

# Gestionarea unui magazin de haine

Proiect pentru cursul Sisteme de Gestiune a Bazelor de Date realizat de Podaru Eduard-Cristian, grupa 231

## Cerința 1.

Am modelat baza de date a unui magazin ce vinde articole de îmbrăcăminte. Utilitatea acesteia constă în facilitarea și eficientizarea a mai multor procese aferente unei astfel de afaceri. Prin folosirea unei baze de date, se pot afla rapid informații legate de clienți, de stocul actual al magazinului, de articole care sunt în stare de livrare sau rezervate pentru o comandă, de detalii despre articole precum colecțiile la care aparțin, de firmele care fabrică articolele, și alte informații folositoare prin cereri mai complexe.

### Baza de date conține cuprinde informații despre:

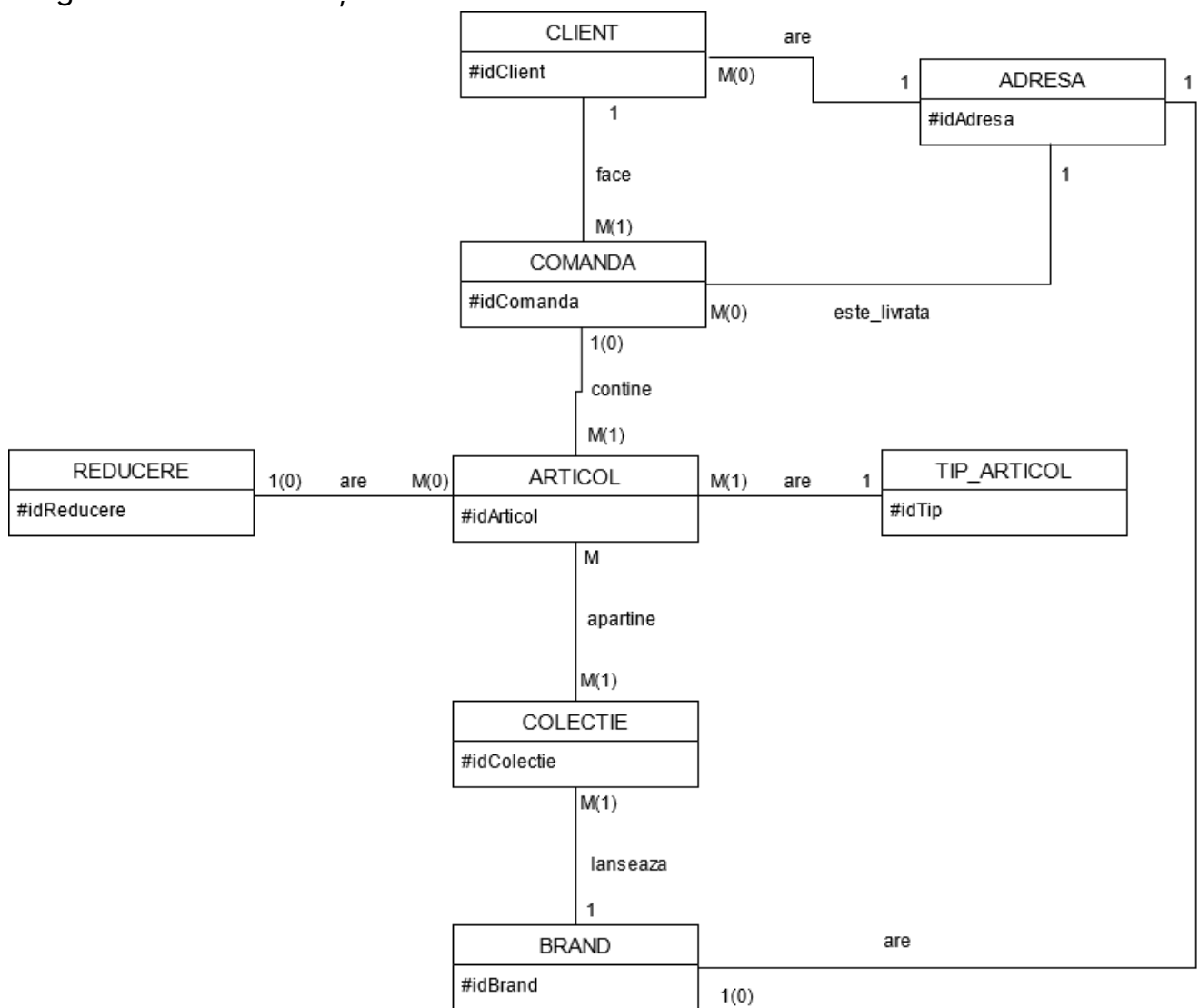
- Articole de îmbrăcăminte și tipul în care se încadrează
- Comenzi și clienții care le-au efectuat
- Colecțiile din care fac parte articolele de îmbrăcăminte
- Firmele care lansează colecțiile de haine
- Adresele firmelor și clienților

### Caracteristici:

- Magazinul deține articole de diferite tipuri care pot aparține mai multor colecții de haine.
- Fiecare articol are un singur tip.
- Pot exista mai multe articole cu aceleași caracteristici în stoc în același timp.
- Tipurile articolelor se referă la categoria de îmbrăcăminte din care fac parte(tricou/cămașă etc) și sexul purtătorului.
- Fiecare articol aparține cel puțin unei colecții.
- Colecțiile de haine sunt lansate de către brand-uri, fiecare colecție e lansată de un singur brand.
- Articolele pot fi la reducere, însă pot face parte dintr-o singură promoție(nu pot fi mai multe reduceri legate de un singur articol).
- Reducerile provin din promoții care au propriile denumiri.
- Clienții pot face comenzi pentru mai multe articole.
- Fiecare articol poate face parte din maxim o comandă.
- Comenzile au o dată de plecare și o dată de livrare.
- Fiecarei comenzi îi este asociat un singur client.

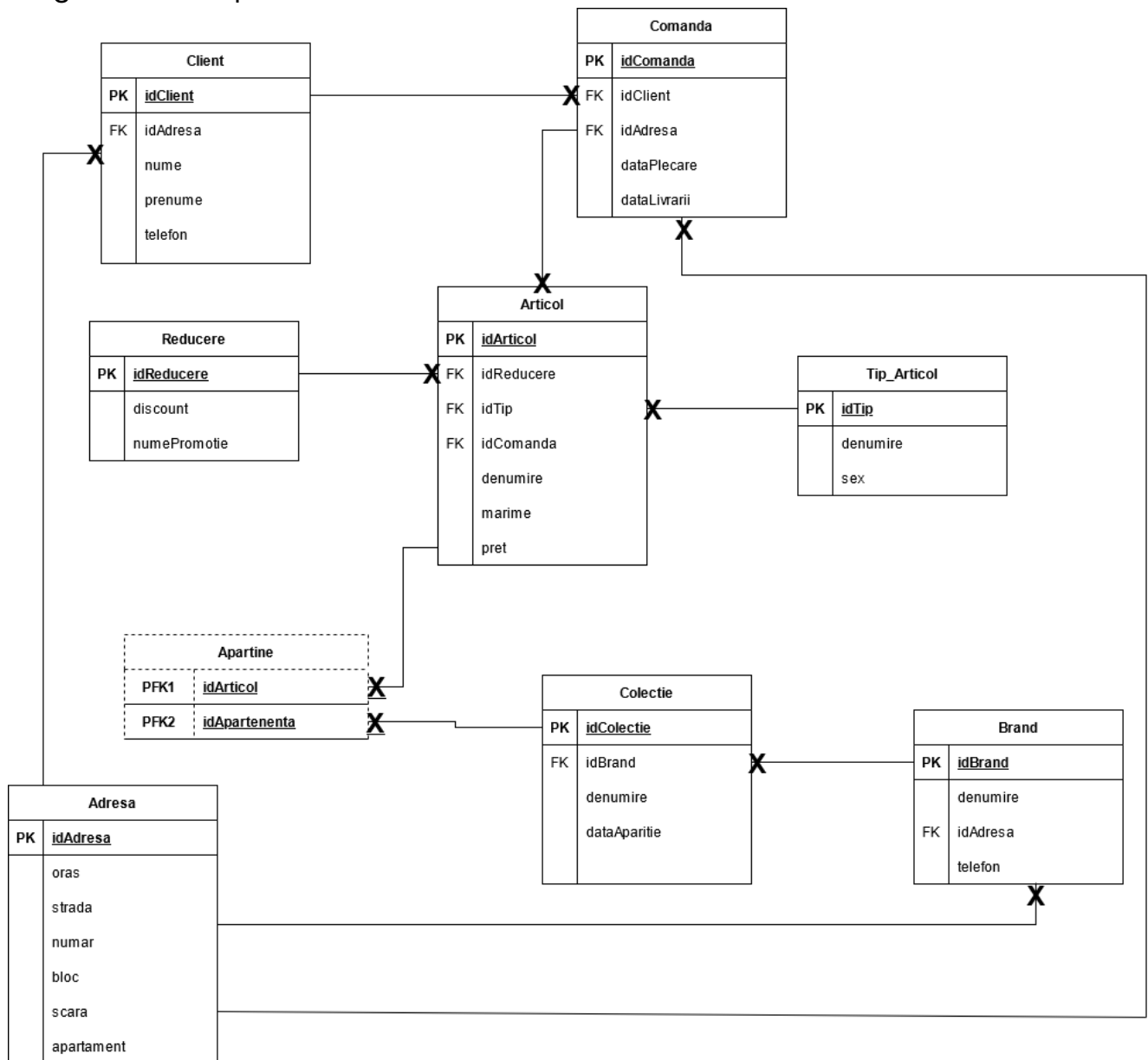
## Cerința 2.

