**ACADEMIA DE STUDII ECONOMICE**

PROIECT BAZE DE DATE

**Profesori Coordonatori:**

Diaconiță Vlad

Puican Florina Camelia

**Facultatea de Cibernetică, Statistică și Informatică Economică**

**Anul II, Seria E, Grupa 1057**

Rotaru Ana-Maria

**CUPRINS**

[**DESCRIEREA TEMEI** 3](#_Toc72850379)

[**SCHEMA BAZEI DE DATE** 4](#_Toc72850380)

[**INTRODUCERE ÎN PL/SQL** 5](#_Toc72850381)

[**EXERCIȚII ADAPTATE PE BAZA MEA DE DATE** 11](#_Toc72850382)

[**VARIABILE** 17](#_Toc72850383)

[**STRUCTURI FUNDAMENTALE DE PROGRAMARE** 29](#_Toc72850384)

[**CURSORUL ÎN PL/SQL** 40](#_Toc72850385)

[**TRATAREA EXCEPȚIILOR** 47](#_Toc72850386)

[**SUBPROGRAME PL/SQL** 50](#_Toc72850387)

[**PACHETE DE SUBPROGRAME** 55](#_Toc72850388)

[**TRIGGERI PE BAZA DE DATE** 60](#_Toc72850389)

[**ANEXE** 64](#_Toc72850390)

[**TABELE** 65](#_Toc72850391)

[**OPERATII DE ACTUALIZARE** 67](#_Toc72850392)

[**CERINTE** 69](#_Toc72850393)

# **DESCRIEREA TEMEI**

Bazele de date reprezintă colecții de informații logic organizate, astfel încât să ne ușureze și să ne eficientizeze diverse acțiuni. În acest sens, am luat în considerare o firma a cărei activitate constă în comercializarea mobilei.

În principal, comenzile venite din partea clienților sunt prelucrate de către angajații firmei, care se ocupă totodată și de urmărirea încasărilor. Articolele cuprinse în comenzi provin de la diferiți furnizori.

Menționez că prelucrarea datelor se face printr-un număr unic de referință (id), ales pentru fiecare furnizor/articol/client/comandă/încasare.

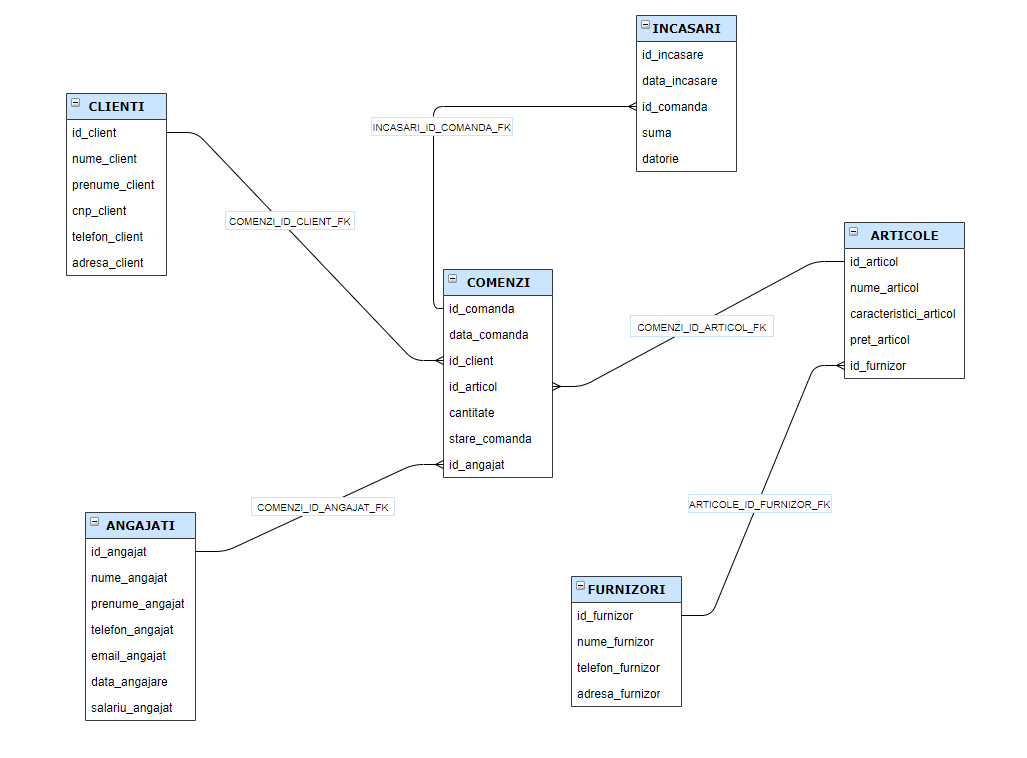
În comenzi se specifică dată, id-ul clientului, denumirea și cantitatea articolului, id-ul angajatului, cel care o întocmește și statusul (livrat, returnat, anulat). Fiecare articol se caracterizează prin denumire, dimensiuni, preț și furnizor.

Baza de date creată cuprinde următoarele caracteristici:

1. funizori: id, denumire companie, telefon, adresa;
2. clienți: id, nume&prenume, cnp, telefon, adresa;
3. angajați: id, nume&prenume, cnp, telefon, email…

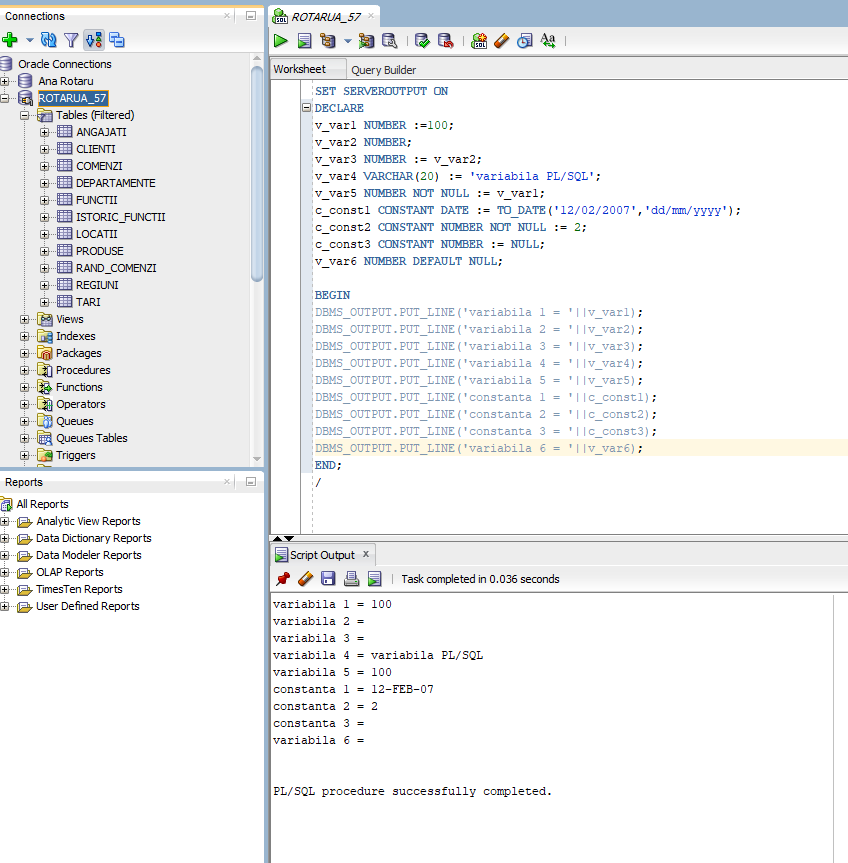
Întocmind această baza de date am urmărit să simplific evidență contabila a firmei, prin crearea unui program care să faciliteze introducerea datelor primare, precum și prelucrarea acestora.

# **SCHEMA BAZEI DE DATE**

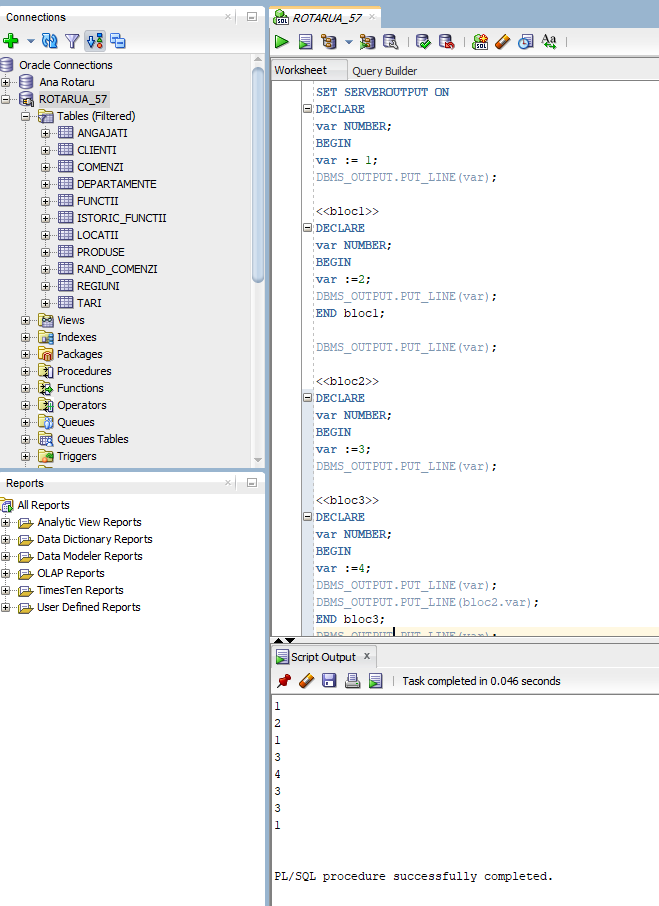


# **INTRODUCERE ÎN PL/SQL**

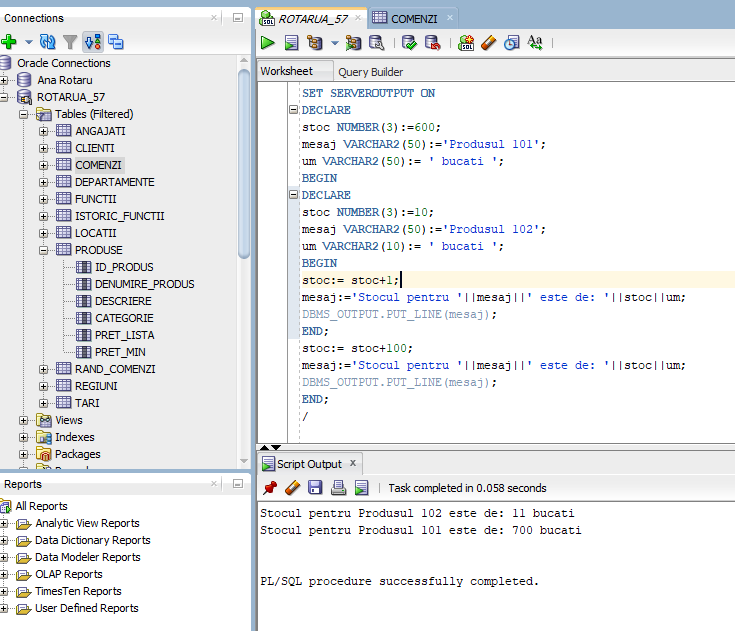
1. Specificaţi ce se va afişa la rularea următorului bloc PL/SQL.



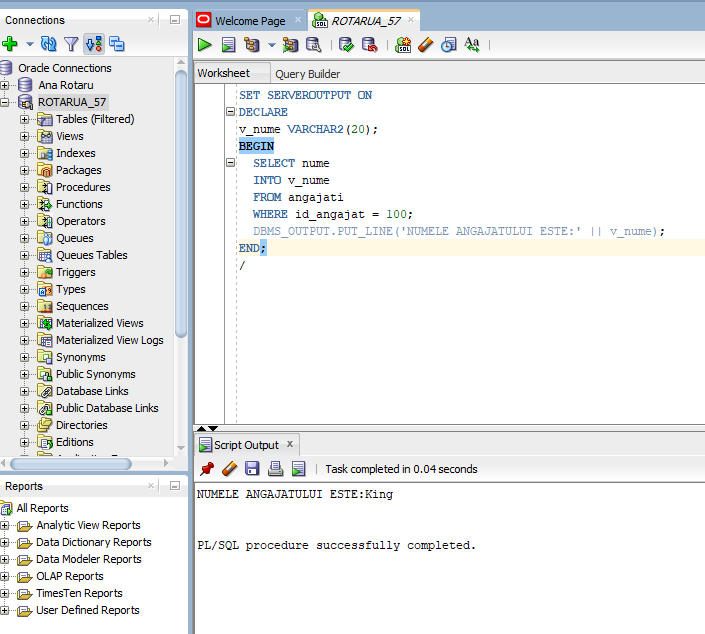
1. Specificaţi ce se va afişa la rularea următorului bloc PL/SQL (care conţine blocuri imbricate, ilustrând domeniul de vizibilitate al unor variabile care au acelaşi nume).

****

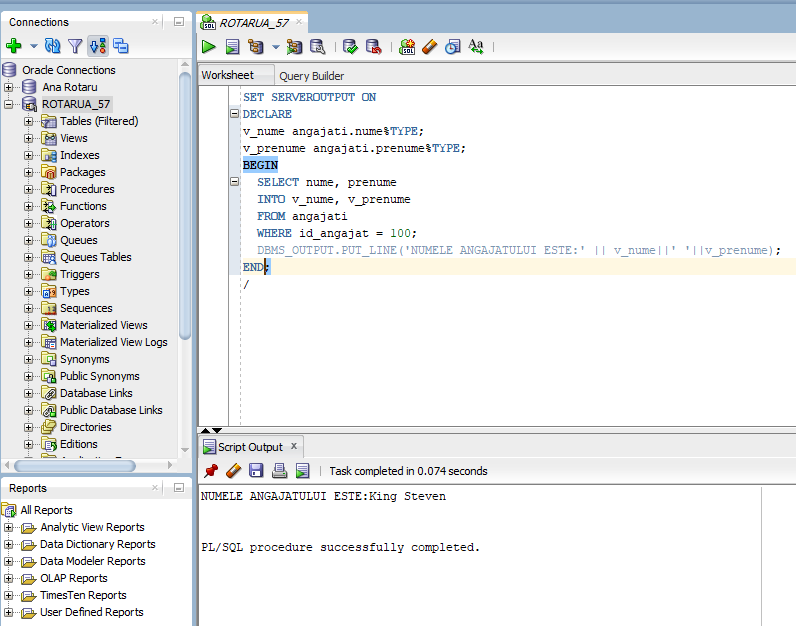
1. Specificaţi ce se va afişa la rularea următorului bloc PL/SQL.

****

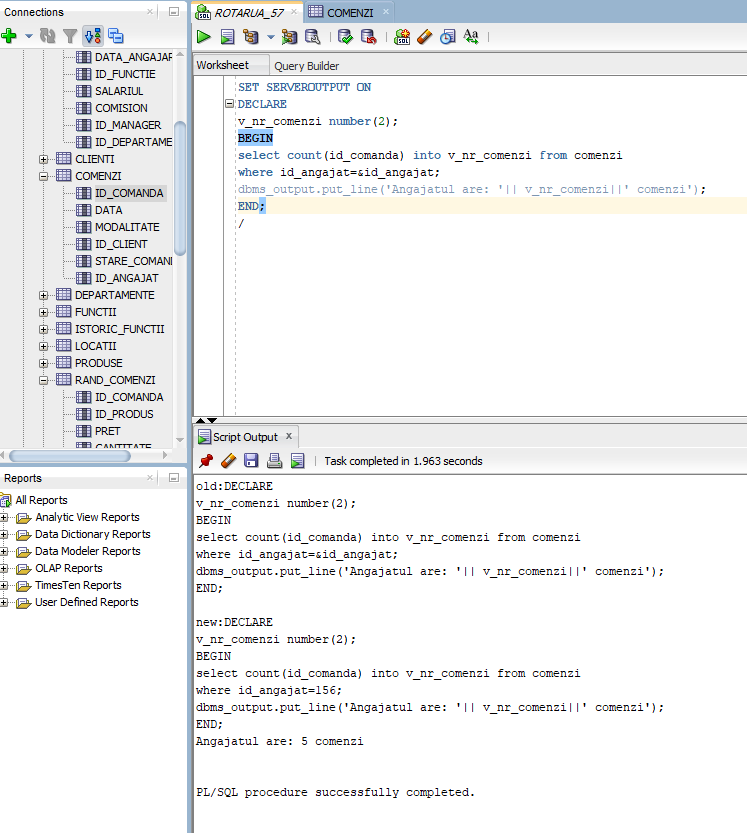
1. Se afișează numele angajatului cu codul 100.

****

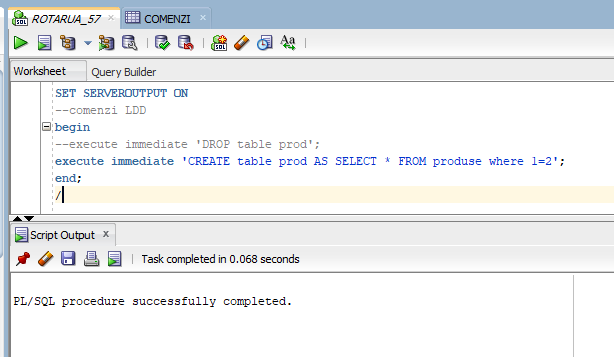
1. Să se afișeze numele și prenumele angajatului cu codul 100.

****

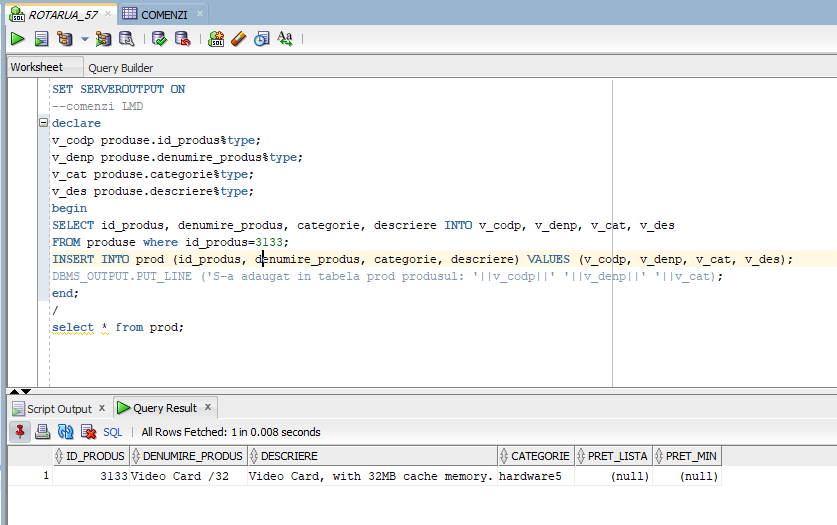
1. Se afișează numărul de comenzi ale angajatului al cărui cod este introdus de utilizator prin intermediul variabilei de substituție &id\_angajat (pt. id\_angajat = 156).

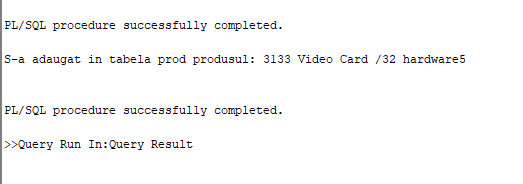
****

1. Se creează tabela **prod** în cadrul unui bloc PL/SQL:

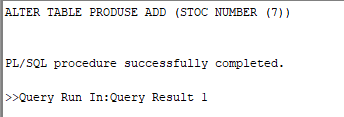
****

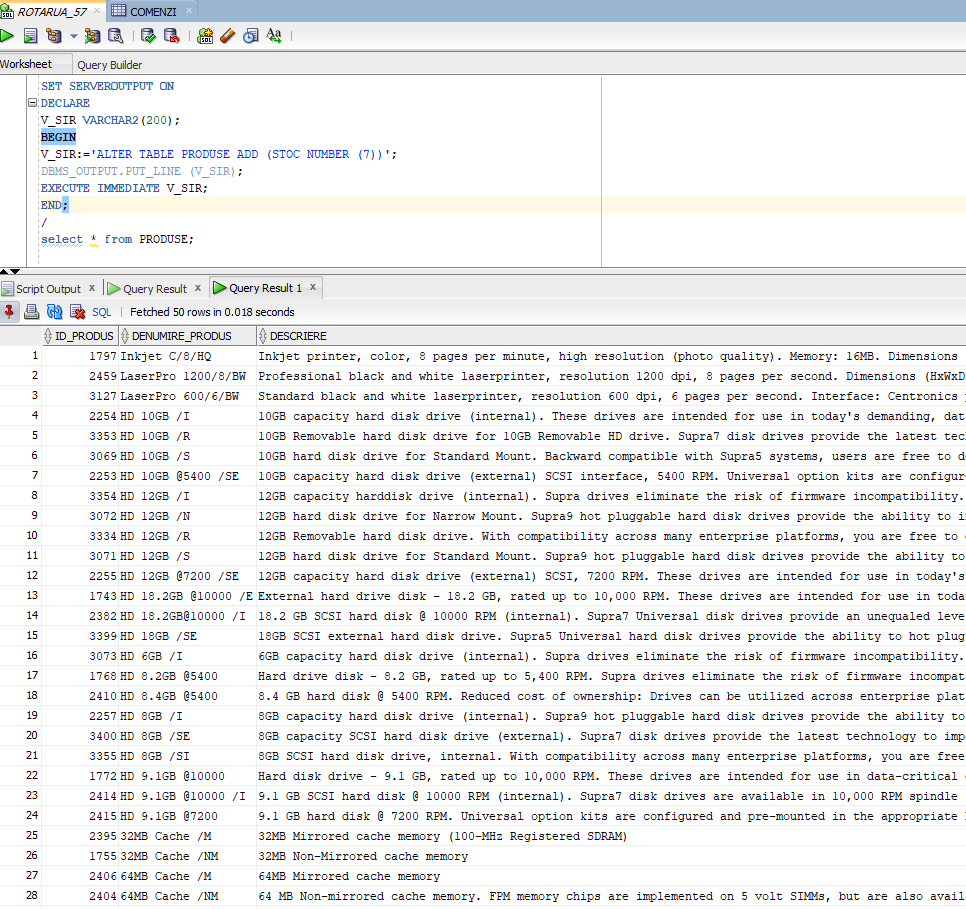
1. Se adaugă în tabela **prod** înregistrări din tabela produse.

