Proiect Sisteme de Gestiune a Bazelor de Date,

# Ghidarcea Tudor-Alexandru, grupa 241

1. Aceasta baza de date va fi utila pentru gestionarea unui magazin si a resurselor sale (angajati, produse).Modelul de date va fi suficient pentru a intelege si controla buna functionare a acestui magazin, asigurand toate conditiile necesare pentru acesta.

Magazinul va avea o locatie fizica, si va reprezenta o entitate. Produsele, comenzile vor reprezenta entitati. O tranzactie poate avea un singur client si unul sau mai multe produse. Un client poate avea mai multe tranzactii, la fel si un produs. Magazinul va avea mai multe firme de curierat, in functie de zonele in care acestea livreaza. Clientii vor beneficia de diverse garantii, modalitati de plata si optiuni de finantare.

Scopul acestei baze de date este ca un magazin sa poata gestiona comenzi.

Un client va avea abilitatea sa faca o comanda, incluzand unul sau mai multe produse, ridicate din magazinul fizic sau prin livrare, utilizand o optiune de finantare/plata conform alegerii sale.

Un produs poate face parte din zero, una sau mai multe comenzi/tranzactii, si beneficiaza de un tip de garantie.

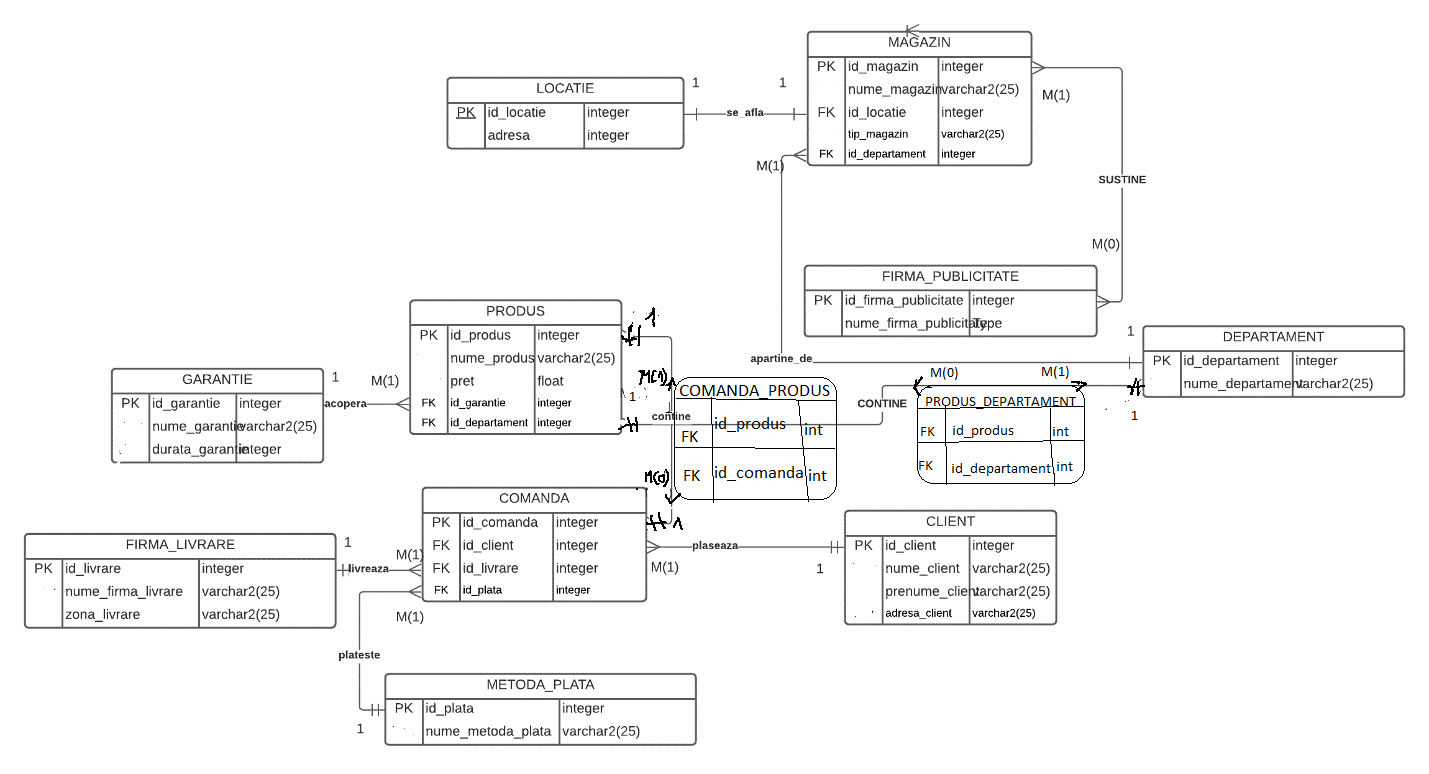
O tranzactie poate include unul sau mai multe produse, o metoda de plata/finantare si un singur client.

Un produs poate face parte din unul sau mai multe departamente(de exemplu un ou Kinder poate face parte si din categoria alimente, si din categoria jucarii; un scaun de masaj poate face parte si din categoria mobila si din categoria electronice).

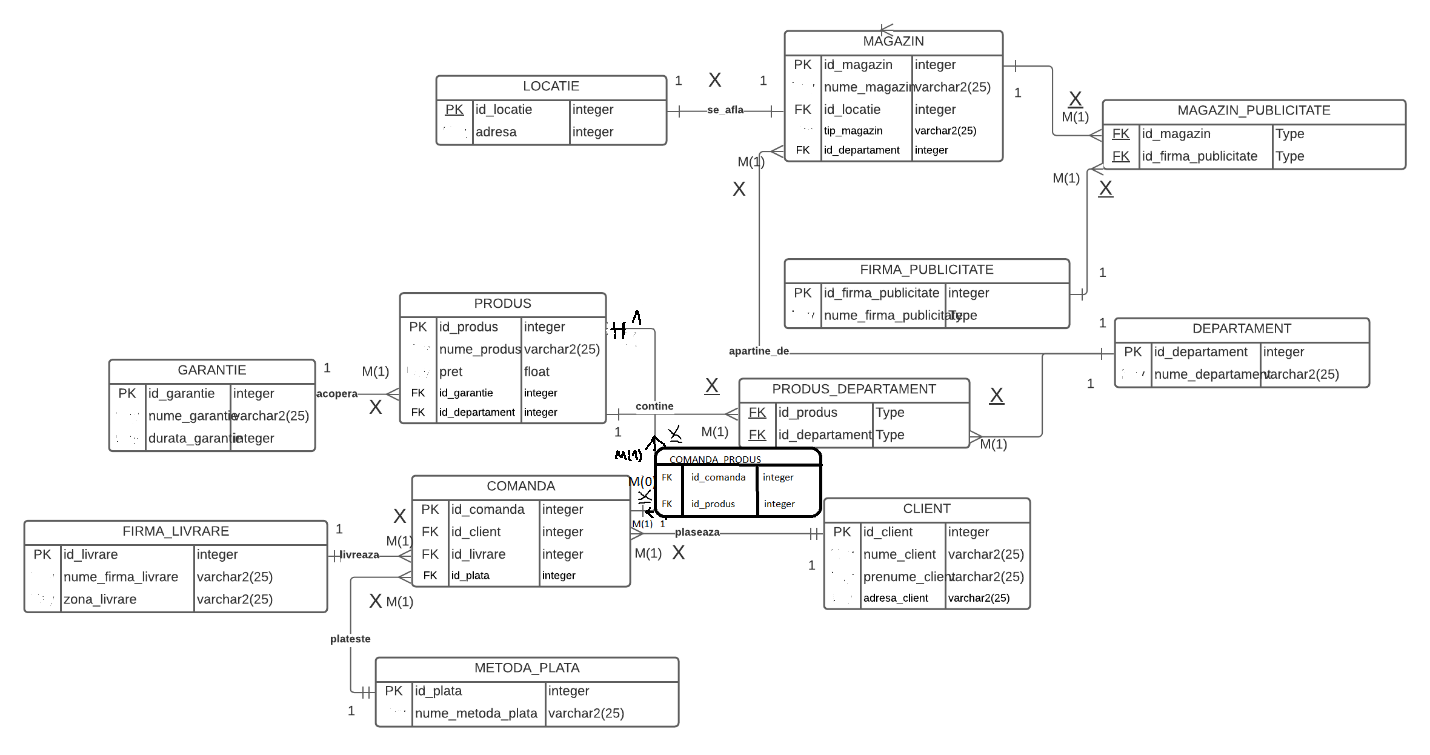
Un magazin poate face parte din unul sau mai multe departamente.

Un magazin poate fi sustinut de una sau mai multe firme de publicitate, care la randul lor pot sustine unul sau mai multe magazine.