Diagrama entitate-relație:



### Cerința 3.

Diagrama conceptuală:



## Cerința 4.

Codul pentru crearea tabelor:

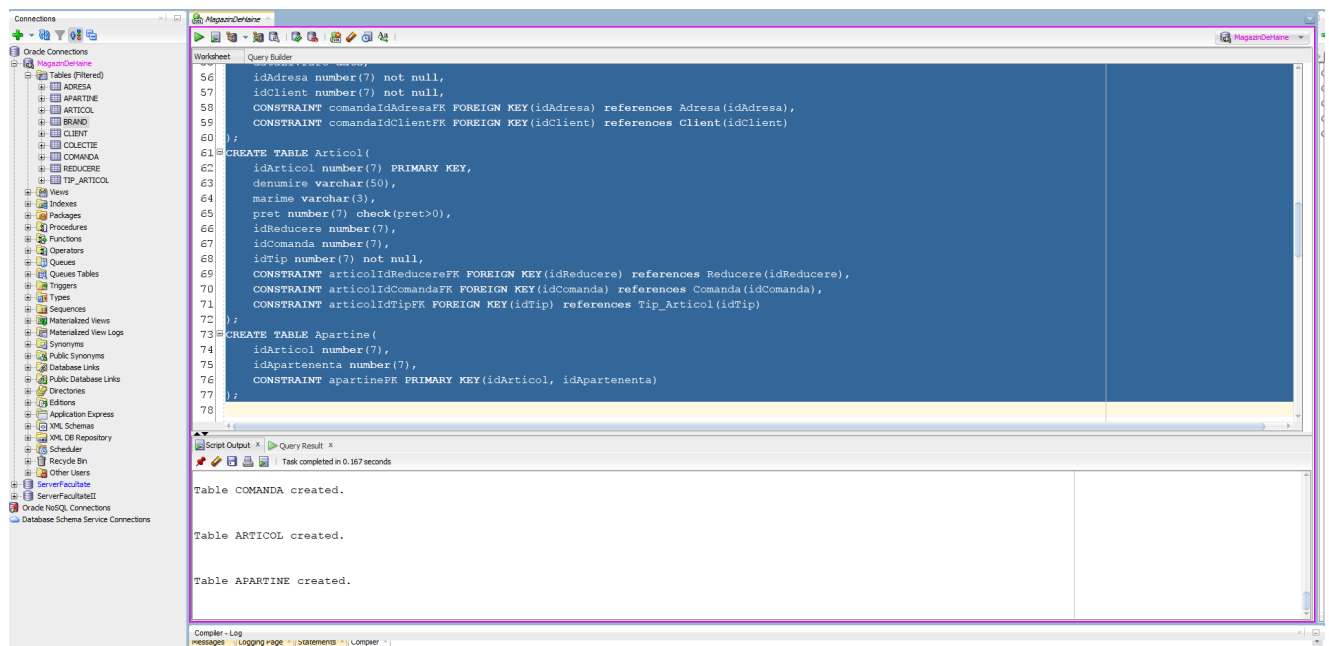
```
CREATE TABLE Adresa(  
    idAdresa number(7) PRIMARY KEY,  
    oras varchar(50),  
    strada varchar(50),  
    numar number(5),  
    bloc varchar(10),  
    scara varchar(10),  
    apartament number(5)  
);  
CREATE TABLE Client(  
    idClient number(7) PRIMARY KEY,  
    nume varchar(50),  
    prenume varchar(50),  
    telefon varchar(15) unique,  
    idAdresa number(7) not null,  
    CONSTRAINT clientIdAdresaFK FOREIGN KEY(idAdresa) references  
Adresa(idAdresa)  
);  
CREATE TABLE Tip_Articol(  
    idTip number(7) PRIMARY KEY,  
    denumire varchar(50) unique,  
    sex char(1)  
  
);  
CREATE TABLE Reducere(  
    idReducere number(7) PRIMARY KEY,  
    discount number(3),  
    numePromotie varchar(100)  
);  
CREATE TABLE Brand(  
    idBrand number(7) PRIMARY KEY,  
    denumire varchar(50) unique,  
    telefon varchar(15) unique,  
    idAdresa number(7) not null unique,  
    CONSTRAINT brandIdAdresaFK FOREIGN KEY(idAdresa) references  
Adresa(idAdresa)  
);  
CREATE TABLE Colectie(  
    idColectie number(7) PRIMARY KEY,  
    denumire varchar(100),  
    dataAparitie date,  
    idBrand number(7) not null,  
    CONSTRAINT colectieIdBrandFK FOREIGN KEY(idBrand) references
```

```

Brand(idBrand)
);
CREATE TABLE Comanda(
    idComanda number(7) PRIMARY KEY,
    dataPlecare date,
    dataLivrare date,
    idAdresa number(7) not null,
    idClient number(7) not null,
    CONSTRAINT comandaIdAdresaFK FOREIGN KEY(idAdresa) references
Adresa(idAdresa),
    CONSTRAINT comandaIdClientFK FOREIGN KEY(idClient) references
Client(idClient)
);
CREATE TABLE Articol(
    idArticol number(7) PRIMARY KEY,
    denumire varchar(50),
    marime varchar(3),
    pret number(7) check(pret>0),
    idReducere number(7),
    idComanda number(7),
    idTip number(7) not null,
    CONSTRAINT articolIdReducereFK FOREIGN KEY(idReducere) references
Reducere(idReducere),
    CONSTRAINT articolIdComandaFK FOREIGN KEY(idComanda) references
Comanda(idComanda),
    CONSTRAINT articolIdTipFK FOREIGN KEY(idTip) references
Tip_Articol(idTip)
);
CREATE TABLE Apartine(
    idArticol number(7),
    idApartenenta number(7),
    CONSTRAINT apartinePK PRIMARY KEY(idArticol, idApartenenta)
);

```

Următoarea captură de ecran conține rularea codului pentru crearea tabelelor:



## Cerința 5.

Codul pentru inserarea datelor:

```
alter session set nls_date_format = 'dd/mm/yyyy';

INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(1, 'Bucuresti', 'Strada Dorobantilor', 2, 'C10',
'2', 20);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(2, 'Bucuresti', 'Strada Revolutiei', 44, 'A1', '3',
33);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(3, 'Bucuresti', 'Strada Mihai Eminescu', 63, 'B6',
'5', 101);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(4, 'Bucuresti', 'Strada Dealului', 1, 'B9', '6',
43);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(5, 'Brasov', 'Strada Ion Creanga', 23, 'F12', '4',
46);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(6, 'Brasov', 'Strada Sforii', 50, null, null,
null);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(7, 'Brasov', 'Strada Mihai', 33, null, null, null);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(8, 'Constanta', 'Strada Mare', 53, null, null,
null);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(9, 'Ploiesti', 'Strada Mihai Viteazul', 52, null,
null, null);
INSERT INTO Adresa(idAdresa, oras, strada, numar, bloc, scara,
apartament) VALUES(10, 'Oradea', 'Strada Calarasilor', 12, null, null,
null);
INSERT INTO Client(idClient, nume, prenume, telefon, idAdresa)
VALUES(1, 'Ionescu', 'Gigel', '0732103441', 1);
INSERT INTO Client(idClient, nume, prenume, telefon, idAdresa)
VALUES(2, 'Popescu', 'Ion', '0730286884', 2);
INSERT INTO Client(idClient, nume, prenume, telefon, idAdresa)
VALUES(3, 'Podaru', 'Andrei', '0734940265', 3);
INSERT INTO Client(idClient, nume, prenume, telefon, idAdresa)
VALUES(4, 'Petcu', 'Alexandru', '0735273197', 4);
INSERT INTO Client(idClient, nume, prenume, telefon, idAdresa)
VALUES(5, 'Radu', 'Robert', '0730784294', 5);

INSERT INTO Tip_Articol(idTip, denumire, sex) VALUES(1, 'palton',
```

```

'M');
INSERT INTO Tip_Articol(idTip, denumire, sex) VALUES(2, 'pantalon',
'M');
INSERT INTO Tip_Articol(idTip, denumire, sex) VALUES(3, 'fusta', 'F');
INSERT INTO Tip_Articol(idTip, denumire, sex) VALUES(4, 'rochie',
'F');
INSERT INTO Tip_Articol(idTip, denumire, sex) VALUES(5, 'tricou',
'M');

INSERT INTO Reducere(idReducere, discount, numePromotie) VALUES(1, 10,
'Reducerile Lunare');
INSERT INTO Reducere(idReducere, discount, numePromotie) VALUES(2, 20,
'Reduceri De Primavara');
INSERT INTO Reducere(idReducere, discount, numePromotie) VALUES(3, 50,
'Mega Promotia');
INSERT INTO Reducere(idReducere, discount, numePromotie) VALUES(4, 15,
'Reducerile saptamanii');
INSERT INTO Reducere(idReducere, discount, numePromotie) VALUES(5, 30,
'Reducerile zilei');

INSERT INTO Brand(idBrand, denumire, telefon, idAdresa) VALUES(1,
'Zara', '0731106672', 6);
INSERT INTO Brand(idBrand, denumire, telefon, idAdresa) VALUES(2,
'HnM', '0733012535', 7);
INSERT INTO Brand(idBrand, denumire, telefon, idAdresa) VALUES(3,
'CnA', '0731373199', 8);
INSERT INTO Brand(idBrand, denumire, telefon, idAdresa) VALUES(4,
'Tommy Hilfiger', '0734168623', 9);
INSERT INTO Brand(idBrand, denumire, telefon, idAdresa) VALUES(5,
'Levis', '0734451270', 10);

INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(1, 'Colectia de Primavara', '01/04/2021', 1);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(2, 'Colectia Johnny Smith', '01/04/2021', 1);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(3, 'Colectia Light', '03/10/2019', 2);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(4, 'Colectia Coloreaza-ti Viata', '03/10/2019', 2);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(5, 'Colectia Culori Deschise', '22/11/2018', 3);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(6, 'Colectia Primaverii', '22/11/2018', 3);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(7, 'Colectia Super', '13/06/2017', 4);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(8, 'Colectia Chique', '13/06/2017', 4);
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(9, 'Colectia Levis Classic', '25/05/2016', 5);