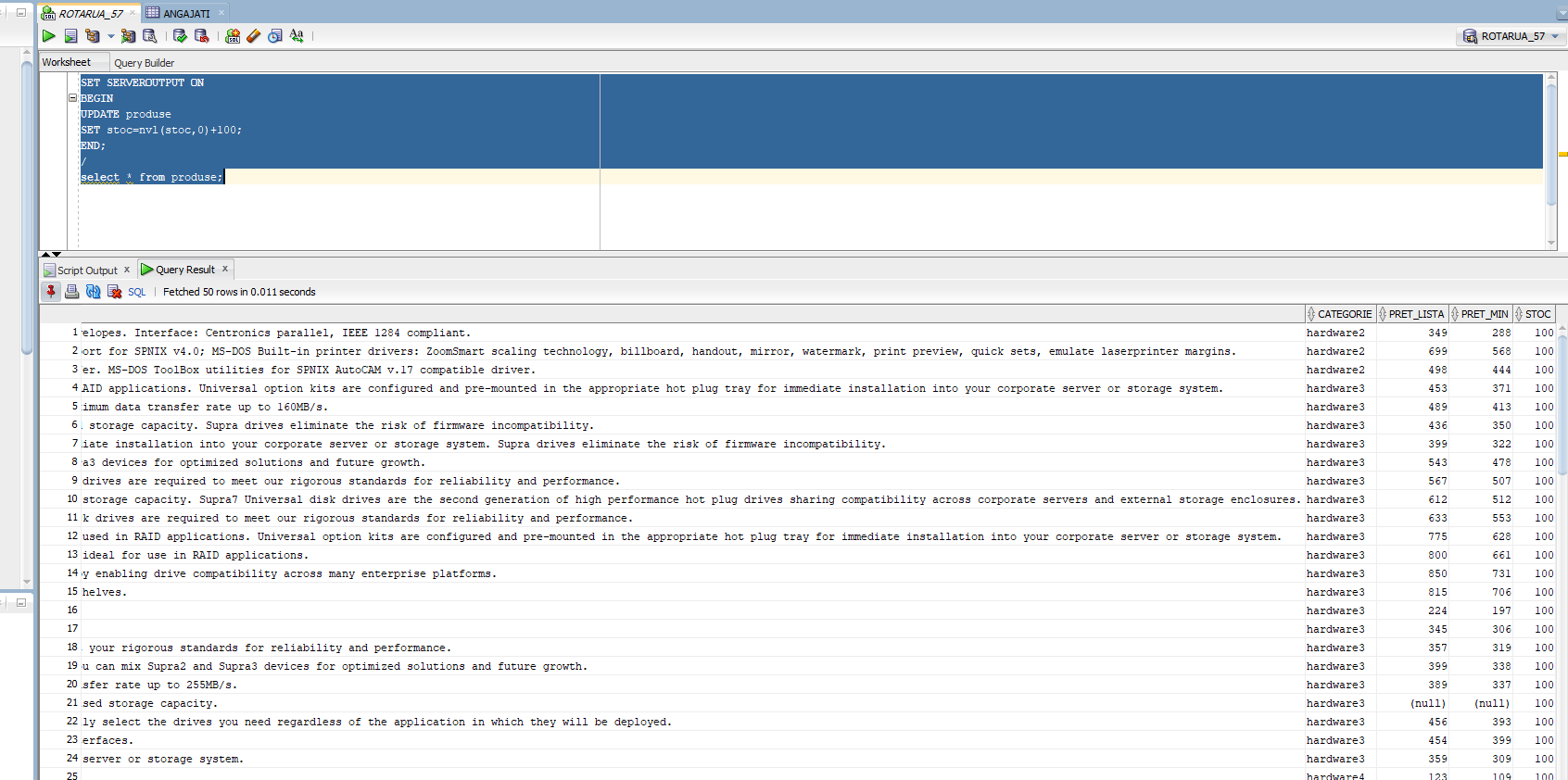
****

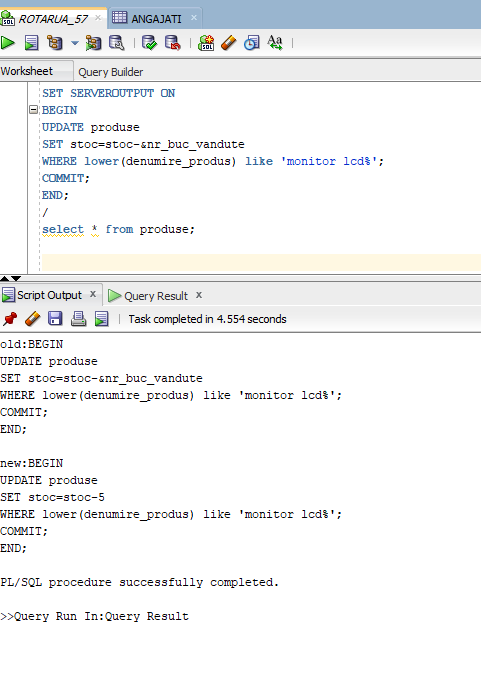
****

1. Se adaugă o nouă coloană STOC în tabela produse.

****

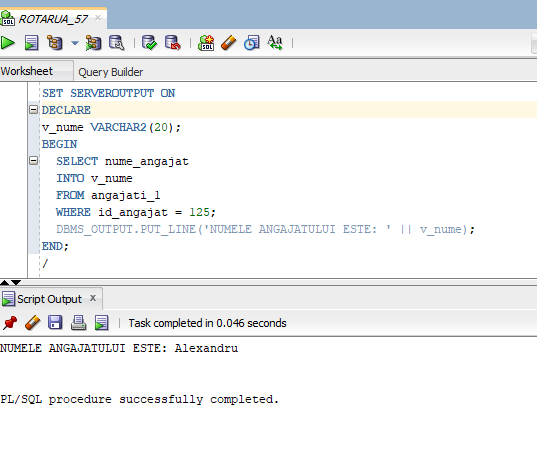
****

1. Se realizează o aprovizionare în cadrul depozitului prin care se măresc socurile tuturor produselor cu 100 bucăţi.
2. În urma comenzilor realizate de clienţi, monitoarele de tipul LCD sunt vândute, se deci scade din stocul existent un număr de monitoare introdus de la tastatură (5).

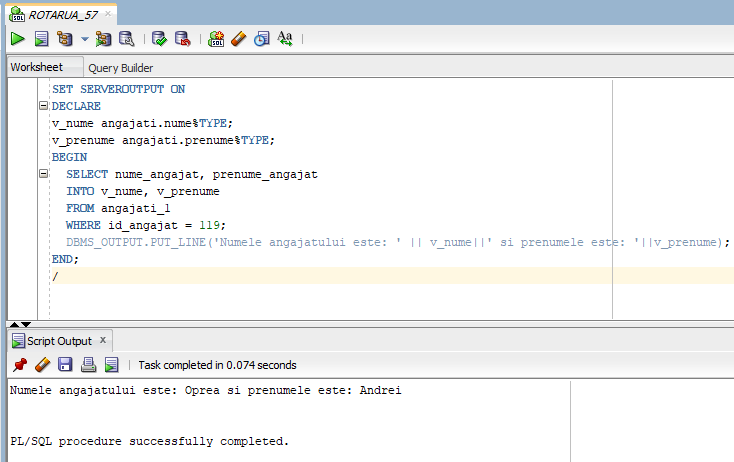
****

## **EXERCIȚII ADAPTATE PE BAZA MEA DE DATE**

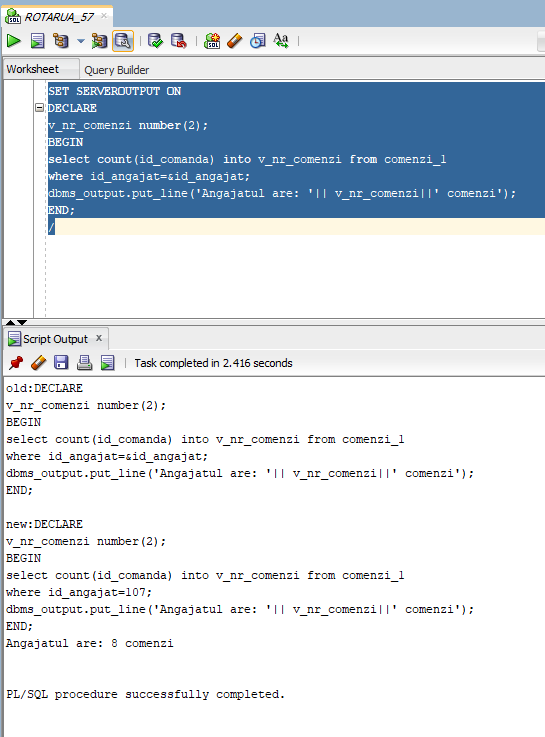
1. Să se afișeze numele angajatului cu id-ul 125.

****

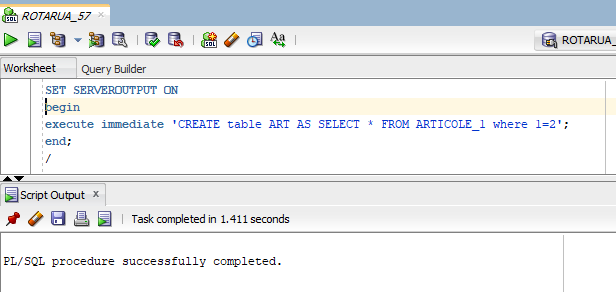
1. Să se afișeze numele și prenumele angajatului cu id-ul 119.

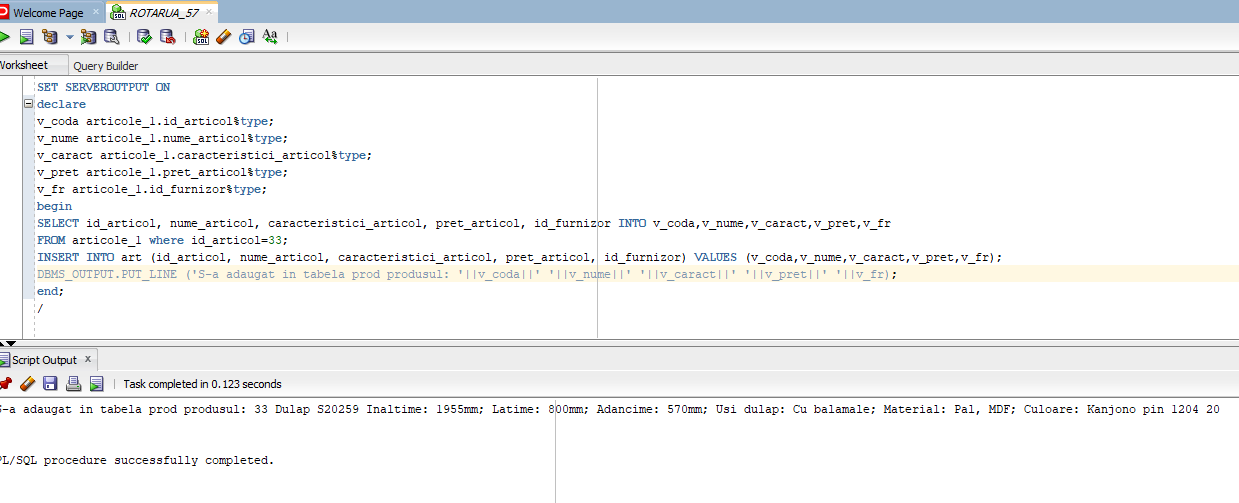
****

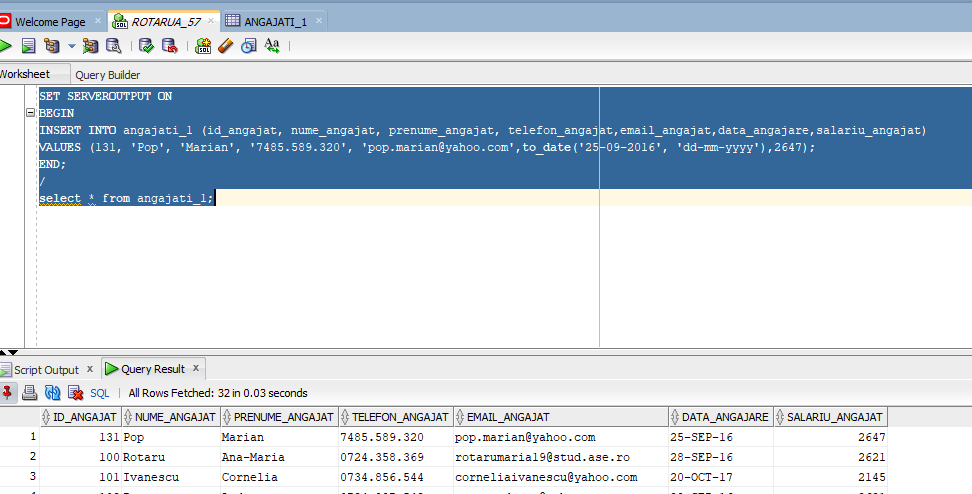
1. Să se afișeaze numarul de comenzi ale angajatului al cărui cod este introdus de utilizator prin intermediul variabilei de substituție &id\_angajat (am introdus 107).

****

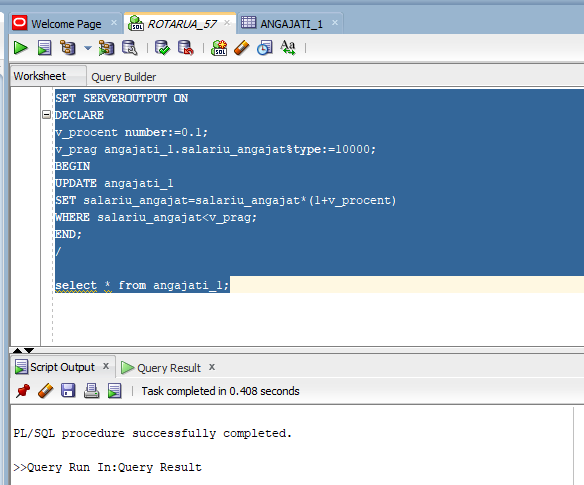
1. Se creează tabela **art** în cadrul unui bloc PL/SQL.

****

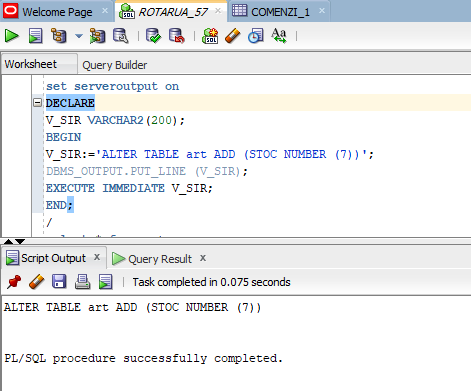
1. ****Se adaugă în tabela **art** înregistrări din tabela articole\_1.
2. Să se adauge în angajati\_1 o noua înregistrare.

****

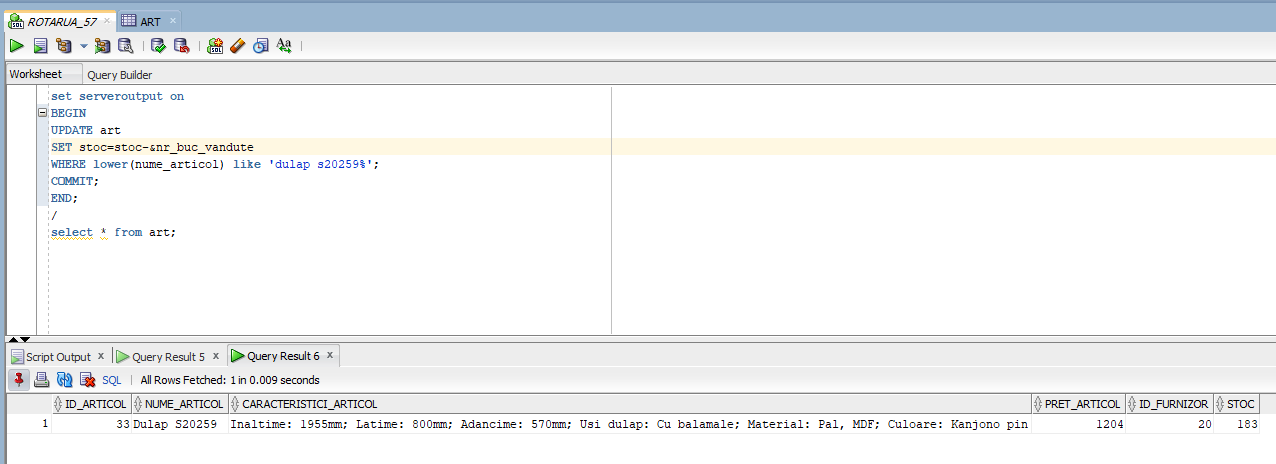
1. Se măreşte cu un procent salariul angajaţilor din tabela emp\_sal care au în prezent salariul mai mic decât o anumită valoare.

****

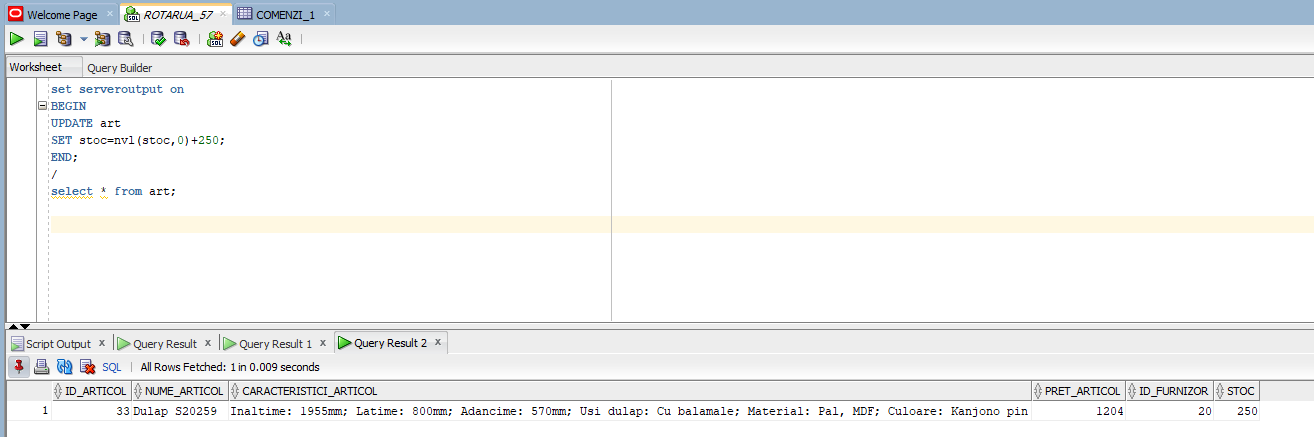
1. Se adaugă o nouă coloană STOC în tabela art.

****

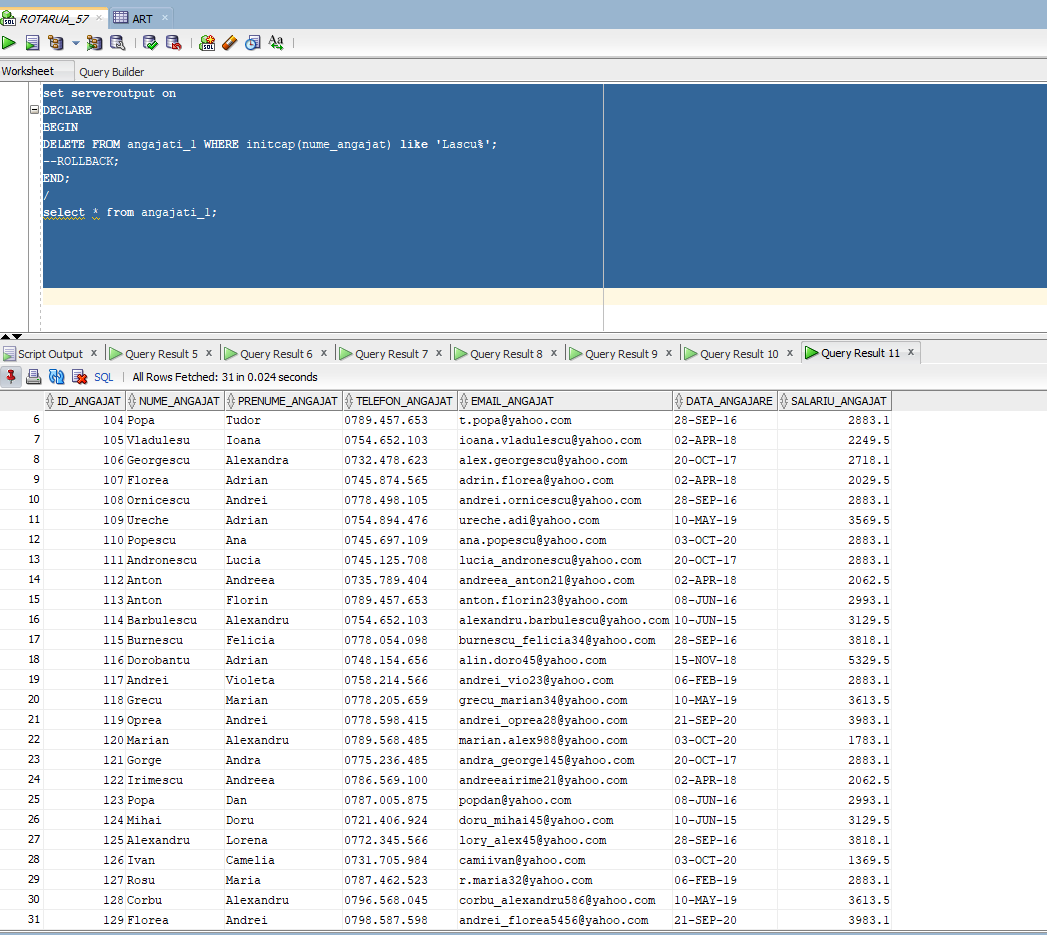
1. Se realizează o aprovizionare în cadrul depozitului prin care se măresc socurile tuturor produselor cu 250 bucăţi.

****

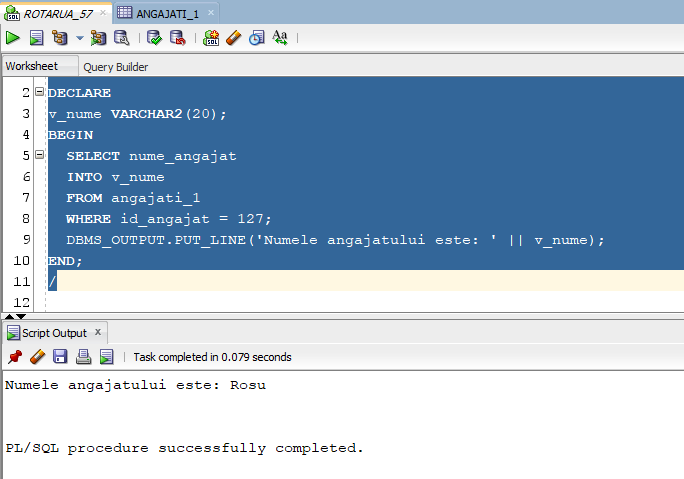
1. În urma comenzilor realizate de clienţi, dulapurile S20259 sunt vândute, se deci scade din stocul existent un număr de dulapuri introdus de la tastatură (67).

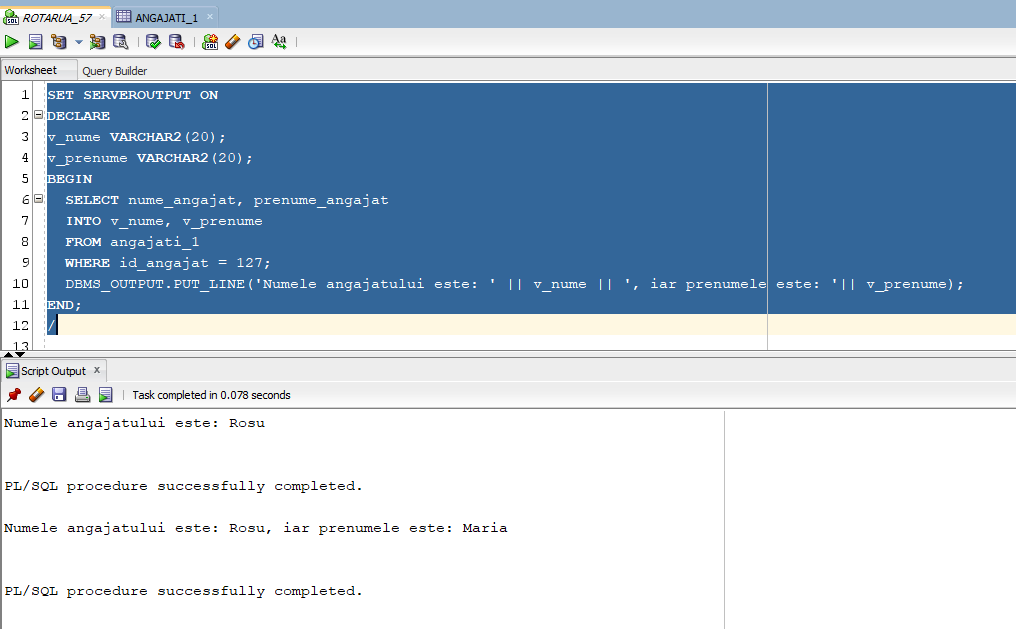


1. Se şterge angajatul cu numele Lascu din tabela angajati\_1.

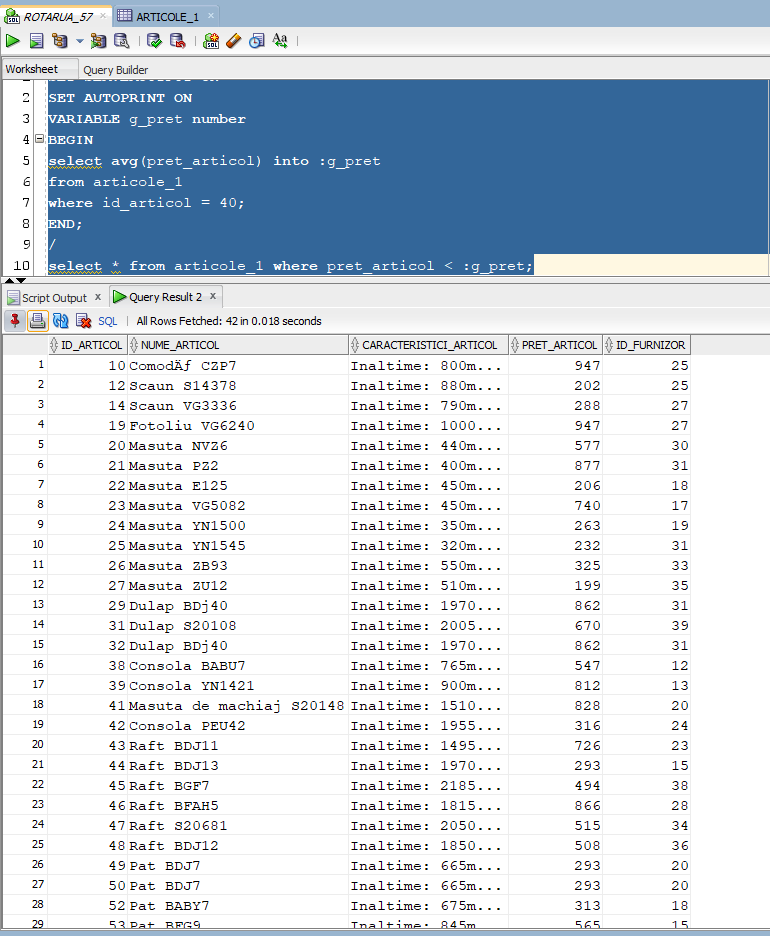
****

# **VARIABILE**

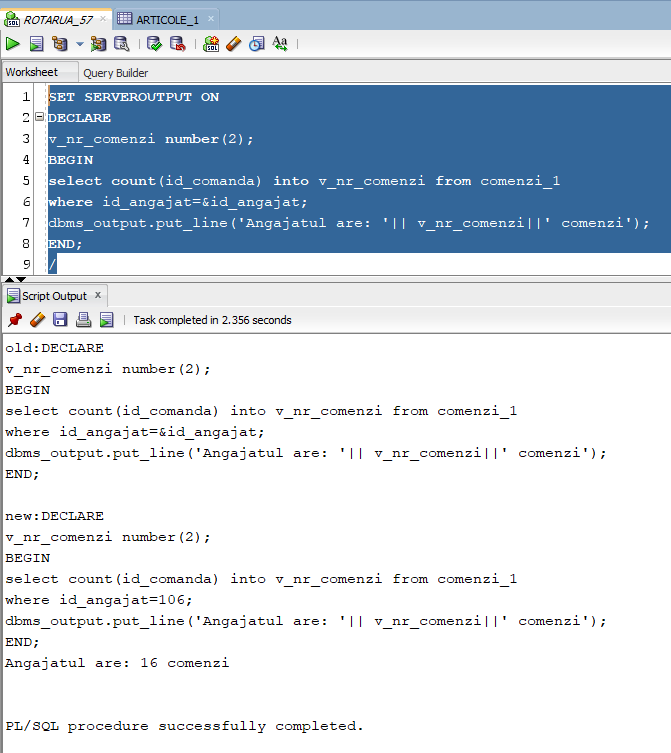
1. Să se afișeze numele angajatului cu codul 127.