1. Diagrama entitate-relatie:



1. Diagrama conceptuala:



1. Creare

create table PRODUS(

id\_produs int primary key,

nume\_produs varchar2(25),

pret float,

id\_garantie int,

id\_departament int)

create table GARANTIE(

id\_garantie int primary key,

nume\_garantie varchar2(25),

durata\_garantie int)

create table CLIENTI(

id\_client int primary key,

nume\_client varchar2(25),

prenume\_client varchar2(25),

adresa\_client varchar2(25))

create table FIRMA\_LIVRARE(

id\_livrare int primary key,

nume\_firma\_livrare varchar2(25),

zona\_livrare varchar2(25)

)

create table LOCATIE(

id\_locatie int primary key,

adresa varchar2(25))

create table FIRMA\_PUBLICITATE (

id\_firma\_publicitate int primary key,

nume\_firma\_publicitate varchar2(25))

create table DEPARTAMENT(

id\_departament int primary key,

nume\_departament varchar2(25))

create table MAGAZIN (

id\_magazin int primary key,

nume\_magazin varchar2(25),

id\_locatie references LOCATIE(id\_locatie),

tip\_magazin varchar2(25),

id\_departament references DEPARTAMENT(id\_departament))

create table METODA\_PLATA(

id\_plata int primary key,

nume\_metoda\_plata varchar2(25))

create table COMANDA (

id\_comanda int primary key,

id\_client references CLIENTI(id\_client),

id\_livrare references FIRMA\_LIVRARE(id\_livrare),

id\_plata references METODA\_PLATA(id\_plata) )

create table COMANDA\_PRODUS(

id\_comanda references COMANDA(id\_comanda),

id\_produs references PRODUS(id\_produs))

create table MAGAZIN\_PUBLICITATE(

id\_magazin references MAGAZIN(id\_magazin),

id\_firma\_publicitate references FIRMA\_PUBLICITATE(id\_firma\_publicitate))

create table PRODUS\_DEPARTAMENT(

id\_produs references PRODUS(id\_produs),

id\_departament references DEPARTAMENT(id\_departament))

1. Inserare

insert into PRODUS values (1, 'Sapun Dove', 5, 0, 2);

insert into PRODUS values (2, 'Televizor Samsung', 899.99, 2, 3);

insert into PRODUS values (3, 'Ghiveci flori', 39, 0, 4);

insert into PRODUS values (4, 'Ciocolata Milka', 4.99, 0, 1);

insert into PRODUS values (6, 'Pereche sosete', 19.99, 0, 2);

insert into PRODUS values (7, 'Manusa cuptor', 49.99, 0, 2);

insert into PRODUS values (8, 'Husa scaun', 89.99, 0, 4);

insert into PRODUS values (9, 'Termostat', 199.99, 1, 2);

insert into PRODUS values (10, 'Saltea gonflabila', 59.99, 0, 5);

insert into PRODUS values (5, 'Cartus imprimanta', 99.99, 1, 3);

insert into GARANTIE values (1, 'Garantie standard', 1);

insert into GARANTIE values (2, 'Garantie un an',12 );

insert into GARANTIE values (3, 'Garantie extinsa', 24);

insert into GARANTIE values (4, 'Garantie extra', 36);

insert into GARANTIE values (5, 'Garantie 5 ani',60 );

insert into CLIENTI values (1, 'Popescu', 'Ion', 'Calea Rahova 17');

insert into CLIENTI values (2, 'Gheorghitoaia', 'Antonie', 'Bulevardul Eroilor 13');

insert into CLIENTI values (3, 'Ionescu', 'Sorin', 'Libertatii 5');

insert into CLIENTI values (4, 'Maria', 'Alexandra', 'Aleea Revolutiei 11');

insert into CLIENTI values (10, 'Cretulescu', 'Alin', 'Calea Lalelelor 8');

insert into FIRMA\_LIVRARE values (1, 'Cargus', 'Timisoara');

insert into FIRMA\_LIVRARE values (2, 'Fan courier', 'Sibiu');

insert into FIRMA\_LIVRARE values (3, 'sc Romtrans srl', 'Gorj');

insert into FIRMA\_LIVRARE values (4, 'Glovo', 'Ramnicu Valcea');

insert into FIRMA\_LIVRARE values (5, 'Foodpanda', 'Tulcea');

insert into LOCATIE values (1, 'calea Bucuresti 33');

insert into LOCATIE values (2, 'ocolul Dumbravitei 21');

insert into LOCATIE values (3, 'Olteniei 51');

insert into LOCATIE values (4, 'Dunarii 75');

insert into LOCATIE values (5, 'Dorohoiului 19');

insert into FIRMA\_PUBLICITATE values (1, 'Brandaffair');

insert into FIRMA\_PUBLICITATE values (2, 'Infodesign');

insert into FIRMA\_PUBLICITATE values (3, '3d Art Agency');

insert into FIRMA\_PUBLICITATE values (4, 'Web Ventures');

insert into FIRMA\_PUBLICITATE values (5, 'Hueman Advertising');

insert into FIRMA\_PUBLICITATE values (6, 'the Syndicate');

insert into FIRMA\_PUBLICITATE values (7, 'FCB Bucharest');

insert into FIRMA\_PUBLICITATE values (8, 'MSA BRANDING AND COMMUNICATION');

insert into FIRMA\_PUBLICITATE values (9, 'Google Adsense');

insert into FIRMA\_PUBLICITATE values (10, 'Facebook');

insert into DEPARTAMENT values (1, 'Alimentare');

insert into DEPARTAMENT values (2, 'Casnice');

insert into DEPARTAMENT values (3, 'Electronice');

insert into DEPARTAMENT values (4, 'Mobila');

insert into DEPARTAMENT values (5, 'Rechizite');

insert into DEPARTAMENT values (6, 'Medicamente');

insert into DEPARTAMENT values (7, 'Produse de ingrijire');

insert into DEPARTAMENT values (8, 'Produse de gradinarit');

insert into DEPARTAMENT values (9, 'Produse de pescuit');

insert into DEPARTAMENT values (10, 'Carti');

insert into MAGAZIN values (1, 'MediaGalaxy', 1, 'Electronice', 3);

insert into MAGAZIN values (2, 'Mobexpert', 2, 'Mobila', 4);

insert into MAGAZIN values (3, 'Carrefour', 3, 'Alimentare', 1);

insert into MAGAZIN values (4, 'Faber Kastell', 4, 'Rechizite', 5);

insert into MAGAZIN values (5, 'Altex', 5, 'Electronice', 3);

insert into MAGAZIN values (6, 'Emag', 4, 'Electronice', 3);

insert into MAGAZIN values (7, 'Domo', 3, 'Casnice', 3);

insert into MAGAZIN values (8, 'House of Games', 2, 'Jocuri video', 3);

insert into MAGAZIN values (9, 'Billa', 1, 'Alimentare',1 );

insert into MAGAZIN values (10, 'Penny', 2, 'Alimentare', 1);

insert into METODA\_PLATA values (1, 'Cash');

insert into METODA\_PLATA values (2, 'Card');

insert into METODA\_PLATA values (3, 'Bonuri de masa');

insert into METODA\_PLATA values (4, 'Credit');

insert into METODA\_PLATA values (5, 'Cec');

insert into COMANDA values (1, 1, 1 , 1);

insert into COMANDA values (2, 2, 2, 2);

insert into COMANDA values (3, 3, 3, 3);

insert into COMANDA values (4, 4, 4 , 4);

insert into COMANDA values (5, 5, 5 , 5);

insert into COMANDA values (6,5,5,5);

insert into COMANDA values (7,4,4,4);

insert into COMANDA values (8,3,3,3);

insert into COMANDA values (9,2,2,2);

insert into COMANDA values (10,1,1,1);

insert into COMANDA\_PRODUS values (1,1);

insert into COMANDA\_PRODUS values (2,2);

insert into COMANDA\_PRODUS values (3,3);

insert into COMANDA\_PRODUS values (4,4);

insert into COMANDA\_PRODUS values (5,5);

insert into COMANDA\_PRODUS values (6,6);

insert into COMANDA\_PRODUS values (7,7);

insert into COMANDA\_PRODUS values (8,8);

insert into COMANDA\_PRODUS values (9,9);

insert into COMANDA\_PRODUS values (10,10);

insert into MAGAZIN\_PUBLICITATE values (1,1);

insert into MAGAZIN\_PUBLICITATE values (2,2);

insert into MAGAZIN\_PUBLICITATE values (3,3);

insert into MAGAZIN\_PUBLICITATE values (4,4);

insert into MAGAZIN\_PUBLICITATE values (5,5);

insert into MAGAZIN\_PUBLICITATE values (6,6);

insert into MAGAZIN\_PUBLICITATE values (7,7);

insert into MAGAZIN\_PUBLICITATE values (8,8);

insert into MAGAZIN\_PUBLICITATE values (9,9);

insert into MAGAZIN\_PUBLICITATE values (10,10);

insert into PRODUS\_DEPARTAMENT values (1,1);

insert into PRODUS\_DEPARTAMENT values (2,2);

insert into PRODUS\_DEPARTAMENT values (3,3);

insert into PRODUS\_DEPARTAMENT values (4,4);

insert into PRODUS\_DEPARTAMENT values (5,5);

insert into PRODUS\_DEPARTAMENT values (6,6);

insert into PRODUS\_DEPARTAMENT values (7,7);

insert into PRODUS\_DEPARTAMENT values (8,8);

insert into PRODUS\_DEPARTAMENT values (9,9);

insert into PRODUS\_DEPARTAMENT values (10,10);

1. Formulare in limbaj natural: O procedura PL/SQL care sa afiseze toate produsele din tabelele comanda\_produs si produs\_departament.