```



```
INSERT INTO Colectie(idColectie, denumire, dataAparitie, idBrand)
VALUES(10, 'Colectia Tricourilor', '25/05/2016', 5);
```

```
INSERT INTO Comanda(idComanda, dataPlecare, dataLivrare, idAdresa,
idClient) VALUES(1, '05/04/2021', '07/04/2021', 1, 1);
INSERT INTO Comanda(idComanda, dataPlecare, dataLivrare, idAdresa,
idClient) VALUES(2, '04/05/2021', '06/05/2021', 2, 2);
INSERT INTO Comanda(idComanda, dataPlecare, dataLivrare, idAdresa,
idClient) VALUES(3, '09/04/2021', '11/04/2021', 3, 3);
INSERT INTO Comanda(idComanda, dataPlecare, dataLivrare, idAdresa,
idClient) VALUES(4, '14/05/2021', '23/05/2021', 4, 4);
INSERT INTO Comanda(idComanda, dataPlecare, dataLivrare, idAdresa,
idClient) VALUES(5, '20/06/2021', '27/06/2021', 5, 5);
```

```
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(1, 'Palton rosu', 'M', 100, 1, 1, 1);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(2, 'Palton rosu', 'XXL', 100, 1, null, 1);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(3, 'Palton alb', 'L', 120, 1, null, 1);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(4, 'Palton alb', 'M', 120, 1, null, 1);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(5, 'Pantaloni stofa negri', 'S', 90, 2, 2,
2);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(6, 'Pantaloni stofa negri', 'XL', 90, 2,
null, 2);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(7, 'Blugi albastri', 'XS', 80, 3, null, 2);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(8, 'Blugi albastri', 'L', 80, 3, 3, 2);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(9, 'Fusta blugi', 'S', 70, 2, null, 3);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(10, 'Fusta blugi', 'M', 70, 2, null, 3);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(11, 'Fusta galbena bumbac', 'M', 65, 1, null,
3);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(12, 'Fusta galbena bumbac', 'L', 65, 1, null,
3);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(13, 'Rochie albastra bumbac', 'XL', 200, 4,
1, 4);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(14, 'Rochie albastra bumbac', 'XXL', 200, 4,
4, 4);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
```

```

idComanda, idTip) VALUES(15, 'Rochie alba elastica', 'XL', 230, 5,
null, 4);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(16, 'Rochie alba elastica', 'S', 230, 5,
null, 4);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(17, 'Tricou alb simplu', 'XS', 50, 5, 5, 5);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(18, 'Tricou alb simplu', 'M', 50, 5, null,
5);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(19, 'Tricou verde imprimat', 'L', 140, 1,
null, 5);
INSERT INTO Articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip) VALUES(20, 'Tricou verde imprimat', 'S', 140, 1,
null, 5);

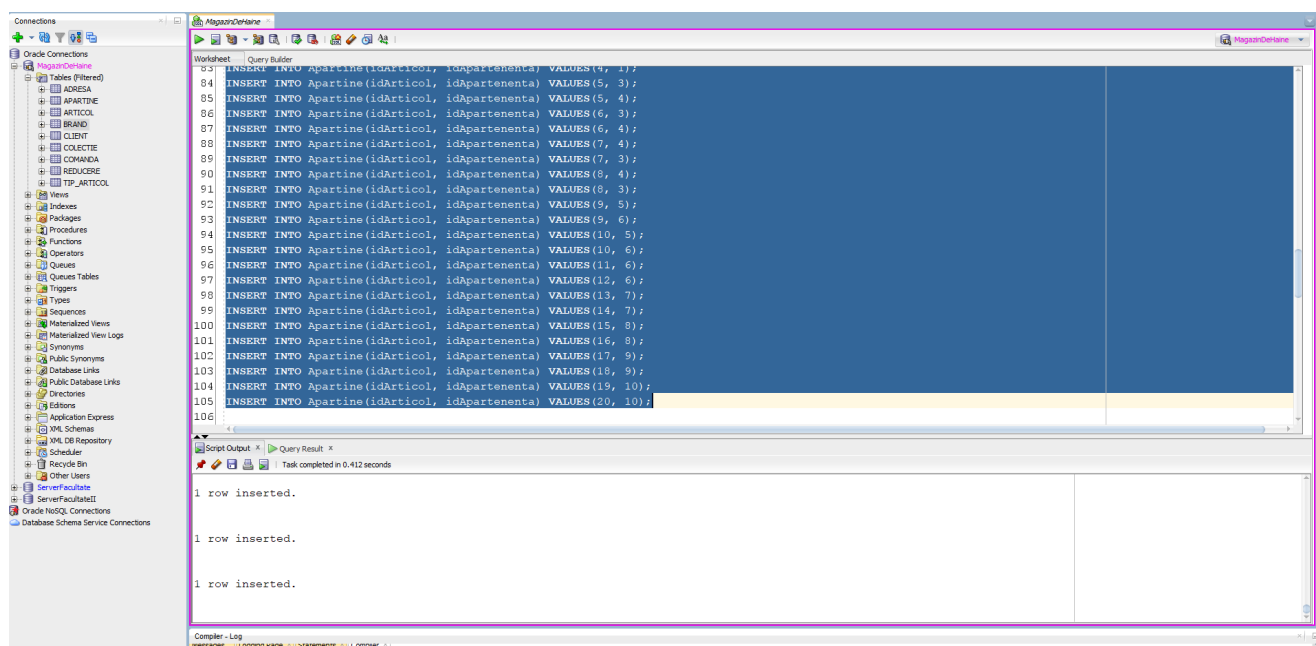
```

```

INSERT INTO Apartine(idArticol, idApartenenta) VALUES(1, 1);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(1, 2);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(2, 1);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(2, 2);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(3, 2);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(3, 1);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(4, 2);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(4, 1);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(5, 3);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(5, 4);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(6, 3);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(6, 4);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(7, 4);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(7, 3);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(8, 4);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(8, 3);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(9, 5);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(9, 6);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(10, 5);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(10, 6);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(11, 6);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(12, 6);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(13, 7);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(14, 7);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(15, 8);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(16, 8);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(17, 9);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(18, 9);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(19, 10);
INSERT INTO Apartine(idArticol, idApartenenta) VALUES(20, 10);

```

Următoare captură de ecran conține rularea codului pentru inserarea datelor:



## Cerința 6.

Cerința în limbaj natural: Să se scrie un subprogram care afișează, pentru fiecare client, prețul total al comenzilor pe care le-a efectuat.

Am folosit un tabel imbricat și un varray pentru a rezolva cerința.

Codul subprogramului:

```
-- Sa se scrie un subprogram care afiseaza, pentru fiecare client,
pretul total al comenzilor pe care le-a efectuat
CREATE OR REPLACE PROCEDURE PretTotalComenziPerClient IS
    TYPE recordClienti IS RECORD (idCl client.idClient%type, nume
client.nume%type, prenume client.prenume%type);
    TYPE tabelClienti IS TABLE OF recordClienti;
    TYPE varrayComenzi IS VARRAY(1000) OF comanda.idComanda%type;

    clienti tabelClienti := tabelClienti();
    comenzi varrayComenzi := varrayComenzi();

    idxCom NUMBER;
    pretTotal articol.pret%type;

BEGIN

    SELECT idClient, nume, prenume
    BULK COLLECT INTO clienti
    FROM Client;

    FOR idxClient IN clienti.first..clienti.last LOOP
        pretTotal := 0;
        idxCom := 0;

        FOR idCom IN (
            SELECT cmd.idComanda
            FROM Comanda cmd
            WHERE cmd.idClient = clienti(idxClient).idCl
        )
        LOOP
            idxCom := idxCom + 1;
            comenzi.extend;
            comenzi(idxCom) := idCom.idComanda;

        END LOOP;

        FOR idx IN comenzi.first..comenzi.last LOOP
            FOR pr IN ( SELECT art.pret FROM Articol art WHERE
comenzi(idx) = art.idComanda ) LOOP
```

```
        pretTotal := pretTotal + pr.pret;
    END LOOP;
END LOOP;

    dbms_output.put_line('Clientul ' || clienti(idxCliet).nume ||
' ' || clienti(idxCliet).prenume || ' are comenzi cu valoarea totala:
' || pretTotal);

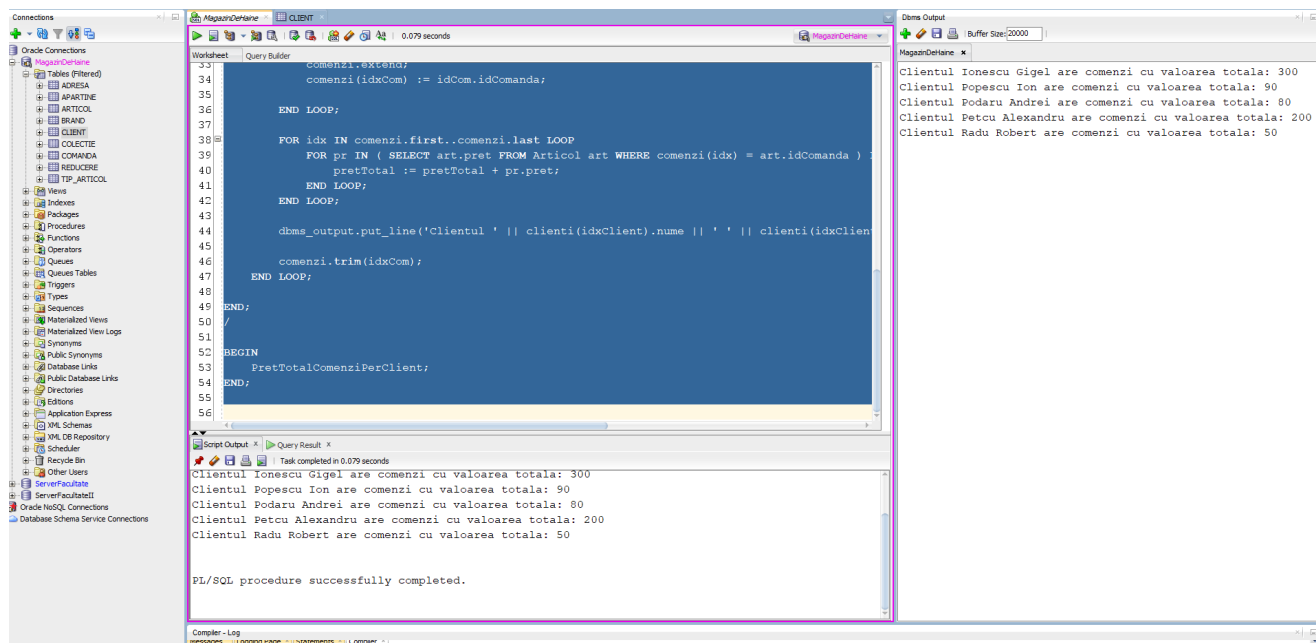
    comenzi.trim(idxCom);
END LOOP;
END;
```