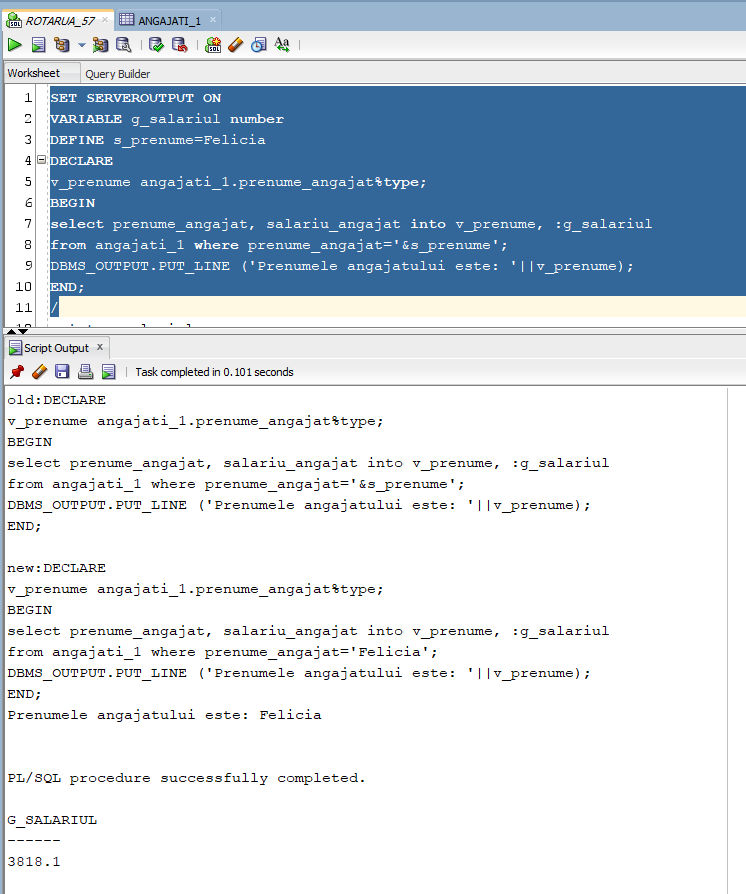
2. Să se afișeze numele și prenumele angajatului cu codul 127.

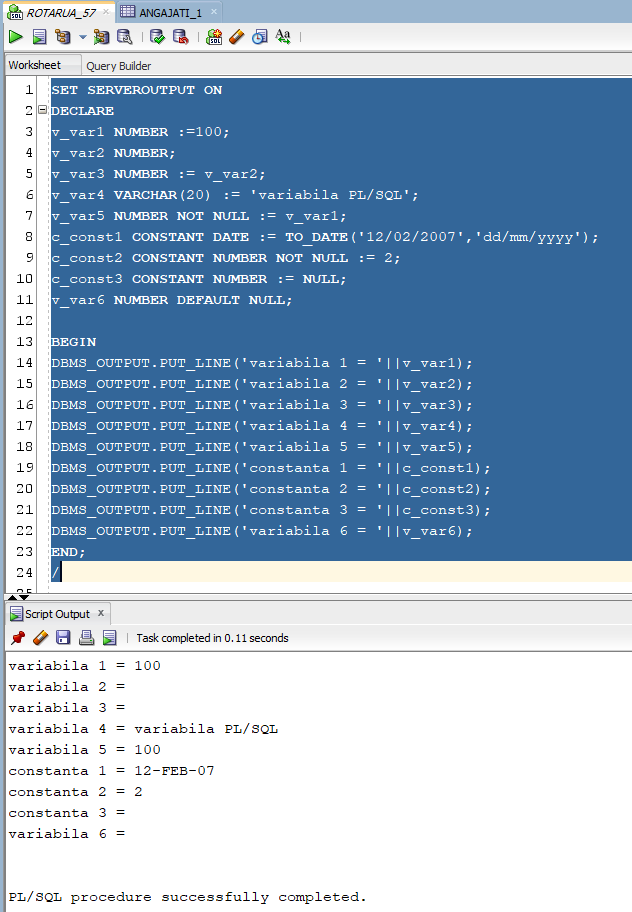
3. Să se selecteze articolele și prețul acestora pentru cele care au prețul < prețul mediu al articolului cu codul 40 fără a utiliza un select imbricat. (prețul id\_articol=40 este 1090)



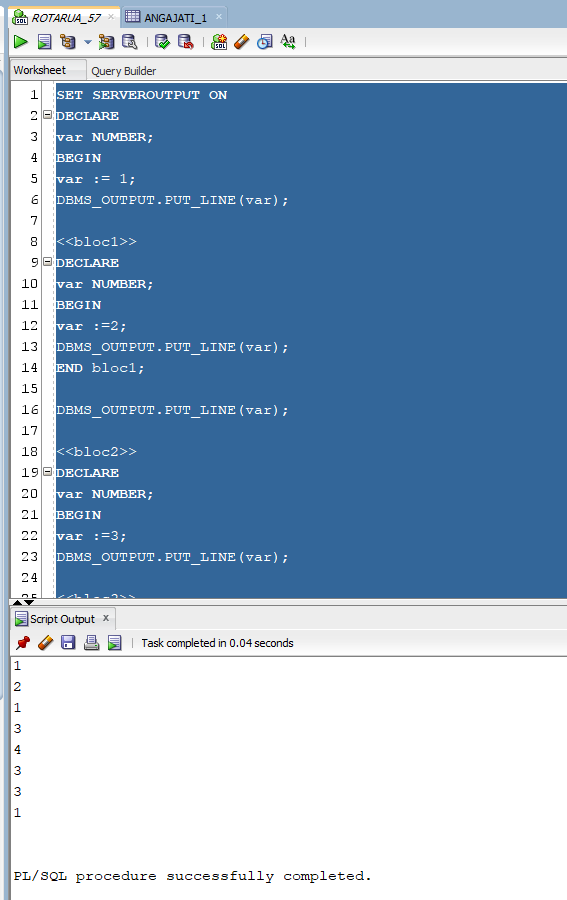
4. Sa se afiseze numarul de comenzi ale angajatului al carui cod este introdus de utilizator prin intermediul variabilei de substitutie &id\_angajat (id-ul introdus este 106).

****

****5. Să se afișeze salariul și prenumele angajatului cu prenumele Felicia.

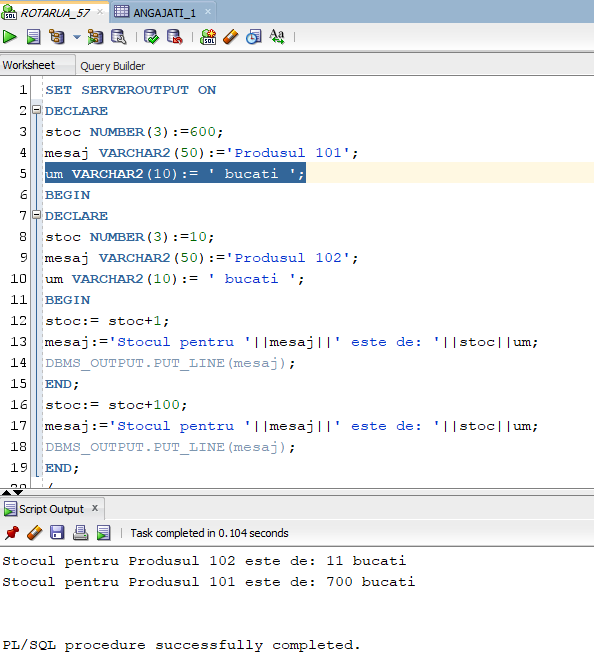
****6. Specificați ce se va afișa la rularea următorului bloc PL/SQL.

7. Specificaţi ce se va afişa la rularea următorului bloc PL/SQL (care conţine blocuri imbricate, ilustrând domeniul de vizibilitate al unor variabile care au acelaşi nume):

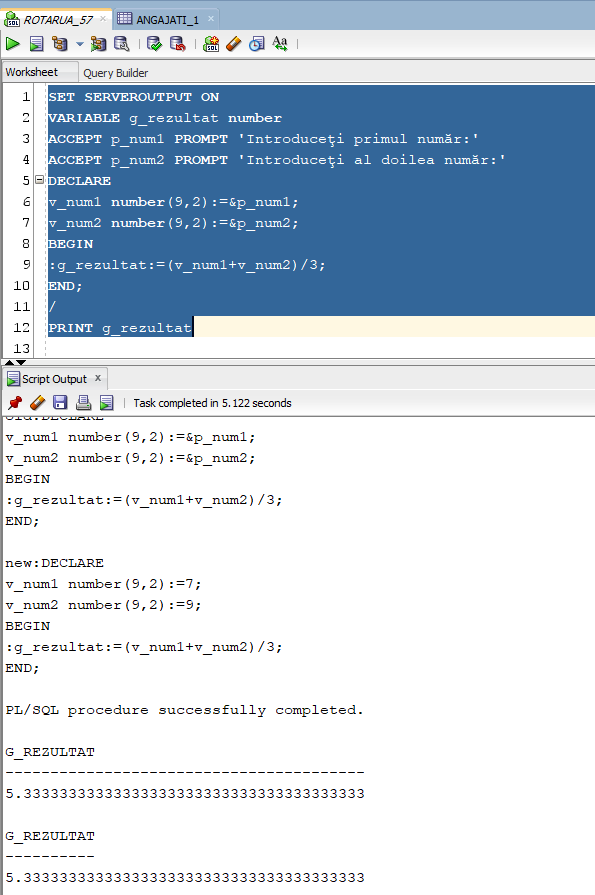
****

8. Specificați de ce apare o eroare la execuția următorului bloc PL/SQL:

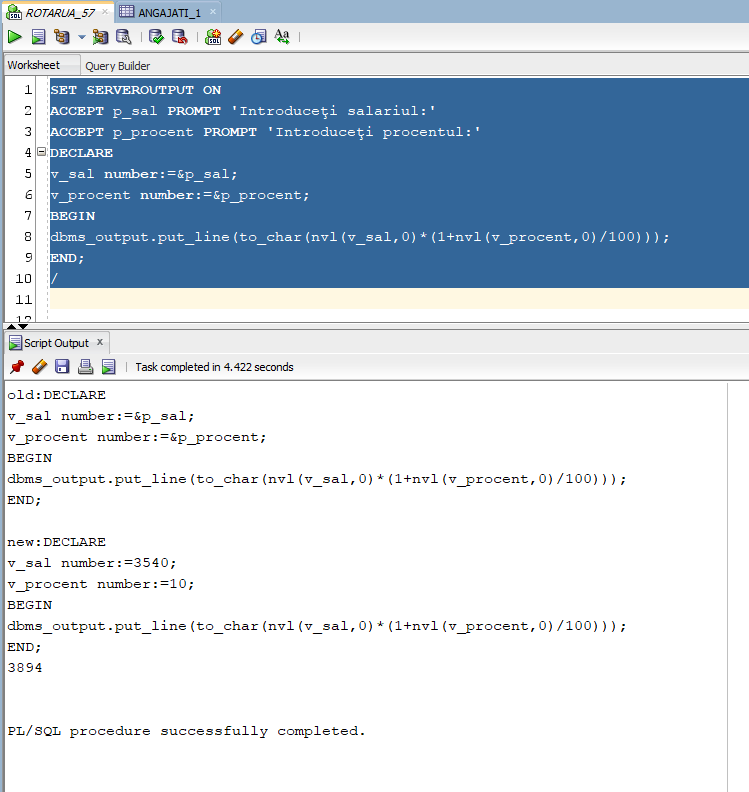
Deoarece **um** nu este declarat

****

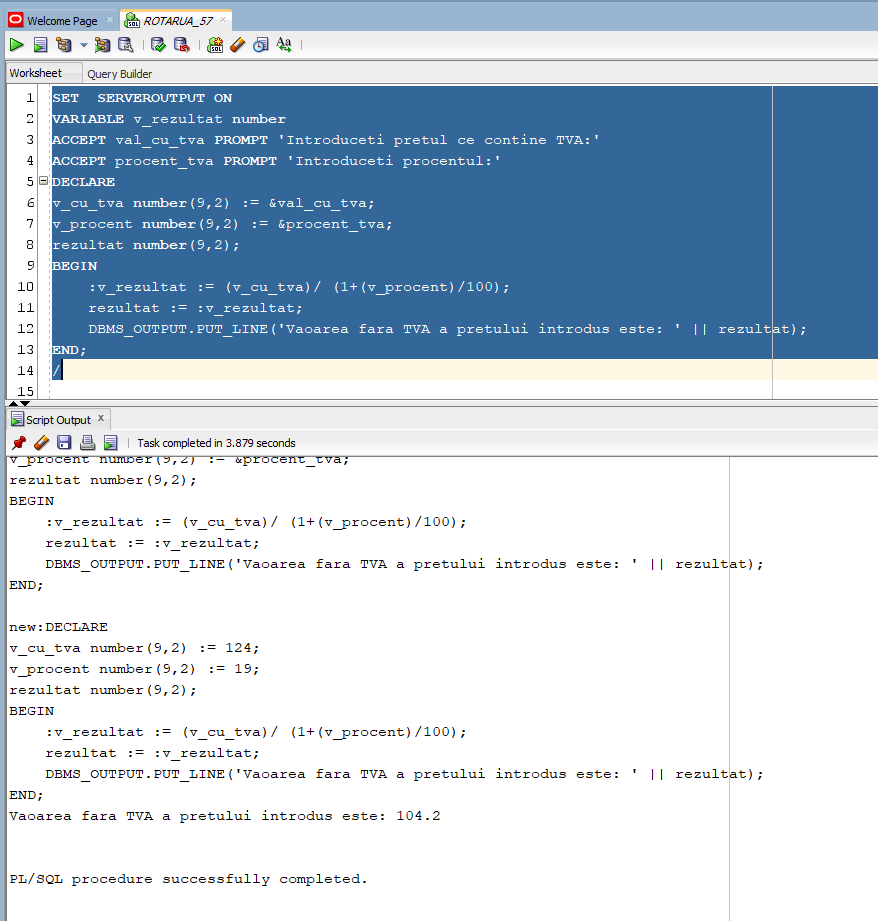
9. Să se calculeze suma a două numere, iar rezultatul să se dividă cu 3. Numerele se vor introduce de la tastatură. (primul nr: 7, al doilea nr: 9)



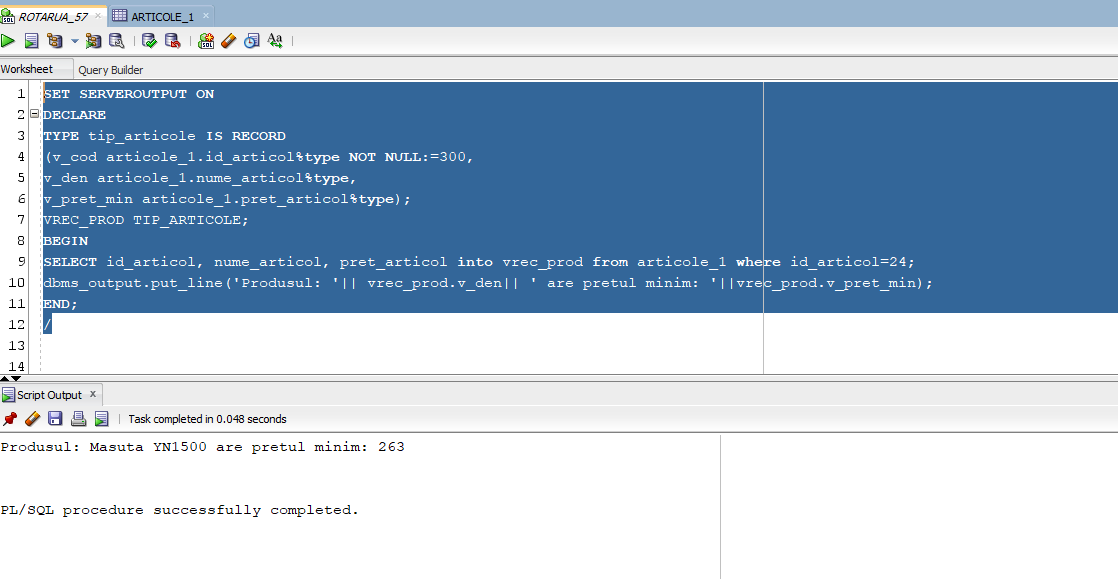
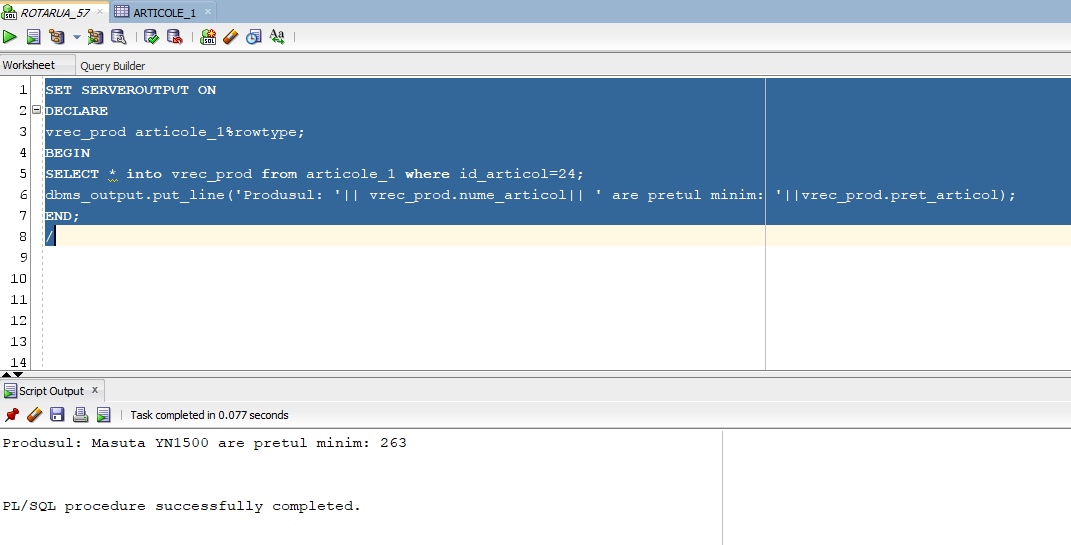
10. Să se afişeze salariul mărit cu un x procente. Salariul şi procentul se citesc de la tastatură. (salariu: 3540, procent: 10)

****

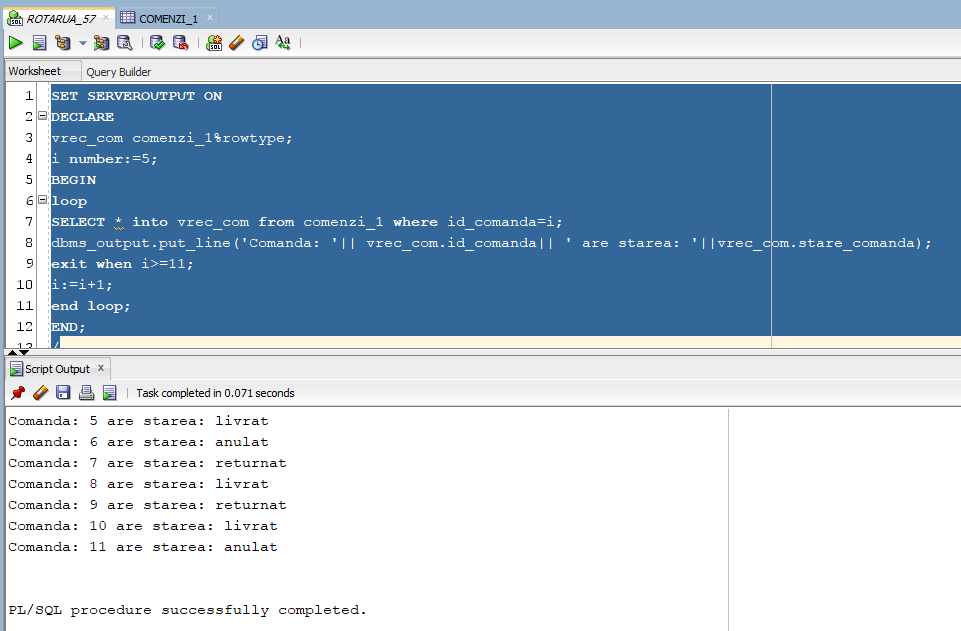
11. Să se calculeze valoarea fără TVA pentru un preț introdus de la tastatură ce conține TVA.

****

12. Utilizând un tip de dată înregistrare definit de utilizator, să se afișeze preţul minim al articolului cu codul 24.

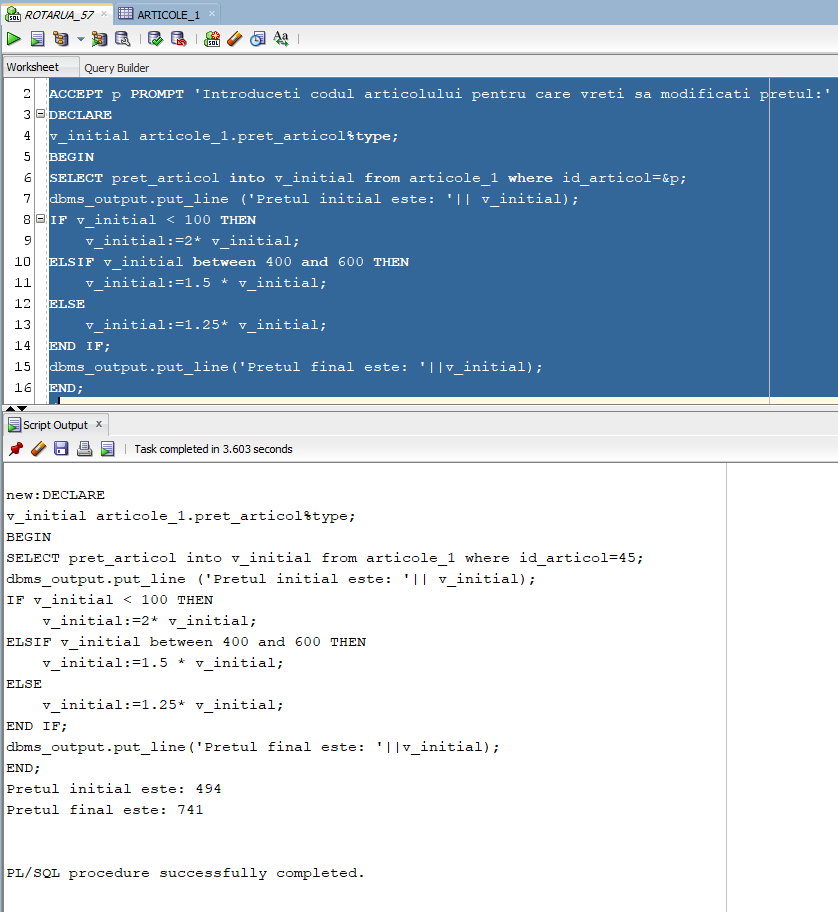
****

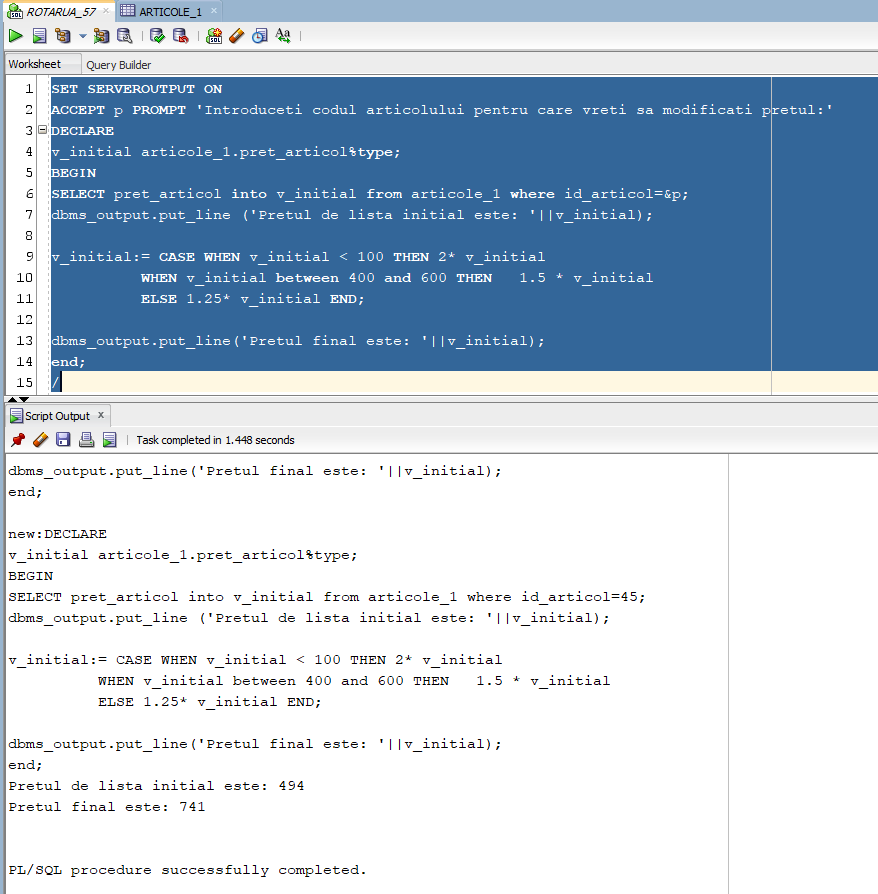
13. Utilizând un tip de dată înregistrare de același tip cu un rând din tabela COMENZI să se afișeze starea fiecarei comenzi cu id-ul: 5, 6, 7, 8, 9, 10, 11.