CREATE OR REPLACE PROCEDURE procedura\_6

(v\_produse\_2 produs\_departament.id\_produs%type, v\_produse\_5 comanda\_produs.id\_produs%type )

IS

TYPE varray\_produse\_2 IS VARRAY(255) OF produs\_departament.id\_produs%type;

TYPE nested\_table\_produse\_5 IS TABLE OF comanda\_produs.id\_produs%type;

varray\_numar\_dept varray\_produse\_2:=varray\_produse\_2();

nested\_table\_numar\_com nested\_table\_produse\_5:=nested\_table\_produse\_5();

BEGIN

SELECT id\_produs BULK COLLECT INTO varray\_numar\_dept FROM produs\_departament;

SELECT id\_produs BULK COLLECT INTO nested\_table\_numar\_com FROM comanda\_produs;

for l\_index in varray\_numar\_dept.FIRST..varray\_numar\_dept.LAST

LOOP

IF (v\_produse\_2>varray\_numar\_dept(l\_index)) THEN

DBMS\_OUTPUT.PUT\_LINE('produs din produs\_departament: '|| varray\_numar\_dept(l\_index));

END IF;

END LOOP;

for l\_index in nested\_table\_numar\_com.FIRST..nested\_table\_numar\_com.LAST

LOOP

IF (v\_produse\_5>nested\_table\_numar\_com(l\_index)) THEN

DBMS\_OUTPUT.PUT\_LINE('produs din comanda\_produs: '|| nested\_table\_numar\_com(l\_index));

END IF;

END LOOP;

END procedura\_6;

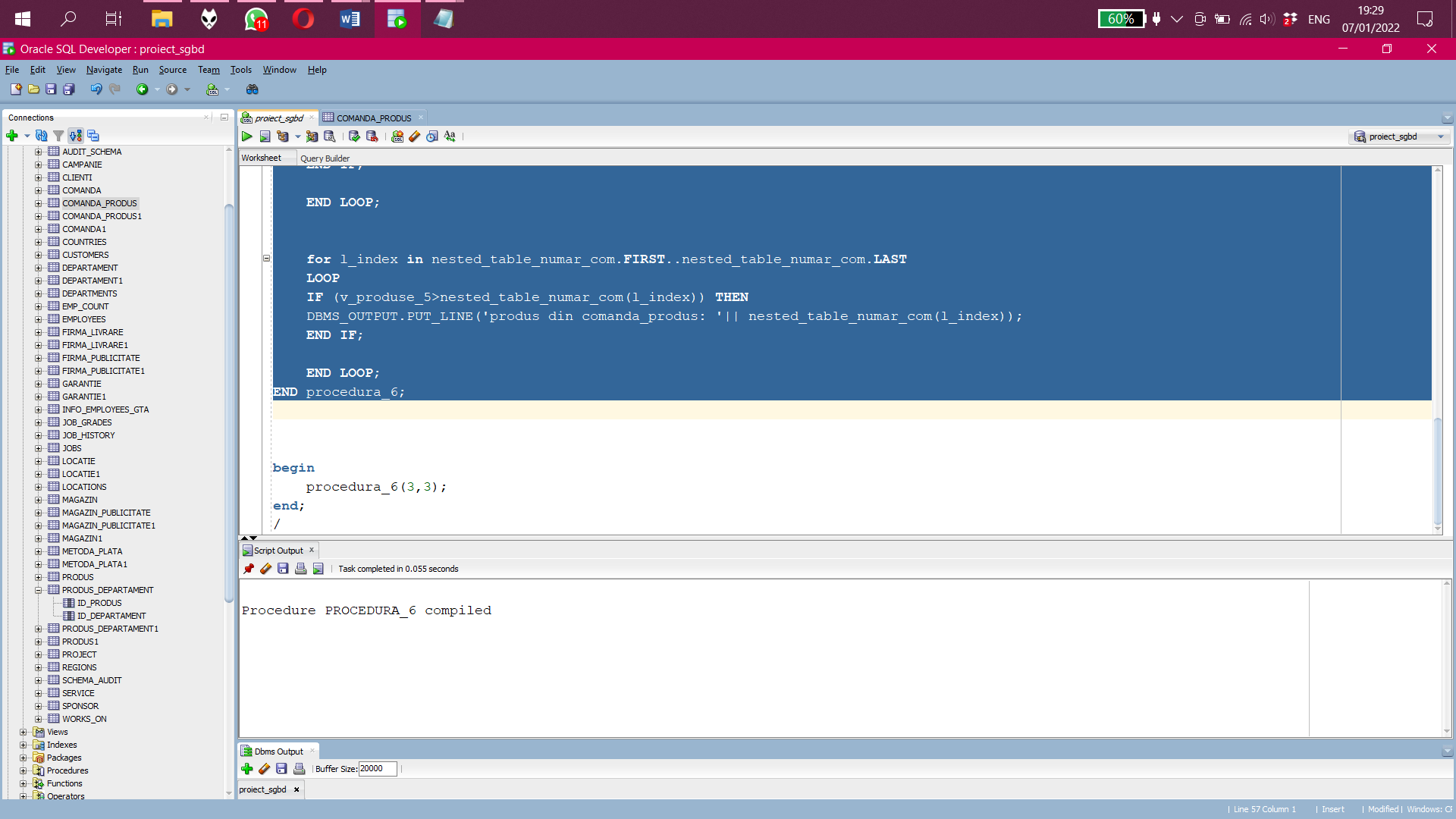
/\*apel:\*/

begin

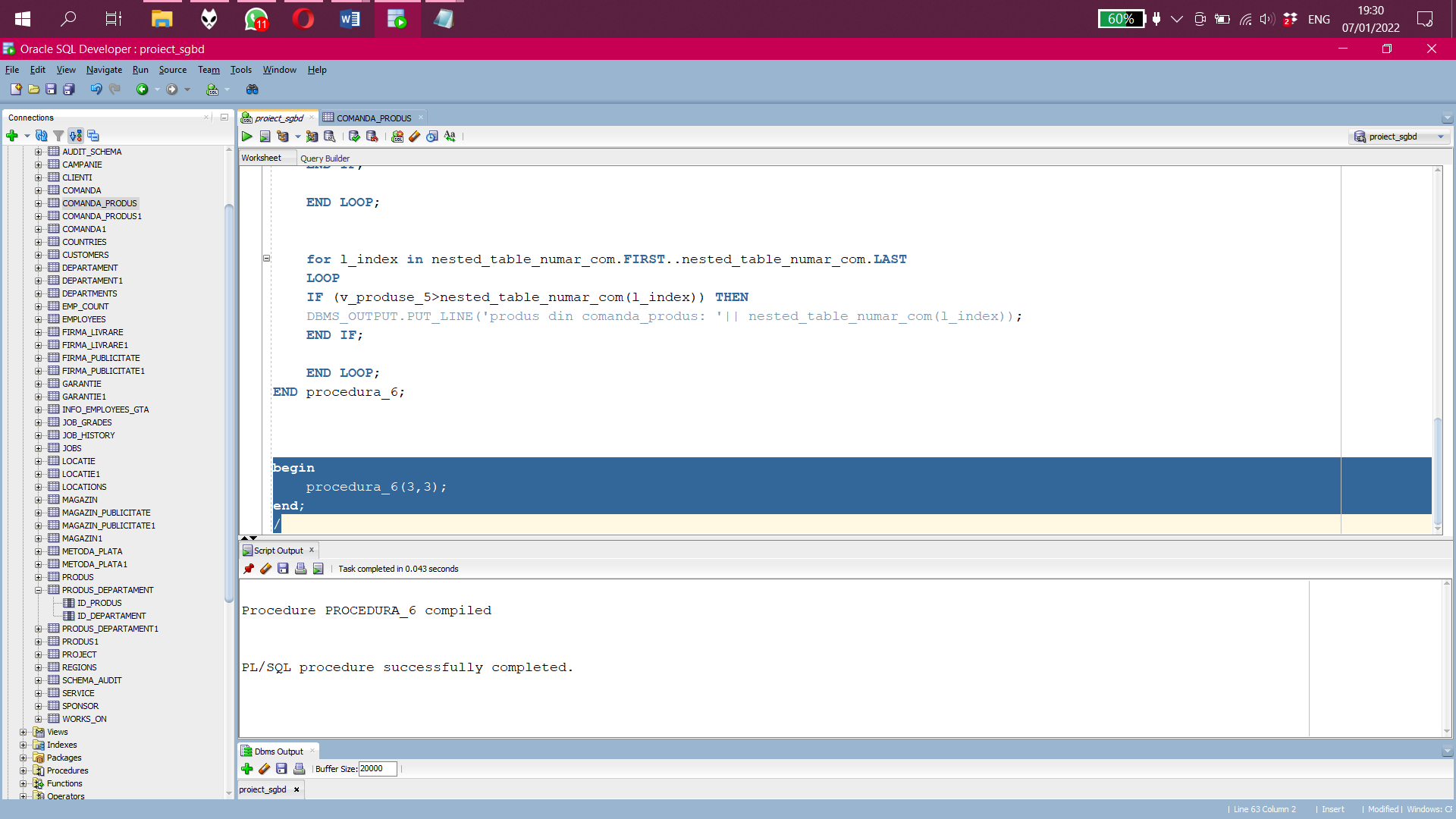
procedura\_6(3,3);

end;

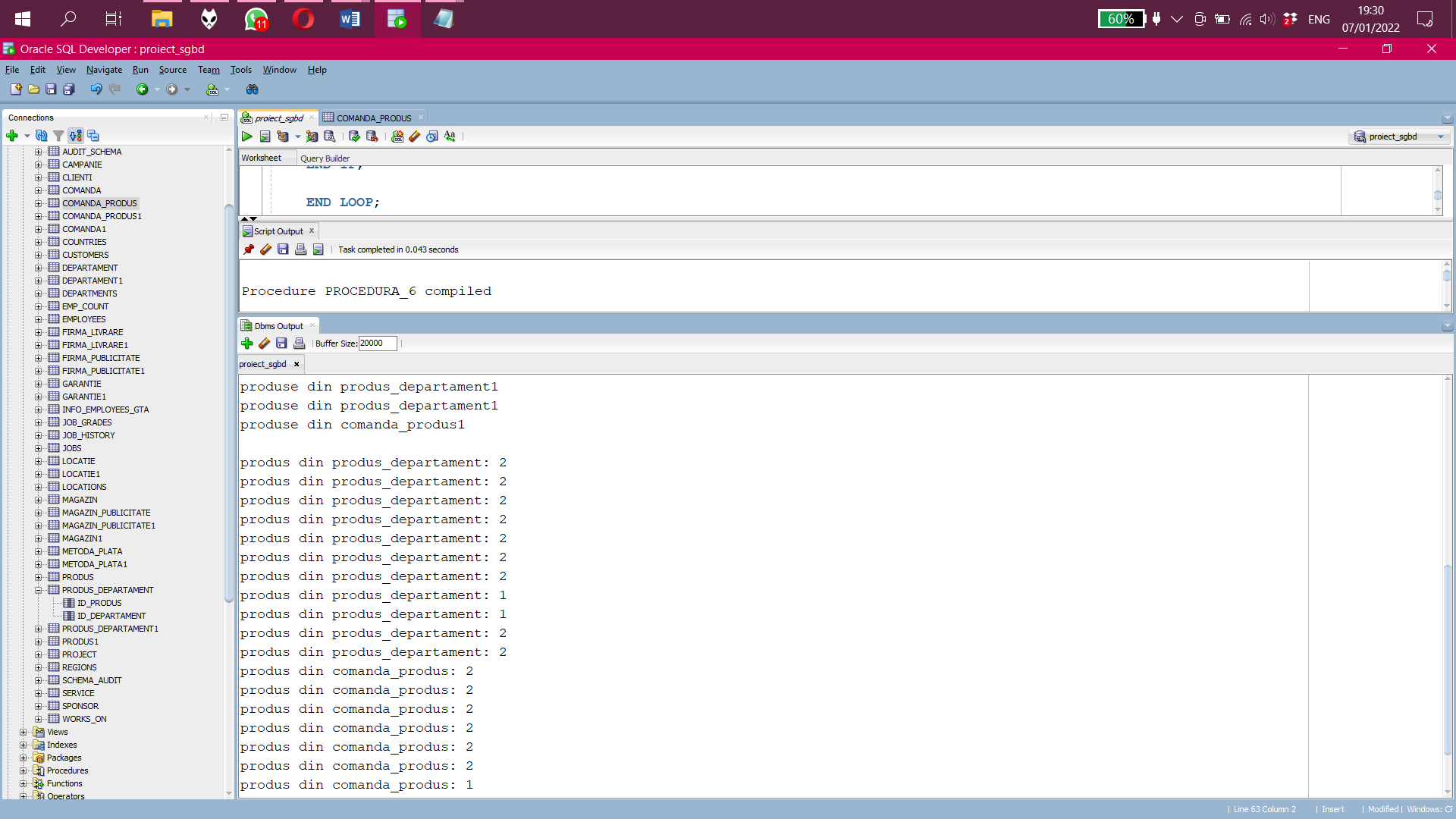
/



/\*procedura compileaza\*/



/\*apelul functioneaza\*/



/\* si in dbms afiseaza ce produse sunt in ce tabele\*/

1. O procedura care contine un cursor, ce parcurge si ulterior afiseaza pe dbms toate produsele din tabela produse cu id mai mic sau egal cu 10.

create or replace procedure procedura\_7

is variabila varchar2(255);

cursor curs\_produse is

select nume\_produs from produs

where id\_produs<=10;

begin

open curs\_produse;

loop

fetch curs\_produse into variabila;

DBMS\_OUTPUT.PUT\_LINE(variabila);

exit when curs\_produse%notfound;

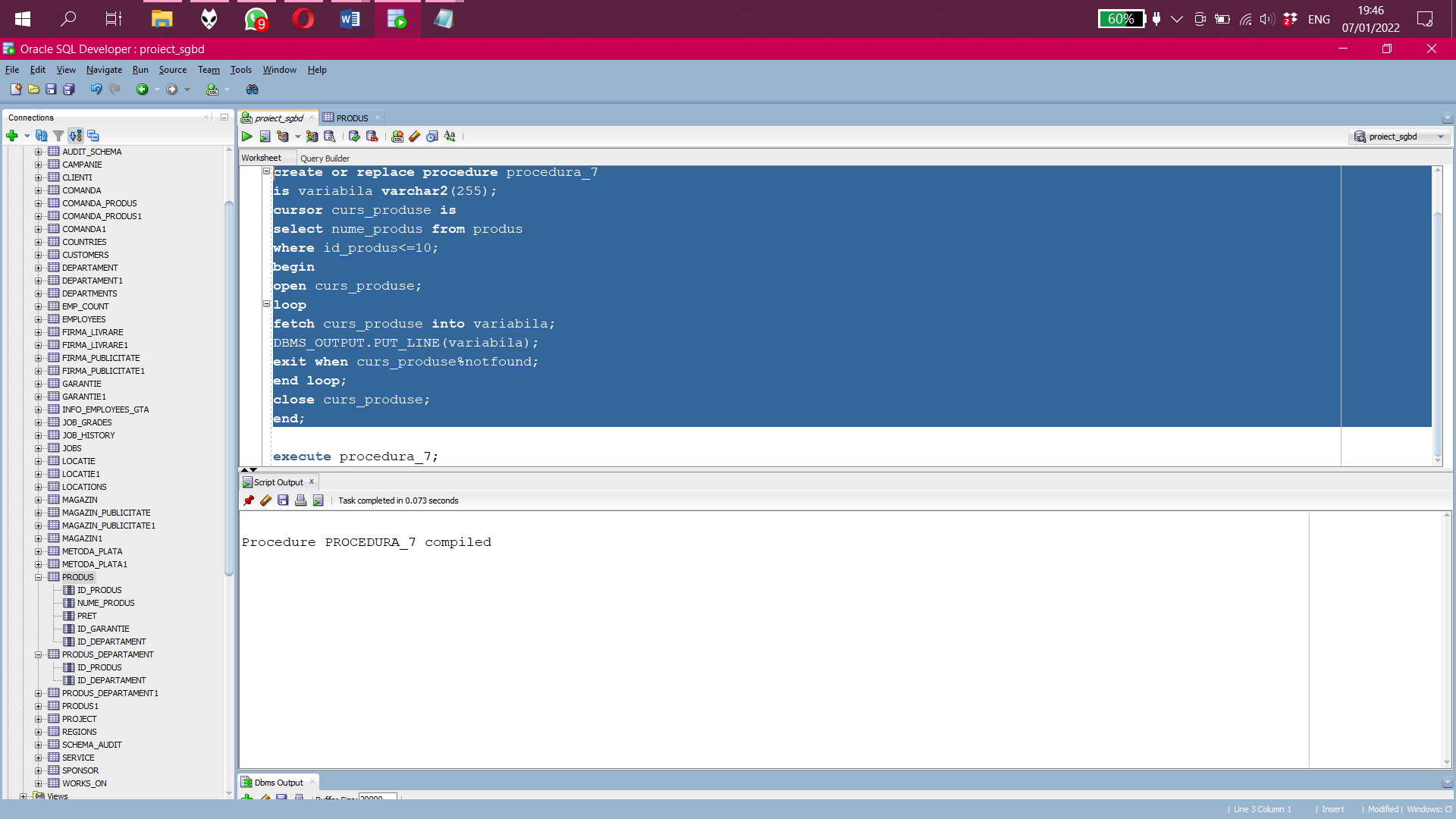
end loop;

close curs\_produse;

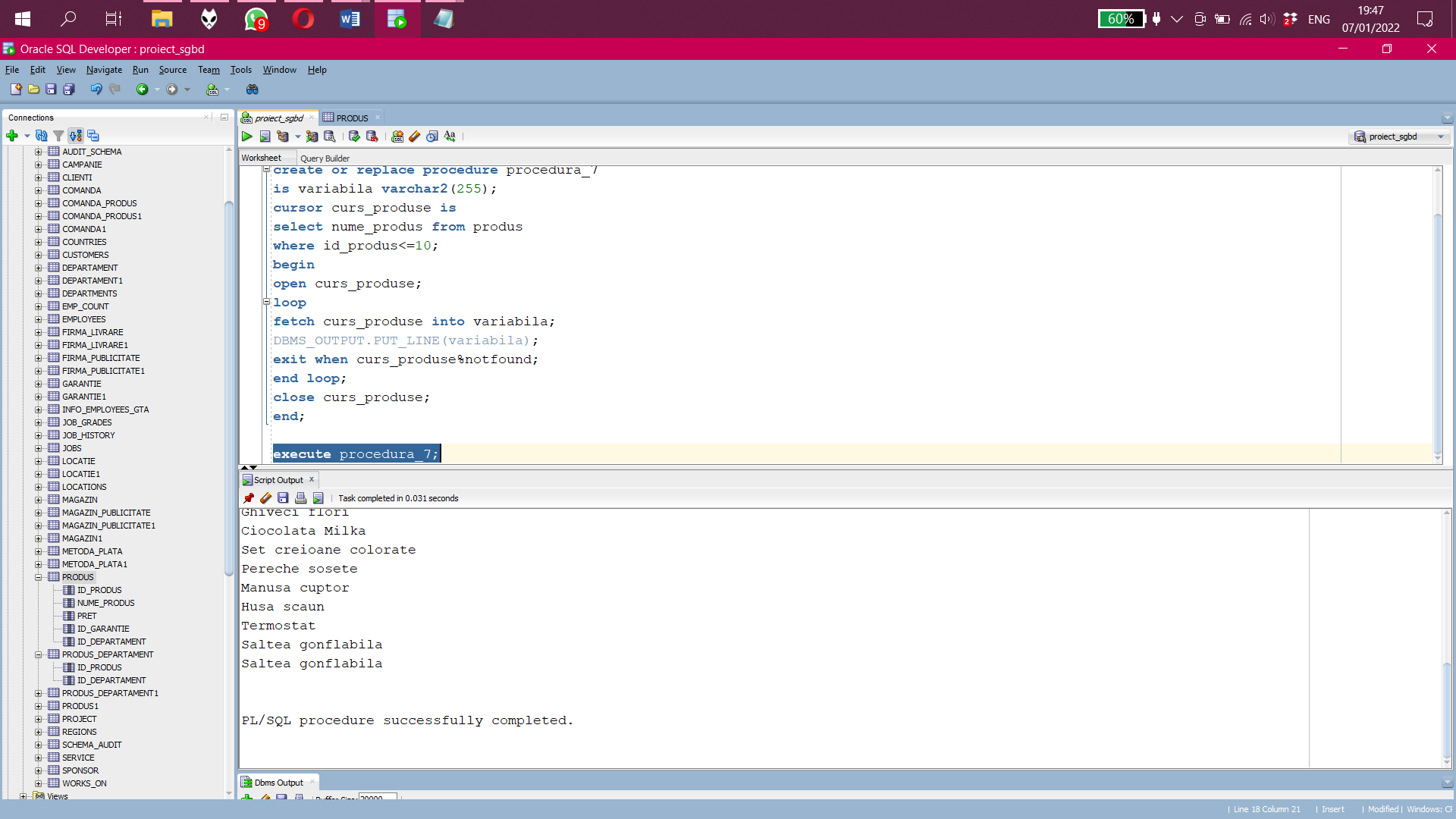
end;

/\*apel:\*/

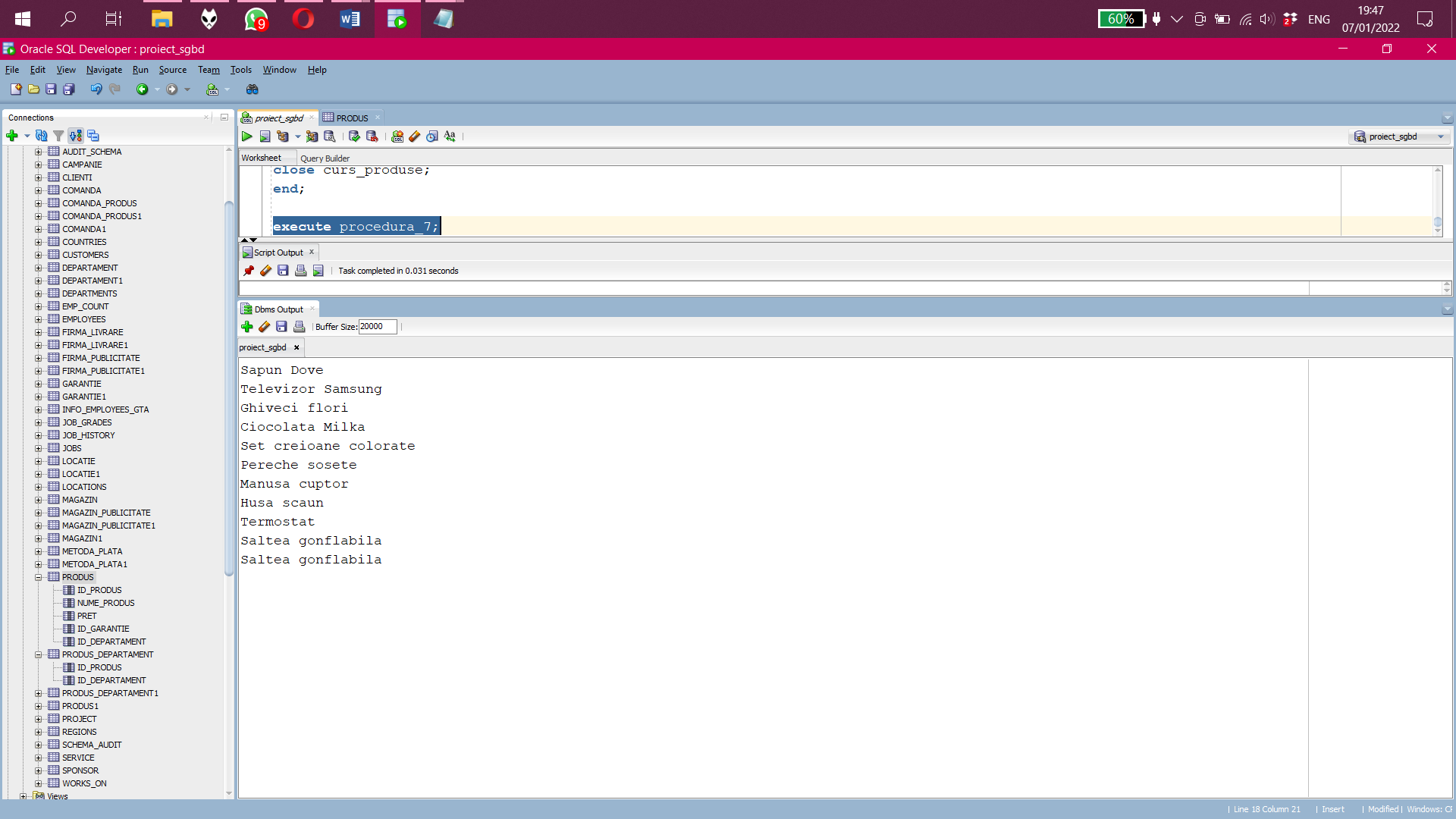
execute procedura\_7;



/\*aici compilez procedura\*/



/\*aici apelez procedura\*/



/\*acesta este outputul cand apelez procedura\*/

8.

/\* sa sa afiseze metoda de plata utilizata in fiecare comanda pentru un anume produs

\*/

create or replace function subprogram\_ex8

(v\_nume\_produs produs.nume\_produs%type)

return sys.odcivarchar2list

is

rezultat sys.odcivarchar2list;

begin

select nume\_metoda\_plata

bulk collect

into rezultat

FROM metoda\_plata mp JOIN comanda c on mp.id\_plata = c.id\_plata

JOIN comanda\_produs cp on cp.id\_comanda = c.id\_comanda

JOIN produs p on p.id\_produs = cp.id\_produs

WHERE upper(p.nume\_produs)=upper(v\_nume\_produs);

return rezultat;

exception

when no\_data\_found

then dbms\_output.put\_line('Nu au fost gasite date!');

when too\_many\_rows

then dbms\_output.put\_line('Eroare too many rows!');

when others

then dbms\_output.put\_line('Eroare!');

end subprogram\_ex8;

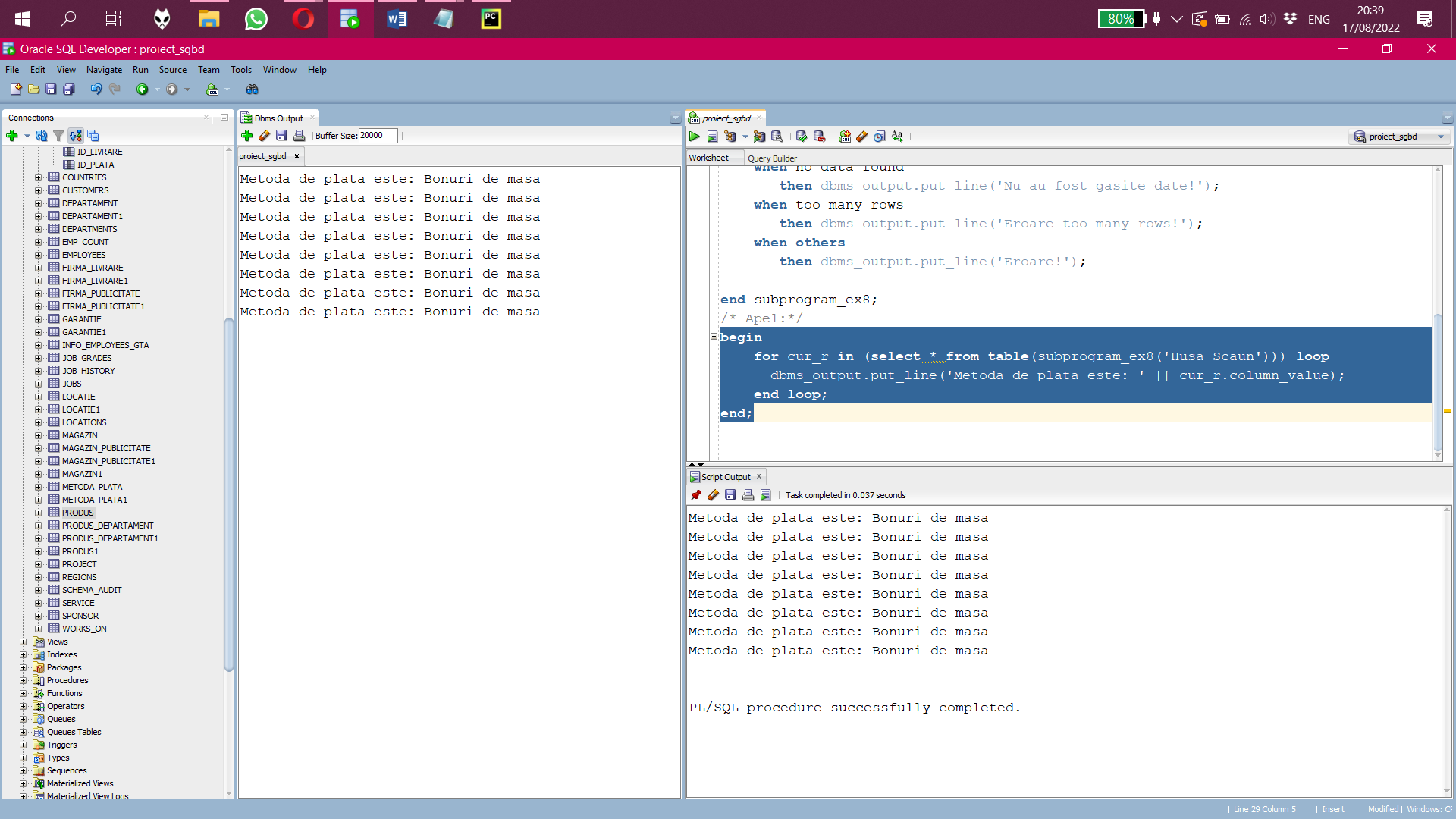
/\* Apel:\*/

begin

for cursor in (select \* from table(subprogram\_ex8('Husa Scaun'))) loop

dbms\_output.put\_line('Metoda de plata este: ' || cursor.column\_value);

end loop;

end;

/\*Exercitiul 9

enunt: Faceti o procedura care primeste id-ul unui client si afiseaza toate garantiile pe care acesta le-a primit pentru produsele achizitionate\*/

create or replace procedure procedura\_ex9

(v\_id\_client clienti.id\_client%type)

is

TYPE tablou IS TABLE OF garantie.nume\_garantie%type INDEX BY PLS\_INTEGER;

v\_tablou tablou;

begin

SELECT nume\_garantie

BULK COLLECT INTO v\_tablou

FROM garantie g JOIN produs p on p.id\_garantie = g.id\_garantie

JOIN comanda\_produs cp on cp.id\_produs = p.id\_produs

JOIN comanda c on c.id\_comanda = cp.id\_comanda

JOIN clienti cl on c.id\_client=cl.id\_client

WHERE cl.id\_client = v\_id\_client;

FOR i in v\_tablou.first..v\_tablou.last loop

dbms\_output.put\_line(v\_tablou(i));

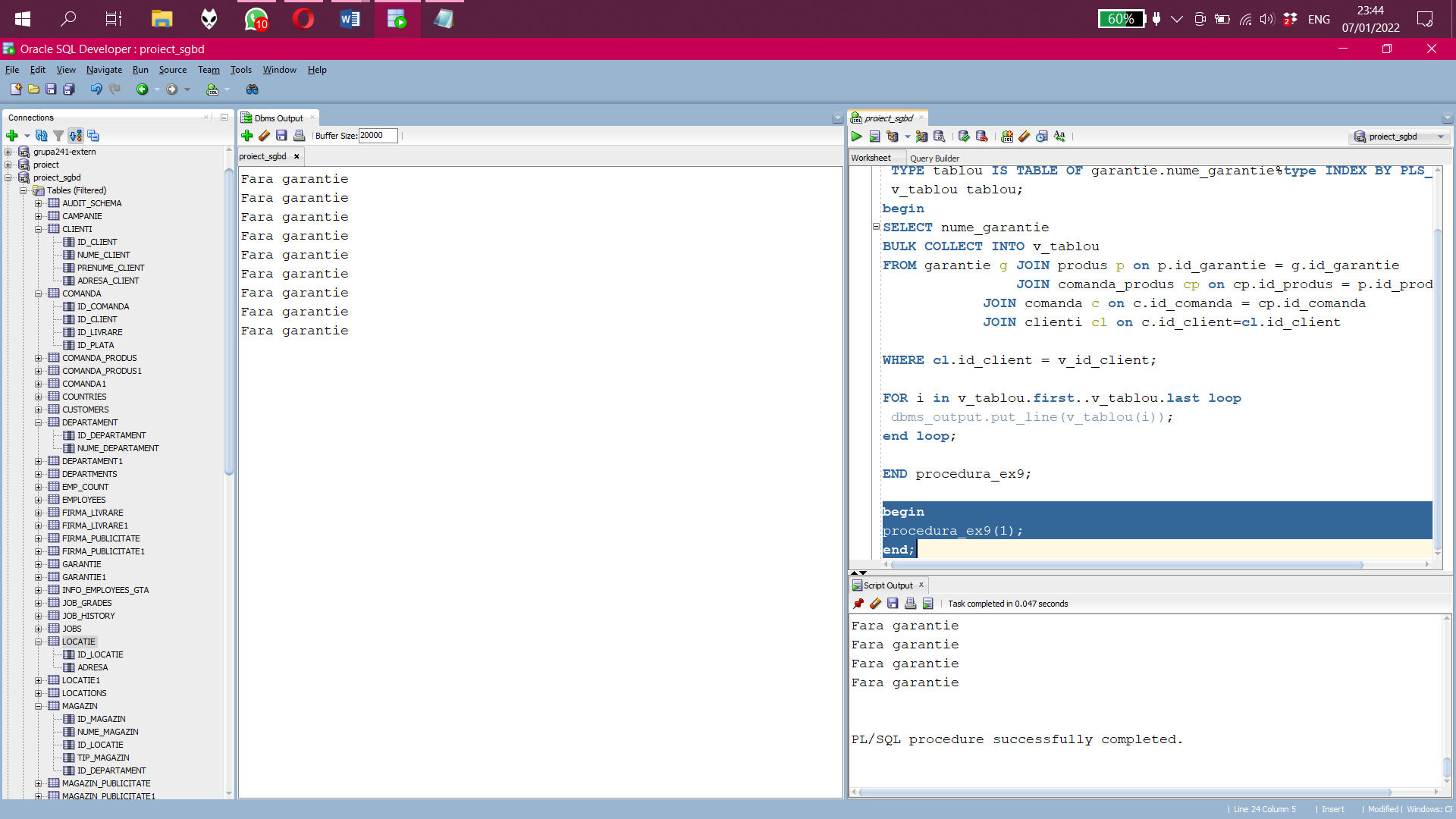
end loop;

END procedura\_ex9;

begin

procedura\_ex9(1);

end;



10. Trigger care sa nu permita inserarea, modificarea sau stergerea datelor din tabela PRODUS in afara programului de lucru al magazinului.

create or replace trigger trigger\_10

before insert or update or delete on produs

begin

if (to\_char(sysdate,'D') = 1) or (to\_char(sysdate, 'D')=6) or (to\_char(sysdate,'HH24') not between 9 and 17)

then

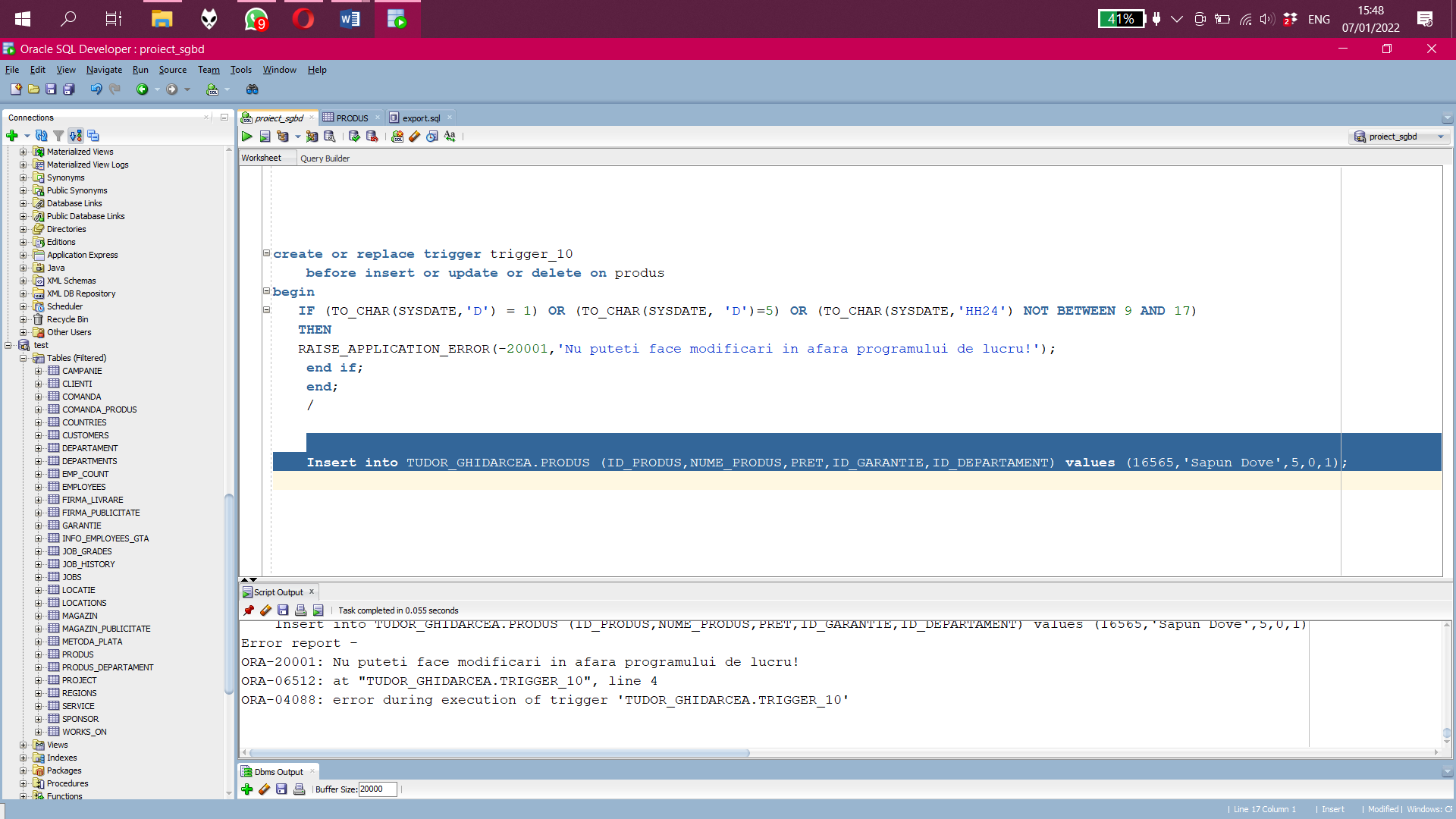
raise\_application\_error(-20001,'Nu puteti face modificari in afara programului de lucru!');

end if;

end;

/

Insert into PRODUS values (123,'Sapun Dove',5,0,1);



Daca modific ziua din trigger astfel incat sa corespunda zilei curente a sistemului, imi da eroarea ORA-20001, asa cum cere

11. Trigger care sa nu imi permita sa cresc pretul unui produs cu mai mult de 25%, ci doar sa il reduc sau sa il cresc cu pana la 25%.

create or replace trigger TRIGGER\_11

before update of PRET on PRODUS

for each row

when (new.PRET > ((old.PRET+25\*old.PRET)/100))

begin

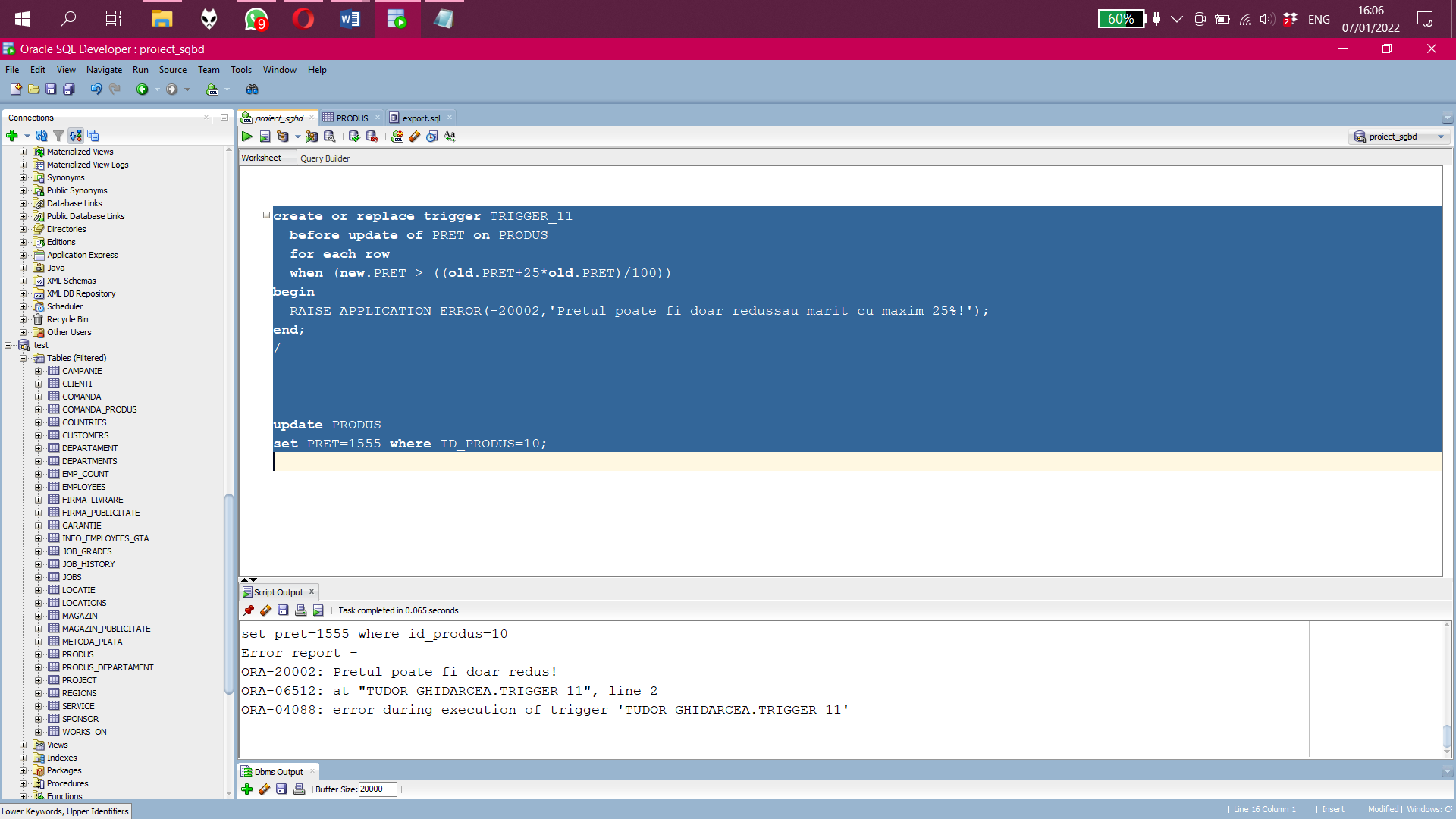
RAISE\_APPLICATION\_ERROR(-20002,'Pretul poate fi doar redus sau marit cu maxim 25%!');

end;

/

update PRODUS

set PRET=1555 where ID\_PRODUS=10;



12. Trigger LDD ce actioneaza asupra schemei, si noteaza toate crearile, stergerile si modificarile facute asupra schemei sau a tabelelor din ea.