Următoarea captură de ecran arată apelarea subprogramului:

BEGIN

PretTotalComenziPerClient;

END;

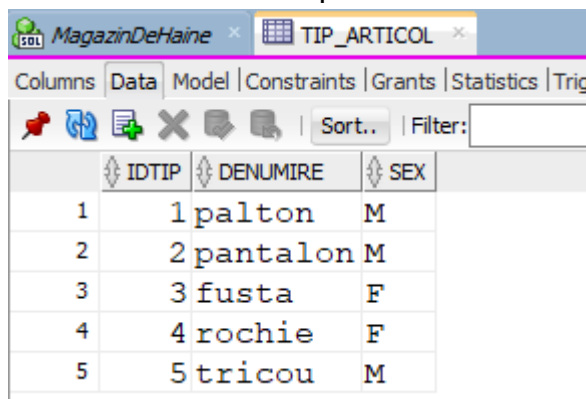


## Cerința 7.

Cerința în limbaj natural: Să se scrie un subprogram care afișează numărul de articole în stoc pentru fiecare tip de articol.

Am folosit un cursor explicit pentru a rezolva cerința.

Datele din tabelul tip\_articol:



The screenshot shows the SQL Developer interface with the 'TIP\_ARTICOL' table selected. The 'Data' tab is active, displaying a table with three columns: IDTIP, DENUMIRE, and SEX. The data is as follows:

| IDTIP | DENUMIRE   | SEX |
|-------|------------|-----|
| 1     | 1 palton   | M   |
| 2     | 2 pantalon | M   |
| 3     | 3 fusta    | F   |
| 4     | 4 rochie   | F   |
| 5     | 5 tricou   | M   |

Codul subprogramului:

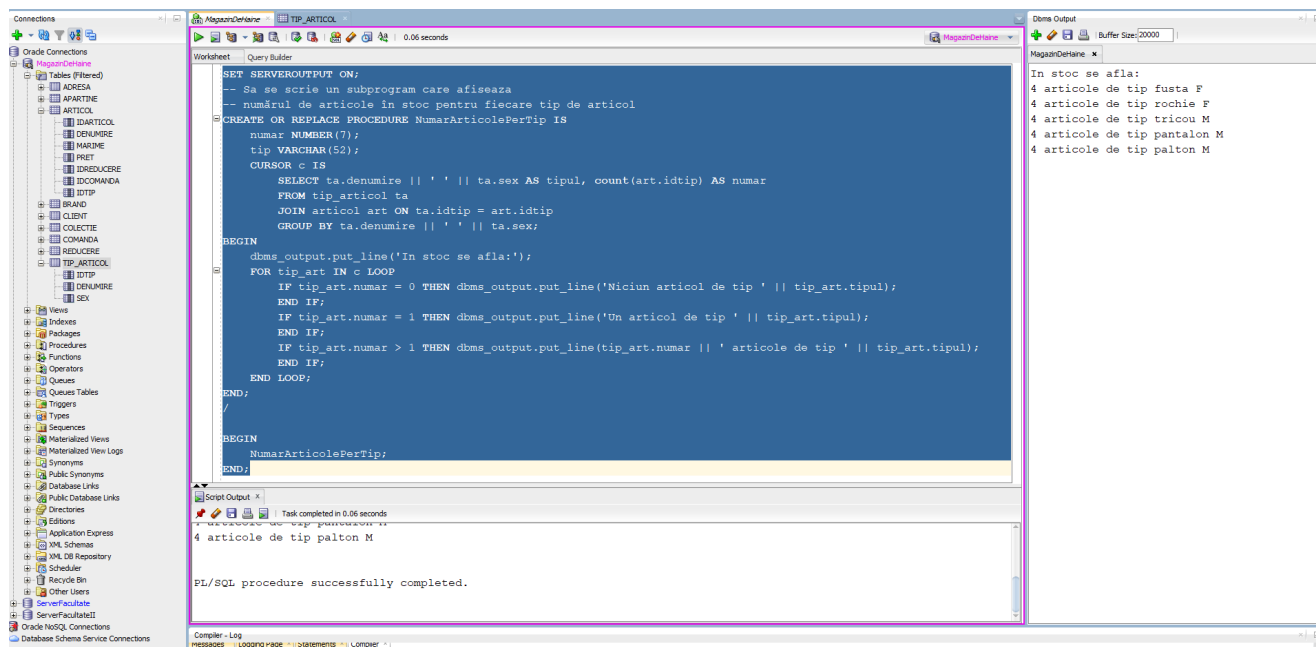
```
-- Sa se scrie un subprogram care afiseaza
-- numărul de articole în stoc pentru fiecare tip de articol
CREATE OR REPLACE PROCEDURE NumarArticolePerTip IS
    numar NUMBER(7);
    tip VARCHAR(52);
    CURSOR c IS
        SELECT ta.denumire || ' ' || ta.sex AS tipul, count(art.idtip)
AS numar
        FROM tip_articol ta
        JOIN articol art ON ta.idtip = art.idtip
        GROUP BY ta.denumire || ' ' || ta.sex;
BEGIN
    dbms_output.put_line('In stoc se afla:');
    FOR tip_art IN c LOOP
        IF tip_art.numar = 0 THEN dbms_output.put_line('Niciun articol
de tip ' || tip_art.tipul);
        END IF;
        IF tip_art.numar = 1 THEN dbms_output.put_line('Un articol de
tip ' || tip_art.tipul);
        END IF;
        IF tip_art.numar > 1 THEN dbms_output.put_line(tip_art.numar
|| ' articole de tip ' || tip_art.tipul);
        END IF;
    END LOOP;
END;
```

Următoarea captură de ecran arată apelarea subprogramului:

```
BEGIN
```

```
    NumarArticolePerTip;
```

```
END;
```





## Cerința 8.

Cerința în limbaj natural: Să se scrie o funcție care returnează numărul de articole care fac parte din colecții lansate într-un interval dat.

Codul funcției:

```
-- Sa se scrie o functie care returneaza numarul de articole
-- care fac parte din colectii lansate intr-un interval dat
CREATE OR REPLACE FUNCTION ArticoleColecțieiInterval(inceput DATE,
sfarsit DATE)
RETURN NUMBER IS

    numar NUMBER(7);
    exceptie_interval EXCEPTION;
BEGIN

    IF inceput > sfarsit THEN RAISE exceptie_interval;
    END IF;

    SELECT count(DISTINCT art.idarticol) INTO numar
    FROM articol art
    JOIN apartine apt ON apt.idarticol = art.idarticol
    JOIN colectie cole ON apt.idapartenenta = cole.idcolectie
    WHERE cole.dataaparitie >= inceput AND cole.dataaparitie <=
sfarsit;

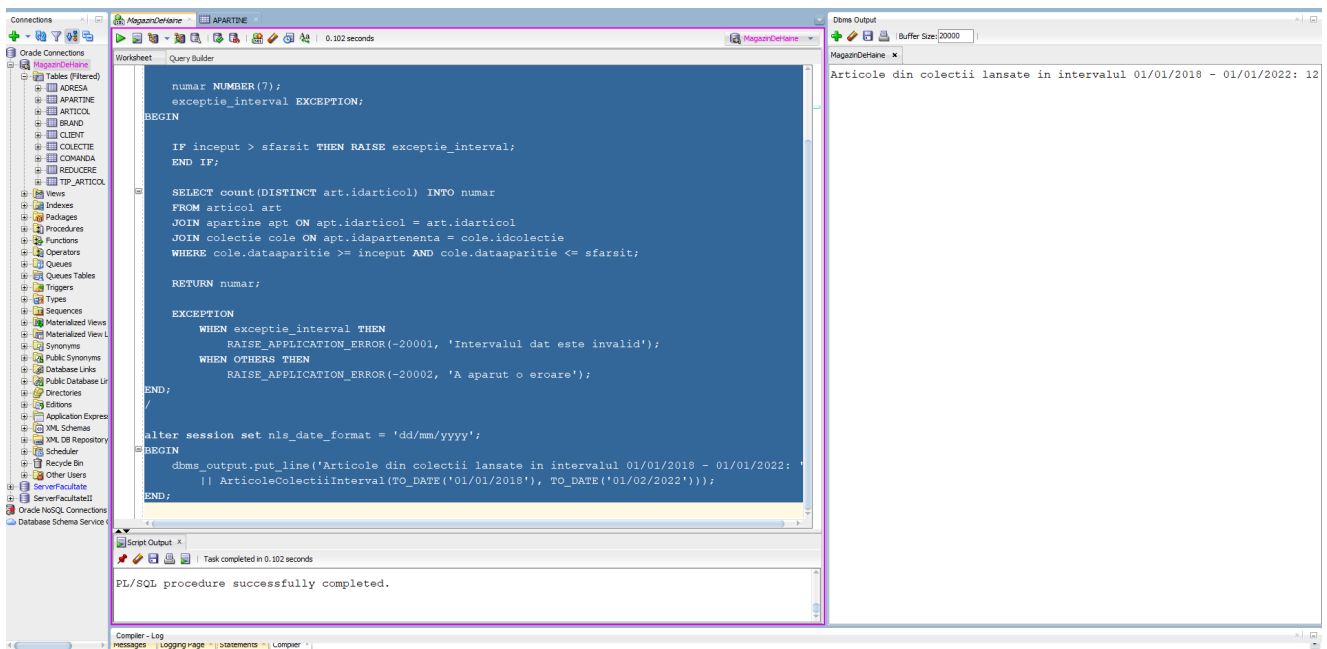
    RETURN numar;

EXCEPTION
    WHEN exceptie_interval THEN
        RAISE_APPLICATION_ERROR(-20001, 'Intervalul dat este
invalid');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'A aparut o eroare');
END;
```