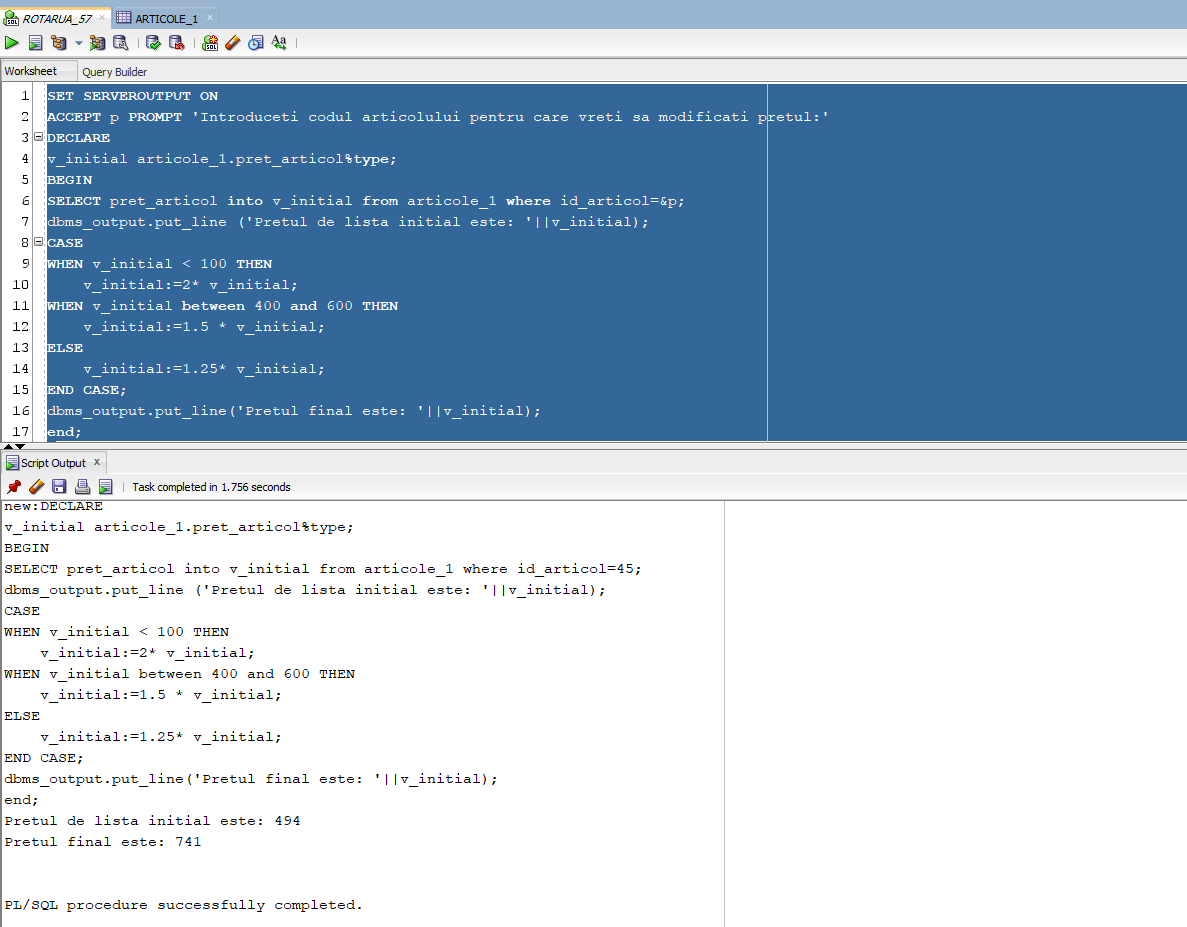
****

# **STRUCTURI FUNDAMENTALE DE PROGRAMARE**

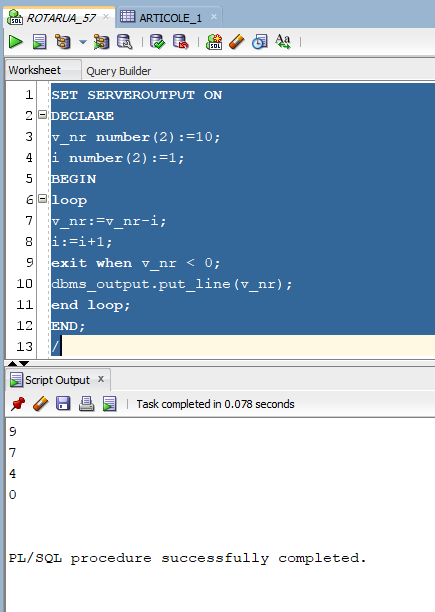
1. În funcție de prețul inițial al articolului, având codul citit de la tastatură, se va afișa modificat pe ecran noua valoare. (id\_articol=45 cu prețul 494)

****

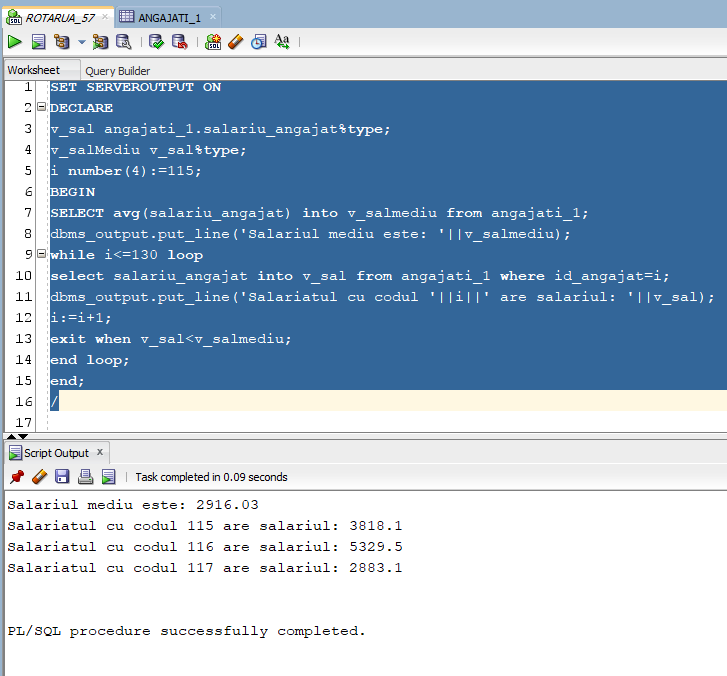
**CASE Expressions:**

**CASE Statement:**

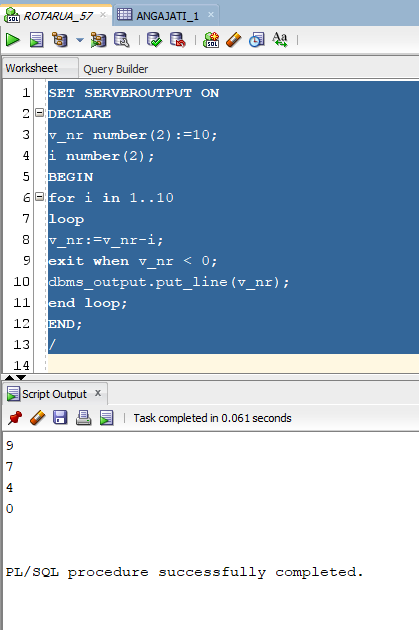
2. Se afişează pe ecran utilizând structura loop…end loop numerele 9, 7, 4, 0.

****

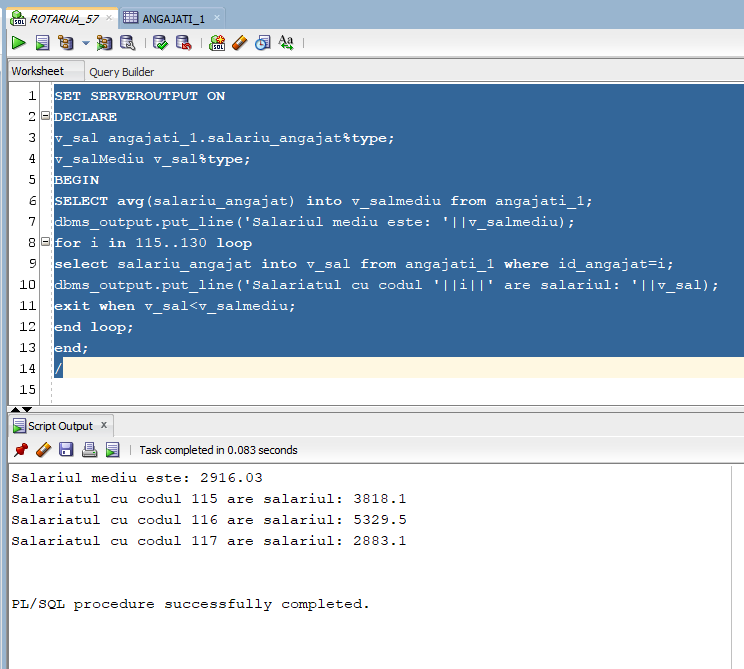
3. Se afişează în ordine angajaţii cu codurile în intervalul 115-130 atât timp cât salariul acestora este mai mic decât media:

****

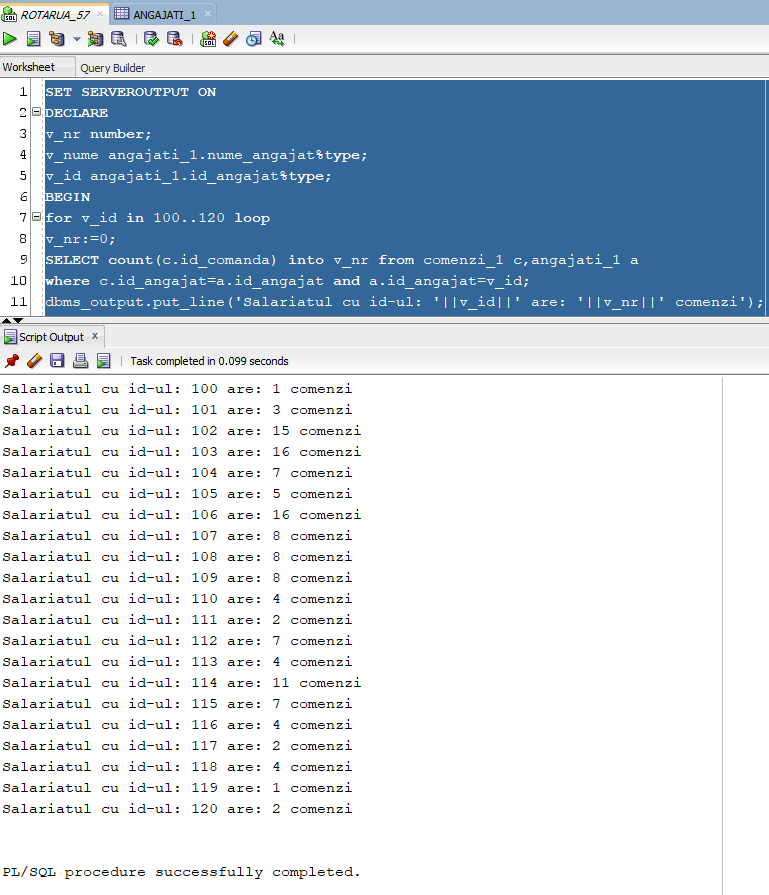
4. Se afişează pe ecran utilizând structura for…end loop numerele 9,7, 4, 0.

****

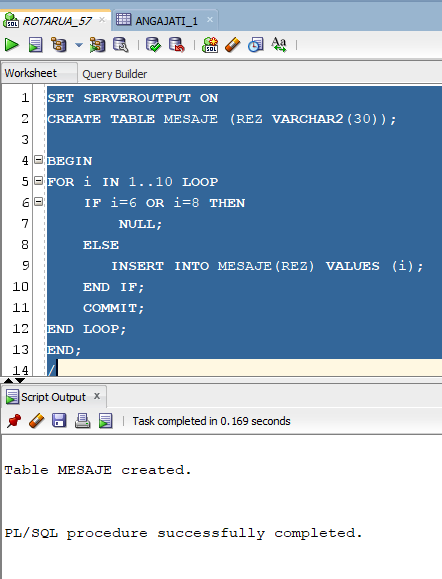
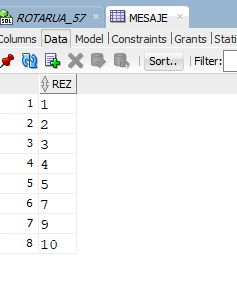
5. Se afişează în ordine angajaţii cu codurile în intervalul 115-130 atât timp cât salariul acestora este mai mic decât media:

****

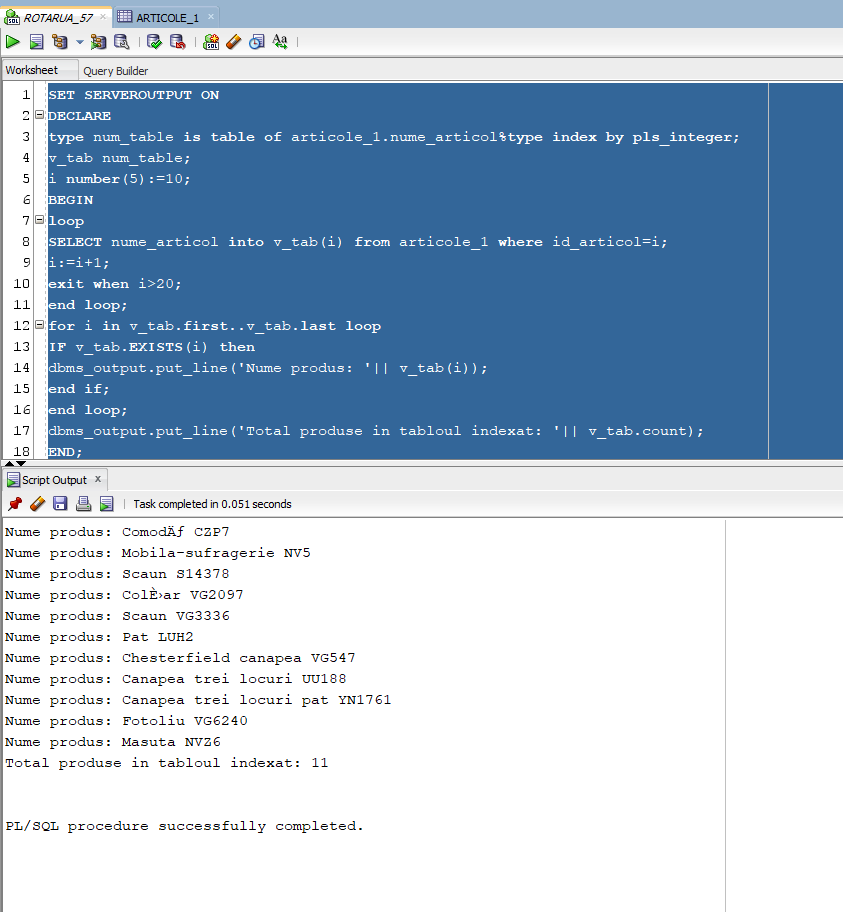
6. Să se afişeze numărul de comenzi ale fiecărui angajat al cărui id este situat în intervalul 100..120, dar să se întrerupă afişarea în cazul în care se găseşte primul angajat din acest interval care nu are nici o comandă:

****

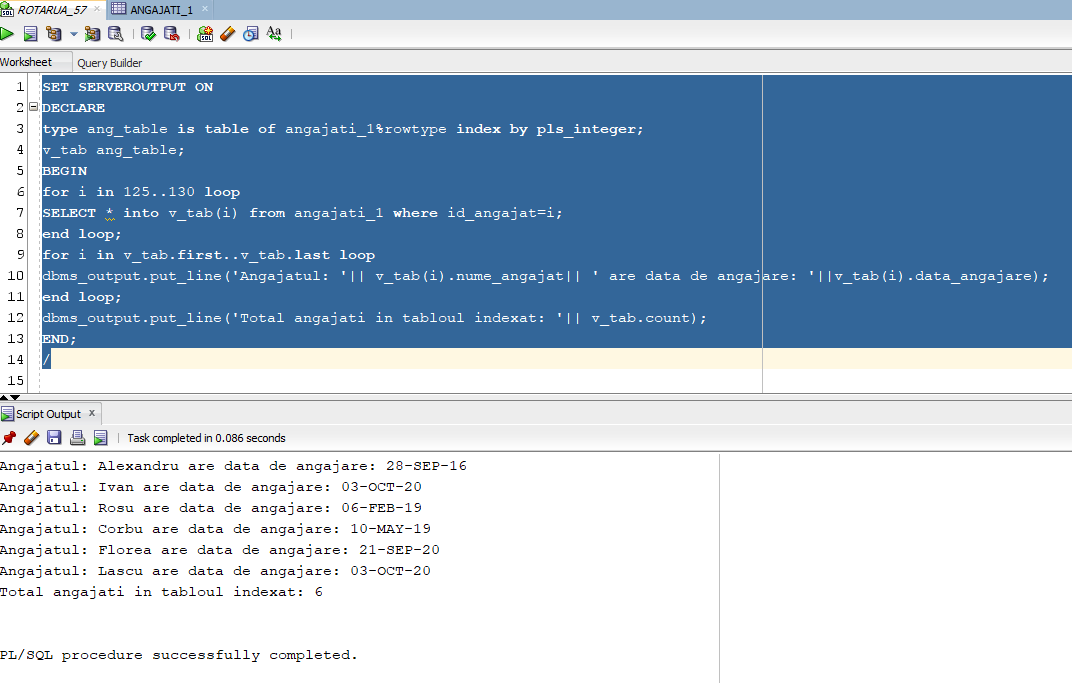
7. Să se încarce în tabela MESAJE numere de la 1…10 cu excepţia lui 6 şi 8.

****

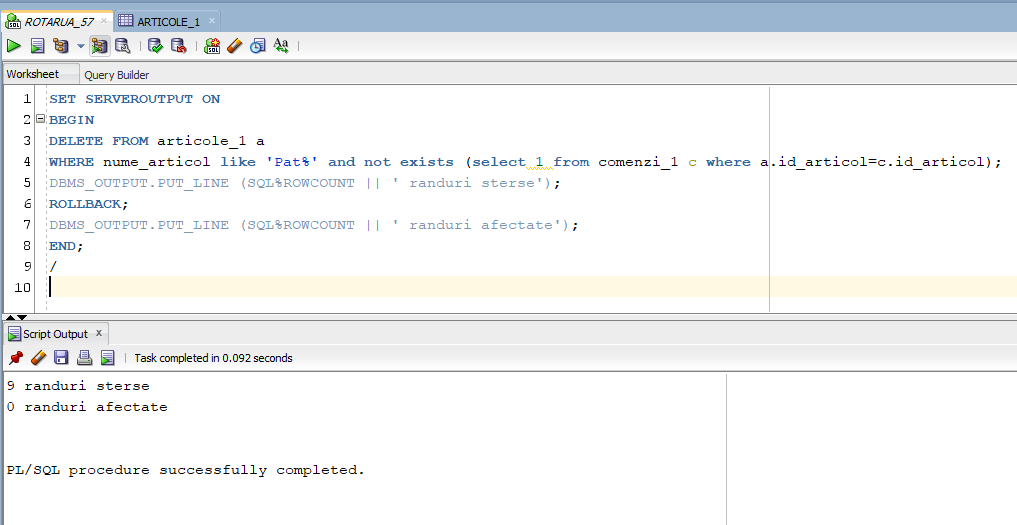
8. Utilizarea unui tablou indexat de tipul ARTICOLE\_1.DENARTICOL

****

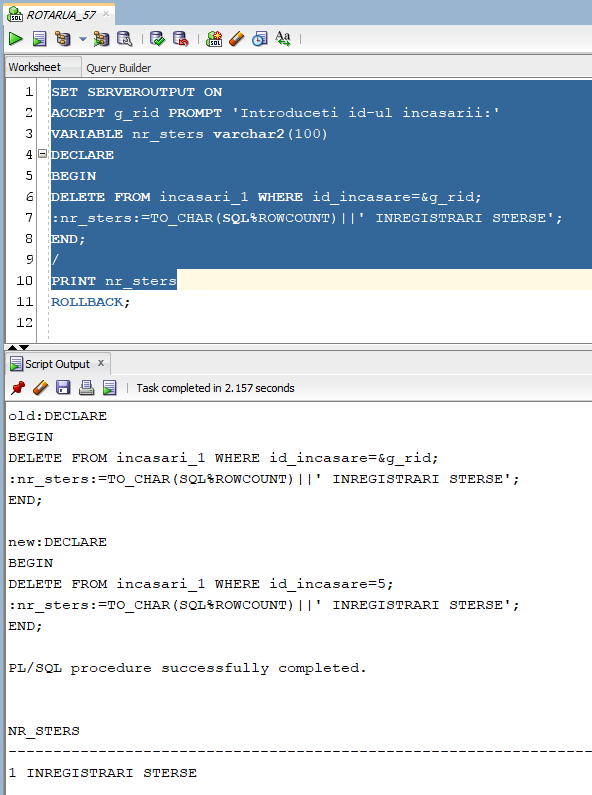
9. Utilizarea unui tablou indexat de același tip cu un rând din tabela ANGAJATI\_1 - %ROWTYPE

