLARE
  V_PRET NUMBER;
BEGIN
  PACHET1.COST_TVA(5,V_PRET);
  DBMS_OUTPUT.PUT_LINE('Pret cu tva:'||V_PRET);
END;
/
```



VIII. Declanșatori

- Sa se creeze un trigger care sa se declanseze inaintea stergerii detelor din tabela furnizori_bd

```
CREATE OR REPLACE TRIGGER FURNIZORI_TRIGG

BEFORE DELETE ON FURNIZORI_BD

BEGIN

  DBMS_OUTPUT.PUT_LINE('Veti sterge date');

END;

/
```

- Sa se creeze un trigger pentru a permite depasirea unei limite minime a pretului sortimentelor de lemn

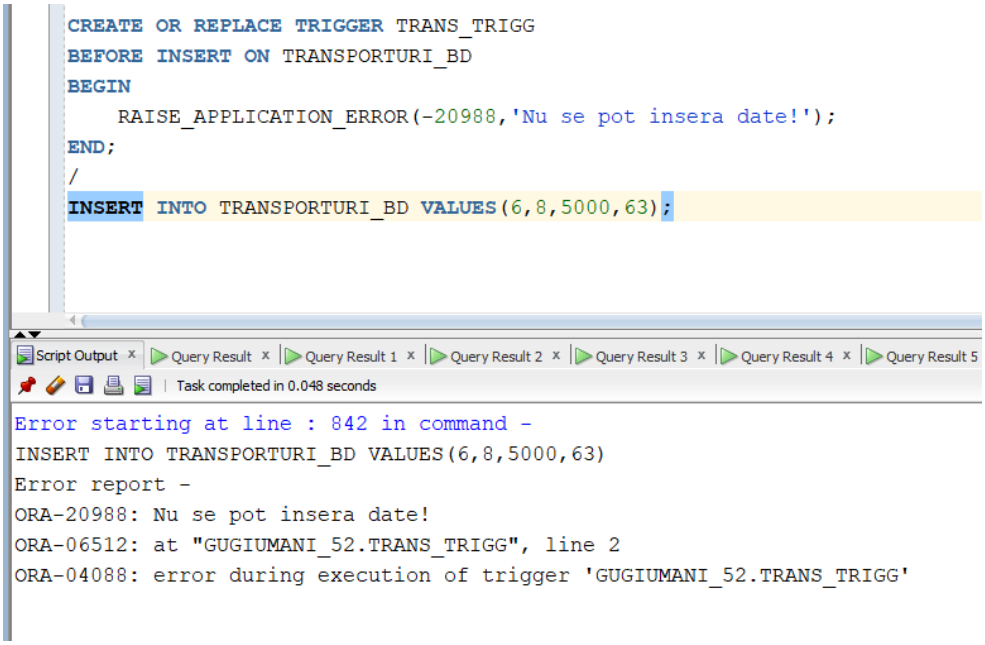
```
CREATE OR REPLACE TRIGGER RESTRICTIE_PRET

BEFORE INSERT OR UPDATE ON SORTIMENTE_LEMN_BD

FOR EACH ROW

DECLARE

  V_PRET_MIN NUMBER:=250;

BEGIN

  IF :NEW.PRET<V_PRET_MIN THEN

    RAISE_APPLICATION_ERROR(-20999,'Pretul nou este prea mic');

  END IF;

END;

/
```

- Sa se creeze un trigger care sa se declaseze inainte de inserarea unui nou transport in tabela transporturi_bd

```
CREATE OR REPLACE TRIGGER TRANS_TRIGG

BEFORE INSERT ON TRANSPORTURI_BD

BEGIN

  RAISE_APPLICATION_ERROR(-20988,'Nu se pot insera date!');

END;
```

/

INSERT INTO TRANSPORTURI_BD VALUES(6,8,5000,63);



```
CREATE OR REPLACE TRIGGER TRANS_TRIGG
BEFORE INSERT ON TRANSPORTURI_BD
BEGIN
    RAISE_APPLICATION_ERROR(-20988,'Nu se pot insera date!');
END;
/
INSERT INTO TRANSPORTURI_BD VALUES(6,8,5000,63);
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5

Task completed in 0.048 seconds

```
Error starting at line : 842 in command -
INSERT INTO TRANSPORTURI_BD VALUES(6,8,5000,63)
Error report -
ORA-20988: Nu se pot insera date!
ORA-06512: at "GUGIUMANI_52.TRANS_TRIGG", line 2
ORA-04088: error during execution of trigger 'GUGIUMANI_52.TRANS_TRIGG'
```