## Apelarea funcției:

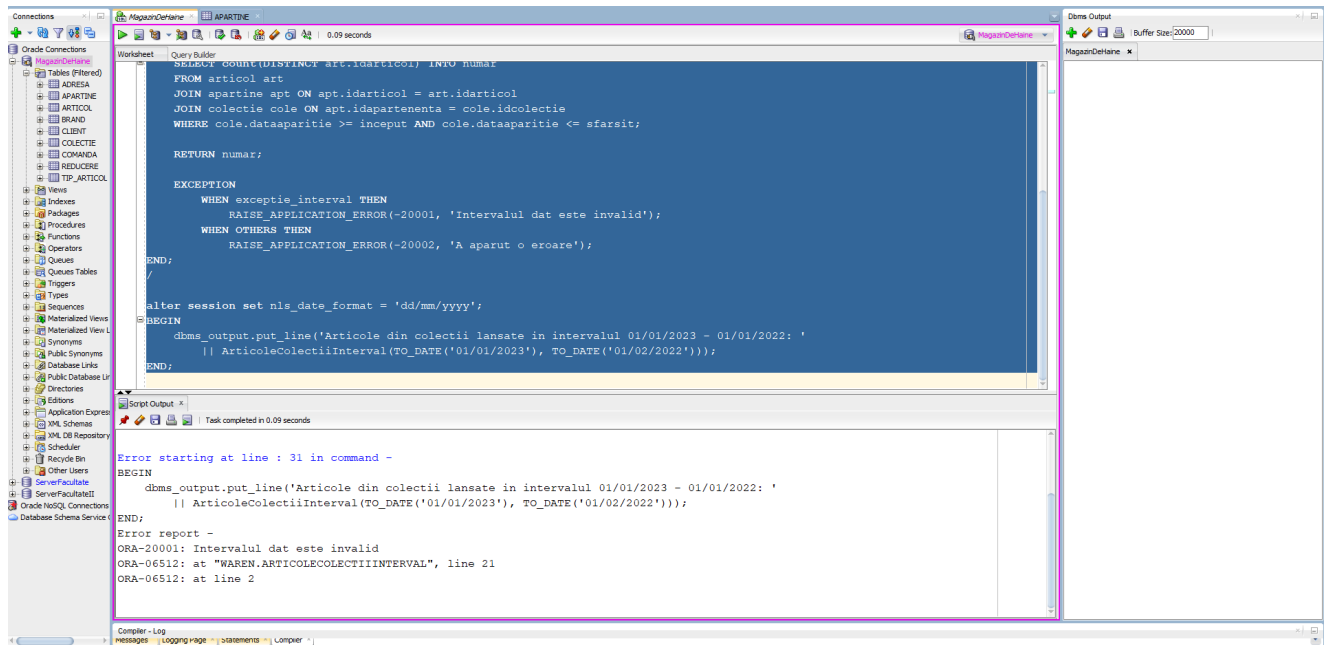
- Apel funcțional

```
alter session set nls_date_format = 'dd/mm/yyyy';
BEGIN
    dbms_output.put_line('Articole din colectii lansate in
intervalul 01/01/2018 - 01/01/2022: '
        || ArticleColectiiInterval(TO_DATE('01/01/2018'),
TO_DATE('01/01/2022')))
END;
```



- Interval invalid

```
alter session set nls_date_format = 'dd/mm/yyyy';
BEGIN
    dbms_output.put_line('Articole din colectii lansate in
intervalul 01/01/2023 - 01/01/2022: '
        || ArticoleColectiiInterval(TO_DATE('01/01/2023'),
TO_DATE('01/01/2022')));
END;
```



## Cerința 9.

Cerința în limbaj natural: Să se scrie o procedură care afișează, pentru un anumit client( nume și prenume), brand-ul de la care a comandat cele mai multe articole.

Codul procedurii:

```
-- Sa se scrie o procedura care afiseaza, pentru un anumit client(nume
si prenume),
-- brand-ul de la care a comandat cele mai multe articole
CREATE OR REPLACE PROCEDURE BrandPreferatClient(numeClient
client.nume%type, prenumeClient client.prenume%type) IS

    idCl client.idClient%type;
    numarMaxim number(7);
    numeBrand brand.denumire%type;
    clientExista BOOLEAN:=false;

BEGIN

    SELECT cli.idClient INTO idCl
    FROM client cli
    WHERE cli.nume = numeClient and cli.prenume = prenumeClient;

    clientExista := true;

    SELECT max(count(bra.denumire)) INTO numarMaxim
    FROM client cli
    JOIN comanda com ON com.idclient = cli.idclient
    JOIN articol art ON art.idcomanda = com.idcomanda
    JOIN apartine apt ON apt.idarticol = art.idarticol
    JOIN colectie cole ON apt.idapartenenta = cole.idcolectie
    JOIN brand bra ON bra.idbrand = cole.idbrand
    WHERE cli.idclient = idCl
    GROUP BY bra.denumire;

    SELECT *
    INTO numarMaxim, numeBrand
    FROM (SELECT count(bra.denumire), bra.denumire
    FROM client cli
    JOIN comanda com ON com.idclient = cli.idclient
    JOIN articol art ON art.idcomanda = com.idcomanda
    JOIN apartine apt ON apt.idarticol = art.idarticol
    JOIN colectie cole ON apt.idapartenenta = cole.idcolectie
    JOIN brand bra ON bra.idbrand = cole.idbrand
    WHERE cli.idclient = idCl
    GROUP BY bra.denumire
```

```

HAVING count(bra.denumire) = numarMaxim)
WHERE rownum = 1;

dbms_output.put_line('Clientul ' || numeClient || ' ' ||
prenumeClient);
dbms_output.put_line('a comandat majoritatea articolelor de la
brand-ul: ' || numeBrand);

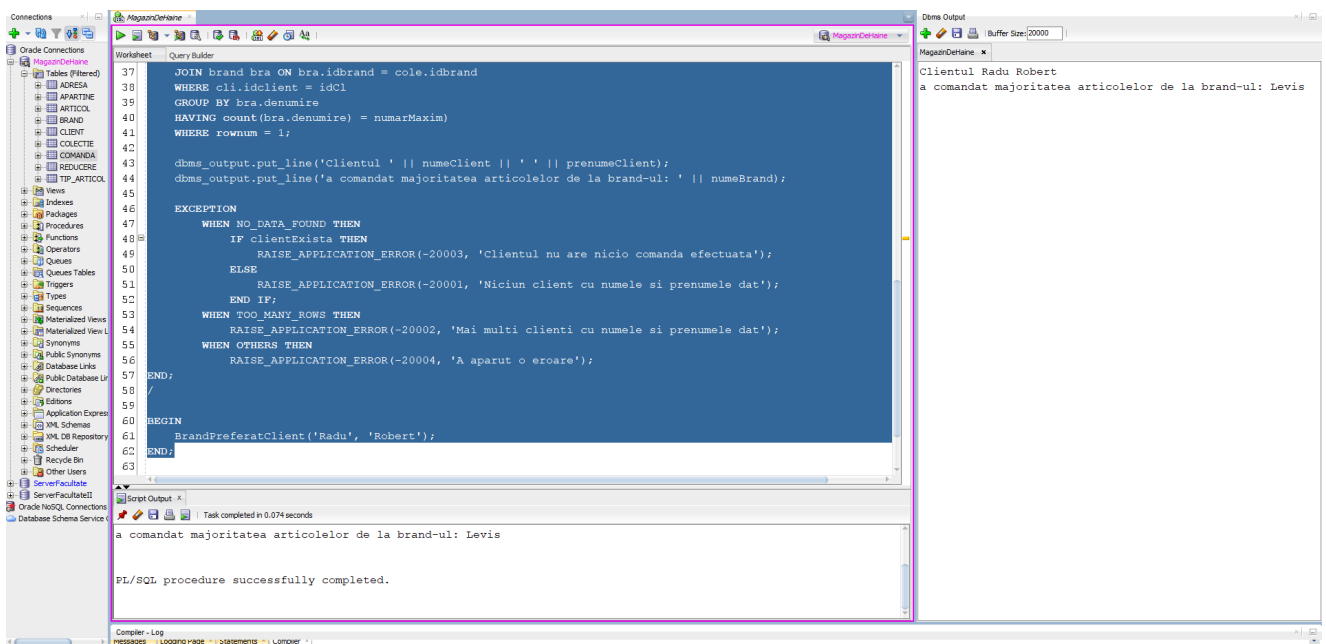
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        IF clientExista THEN
            RAISE_APPLICATION_ERROR(-20003, 'Clientul nu are nicio
comanda efectuata');
        ELSE
            RAISE_APPLICATION_ERROR(-20001, 'Niciun client cu
numele si prenumele dat');
        END IF;
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20002, 'Mai multi clienti cu
numele si prenumele dat');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20004, 'A aparut o eroare');
END;