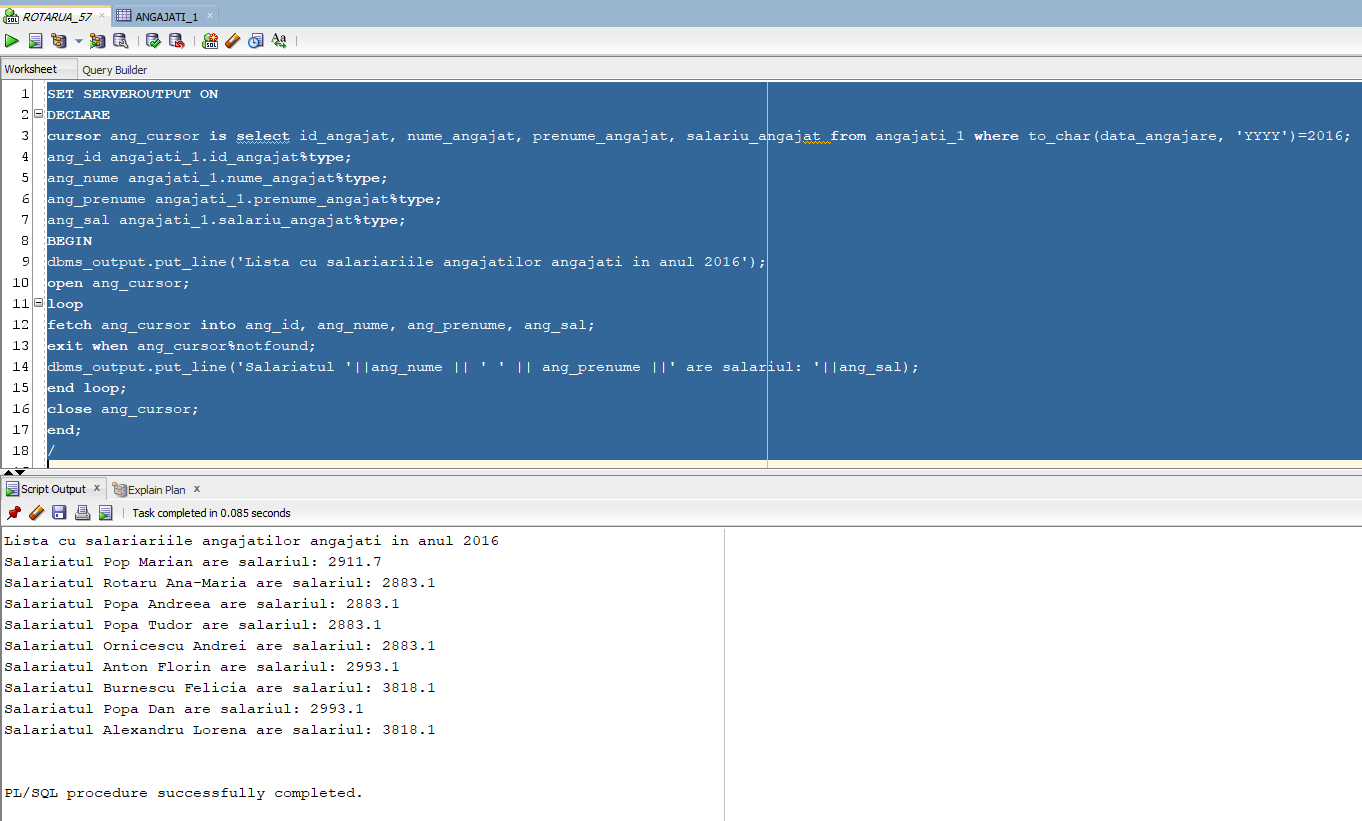
****

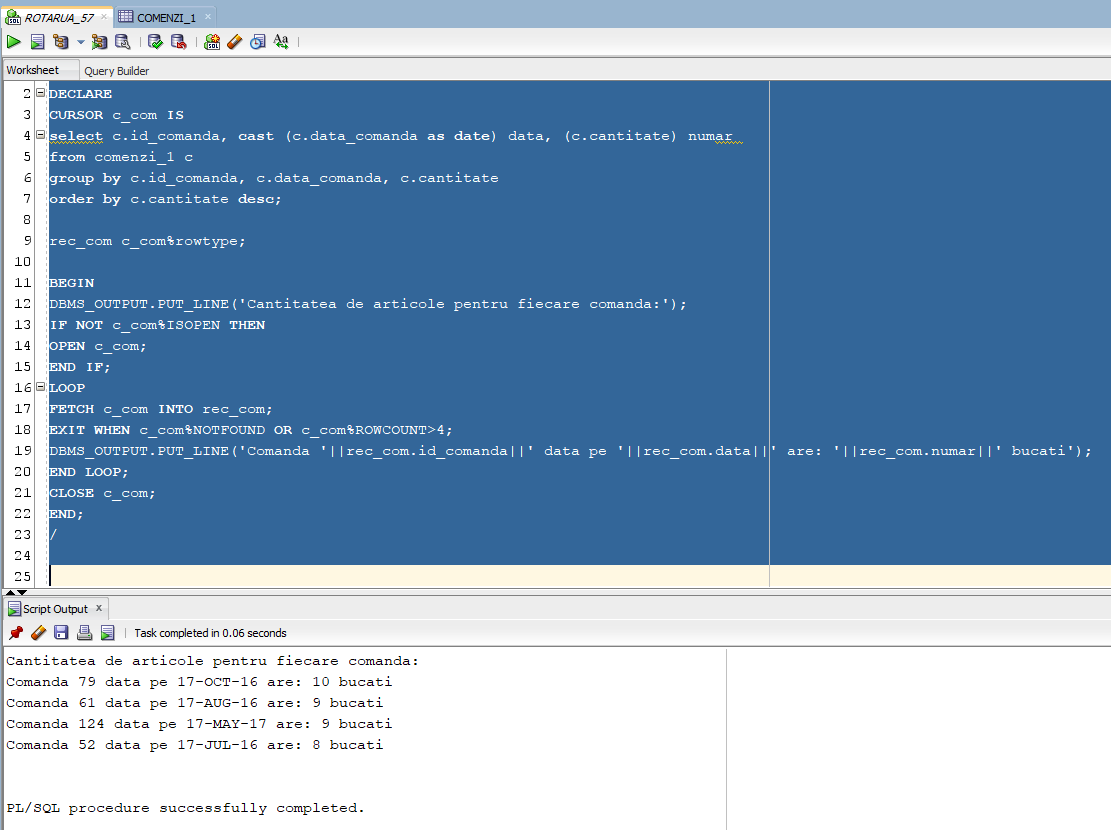
# **CURSORUL ÎN PL/SQL**

1. Se șterg articolele ‘Pat’ care nu au fost comandate. Se afișează numărul de rânduri şterse.

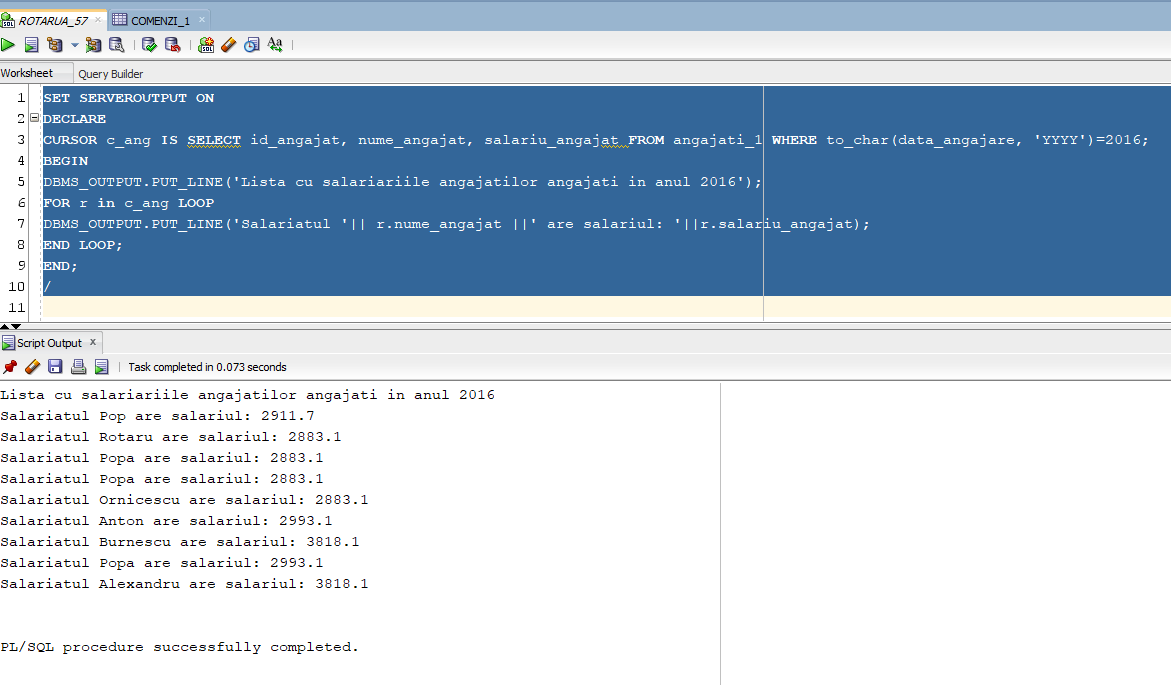
2. Se șterge din tabela INCASARI\_1, încasarea a cărei ID este introdus de utilizator prin intermediul variabilei de substituție *g\_rid.* Mesajul este afișat folosind variabila de mediu *nr\_sters*.

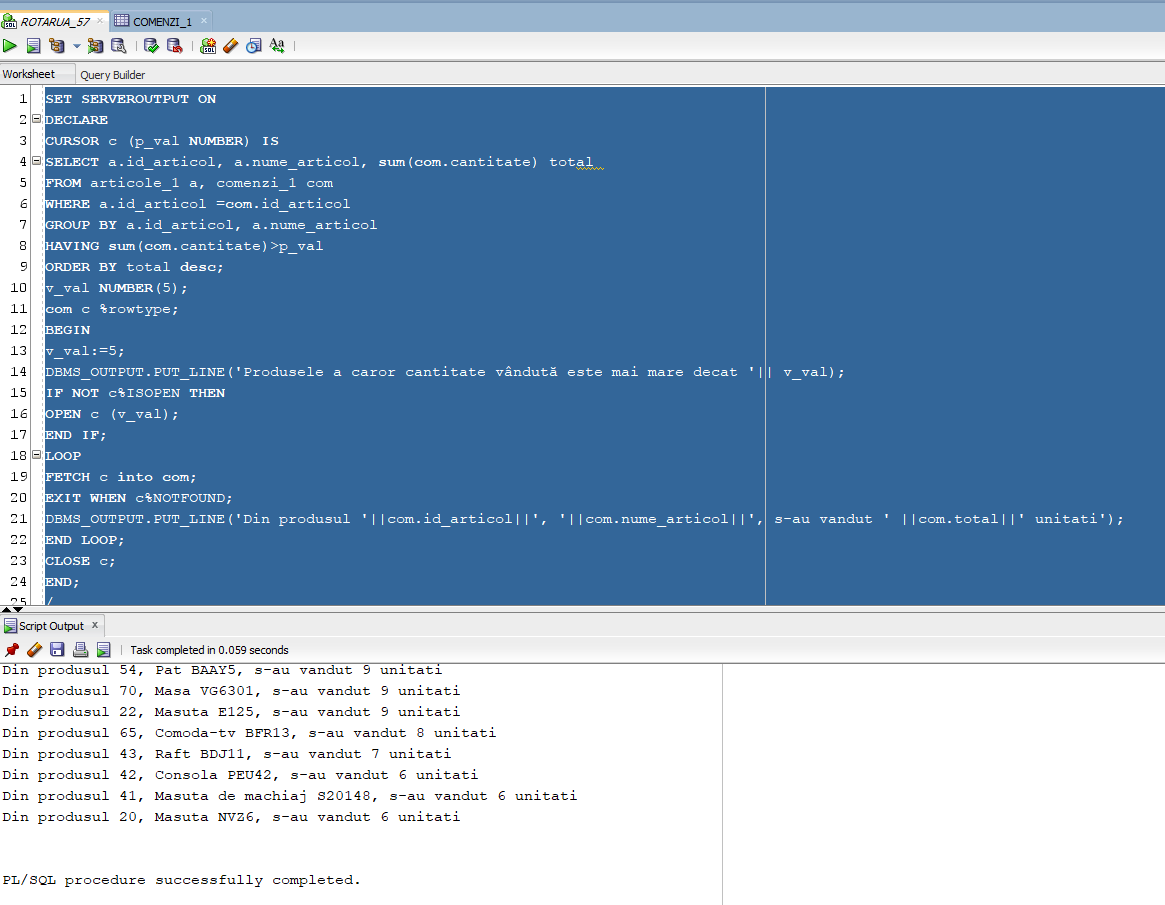


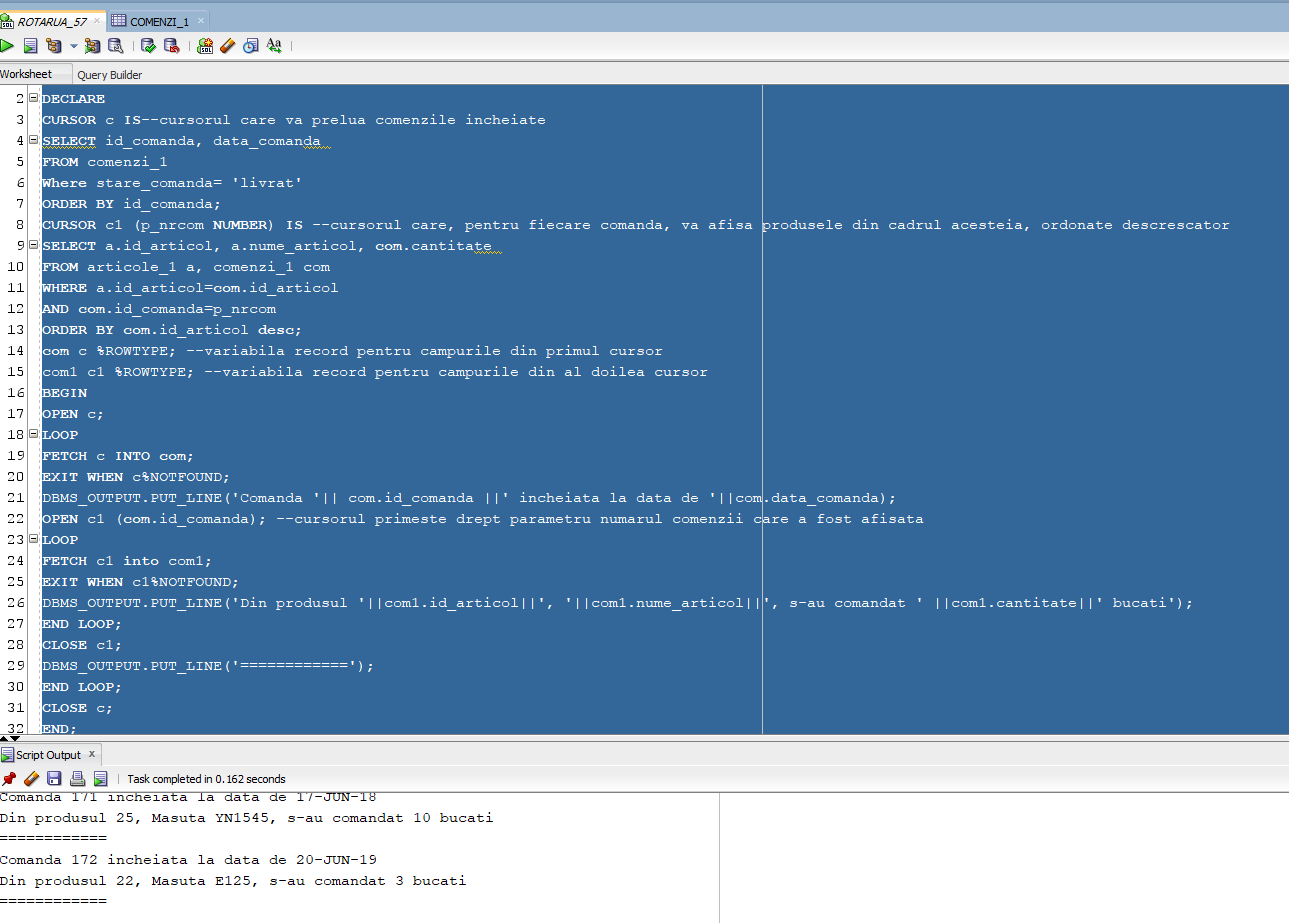
3. Să se afişeze lista cu numele şi salariul angajaţilor din anul de angajare “2016” folosind un cursor explicit și patru variabile scalare.

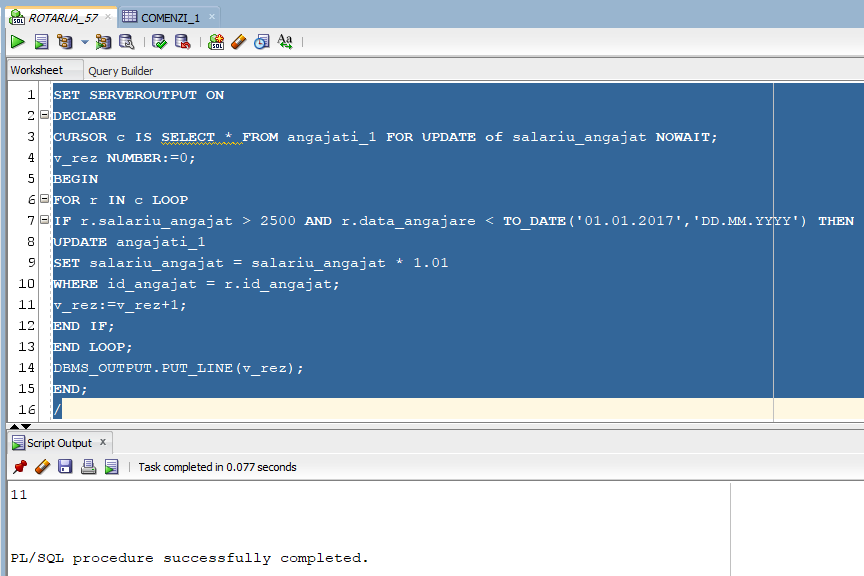


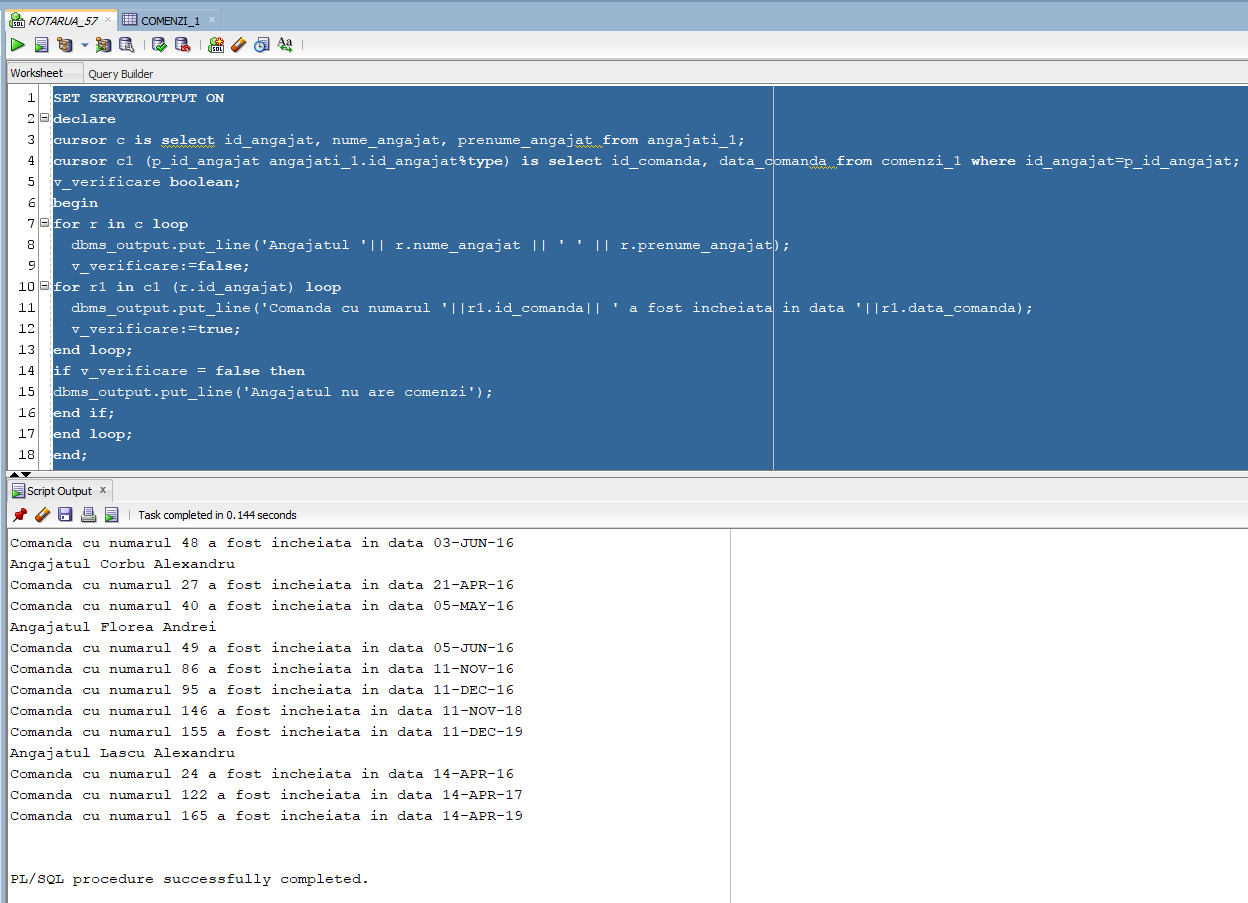
4. Să se afişeze primele 4 comenzi care au cea mai mare cantitate de articole comandate. În acest caz înregistrările vor fi ordonate descrescător în funcţie de număr.

5. Se afişează printr-un ciclu FOR numele şi salariile angajaţilor din anul angajarii 2016.

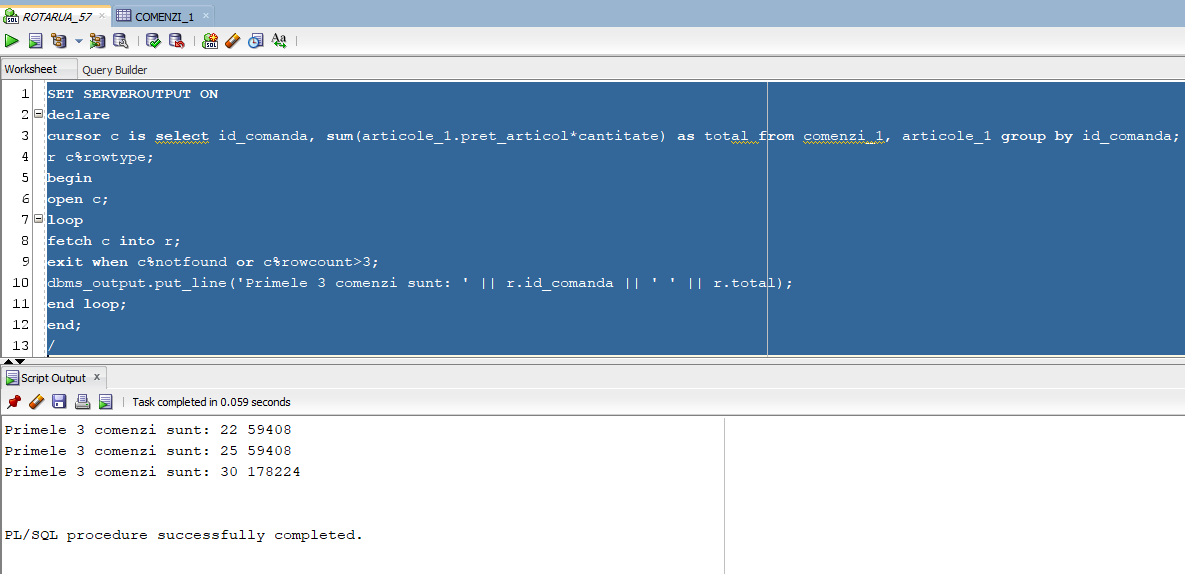
6. Să se afişeze produsele al căror cantitate totală comandată este mai mare decât o valoare primită drept parametru.

7. Să se afişeze pentru fiecare comanda produsele comandate. În acest caz se utilizează două variabile de tip cursor.

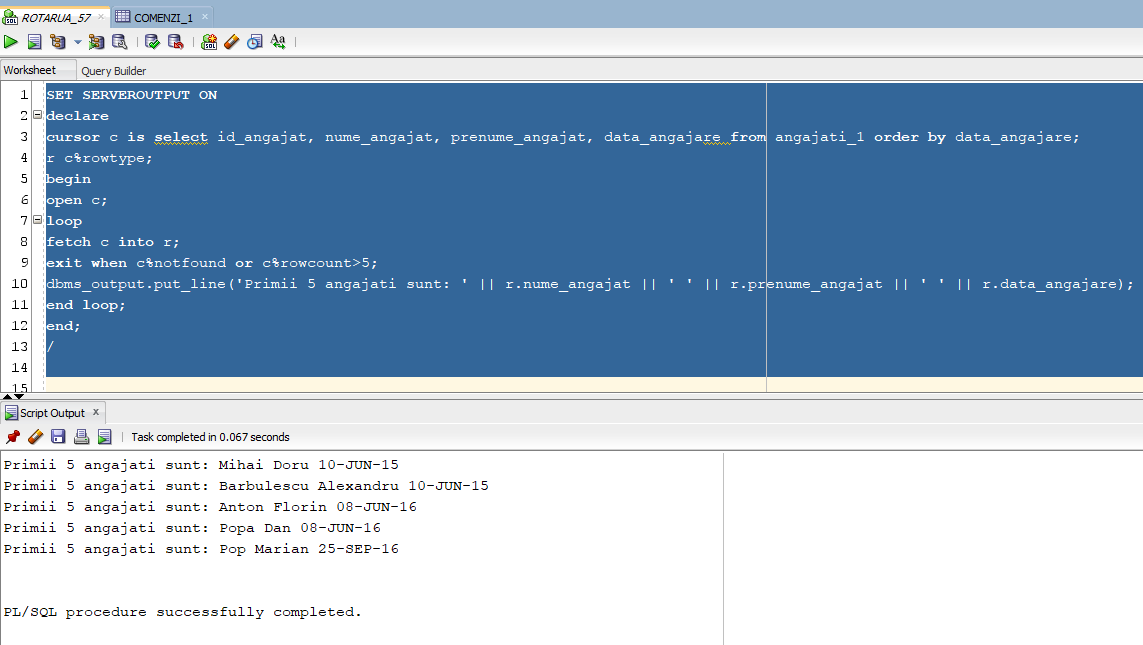
8. Se actualizează salariul dacă acesta este mai mare de 2500 şi au fost angajati înainte de 2017. Se afişează numărul de înregistrări modificate.

9. Afişaţi toţi angajaţii şi comenzile încheiate de fiecare dintre aceştia. Folosiţi un cursor pentru a încărca numele angajaţilor şi un cursor parametrizat pentru încărcarea comenzilor încheiate de aceştia. Pentru angajatii care nu au comenzi se va afisa mesajul corespunzător.