CREATE TABLE audit\_schema

(

ddl\_date DATE,

ddl\_user VARCHAR2(15),

object\_created VARCHAR2(15),

object\_name VARCHAR2(15),

ddl\_operation VARCHAR2(15)

);

CREATE OR REPLACE TRIGGER audit\_schema\_tr

AFTER DDL ON SCHEMA

BEGIN

INSERT INTO audit\_schema VALUES (

sysdate,

sys\_context('USERENV','CURRENT\_USER'),

ora\_dict\_obj\_type,

ora\_dict\_obj\_name,

ora\_sysevent);

END;

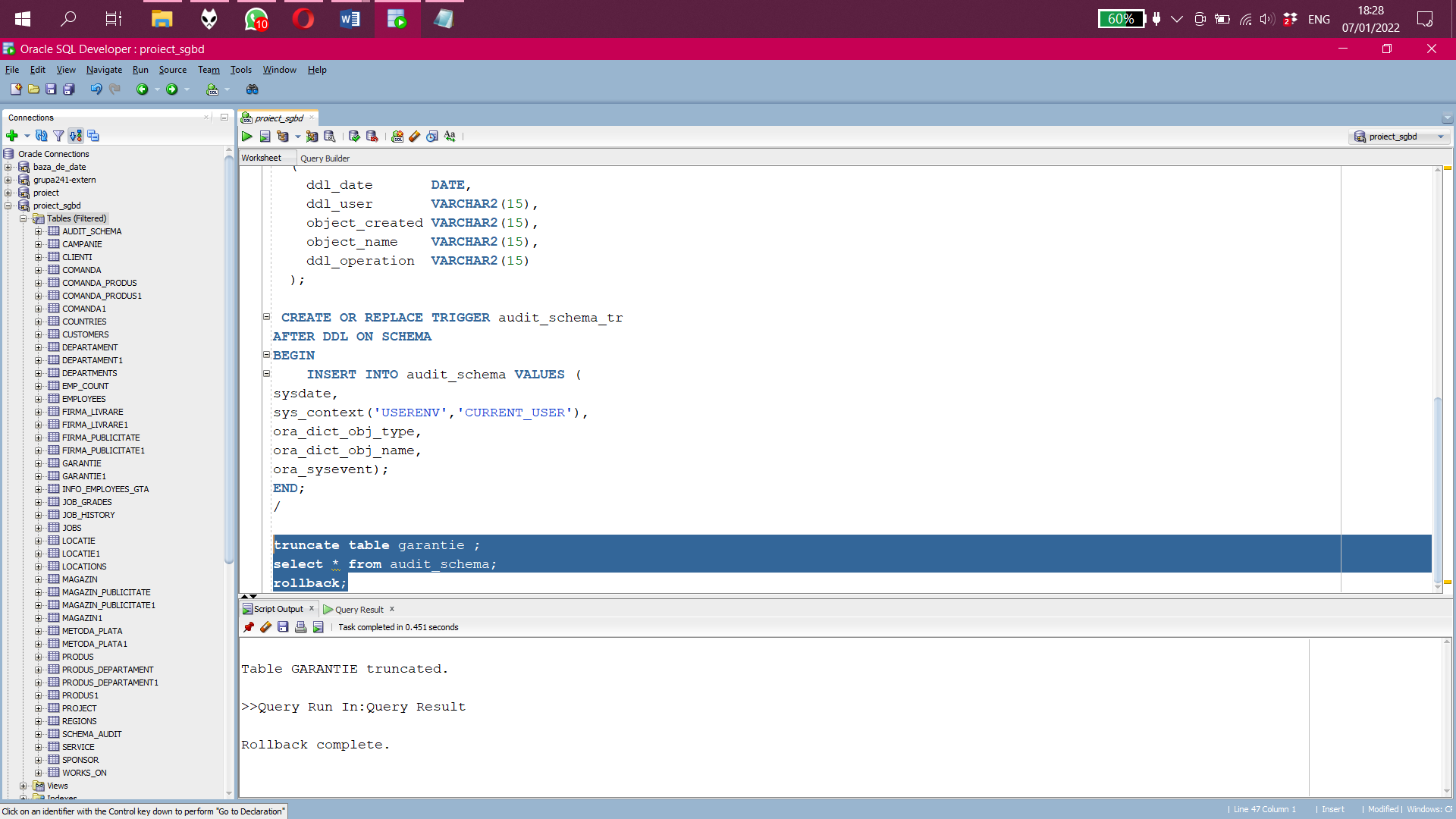
/

truncate table garantie ;

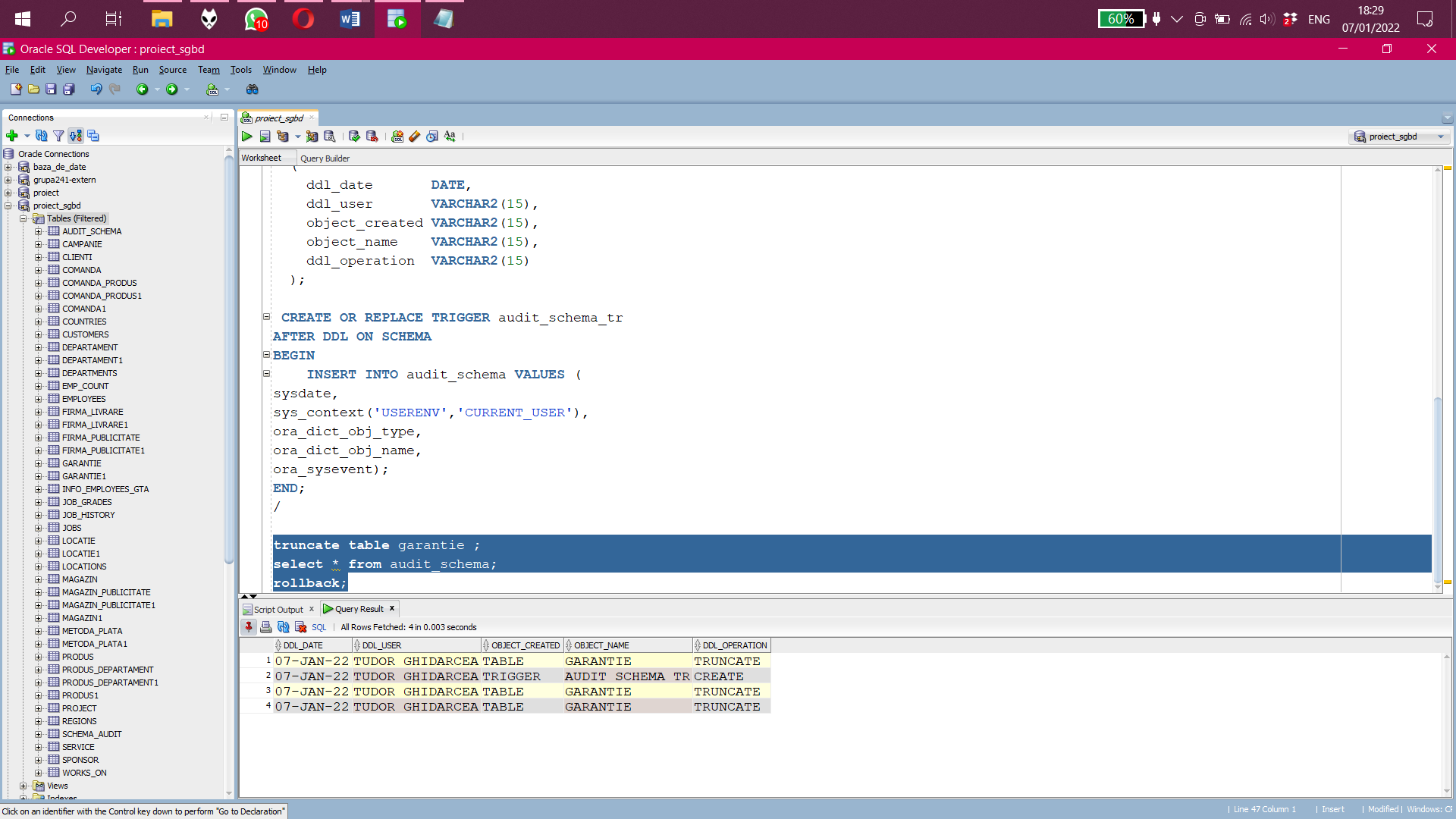
select \* from audit\_schema;

rollback;

/\*creez o tabela speciala pentru a stoca aceste modificari, iar triggerul le noteaza in ea\*/



/\*aici se poate vedea ca am efectuat o operatiune de truncate asupra unei tabele din schema, urmata de un rollback pentru a nu strica baza de date\*/



/\*iar aici, in query result, sunt afisate toate schimbarile facute asupra schemei, notate in tabela dedicata, impreuna cu informatii precum data la care au avut loc aceste schimbari\*/.