```

## Apelarea procedurii:

- Apel funcțional

```
BEGIN  
    BrandPreferatClient('Radu', 'Robert');  
END;
```

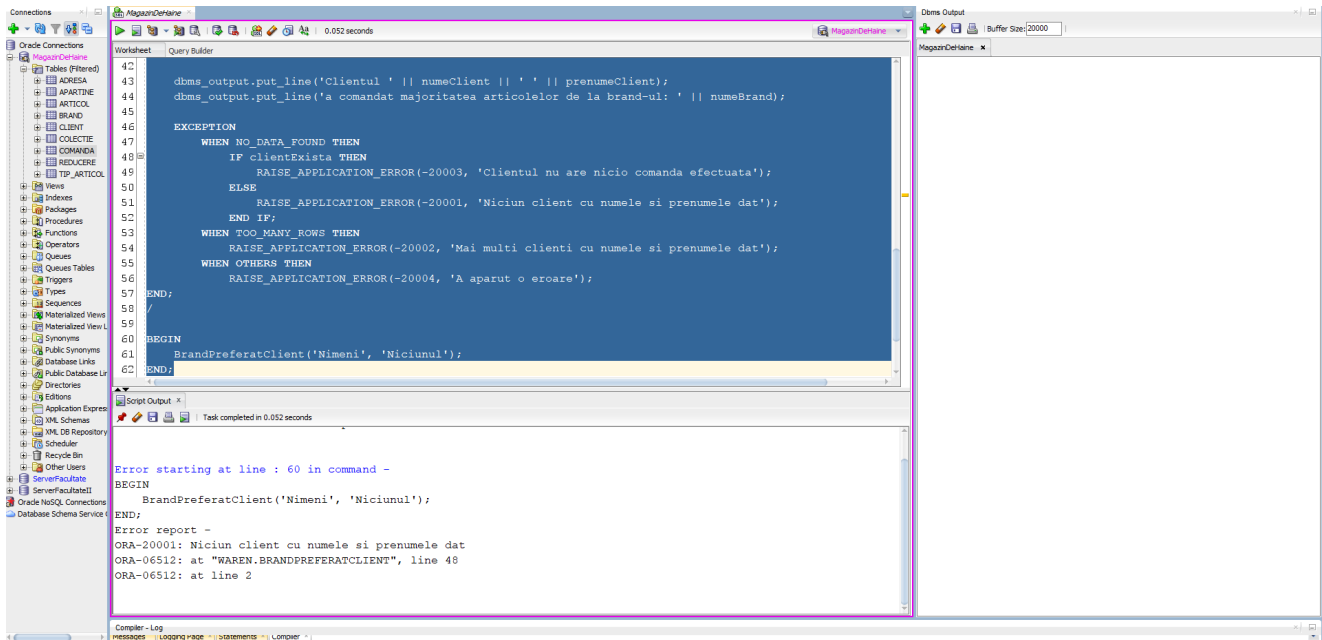


- Client inexistent

BEGIN

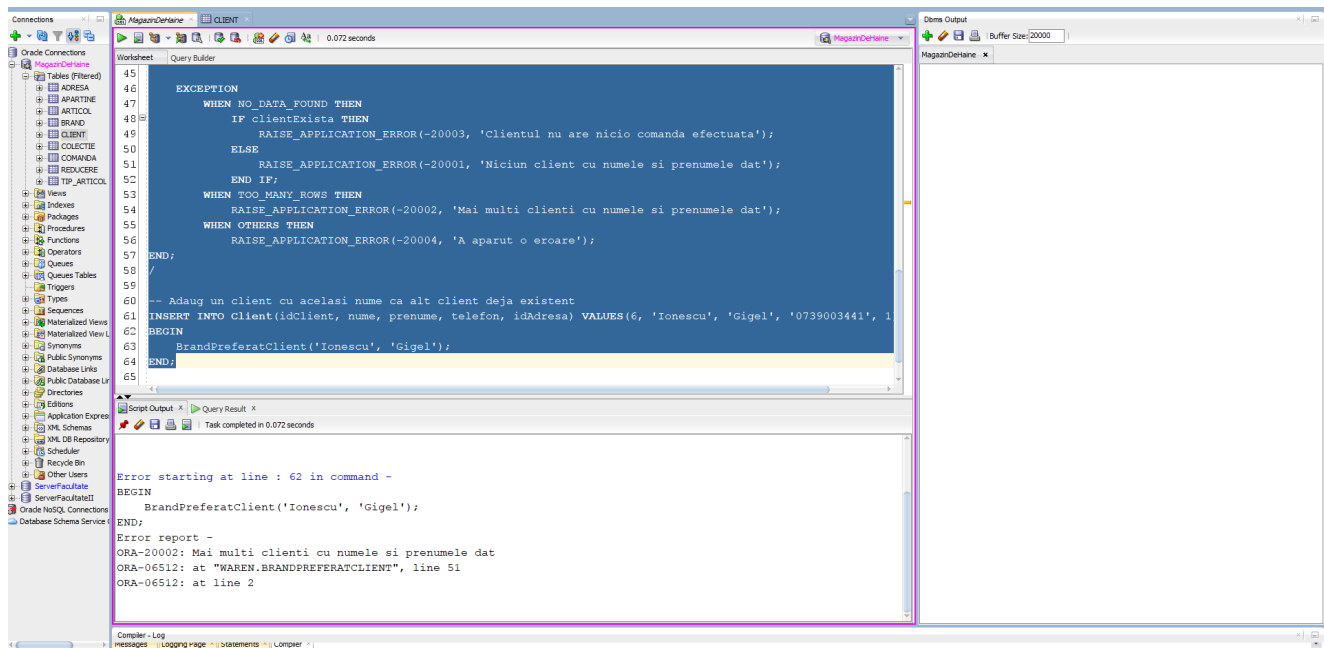
BrandPreferatClient('Nimeni', 'Niciunul');

END;



- Clienți cu același nume

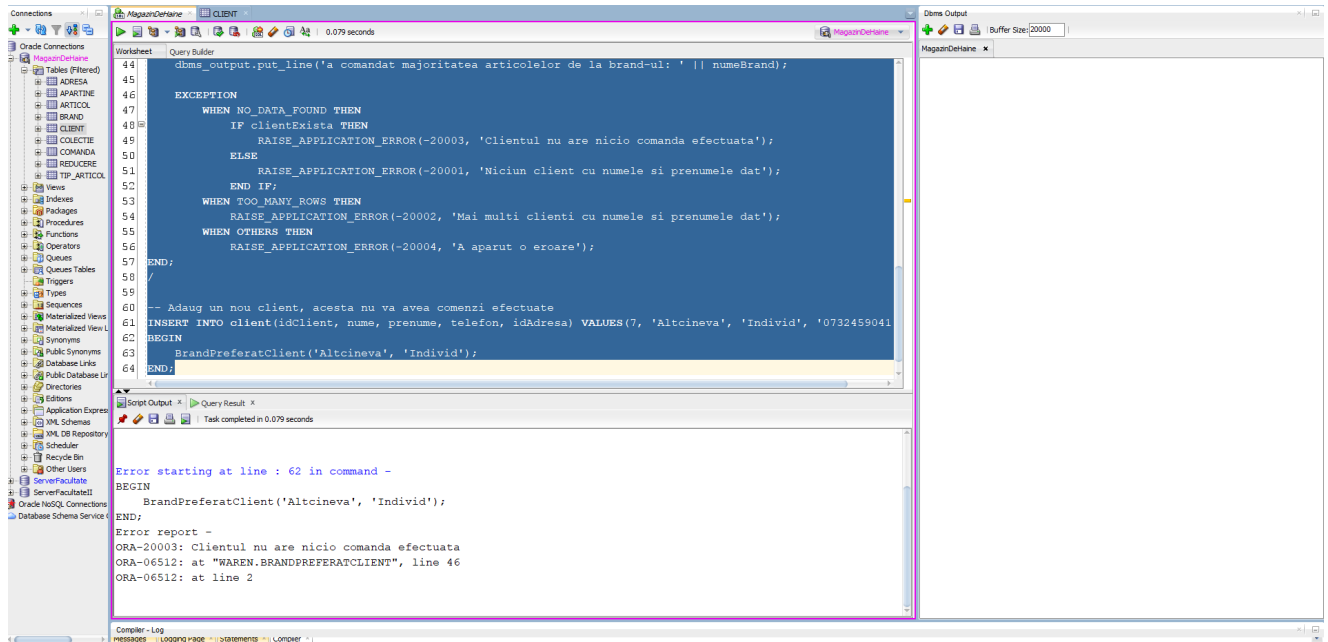
```
-- Adaug un client cu acelasi nume ca alt client deja existent
INSERT INTO Client(idClient, nume, prenume, telefon, idAdresa)
VALUES(6, 'Ionescu', 'Gigel', '0739003441', 1);
BEGIN
    BrandPreferatClient('Ionescu', 'Gigel');
END;
```