10. Afişaţi informaţii despre primele 3 comenzi care au cea mai mare valoare.

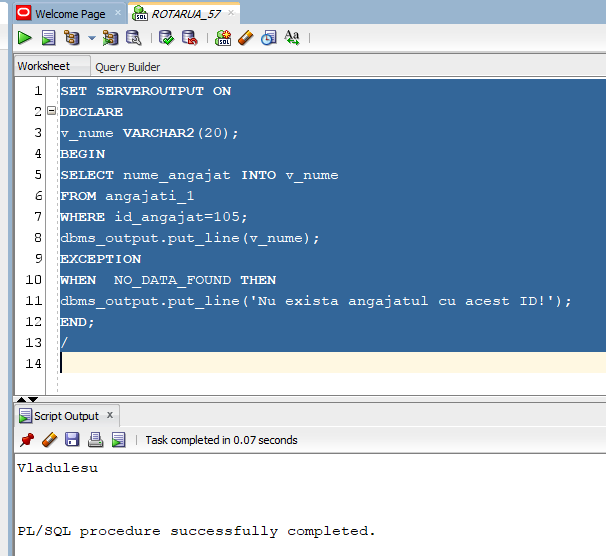


11. Afişaţi informaţii despre primii 5 salariaţi angajaţi (se va realiza filtrarea în funcţie de câmpul Data\_Angajare).

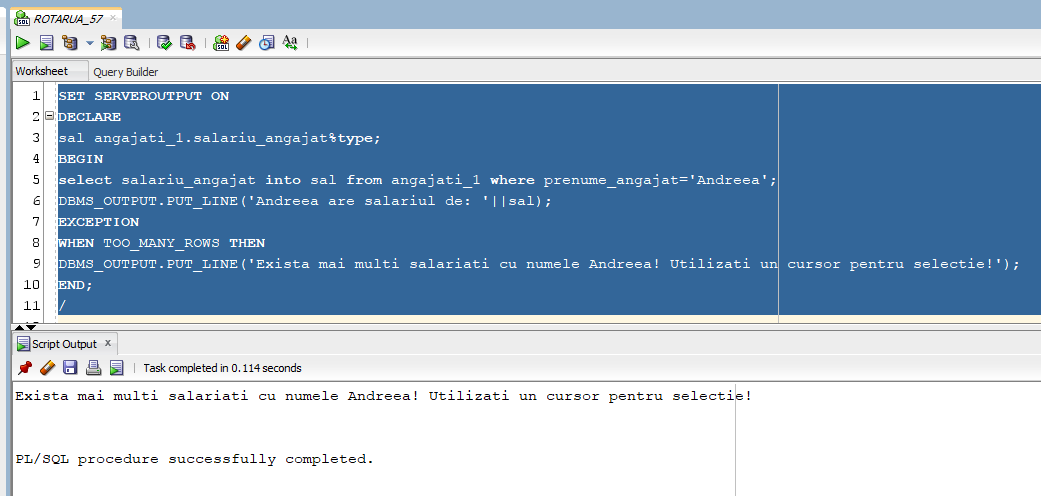


# **TRATAREA EXCEPȚIILOR**

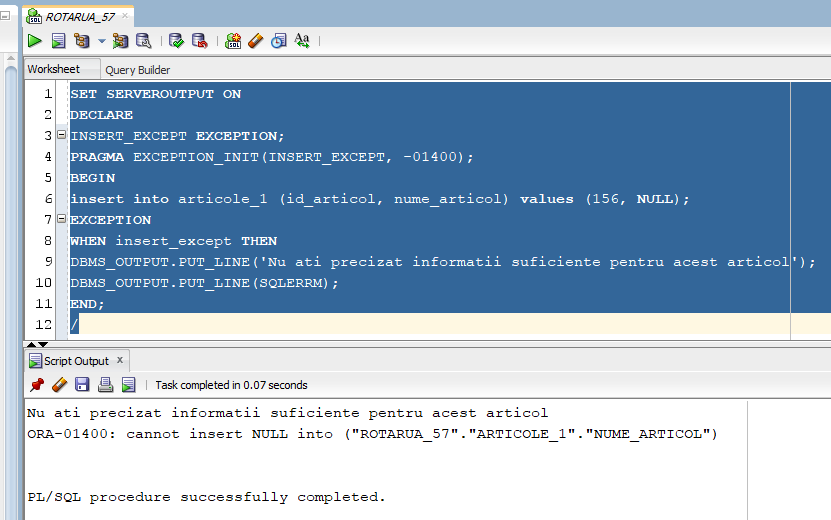
1. Să se afişeze angajatul cu codul 105. Să se trateze eroarea apărută în cazul în care nu există nici un angajat cu acest cod.



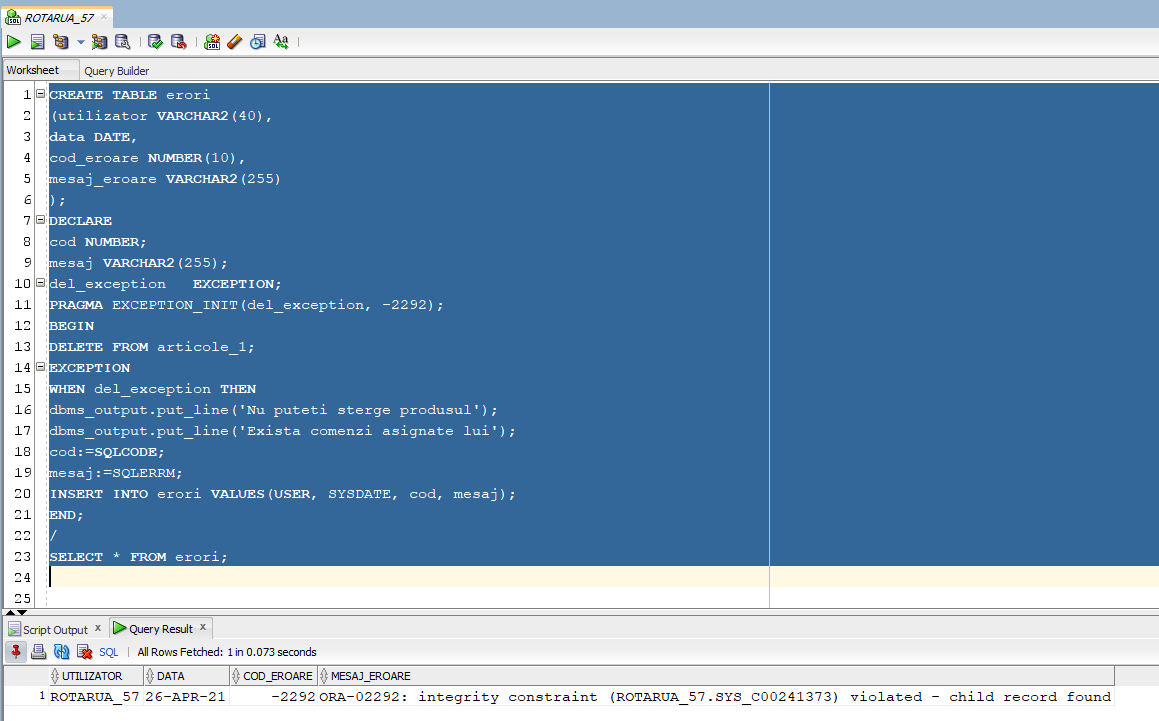
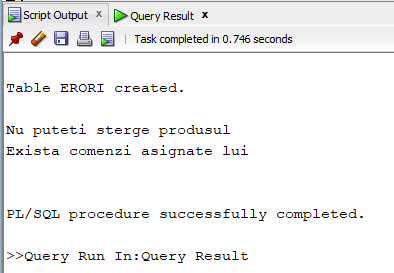
2. Să se afişeze salariul angajatului cu prenumele Andreea. Să se trateze eroare apărută în cazul în care există mai mulţi angajaţi cu acelaşi nume.



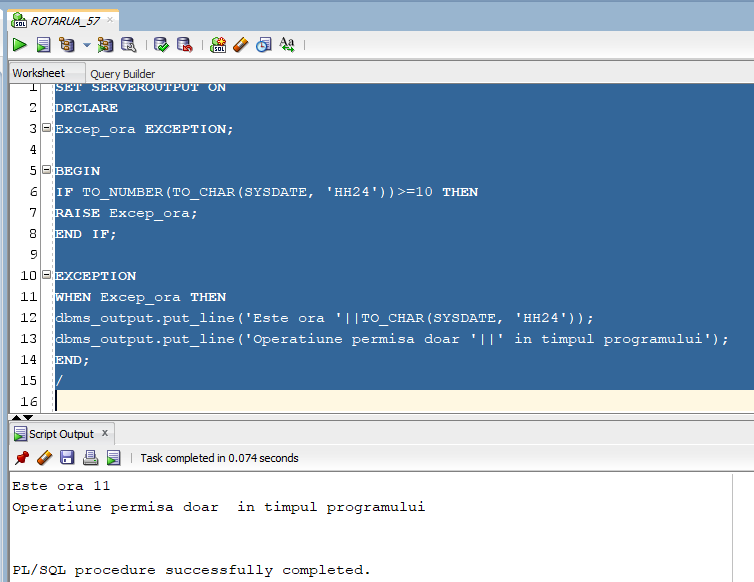
3. Să se insereze în tabela articole un nou articol cu ID-ul 156, fără a preciza denumirea acestuia.

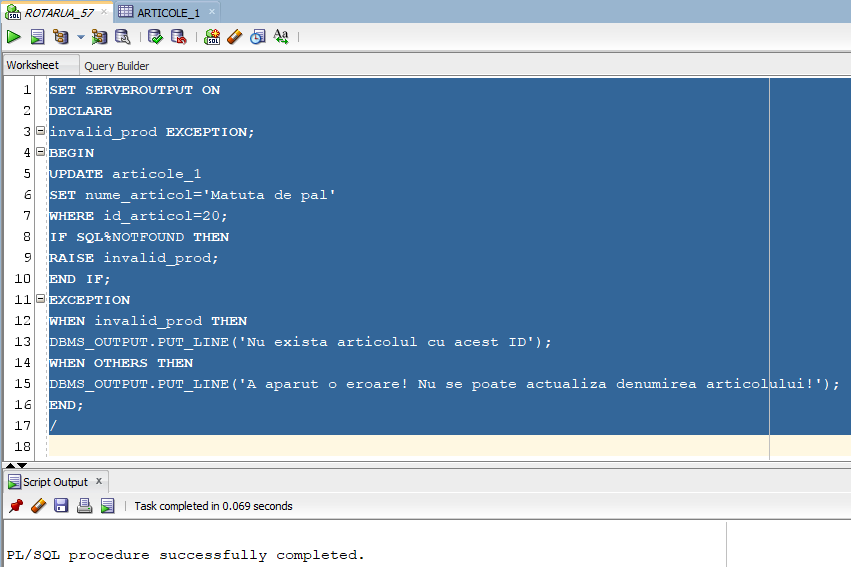


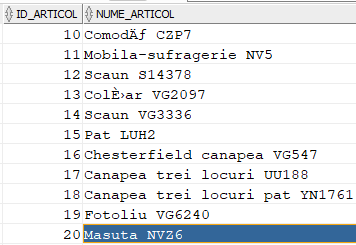
4. Să se şteargă toate înregistrările din tabela articole. Acest lucru va duce la apariţia erorii cu codul –2292, reprezentând încălcarea restricţiei referenţiale. Valorile SQLCODE şi SQLERRM vor fi inserate în tabela ERORI.

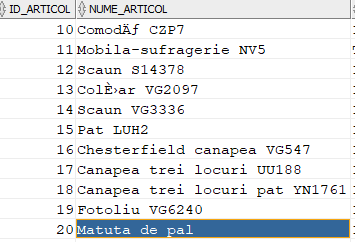


5. Să se invoce o excepție în cazul în care utilizatorul încearcă să execute blocul PL/SQL după ora 10:00.



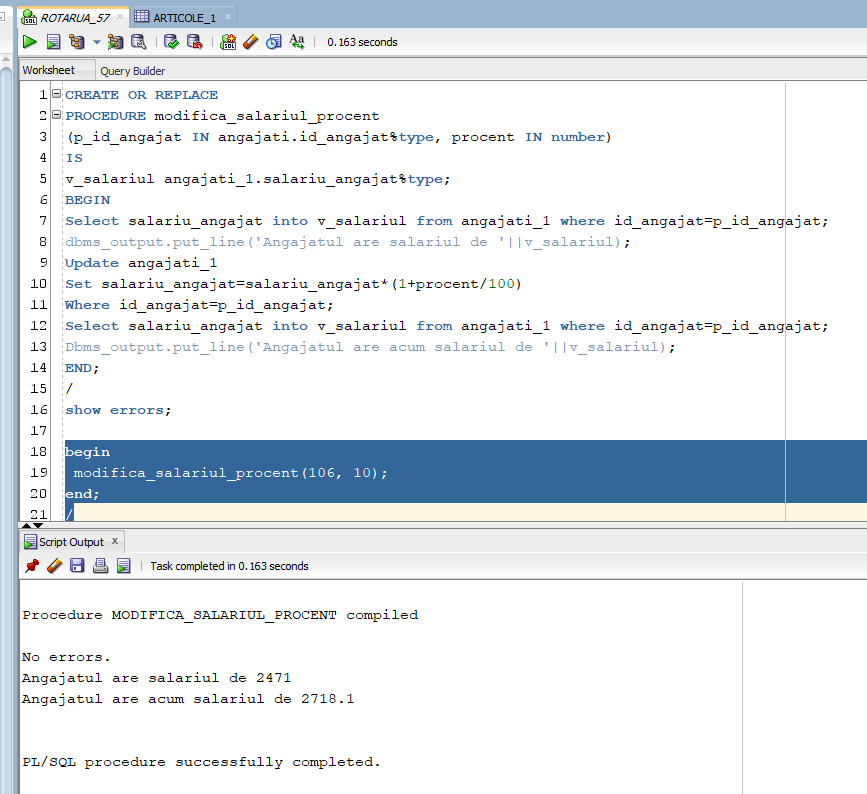
6. Să se modifice denumirea articolului cu id-ul 25. Dacă nu se produce nici o actualizare sau dacă apare o altă excepție atunci să se declanşeze o excepţie prin care să fie avertizat utilizatorul.



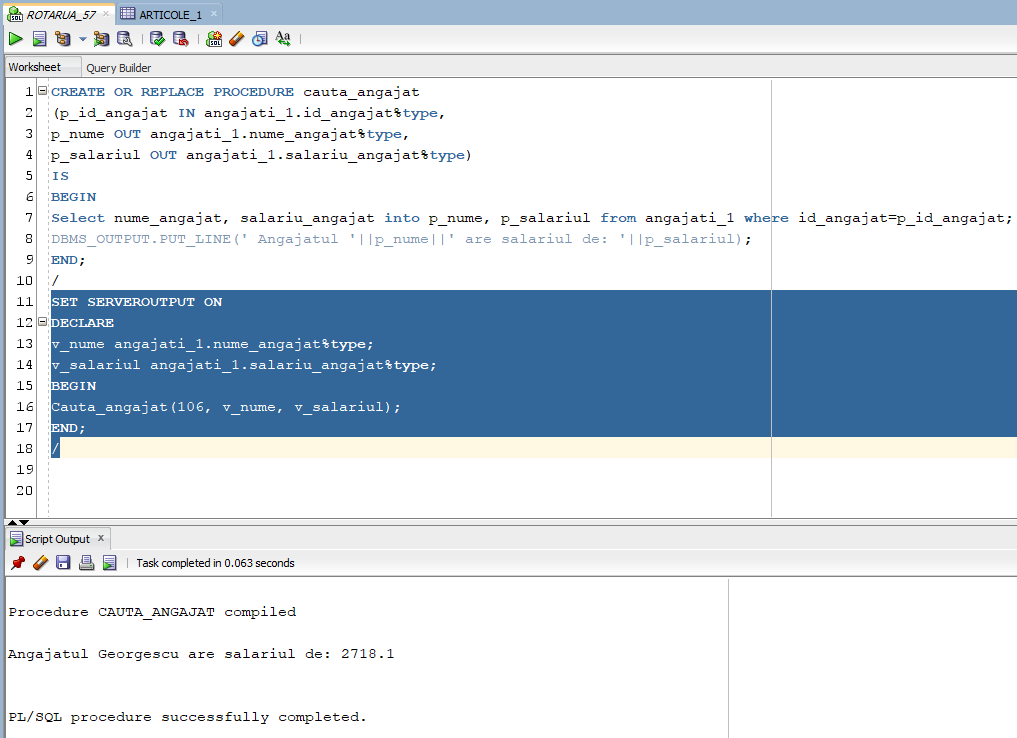


# **SUBPROGRAME PL/SQL**

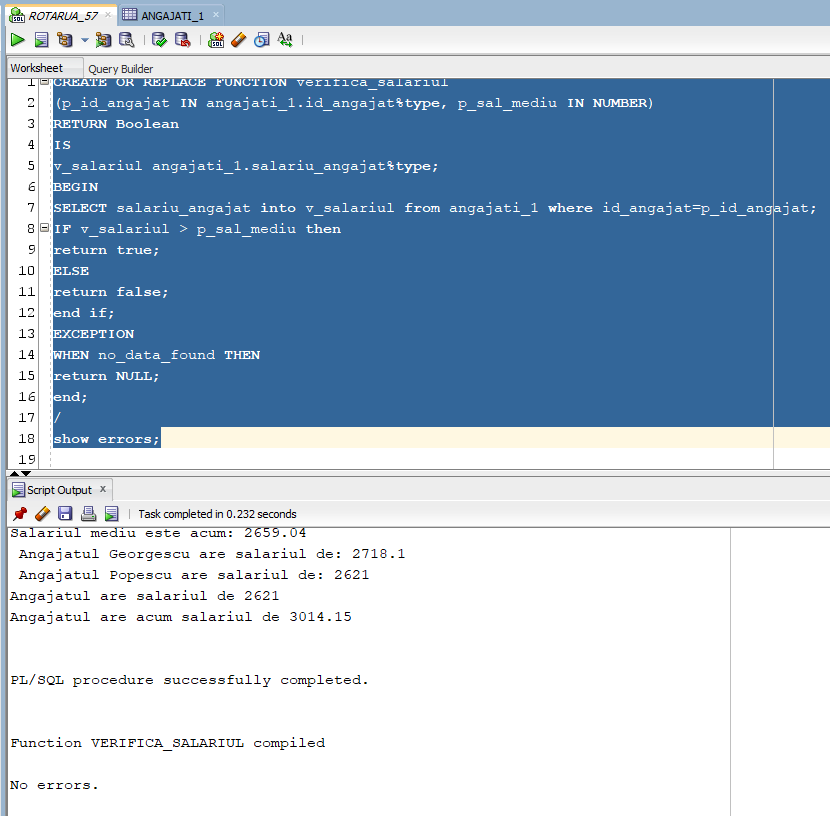
1. Procedura modifica\_salariul primeşte doi parametrii: p\_id\_angajat şi procent şi majorează cu procentul specificat salariul angajatului cu id\_angajat=p\_id\_angajat.

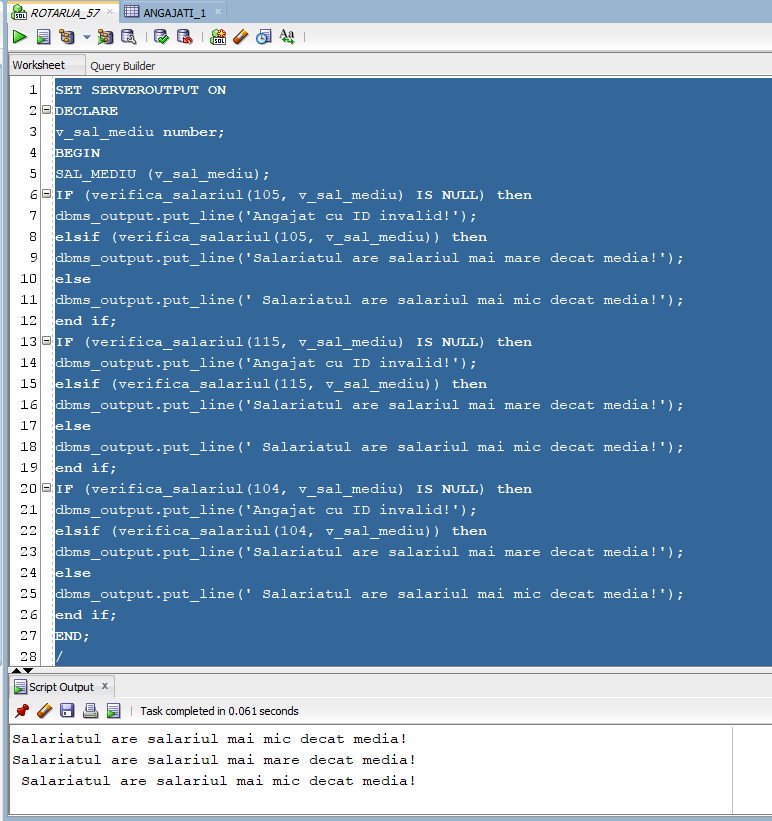


2. Procedura primeşte ca parametru de tip IN id\_ul unui angajat şi returnează prin parametrii de tip OUT numele şi salariul acestuia.

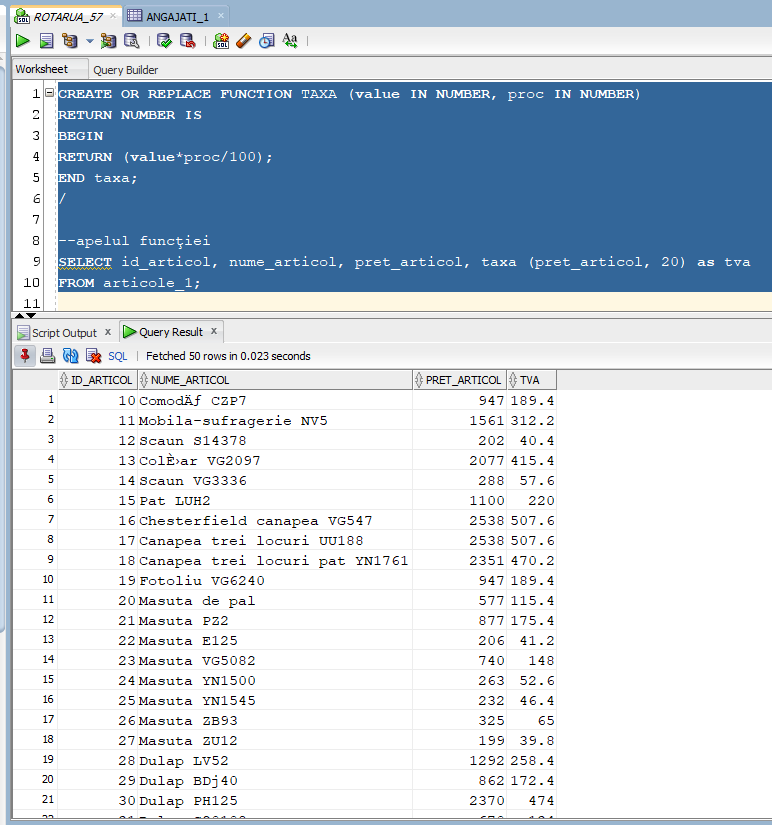


3. Funcția verifica\_salariul returnează TRUE/FALSE daca salariatul are salariul mai mare/mai mic sau egal cu salariul mediu si NULL daca salariatul nu exista

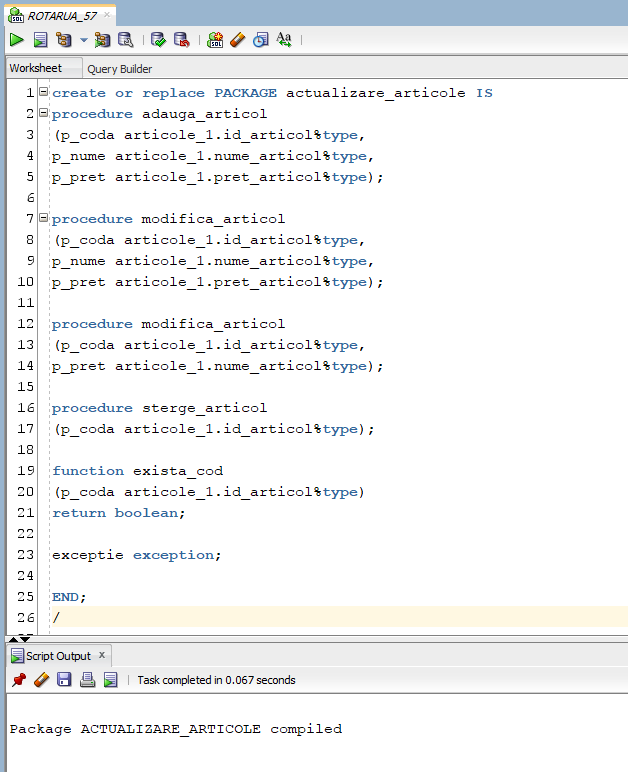




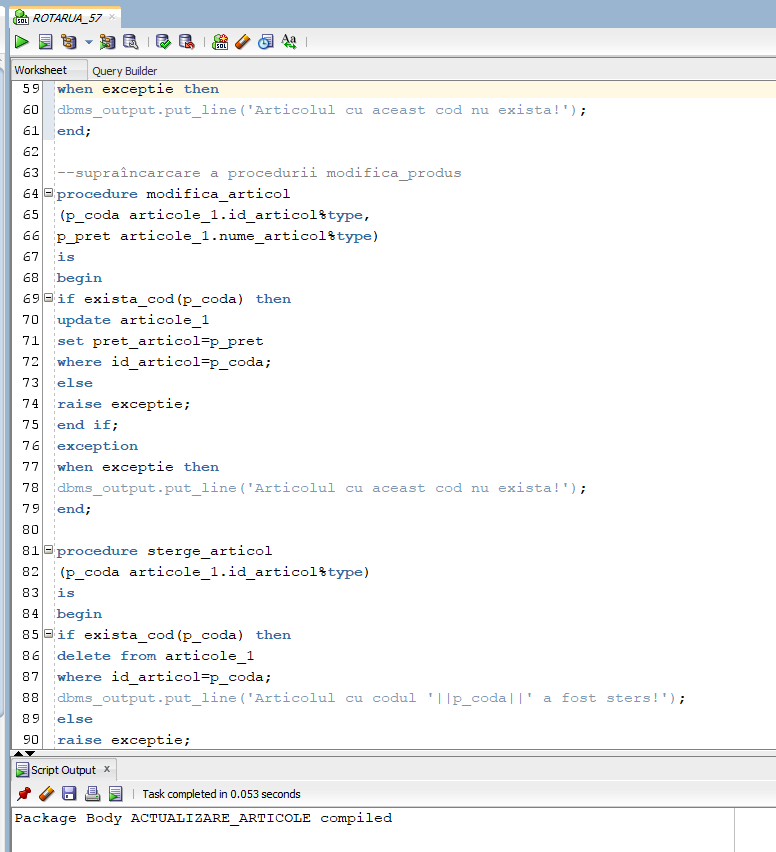
4. Adăugați tva pentru produsele din tabela ARTICOLE.

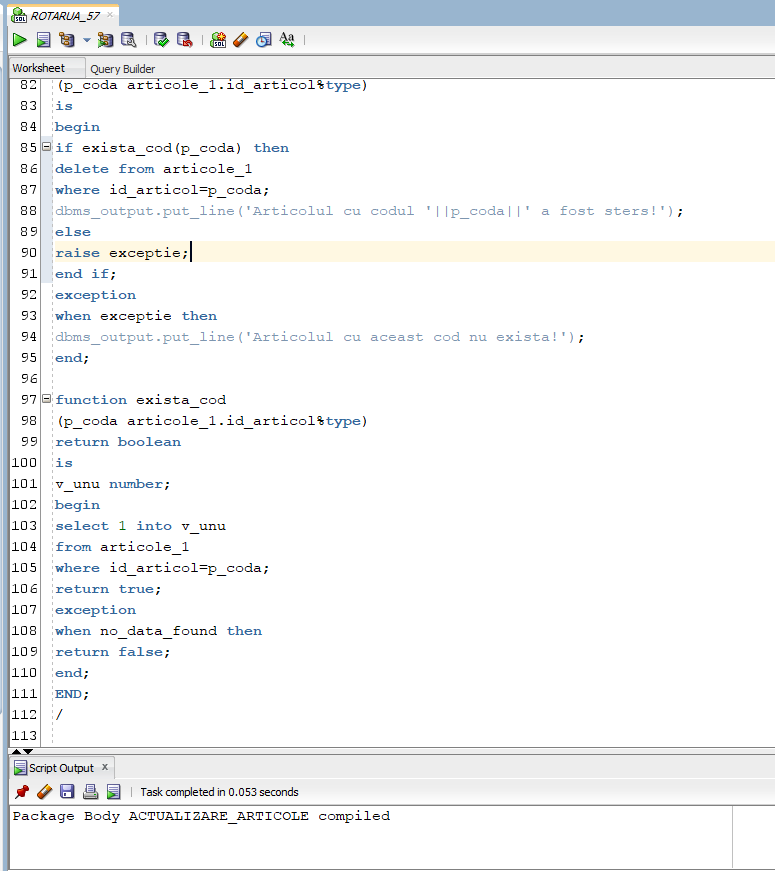


# **PACHETE DE SUBPROGRAME**

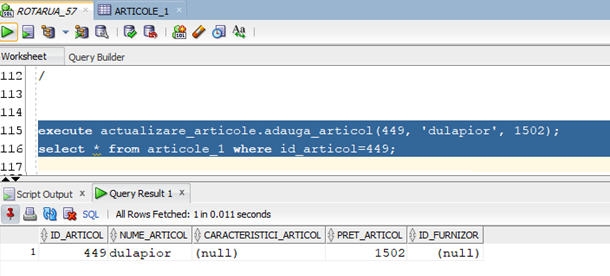
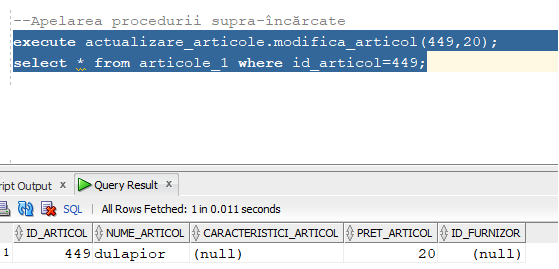


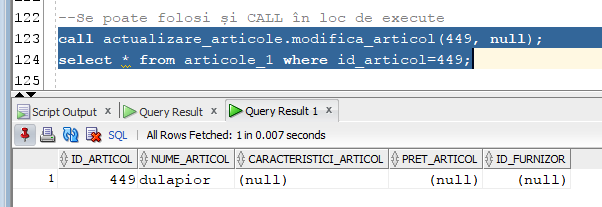




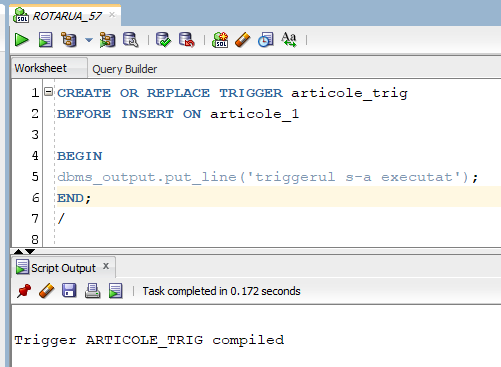


2. Apelul procedurilor / funcțiilor din cadrul pachetului:

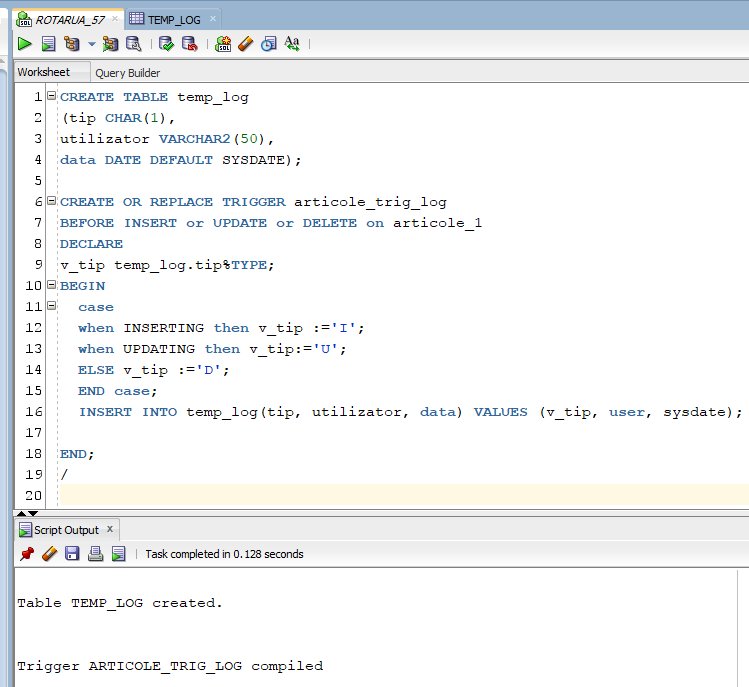


****

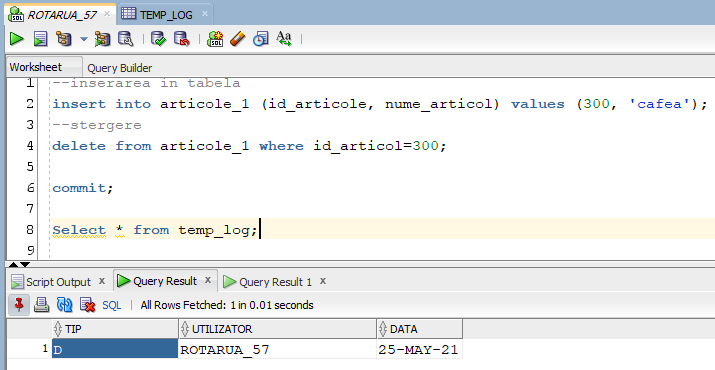
# **TRIGGERI PE BAZA DE DATE**

1. Se creează un trigger care se declanşează înaintea fiecărei operaţii de inserare în tabela articole\_1.

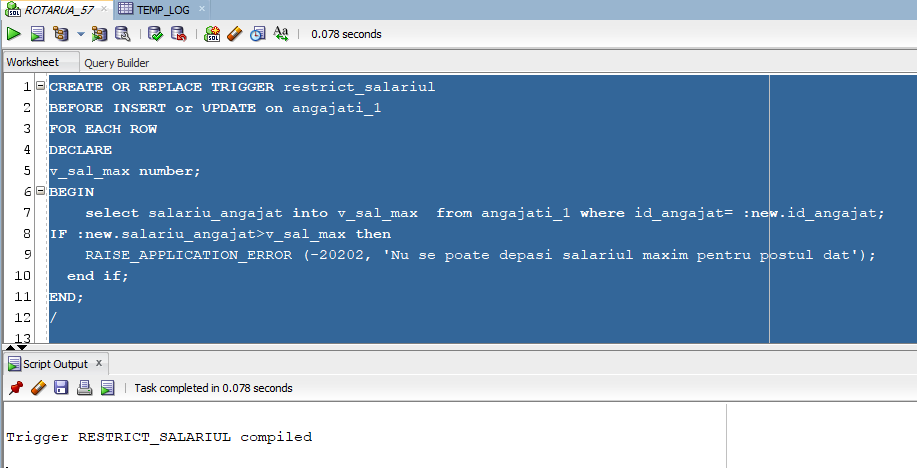
2. In tabela TEMP\_LOG se introduce tipul operaţiei, utilizatorul care a executat-o, data curentă.



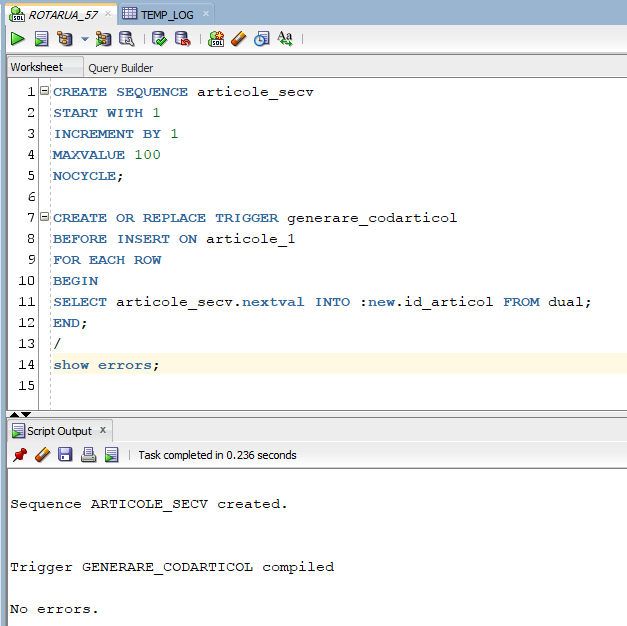
**Verificarea execuției:**



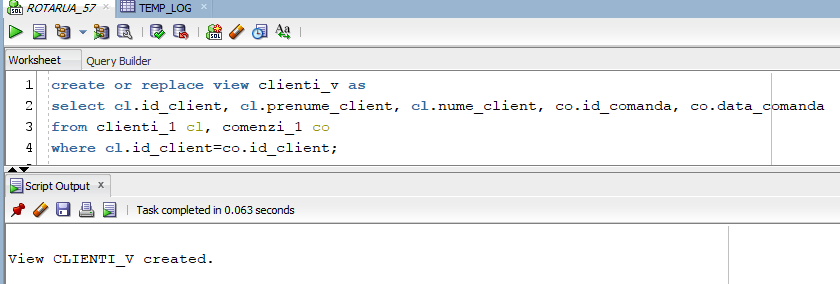
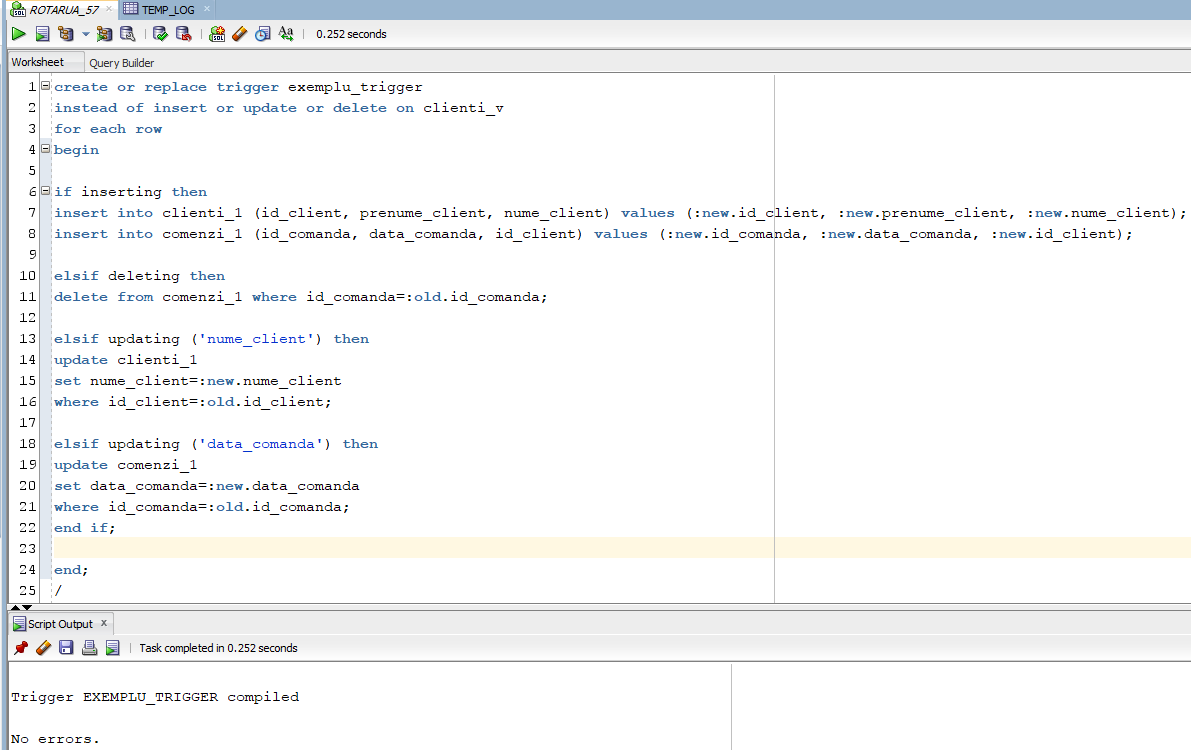
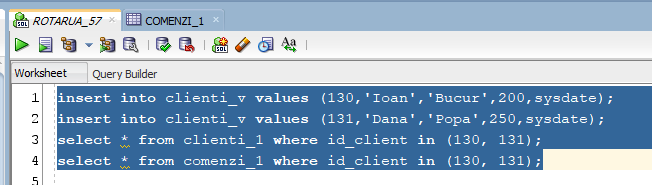
3. Se creează un trigger pentru a nu se permite depăşirea unei limite maxime a salariului unui angajat.



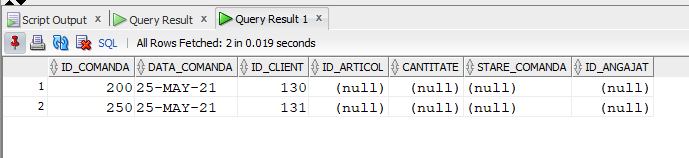
4. Se creează un trigger care asigură unicitatea codului produsului folosind valorile generate de o secvenţă

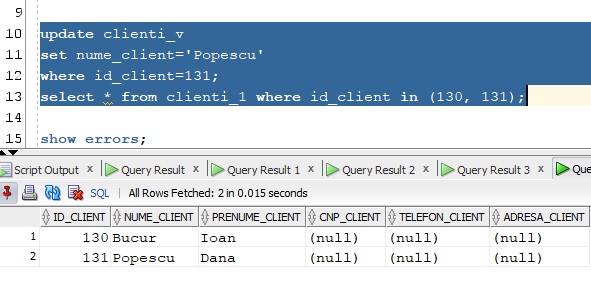


5. Clauza INSTEAD OF

* crearea tabelei virtuale
* crearea triggerului
* Testare trigger







# **ANEXE**

## **TABELE**

**--Sa se creeze tabela ANGAJATI**

CREATE TABLE ANGAJATI (

id\_angajat NUMBER(3) PRIMARY KEY not null,

nume\_angajat VARCHAR2(20) not null,

prenume\_angajat VARCHAR2(20) not null,

telefon\_angajat VARCHAR2(20) not null,

email\_angajat VARCHAR2(30) unique not null,

data\_angajare DATE,

salariu\_angajat NUMBER(8,2) not null

);

alter table ANGAJATI

add constraint email\_angajat check (email\_angajat like '\_%@\_%.\_%');

**--Sa se creeze tabela CLIENTI**

CREATE TABLE CLIENTI (

id\_client NUMBER(3) PRIMARY KEY not null,

nume\_client VARCHAR2(20) not null,

prenume\_client VARCHAR2(20) not null,

cnp\_client VARCHAR2(20) not null,

telefon\_client VARCHAR2(20) not null,

adresa\_client VARCHAR2(50) not null

);

**--Sa se creeze tabela FURNIZORI**

CREATE TABLE FURNIZORI (

id\_furnizor NUMBER(3) PRIMARY KEY not null,

nume\_furnizor VARCHAR2(20) not null,

telefon\_furnizor VARCHAR2(20) not null,

adresa\_furnizor VARCHAR2(30) not null

);

**--Sa se creeze tabela ARTICOLE**

CREATE TABLE ARTICOLE (

id\_articol NUMBER(3) PRIMARY KEY not null,

nume\_articol VARCHAR2(200) not null,

caracteristici\_articol VARCHAR2(500) not null,

stoc\_articol NUMBER(3) not null,

pret\_articol NUMBER(8,2),

id\_furnizor NUMBER(3) not null

);

alter table ARTICOLE

add constraint ARTICOLE\_ID\_FURNIZOR\_FK foreign key (id\_furnizor) references FURNIZORI (id\_furnizor);

alter table ARTICOLE

drop column stoc\_articol;

**--Sa se creeze tabela COMENZI**

CREATE TABLE COMENZI (

id\_comanda NUMBER(3) PRIMARY KEY not null,

data\_comanda DATE,

id\_client NUMBER(3), FOREIGN KEY (id\_client) REFERENCES CLIENTI (id\_client),

id\_articol NUMBER(3), FOREIGN KEY (id\_articol) REFERENCES ARTICOLE (id\_articol),

cantitate NUMBER(5) not null,

stare\_comanda VARCHAR2(20)

);

alter table COMENZI

add id\_angajat number(3);

**--Sa se creeze tabela INCASARI**

CREATE TABLE INCASARI (

id\_incasare NUMBER(4) PRIMARY KEY not null,

data\_incasare DATE,

id\_comanda NUMBER(3),

suma VARCHAR2(20),

datorie NUMBER(8,2)

);

alter table INCASARI

add constraint INCASARI\_ID\_COMANDA\_FK foreign key (id\_comanda) references COMENZI (id\_comanda);

## **OPERATII DE ACTUALIZARE**

**-- Introduceti un rand in tabela ANGAJATI**

insert into ANGAJATI (id\_angajat, nume\_angajat, prenume\_angajat, telefon\_angajat, email\_angajat, data\_angajare, salariu\_angajat)

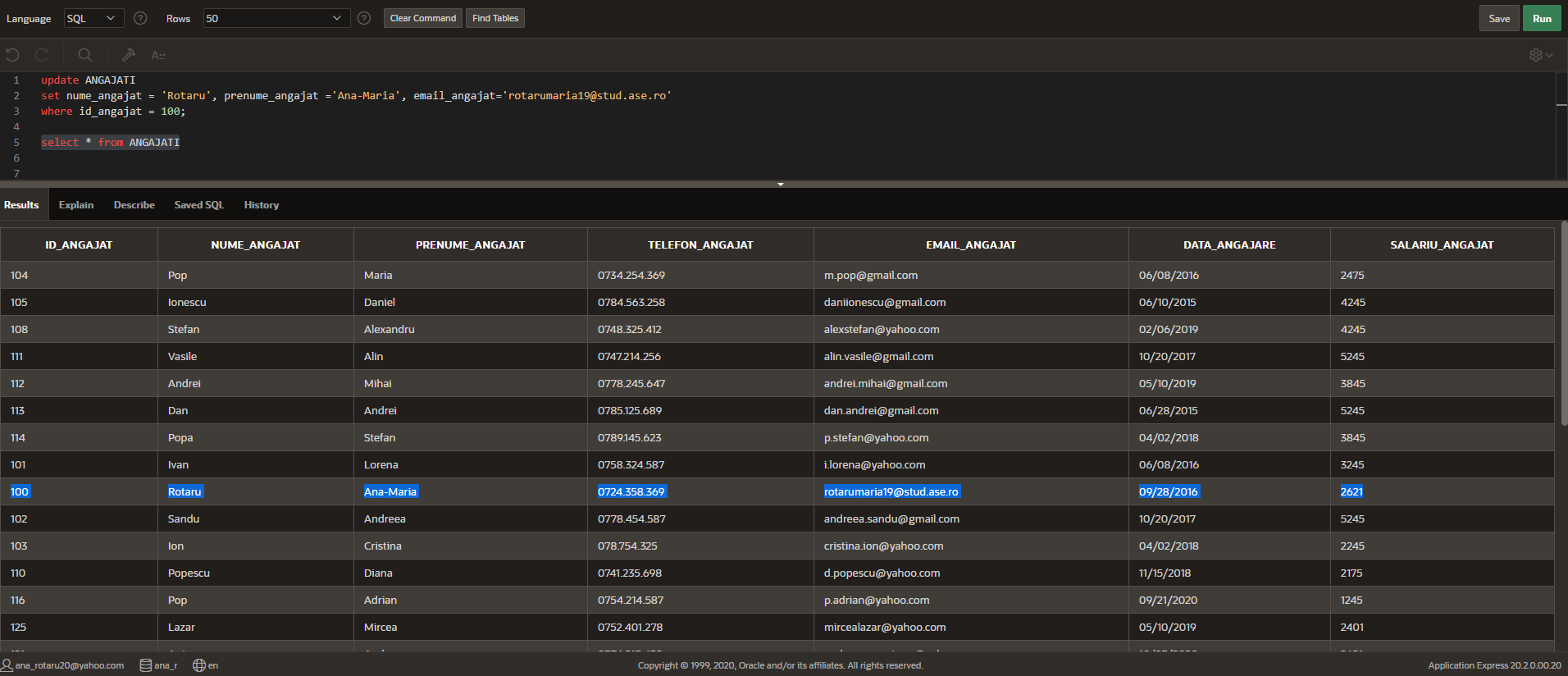
values (100, 'Popa', 'Oana', '0724.358.369', 'oana.popa@yahoo.com', to\_date('28-09-2016', 'dd-mm-yyyy'), 2621);

**-- Modificati numele, prenumele si emailul id-ului angajatului 100 cu datele dvs.**

update ANGAJATI

set nume\_angajat = 'Rotaru', prenume\_angajat ='Ana-Maria', email\_angajat='rotarumaria19@stud.ase.ro'

where id\_angajat = 100;



**-- Introduceti un rand in tabela FURNIZORI**

insert into FURNIZORI (id\_furnizor, nume\_furnizor, telefon\_furnizor, adresa\_furnizor)

values(10, 'AXM PROD', '0315.145.658', 'Strada A. C. Leonte, Bucuresti');

**-- Introduceti doua randuri in tabela CLIENTI**

insert into CLIENTI (id\_client, nume\_client, prenume\_client, cnp\_client, telefon\_client, adresa\_client)

values(10, 'Ionescu', 'Maria', '0251123450096', '0734.852.986, 'Str. Lalelelor, nr.7, Bucuresti');

insert into CLIENTI (id\_client, nume\_client, prenume\_client, cnp\_client, telefon\_client, adresa\_client)

values(22, 'Zavoranu', 'Ionel', '0421458795412', '0728.965.745', 'Strada Meritelor nr.1, Suceava')

**-- Introduceti doua randuri in tabela ARTICOLE**

insert into ARTICOLE (id\_articol, nume\_articol, caracteristici\_articol, pret\_articol, id\_furnizor)

values(10, 'Comodă CZP7', 'Inaltime: 800mm; Latime: 1650mm; Adancime: 400mm', 947, 25);

insert into ARTICOLE (id\_articol, nume\_articol, caracteristici\_articol, pret\_articol, id\_furnizor)

values(40, 'Masuta de machiaj YN1435', 'Inaltime: 1200mm; Latime: 518mm; Lungime: 1160mm; Material: Pal; Culoare: Nuca & Alb', 1090, 11);

**-- Introduceti doua randuri in tabela COMENZI**

insert into COMENZI (id\_comanda, data\_comanda, id\_client, id\_articol, cantitate, stare\_comanda, id\_angajat)

values(5, to\_date('11-02-2016', 'dd-mm-yyyy'), 15, 14, 4, 'livrat', 102);

insert into COMENZI (id\_comanda, data\_comanda, id\_client, id\_articol, cantitate, stare\_comanda, id\_angajat)

values(66, to\_date('03-08-2016', 'dd-mm-yyyy'), 26, 55, 3, 'anulat', 114);

**-- Introduceti un rand in tabela INCASARI**

insert into INCASARI\_1 (id\_incasare, data\_incasare, id\_comanda, suma, datorie)

values(64, to\_date('13-12-2018', 'dd-mm-yyyy'), 154, 'incasat', 0);

**-- Stergeti randul corespunzator al id\_angajat = 101 din tabela ANGAJATI**

delete from ANGAJATI

where id\_angajat = 101;

**-- Modificati in Albert prenumele angajatului cu id\_angajat egal 115**

update ANGAJATI

set prenume\_angajat = 'Albert'

where id\_angajat=115;

**-- Cresteti cu 20% salariile angajatilor care au fost angajati in anul 2015**

update ANGAJATI

set salariu\_angajat = salariu\_angajat \* 1.2

where data\_angajare < to\_date('01-01-2016', 'DD-MM-YYYY') and data\_angajare >= to\_date('01-01-2015', 'DD-MM-YYYY');

**-- Se sterge comanda facuta in data de 25-02-2016**

delete from COMENZI where data\_comanda = to\_date('25-02-2016', 'DD-MM-YYYY');

## **CERINTE**

**-- Sa se selecteze comenzile incheiate de salariatul cu id\_angajat = 124**

select \* from COMENZI

where id\_angajat = 124

order by id\_comanda;

**-- Sa se selecteze toate comezile care au fost livrate dupa 10 februarie 2016**

select \* from COMENZI

where lower(stare\_comanda) like '%livrat%'

and data\_comada > to\_date ('10-02-2000', 'DD-MM-YYYY');

**-- Sa se afiseze furnizorii de la care nu s-au luat articole**

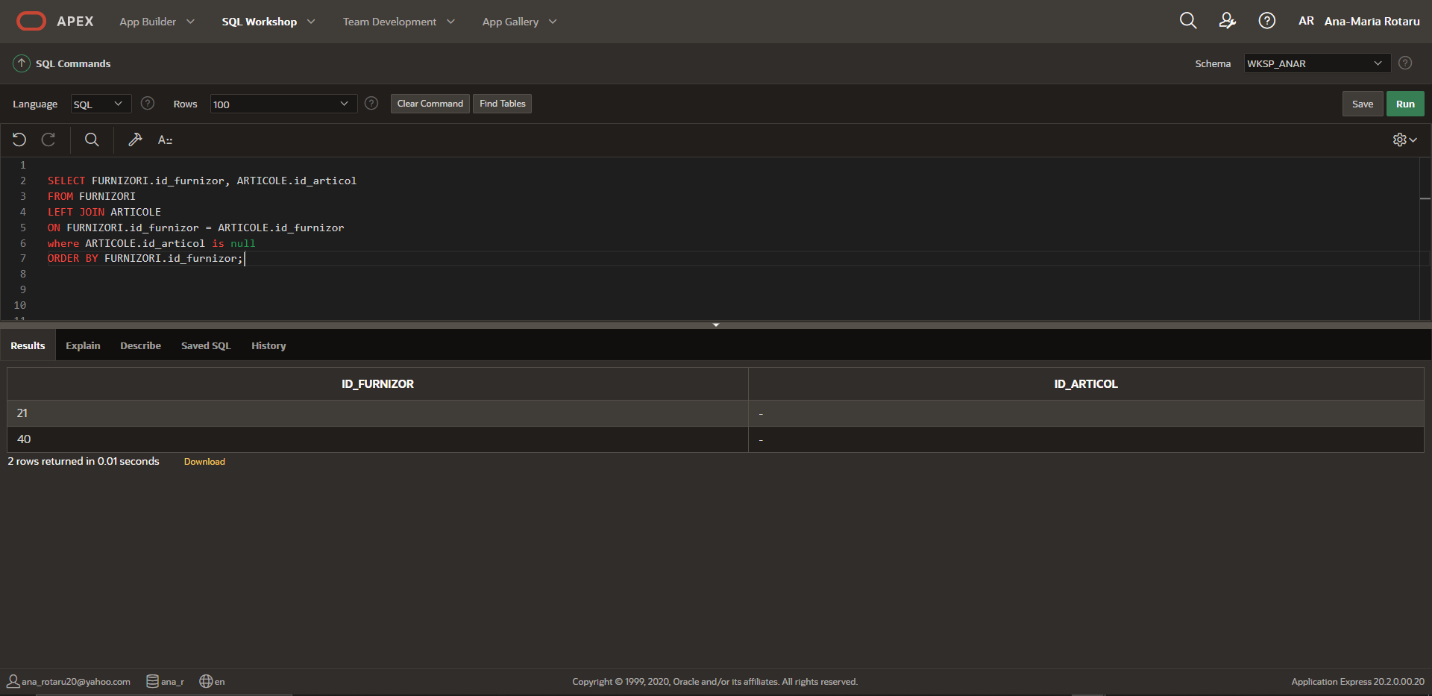
select FURNIZORI.id\_furnizor, ARTICOLE.id\_articol

from FURNIZORI

left join ARTICOLE

on FURNIZORI.id\_furnizor = ARTICOLE.id\_furnizor

where ARTICOLE.id\_articol is null

order by FURNIZORI.id\_furnizor;

**-- Sa se afiseze tipul de persoana, atat din tabelul clienti, cat si din tabelul furnizori**

select 'Client' as Type, nume\_client, adresa\_client

from CLIENTI

union

select 'Furnizor', nume\_furnizor, adresa\_furnizor

from FURNIZORI;

**-- Sa se afiseze numai comenzile care au valoarea cuprinsa intre 100 si 500 lei**

select COMENZI.id\_comanda, sum(COMENZI.cantitate \* ARTICOLE.pret\_articol)total\_comanda

from COMENZI, ARTICOLE

where COMENZI.id\_comanda = ARTICOLE.id\_articol

group by COMENZI.id\_comanda

having sum(COMENZI.cantitate \* ARTICOLE.pret\_articol) between 100 and 500

order by total\_comanda desc;

**-- Sa se selecteze numai articole cu valoarea cuprinsa intre 1000 si 2000 lei**

select ARTICOLE.nume\_articol, COMENZI.cantitate, ARTICOLE.pret\_articol,

COMENZI.cantitate \* ARTICOLE.pret\_articol as valoare

from ARTICOLE, COMENZI

where ARTICOLE.id\_articol = COMENZI.id\_articol

and COMENZI.cantitate \* ARTICOLE.pret\_articol between 1500 and 2500;

**-- Sa se afiseze perioada de timp corespunzatoare (in saptamani) intre data incheierii comenzii si data curenta**

select id\_comanda, (sysdate-data\_comanda)/7 saptamani

from COMENZI;

**-- Sa se afiseze comenzile incheiate intre 15 februarie si 20 aprilie 2016 --**

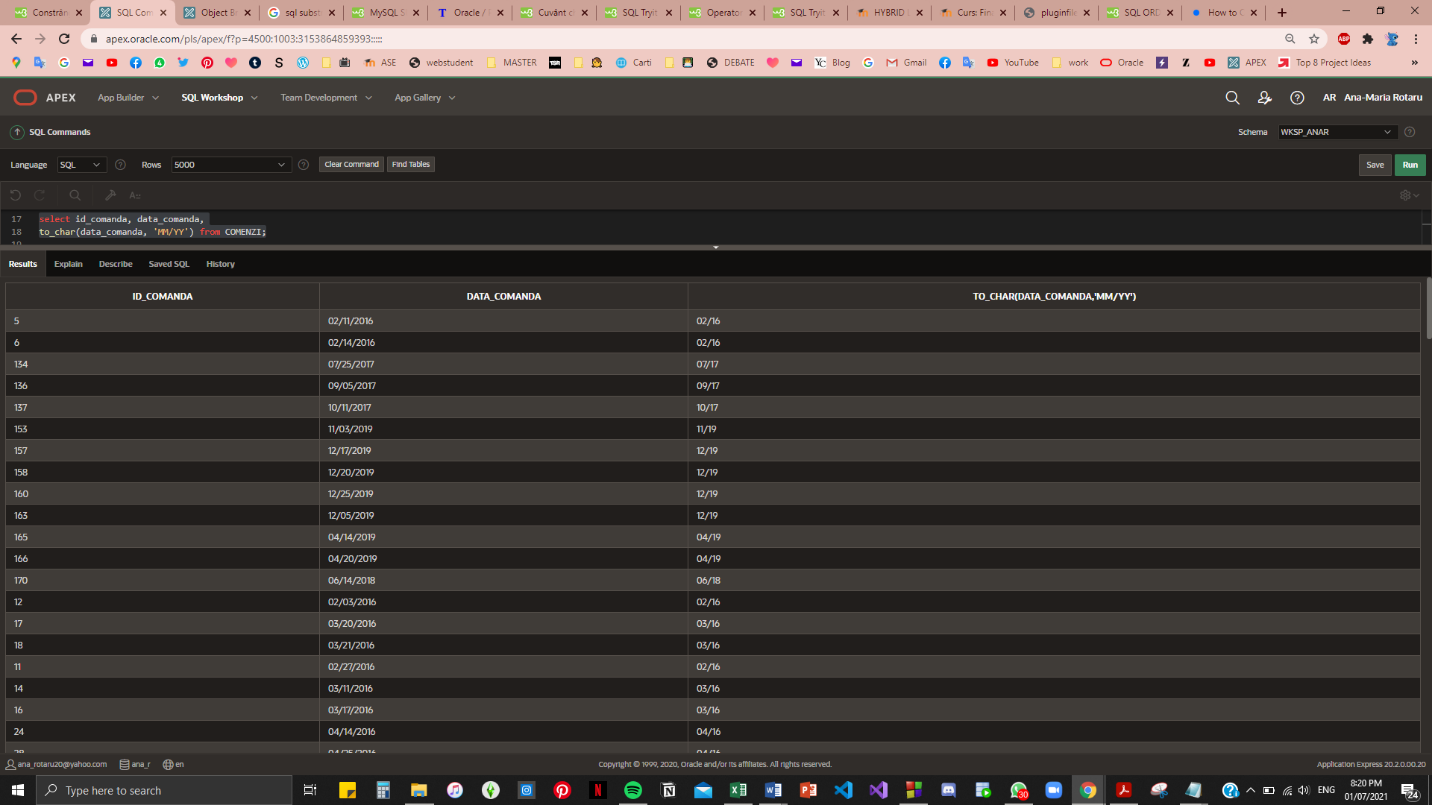
select id\_comanda, data\_comanda

from COMENZI

where data\_comanda between to\_date ( 'February 15, 2016', 'Month dd,YYYY') AND TO\_DATE( 'April 20, 2016', 'Month dd,YYYY');

**-- Sa se afiseze comenzile in formatul initial si in format „MM/YY”**

select id\_comanda, data\_comanda,

to\_char(data\_comanda, 'MM/YY') from COMENZI;

**-- Sa se afiseze comenzile incheiate in anii 2019 si 2020**

select id\_comanda, data\_comanda

from COMENZI

where extract(year from data\_comanda) in (2019,2020);

**-- Sa se afiseze comenzile incheiate in lunile iulie si august**

select id\_comanda, data\_comanda

from COMENZI

where extract(month from data\_comanda) in (7,8);

-- **Sa se afiseze angajatii care au salariul intre 1000 si 5000 fara cei care au salariul 1875 si 2145**

select \* from ANGAJATI where salariu\_angajat between 1000 and 5000

minus

select \* from angajati where salariu\_angajat in (1875,2145);

**-- Sa se puna conditiile urmatoare pentru cantitatea articolelor din comenzi:**

* **cantitatea este mai mare decat 4 „Peste 4 bucati”**
* **cantitatea este egala cu 4 „4 bucati”**
* **cantitatea este sub 4 „Sub 4 bucati”**

select id\_comanda, cantitate,

case

when cantitate > 4 then 'Peste 4 bucati'

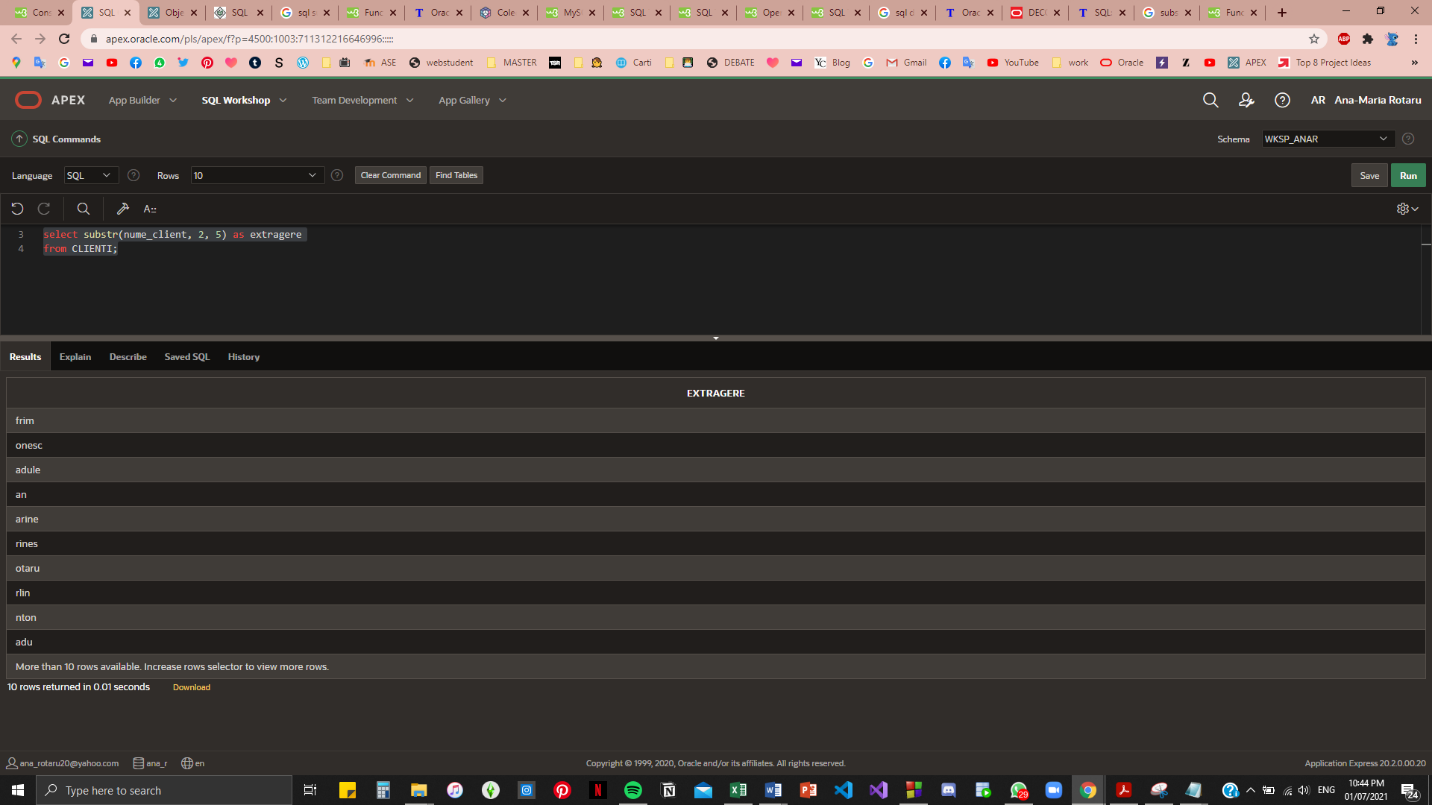
when cantitate = 4 THEN '4 bucati'

else 'Sub 4 bucati'

end as Caracteristici\_Cantitate

from COMENZI;

**-- Extrageți un șir din textul dintr-o coloană (începeți de la poziția 2, extrageți 5 caractere)**

select substr(nume\_client, 2, 5) as extragere from CLIENTI;

**-- Sa se afiseze valoarea totala a comenzilor pentru fiecare articol “dulap”**

select nume\_articol, sum(a.pret\_articol\*c.cantitate)

from ARTICOLE a join COMENZI c on a.id\_articol = c.id\_articol

WHERE lower(nume\_articol) LIKE '%dulap%'

group by nume\_articol;

**-- Sa se selecteze id\_client, cantitatea si data\_comanda, unde cantitatea este mai mare decat 1 si mai mica decat 5**

select COMENZI.id\_client, cantitate, data\_comanda

from comenzi

left join CLIENTI

on COMENZI.id\_client=CLIENTI.id\_client

where cantitate > 1

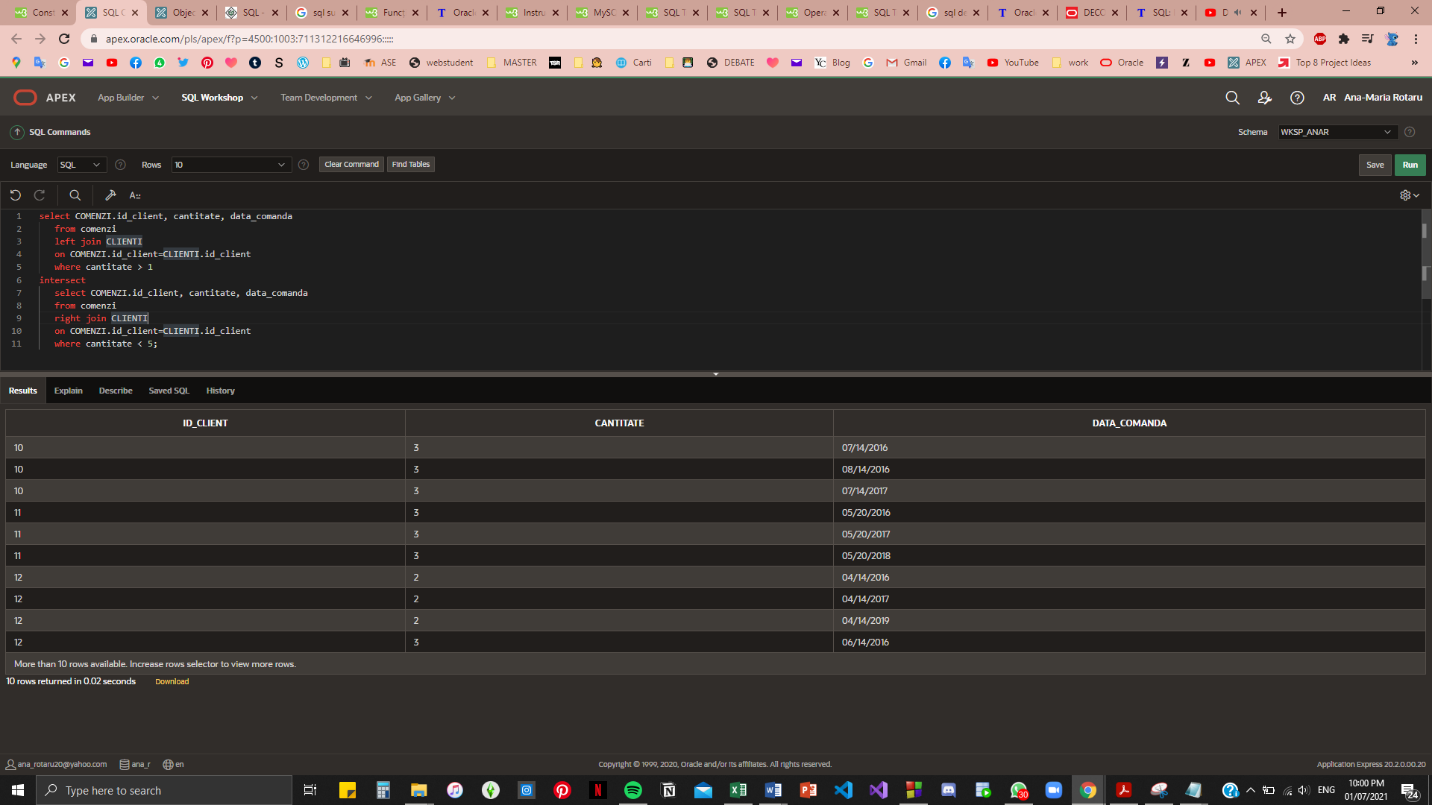
intersect

select COMENZI.id\_client, cantitate, data\_comanda

from comenzi

right join CLIENTI

on COMENZI.id\_client=CLIENTI.id\_client

 where cantitate < 5;

**-- Sa se creeze o coloana cu numele nivel\_angajat in tabela angajati si intoduceti nivelul acestora (incepator (2019-2020), mediu(2017-2018), avansat(2015-2016)**

alter table ANGAJATI

add nivel\_angajat VARCHAR2(20) default ('incepator');

update ANGAJATI

set nivel\_angajat = 'incepator'

where data\_angajare <= to\_date('30-10-2020', 'DD-MM-YYYY') and data\_angajare >= to\_date('10-01-2019', 'DD-MM-YYYY');

update ANGAJATI

set nivel\_angajat = 'mediu'

where data\_angajare <= to\_date('30-10-2018', 'DD-MM-YYYY') and data\_angajare >= to\_date('10-01-2017', 'DD-MM-YYYY');

update ANGAJATI

set nivel\_angajat = 'avansat'

where data\_angajare <= to\_date('30-10-2016', 'DD-MM-YYYY') and data\_angajare >= to\_date('10-01-2015', 'DD-MM-YYYY');

**-- Sa se adauge la salariile angajatiilor profit dupa urmatoarele criterii:**

* **0.1% din valoarea comenzilor daca nivelul angajatului este incepator**
* **0.3% din valoarea comenzilor daca nivelul angajatului este mediu**
* **0.5% din valoarea comenzilor daca nivelul angajatului este avansat**

select angajati.salariu\_angajat,

decode(nivel\_angajat, 'incepator', salariu\_angajat+((COMENZI.cantitate\*ARTICOLE.pret\_articol)\*0.1),

'mediu', salariu\_angajat+((COMENZI.cantitate\*ARTICOLE.pret\_articol)\*0.3),

'avansat', salariu\_angajat+((COMENZI.cantitate\*ARTICOLE.pret\_articol)\*0.5)

) salariu\_profit

from ANGAJATI, COMENZI, ARTICOLE;

**-- Sa se creeze un index pe tabela clienti, coloana prenume, apoi stergeti indexul creat**

create index idx\_nume

ON CLIENTI(nume\_client, prenume\_client);

select \* from user\_indexes;

drop index idx\_nume;

**-- Sa se afiseze angajatii care au vandut si comenzile care nu au avut angajat in anul 2016**

select COMENZI.id\_comanda, nvl(ANGAJATI.id\_angajat, -1)

from COMENZI

left join ANGAJATI

on ANGAJATI.id\_angajat = COMENZI.id\_angajat

where extract(year from COMENZI.data\_comanda) = 2016

order by COMENZI.data\_comanda;

**-- Sa se afiseze articolele cu pretul cel mai mic sau pretul cel mai mare**

select nume\_articol, pret\_articol

from ARTICOLE

where articole.pret\_articol=(select min (pret\_articol) as pretul\_cel\_mai\_mic from articole) or articole.pret\_articol=(select max (pret\_articol) as pretul\_cel\_mai\_mare from articole);

**-- Sa se creeze un sinonim pentru comenzi, iar apoi sa se stearga**

create synonym COM for COMENZI;

select \* from COM;

drop synonym COM;

**-- Sa realizeze o tabela virtuala cu toti angajatii cu un nivel mediu, actualiati salariul cu 100 lei, apoi stergeti tabela.**

create or replace view v\_angajati

as select \* from angajati

where nivel\_angajat = 'mediu';

select \* from v\_angajati;

update v\_angajati

set salariu\_angajat = salariu\_angajat + 100;

drop view v\_angajati;

**-- Sa se afiseze cu majuscule numele clientilor**

select upper(nume\_client) as nume\_client from clienti

**-- Sa se afiseze articolele cu cel mai mic pret de la fiecare furnizor**

select nume\_articol, pret\_articol, id\_furnizor

from ARTICOLE a

where a.pret\_articol=(select min (pret\_articol) as pretul\_cel\_mai\_mic from articole ab where a.id\_furnizor = a.Id\_furnizor);