- Client fără comenzi

```
-- Adaug un nou client, acesta nu va avea comenzi efectuate
INSERT INTO client(idClient, nume, prenume, telefon, idAdresa)
VALUES(7, 'Altcineva', 'Individ', '0732459041', 1);
BEGIN
    BrandPreferatClient('Altcineva', 'Individ');
END;
```



Am șters apoi cei doi clienti adăugați pentru ultimele două exemple:

```
DELETE FROM client WHERE idClient > 5;
```

## Cerinta 10.

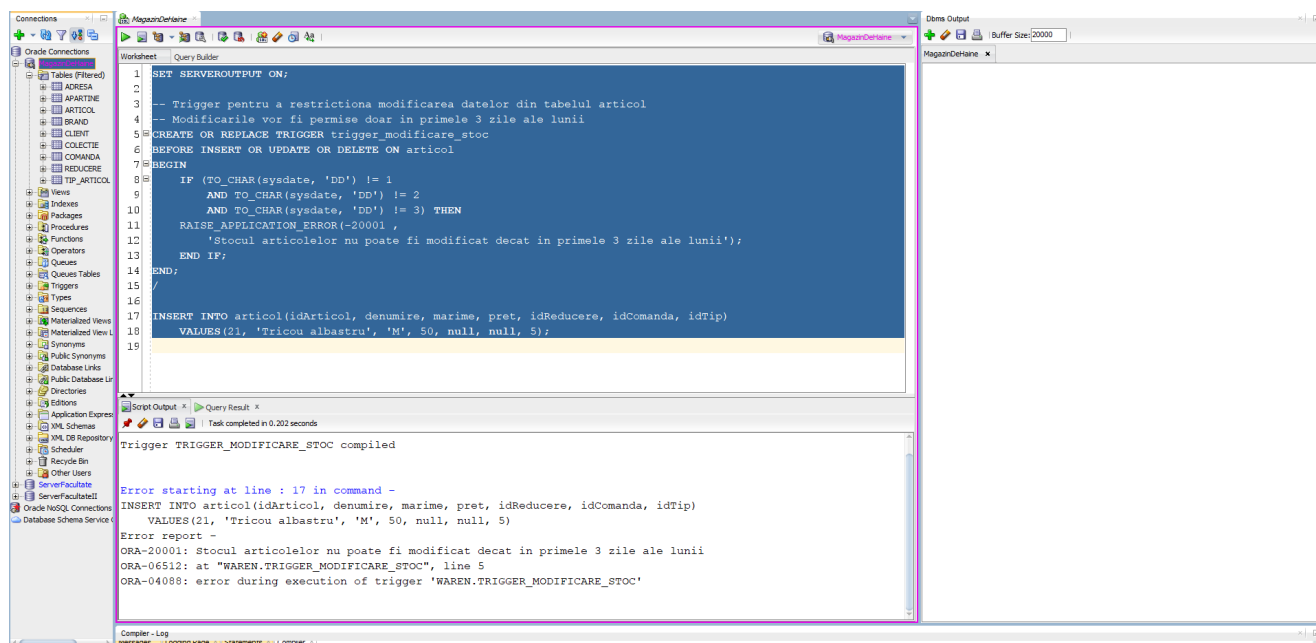
Am creat un trigger la nivel de comandă care să restricționeze modificările asupra tabelului articol astfel încat acestea să fie permise doar în primele 3 zile ale fiecărei luni.

Codul trigger-ului:

```
-- Trigger pentru a restrictiona modificarea datelor din tabelul
articol
-- Modificarile vor fi permise doar in primele 3 zile ale lunii
CREATE OR REPLACE TRIGGER trigger_modificare_stoc
BEFORE INSERT OR UPDATE OR DELETE ON articol
BEGIN
    IF (TO_CHAR(sysdate, 'DD') != 1
        AND TO_CHAR(sysdate, 'DD') != 2
        AND TO_CHAR(sysdate, 'DD') != 3) THEN
        RAISE_APPLICATION_ERROR(-20001 ,
            'Stocul articolelor nu poate fi modificat decat in primele 3
zile ale lunii');
    END IF;
END;
```

## Declanșarea trigger-ului:

```
INSERT INTO articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip)
VALUES(21, 'Tricou albastru', 'M', 50, null, null, 5);
```



## Cerinta 11.

Am creat un trigger la nivel de linie care să permită doar litere și spații în denumirile articolelor din tabelul articol.

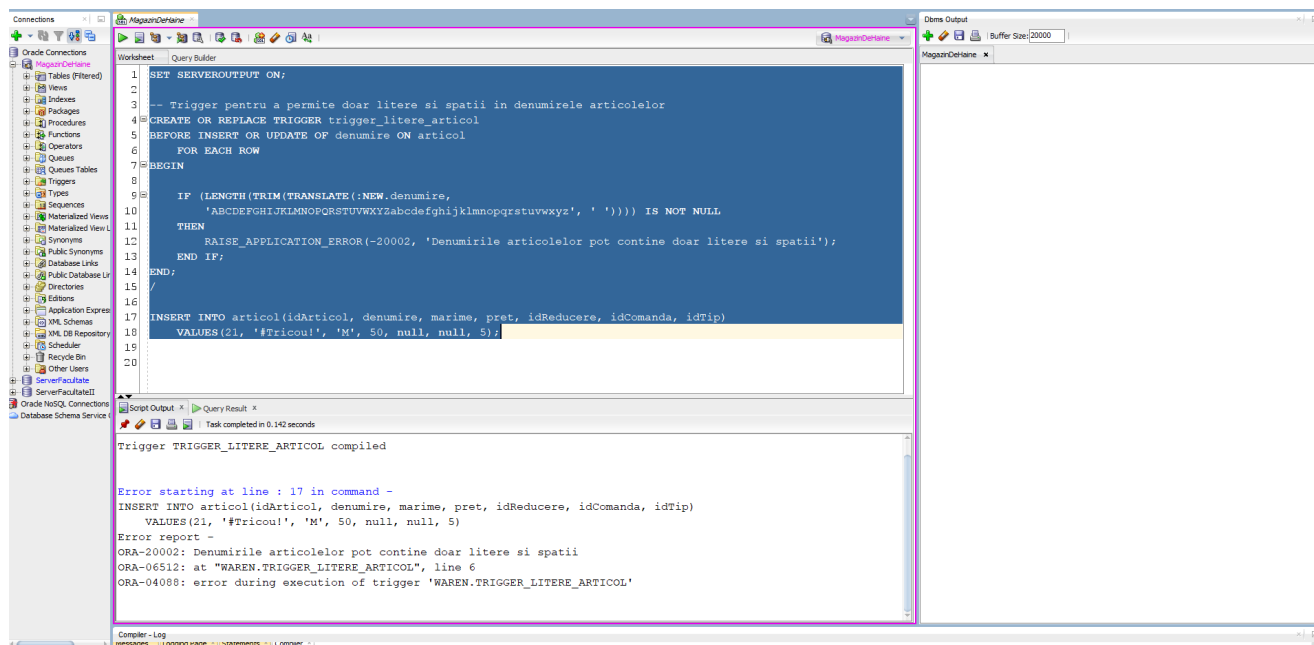
Codul trigger-ului:

```
-- Trigger pentru a permite doar litere si spatii in denumirele articolelor
CREATE OR REPLACE TRIGGER trigger_litere_articol
BEFORE INSERT OR UPDATE OF denumire ON articol
    FOR EACH ROW
BEGIN

    IF (LENGTH(TRIM(TRANSLATE(:NEW.denumire,
        'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz', '
    ')))) IS NOT NULL
    THEN
        RAISE_APPLICATION_ERROR(-20002, 'Denumirile articolelor pot
        contine doar litere si spatii');
    END IF;
END;
```

## Declanșarea trigger-ului:

```
INSERT INTO articol(idArticol, denumire, marime, pret, idReducere,
idComanda, idTip)
VALUES(21, '#Tricou!', 'M', 50, null, null, 5);
```



Este de menționat că trigger-ul acesta nu poate fi declanșat decât în primele trei zile ale lunii dacă trigger-ul de la cerința 10 nu este disabled/dropped.

## Cerința 12.

Am creat un tabel în care se vor înregistra informații despre modificări asupra bazei de date în urma acțiunilor LDD. Se va reține data la momentul acțiunii, acțiunea, numele tabelului și utilizatorul care a efectuat acțiunea.

Crearea tabelului:

```
CREATE TABLE log_ldd(  
    data DATE,  
    actiune VARCHAR2(50),  
    tabel VARCHAR2(50),  
    utilizator VARCHAR2(50)  
);
```

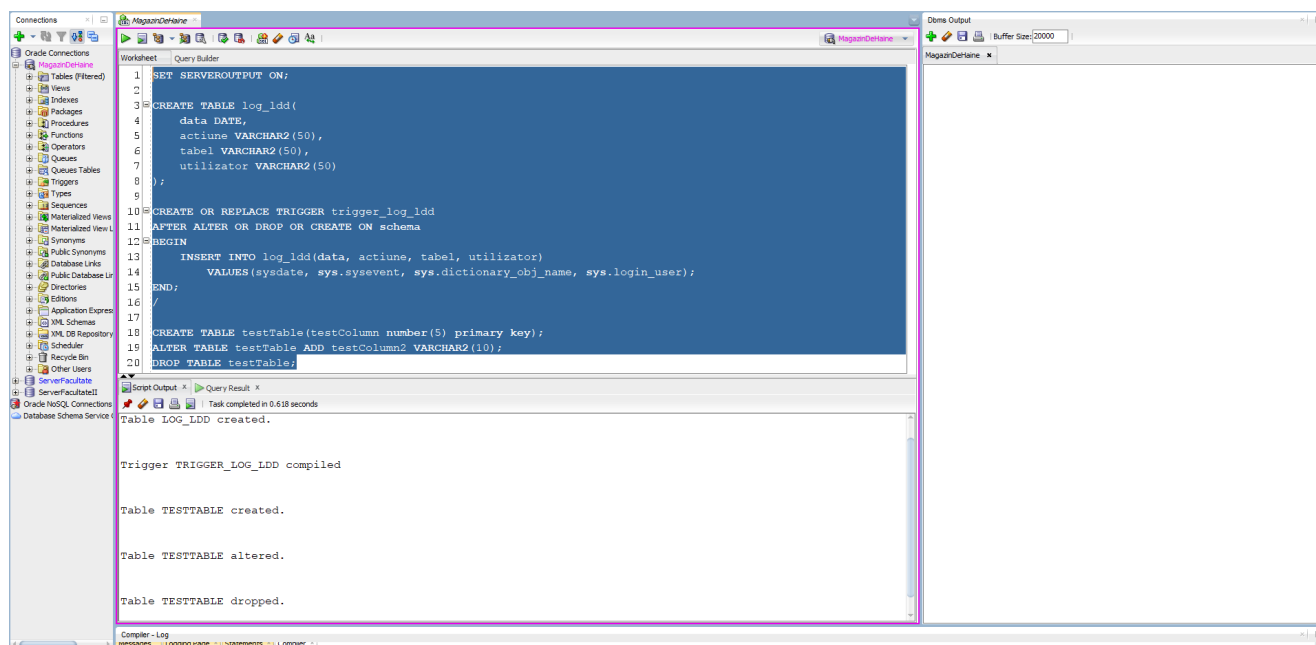
Pentru a insera înregistrări în tabelul de mai sus am folosit un trigger de tip LDD.

Codul trigger-ului:

```
CREATE OR REPLACE TRIGGER trigger_log_ldd  
AFTER ALTER OR DROP OR CREATE ON schema  
BEGIN  
    INSERT INTO log_ldd(data, actiune, tabel, utilizator)  
        VALUES(sysdate, sys.sysevent, sys.dictionary_obj_name,  
sys.login_user);  
END;
```

## Declanșarea trigger-ului:

```
CREATE TABLE testTable(testColumn number(5) primary key);  
ALTER TABLE testTable ADD testColumn2 VARCHAR2(10);  
DROP TABLE testTable;
```



Tabelul log\_ldd după ce am declanșat trigger-ul cu exemplul de mai sus:

The screenshot shows the 'Table Data' view for the 'LOG\_LDD' table. The table has four columns: DATA, ACTIUNE, TABEL, and UTILIZATOR. The data is as follows:

|   | DATA       | ACTIUNE | TABEL       | UTILIZATOR |
|---|------------|---------|-------------|------------|
| 1 | 04/01/2022 | CREATE  | SYS_C007125 | WAREN      |
| 2 | 04/01/2022 | CREATE  | TESTTABLE   | WAREN      |
| 3 | 04/01/2022 | ALTER   | TESTTABLE   | WAREN      |
| 4 | 04/01/2022 | DROP    | TESTTABLE   | WAREN      |

## Cerința 13.

Am creat un pachet în care am integrat toate obiectele definite în cadrul proiectului.

Codul pachetului:

```
CREATE OR REPLACE PACKAGE package_magazindehaine
IS
    PROCEDURE PretTotalComenziPerClient;
    PROCEDURE NumarArticolePerTip;
    FUNCTION ArticoleColectiiInterval(inceput DATE, sfarsit DATE)
        RETURN NUMBER;
    PROCEDURE BrandPreferatClient(umeClient client.ume%type,
prenumeClient client.prenume%type);

END;
/

CREATE OR REPLACE PACKAGE BODY package_magazindehaine
IS
    PROCEDURE PretTotalComenziPerClient IS
        TYPE recordClienti IS RECORD (idCl client.idClient%type,
ume client.ume%type, prenume client.prenume%type);
        TYPE tabelClienti IS TABLE OF recordClienti;
        TYPE varrayComenzi IS VARRAY(1000) OF
comanda.idComanda%type;

        clienti tabelClienti := tabelClienti();
        comenzi varrayComenzi := varrayComenzi();

        idxCom NUMBER;
        pretTotal articol.pret%type;

    BEGIN

        SELECT idClient, ume, prenume
        BULK COLLECT INTO clienti
        FROM Client;

        FOR idxClient IN clienti.first..clienti.last LOOP
            pretTotal := 0;
            idxCom := 0;

            FOR idCom IN (
                SELECT cmd.idComanda
                FROM Comanda cmd
```



```

        WHERE cmd.idClient = clienti(idxCliet).idCl
    )
    LOOP
        idxCom := idxCom + 1;
        comenzi.extend;
        comenzi(idxCom) := idCom.idComanda;

    END LOOP;

    FOR idx IN comenzi.first..comenzi.last LOOP
        FOR pr IN ( SELECT art.pret FROM Articol art WHERE
comenzi(idx) = art.idComanda ) LOOP
            pretTotal := pretTotal + pr.pret;
        END LOOP;
    END LOOP;

    dbms_output.put_line('Clientul ' ||
    clienti(idxCliet).nume || ' ' || clienti(idxCliet).prenume || ' are
comenzi cu valoarea totala: ' || pretTotal);

    comenzi.trim(idxCom);
END LOOP;
END;

PROCEDURE NumarArticolePerTip IS
    numar NUMBER(7);
    tip VARCHAR(52);
    CURSOR c IS
        SELECT ta.denumire || ' ' || ta.sex AS tipul,
count(art.idtip) AS numar
        FROM tip_articol ta
        JOIN articol art ON ta.idtip = art.idtip
        GROUP BY ta.denumire || ' ' || ta.sex;
BEGIN
    dbms_output.put_line('In stoc se afla:');
    FOR tip_art IN c LOOP
        IF tip_art.numar = 0 THEN dbms_output.put_line('Niciun
articol de tip ' || tip_art.tipul);
        END IF;
        IF tip_art.numar = 1 THEN dbms_output.put_line('Un
articol de tip ' || tip_art.tipul);
        END IF;
        IF tip_art.numar > 1 THEN
dbms_output.put_line(tip_art.numar || ' articole de tip ' ||
tip_art.tipul);
        END IF;
    END LOOP;
END;

```

```

FUNCTION ArticoleColectiiInterval(inceput DATE, sfarsit DATE)
RETURN NUMBER IS

    numar NUMBER(7);
    exceptie_interval EXCEPTION;
BEGIN

    IF inceput > sfarsit THEN RAISE exceptie_interval;
    END IF;

    SELECT count(DISTINCT art.idarticol) INTO numar
    FROM articol art
    JOIN apartine apt ON apt.idarticol = art.idarticol
    JOIN colectie cole ON apt.idapartenenta = cole.idcolectie
    WHERE cole.dataaparitie >= inceput AND cole.dataaparitie
<= sfarsit;

    RETURN numar;

EXCEPTION
    WHEN exceptie_interval THEN
        RAISE_APPLICATION_ERROR(-20001, 'Intervalul dat
este invalid');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'A aparut o
eroare');
END;

PROCEDURE BrandPreferatClient(numClient client.nume%type,
prenumeClient client.prenume%type) IS

    idCl client.idClient%type;
    numarMaxim number(7);
    numeBrand brand.denumire%type;
    clientExista BOOLEAN:=false;

BEGIN

    SELECT cli.idClient INTO idCl
    FROM client cli
    WHERE cli.nume = numClient and cli.prenume =
prenumeClient;

    clientExista := true;

    SELECT max(count(bra.denumire)) INTO numarMaxim
    FROM client cli
    JOIN comanda com ON com.idclient = cli.idclient
    JOIN articol art ON art.idcomanda = com.idcomanda

```

```

JOIN apartine apt ON apt.idarticol = art.idarticol
JOIN colectie cole ON apt.idapartenenta = cole.idcolectie
JOIN brand bra ON bra.idbrand = cole.idbrand
WHERE cli.idclient = idCl
GROUP BY bra.denumire;

```

```

SELECT *
INTO numarMaxim, numeBrand
FROM (SELECT count(bra.denumire), bra.denumire
FROM client cli
JOIN comanda com ON com.idclient = cli.idclient
JOIN articol art ON art.idcomanda = com.idcomanda
JOIN apartine apt ON apt.idarticol = art.idarticol
JOIN colectie cole ON apt.idapartenenta = cole.idcolectie
JOIN brand bra ON bra.idbrand = cole.idbrand
WHERE cli.idclient = idCl
GROUP BY bra.denumire
HAVING count(bra.denumire) = numarMaxim)
WHERE rownum = 1;

```

```

        dbms_output.put_line('Clientul ' || numeClient || ' ' ||
prenumeClient);

```

```

        dbms_output.put_line('a comandat majoritatea articolelor
de la brand-ul: ' || numeBrand);

```

```

EXCEPTION

```

```

    WHEN NO_DATA_FOUND THEN

```

```

        IF clientExista THEN

```

```

            RAISE_APPLICATION_ERROR(-20003, 'Clientul nu
are nicio comanda efectuata');

```

```

        ELSE

```

```

            RAISE_APPLICATION_ERROR(-20001, 'Niciun client
cu numele si prenumele dat');

```

```

        END IF;

```

```

    WHEN TOO_MANY_ROWS THEN

```

```

        RAISE_APPLICATION_ERROR(-20002, 'Mai multi clienti
cu numele si prenumele dat');

```

```

    WHEN OTHERS THEN

```

```

        RAISE_APPLICATION_ERROR(-20004, 'A aparut o
eroare');

```

```

    END;

```

```

END;

```

## Compilarea pachetului:

