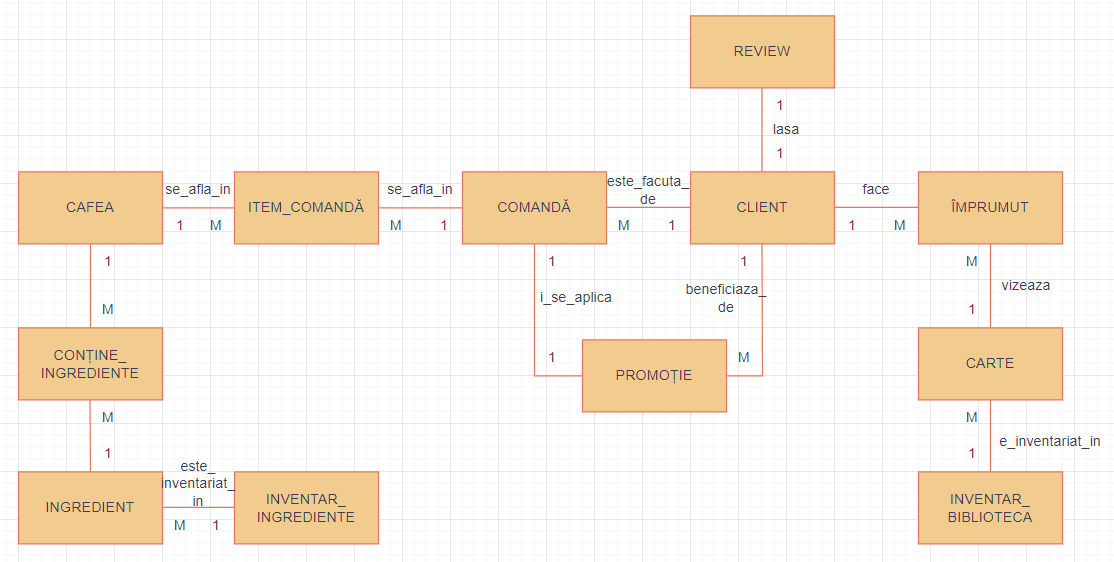
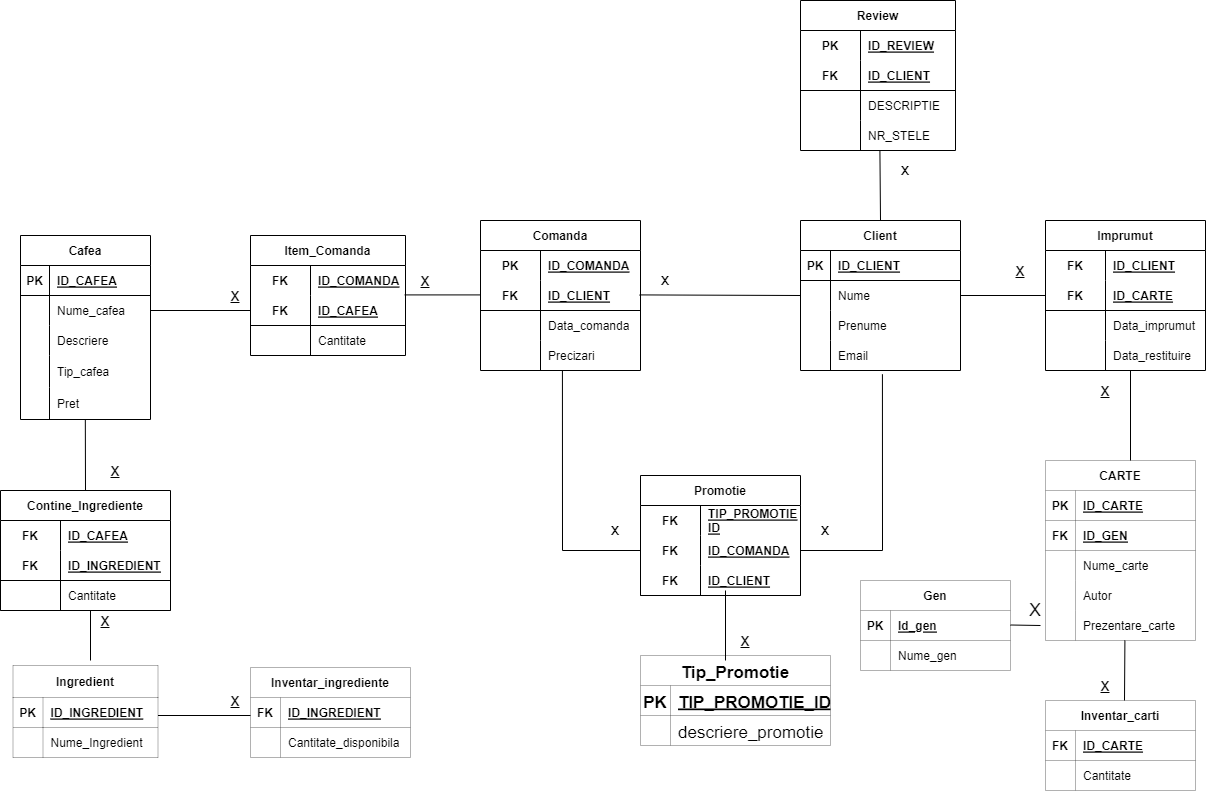
Coffee shop

1. Baza de date pe care am conceput-o mediaza serviciile prestate de un local tip cafenea in care clientii pot comanda produse derivate din cafea si imprumuta carti in incinta cafenelei, in limita disponibilitatii inventarului de ingrediente si bibliotecii de lectura. Un client poate lasa un review pentru eventualul feedback pe care l-a avut, si poate beneficia de eventuale promotii, dar care nu se vor aplica in numar mai mare de una per comanda.
2. Diagrama ERD:



3. Diagrama conceptuala:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

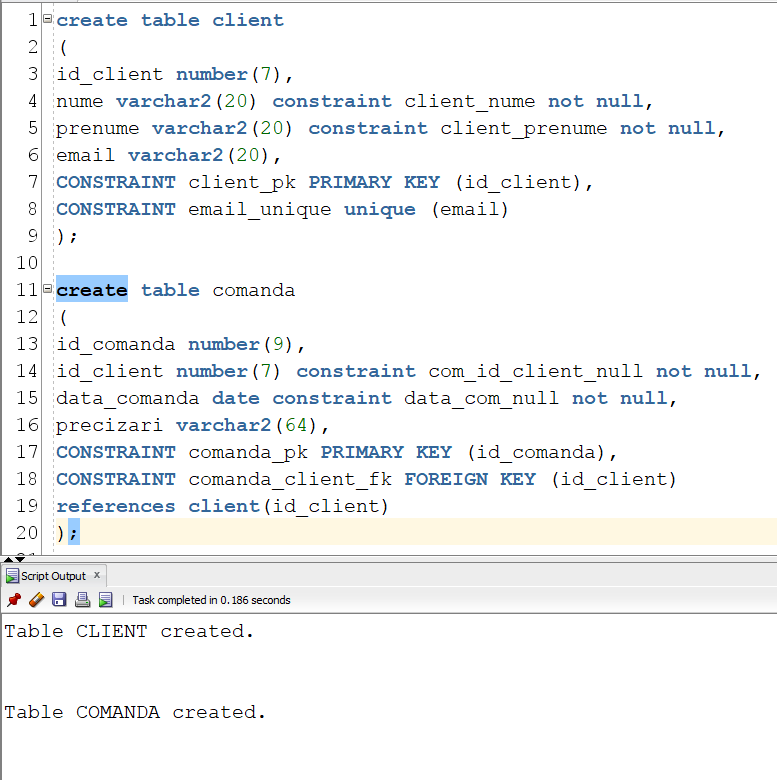


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

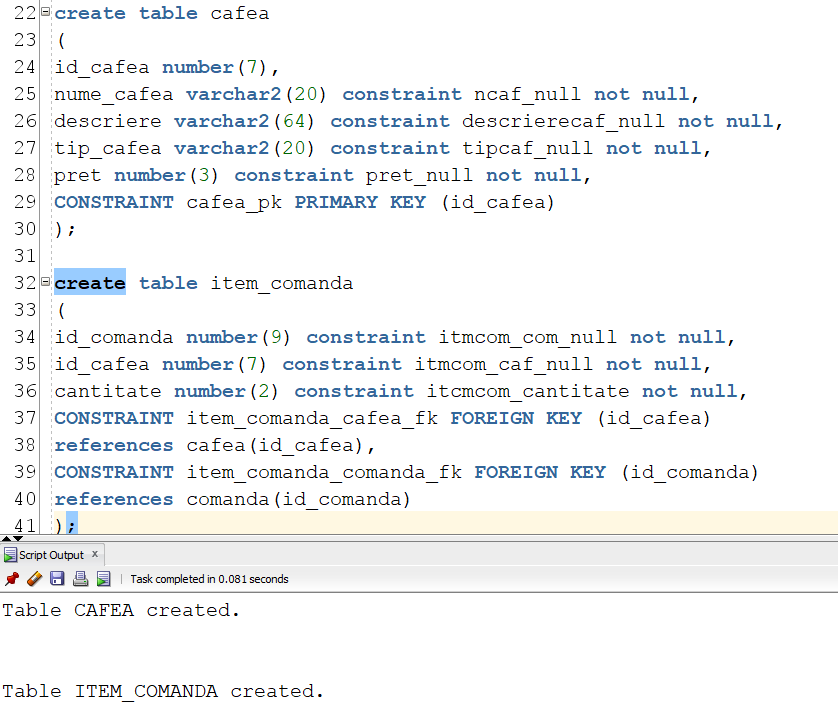
4.

Crearea tabelelor:

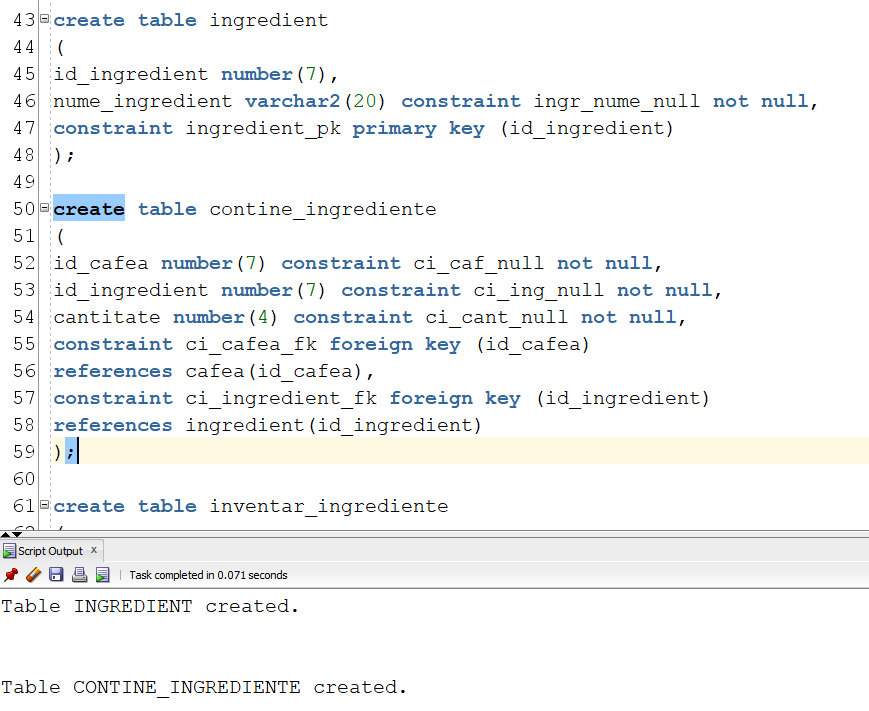
**create** **table** **client**  
(  
id\_client number(7),  
nume varchar2(20) **constraint** client\_nume **not** null,  
prenume varchar2(20) **constraint** client\_prenume **not** null,  
email varchar2(20),  
**CONSTRAINT** client\_pk PRIMARY **KEY** (id\_client),  
**CONSTRAINT** email\_unique **unique** (email)  
);  
   
**create** **table** comanda  
(  
id\_comanda number(9),  
id\_client number(7) **constraint** com\_id\_client\_null **not** null,  
data\_comanda date **constraint** data\_com\_null **not** null,  
precizari varchar2(64),  
**CONSTRAINT** comanda\_pk PRIMARY **KEY** (id\_comanda),  
**CONSTRAINT** comanda\_client\_fk **FOREIGN** **KEY** (id\_client)  
**references** **client**(id\_client)  
);



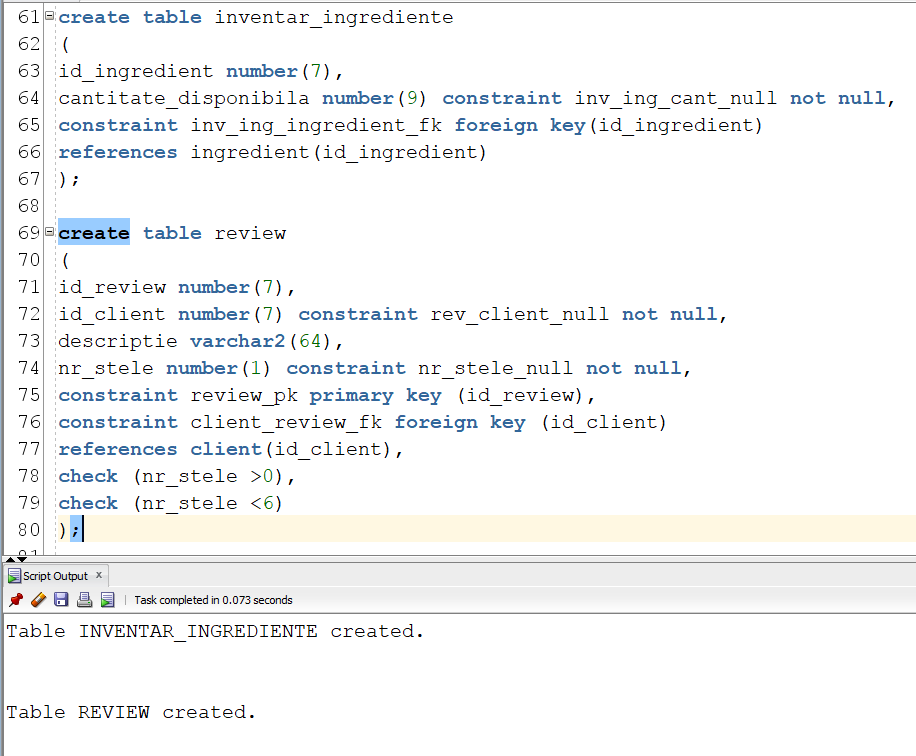
**create** **table** cafea  
(  
id\_cafea number(7),  
nume\_cafea varchar2(20) **constraint** ncaf\_null **not** null,  
descriere varchar2(64) **constraint** descrierecaf\_null **not** null,  
tip\_cafea varchar2(20) **constraint** tipcaf\_null **not** null,  
pret number(3) **constraint** pret\_null **not** null,  
**CONSTRAINT** cafea\_pk PRIMARY **KEY** (id\_cafea)  
);  
   
**create** **table** item\_comanda  
(  
id\_comanda number(9) **constraint** itmcom\_com\_null **not** null,  
id\_cafea number(7) **constraint** itmcom\_caf\_null **not** null,  
cantitate number(2) **constraint** itcmcom\_cantitate **not** null,  
**CONSTRAINT** item\_comanda\_cafea\_fk **FOREIGN** **KEY** (id\_cafea)  
**references** cafea(id\_cafea),  
**CONSTRAINT** item\_comanda\_comanda\_fk **FOREIGN** **KEY** (id\_comanda)  
**references** comanda(id\_comanda)  
);



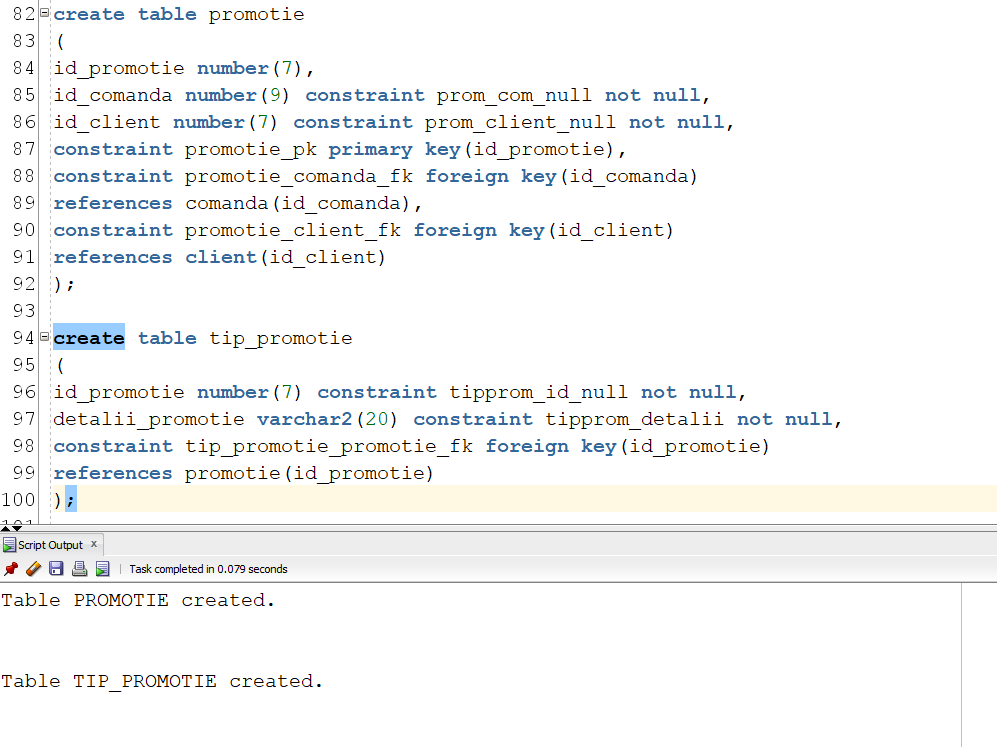
**create** **table** ingredient  
(  
id\_ingredient number(7),  
nume\_ingredient varchar2(20) **constraint** ingr\_nume\_null **not** null,  
**constraint** ingredient\_pk primary **key** (id\_ingredient)  
);  
   
**create** **table** contine\_ingrediente  
(  
id\_cafea number(7) **constraint** ci\_caf\_null **not** null,  
id\_ingredient number(7) **constraint** ci\_ing\_null **not** null,  
cantitate number(4) **constraint** ci\_cant\_null **not** null,  
**constraint** ci\_cafea\_fk **foreign** **key** (id\_cafea)  
**references** cafea(id\_cafea),  
**constraint** ci\_ingredient\_fk **foreign** **key** (id\_ingredient)  
**references** ingredient(id\_ingredient)  
);



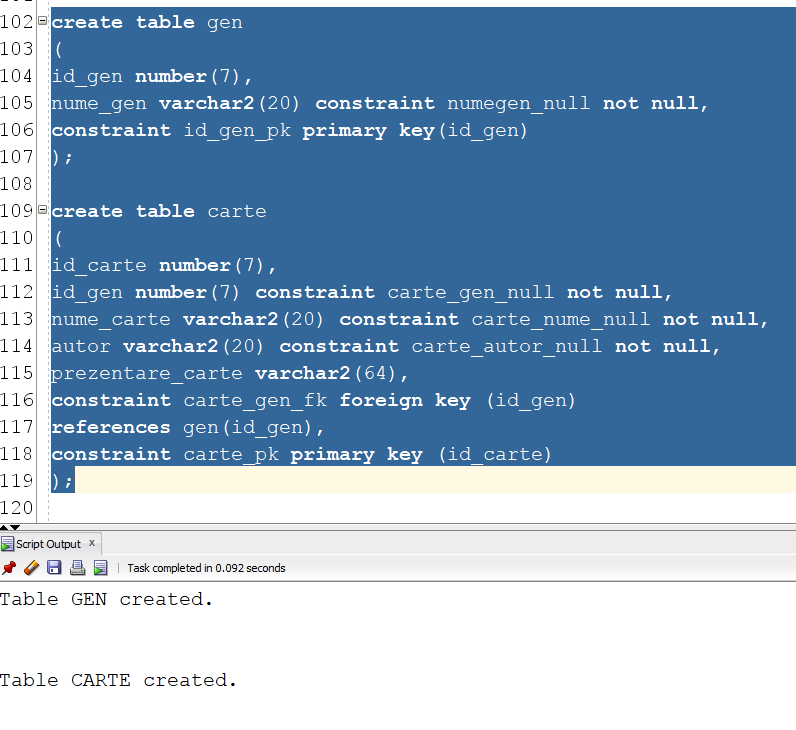
**create** **table** inventar\_ingrediente  
(  
id\_ingredient number(7),  
cantitate\_disponibila number(9) **constraint** inv\_ing\_cant\_null **not** null,   
**constraint** inv\_ing\_ingredient\_fk **foreign** **key**(id\_ingredient)  
**references** ingredient(id\_ingredient)  
);  
   
**create** **table** review  
(  
id\_review number(7),  
id\_client number(7) **constraint** rev\_client\_null **not** null,  
descriptie varchar2(64),  
nr\_stele number(1) **constraint** nr\_stele\_null **not** null,  
**constraint** review\_pk primary **key** (id\_review),  
**constraint** client\_review\_fk **foreign** **key** (id\_client)   
**references** **client**(id\_client),  
**check** (nr\_stele >0),  
**check** (nr\_stele <6)  
);



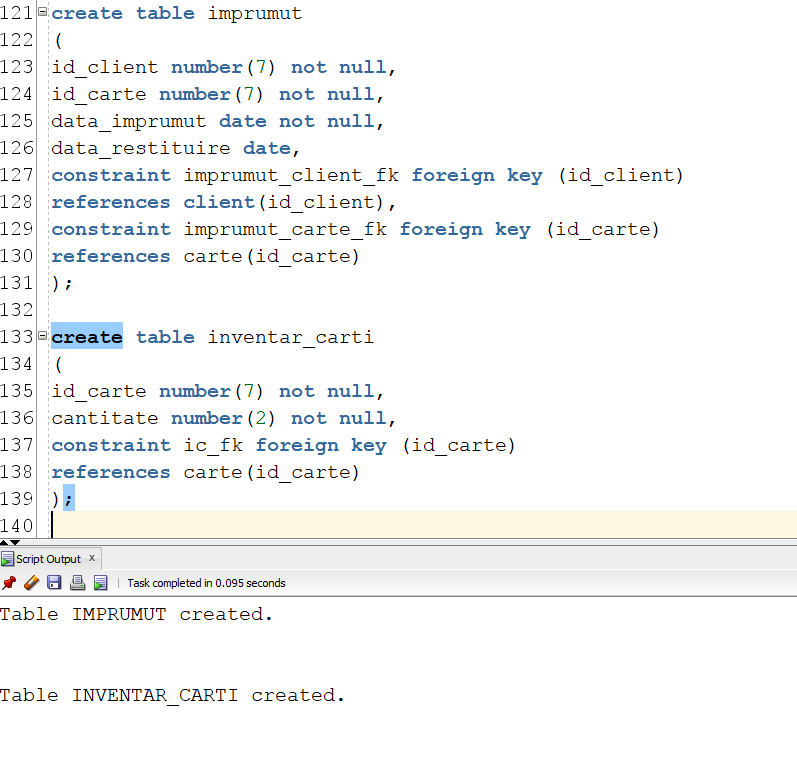
**create** **table** promotie  
(  
id\_promotie number(7),  
id\_comanda number(9) **constraint** prom\_com\_null **not** null,  
id\_client number(7) **constraint** prom\_client\_null **not** null,  
**constraint** promotie\_pk primary **key**(id\_promotie),  
**constraint** promotie\_comanda\_fk **foreign** **key**(id\_comanda)  
**references** comanda(id\_comanda),  
**constraint** promotie\_client\_fk **foreign** **key**(id\_client)  
**references** **client**(id\_client)  
);  
   
**create** **table** tip\_promotie  
(  
id\_promotie number(7) **constraint** tipprom\_id\_null **not** null,  
detalii\_promotie varchar2(20) **constraint** tipprom\_detalii **not** null,  
**constraint** tip\_promotie\_promotie\_fk **foreign** **key**(id\_promotie)  
**references** promotie(id\_promotie)  
);



**create** **table** gen  
(  
id\_gen number(7),  
nume\_gen varchar2(20) **constraint** numegen\_null **not** null,  
**constraint** id\_gen\_pk primary **key**(id\_gen)  
);  
   
**create** **table** carte  
(  
id\_carte number(7),  
id\_gen number(7) **constraint** carte\_gen\_null **not** null,  
nume\_carte varchar2(20) **constraint** carte\_nume\_null **not** null,  
autor varchar2(20) **constraint** carte\_autor\_null **not** null,  
prezentare\_carte varchar2(64),  
**constraint** carte\_gen\_fk **foreign** **key** (id\_gen)   
**references** gen(id\_gen),  
**constraint** carte\_pk primary **key** (id\_carte)   
);



**create** **table** imprumut  
(  
id\_client number(7) **not** null,  
id\_carte number(7) **not** null,  
data\_imprumut date **not** null,  
data\_restituire date,  
**constraint** imprumut\_client\_fk **foreign** **key** (id\_client)   
**references** **client**(id\_client),  
**constraint** imprumut\_carte\_fk **foreign** **key** (id\_carte)   
**references** carte(id\_carte)  
);  
   
**create** **table** inventar\_carti  
(  
id\_carte number(7) **not** null,  
cantitate number(2) **not** null,  
**constraint** ic\_fk **foreign** **key** (id\_carte)  
**references** carte(id\_carte)  
);



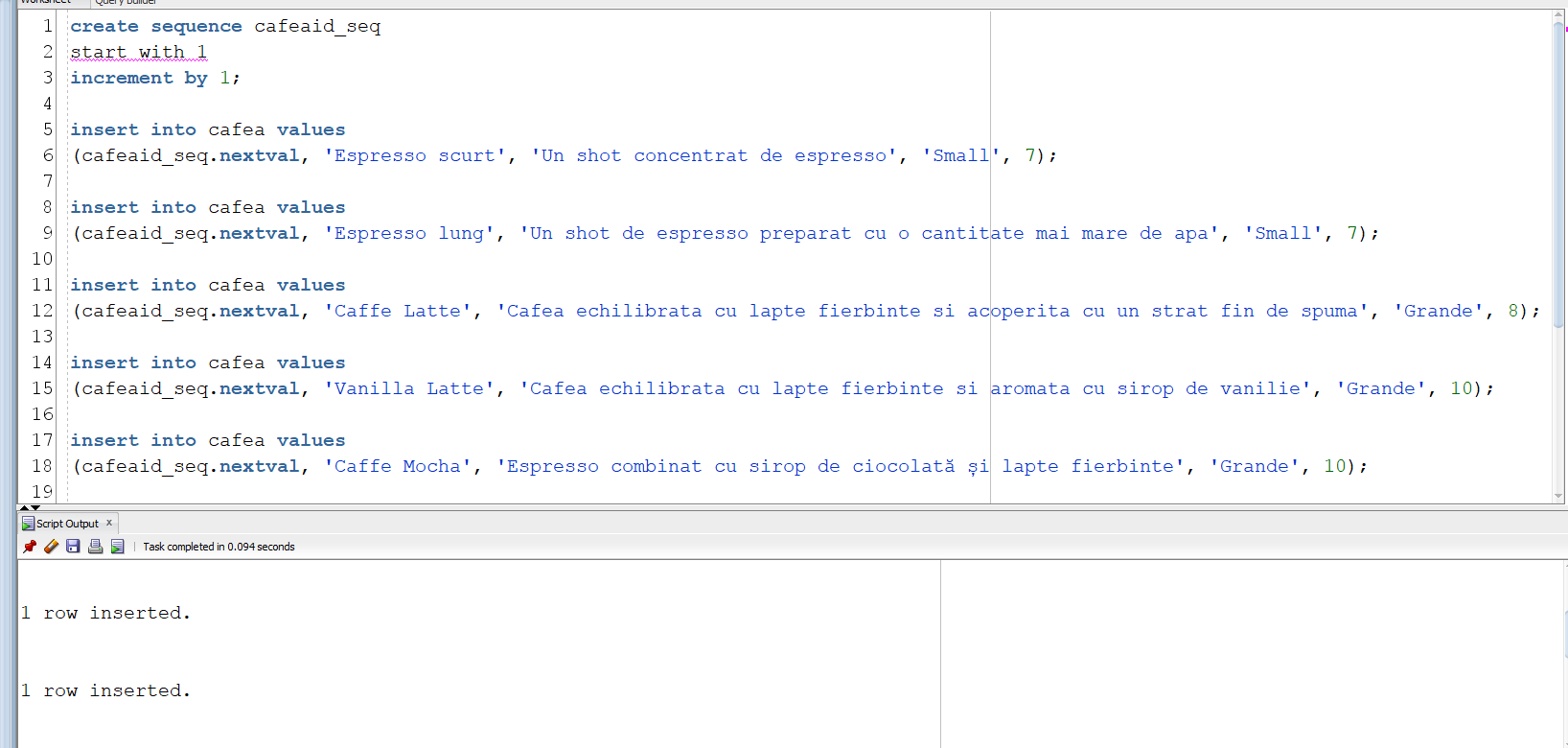
Modificari (pentru inserari):

**alter** **table** cafea  
**modify** nume\_cafea varchar2(40);  
   
**alter** **table** cafea  
**modify** descriere varchar2(80);

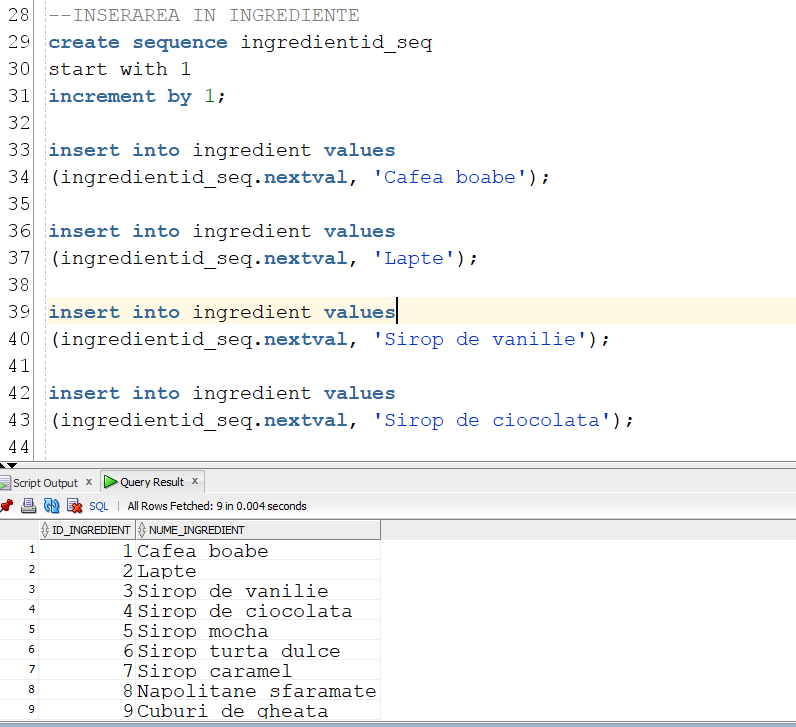
**alter** **table** tip\_promotie  
**modify** detalii\_promotie varchar2(80);  
   
**alter** **table** tip\_promotie  
**add** tip\_promotie\_id number(3) **not** null;  
   
**alter** **table** tip\_promotie  
**drop** **column** id\_promotie;  
   
**alter** **table** tip\_promotie  
**add** tip\_promotie\_id number(3) **not** null;  
   
**alter** **table** tip\_promotie  
**add** primary **key** (tip\_promotie\_id);  
   
**alter** **table** promotie  
**drop** **column** id\_promotie;  
   
**alter** **table** promotie  
**add** tip\_promotie\_id number(3) **not** null;  
   
**alter** **table** promotie  
**add** **foreign** **key** (tip\_promotie\_id) **references** tip\_promotie(tip\_promotie\_id);

5.

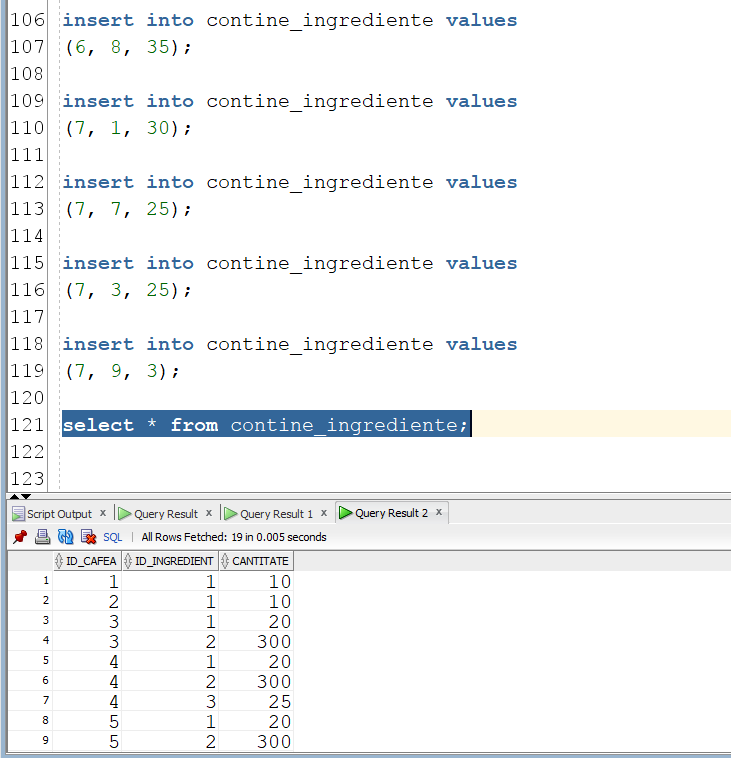
**create** **sequence** cafeaid\_seq   
**start** **with** 1  
**increment** **by** 1;  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Espresso scurt', 'Un shot concentrat de espresso', 'Small', 7);  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Espresso lung', 'Un shot de espresso preparat cu o cantitate mai mare de apa', 'Small', 7);  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Caffe Latte', 'Cafea echilibrata cu lapte fierbinte si acoperita cu un strat fin de spuma', 'Grande', 8);  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Vanilla Latte', 'Cafea echilibrata cu lapte fierbinte si aromata cu sirop de vanilie', 'Grande', 10);  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Caffe Mocha', 'Espresso combinat cu sirop de ciocolată și lapte fierbinte', 'Grande', 10);  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Gingerbread Latte',   
'Siropul de turtă dulce și siropul mocha sunt amestecate cu espresso si lapte, topping de napolitane', 'Venti', 20);  
   
**insert** **into** cafea **values**  
(cafeaid\_seq.nextval, 'Iced Caramel Macchiato', 'Espresso cu sirop de caramel, vanilie si cuburi de gheata', 'Venti', 18);



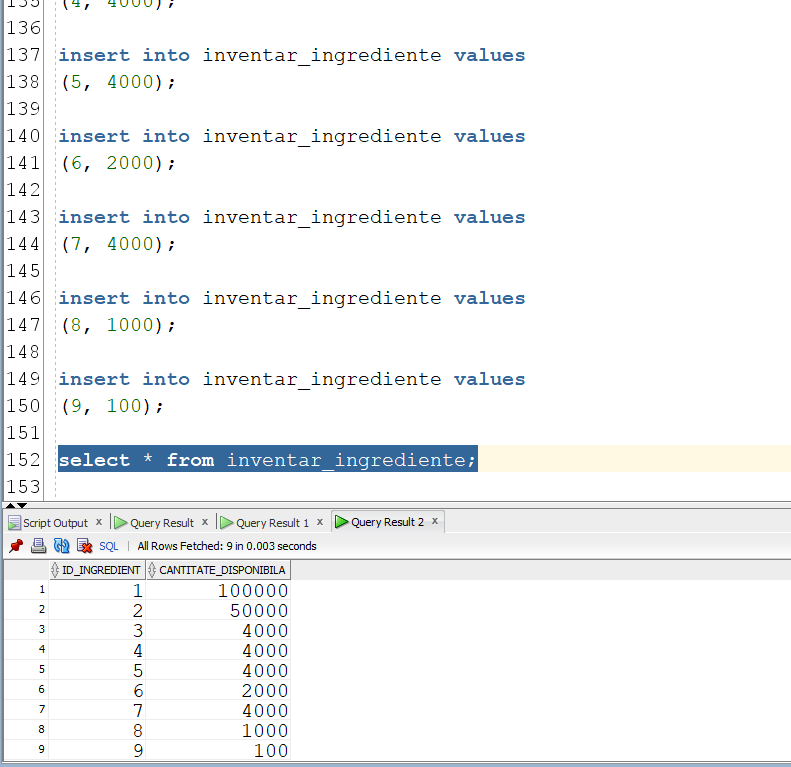
**create** **sequence** ingredientid\_seq   
**start** **with** 1  
**increment** **by** 1;  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Cafea boabe');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Lapte');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Sirop de vanilie');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Sirop de ciocolata');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Sirop mocha');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Sirop turta dulce');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Sirop caramel');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Napolitane sfaramate');  
   
**insert** **into** ingredient **values**  
(ingredientid\_seq.nextval, 'Cuburi de gheata');



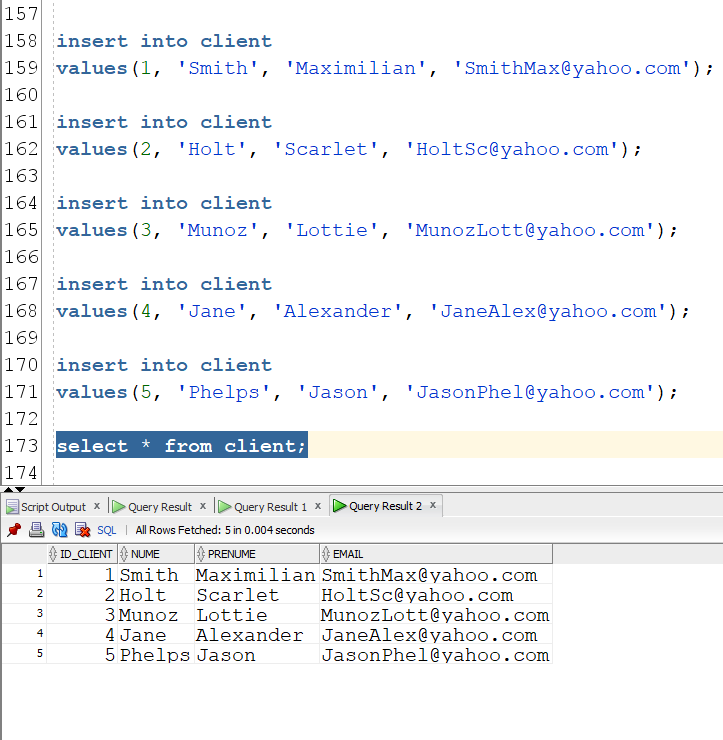
insert into contine\_ingrediente values  
(1, 1, 10);  
   
insert into contine\_ingrediente values  
(2, 1, 10);  
   
insert into contine\_ingrediente values  
(3, 1, 20);  
   
insert into contine\_ingrediente values  
(3, 2, 300);  
   
insert into contine\_ingrediente values  
(4, 1, 20);  
   
insert into contine\_ingrediente values  
(4, 2, 300);  
   
insert into contine\_ingrediente values  
(4, 3, 25);  
   
insert into contine\_ingrediente values  
(5, 1, 20);  
   
insert into contine\_ingrediente values  
(5, 2, 300);  
   
insert into contine\_ingrediente values  
(5, 4, 25);  
   
insert into contine\_ingrediente values  
(6, 1, 30);  
   
insert into contine\_ingrediente values  
(6, 2, 300);  
   
insert into contine\_ingrediente values  
(6, 6, 25);  
   
insert into contine\_ingrediente values  
(6, 5, 25);  
   
insert into contine\_ingrediente values  
(6, 8, 35);  
   
insert into contine\_ingrediente values  
(7, 1, 30);  
   
insert into contine\_ingrediente values  
(7, 7, 25);  
   
insert into contine\_ingrediente values  
(7, 3, 25);  
   
insert into contine\_ingrediente values  
(7, 9, 3);



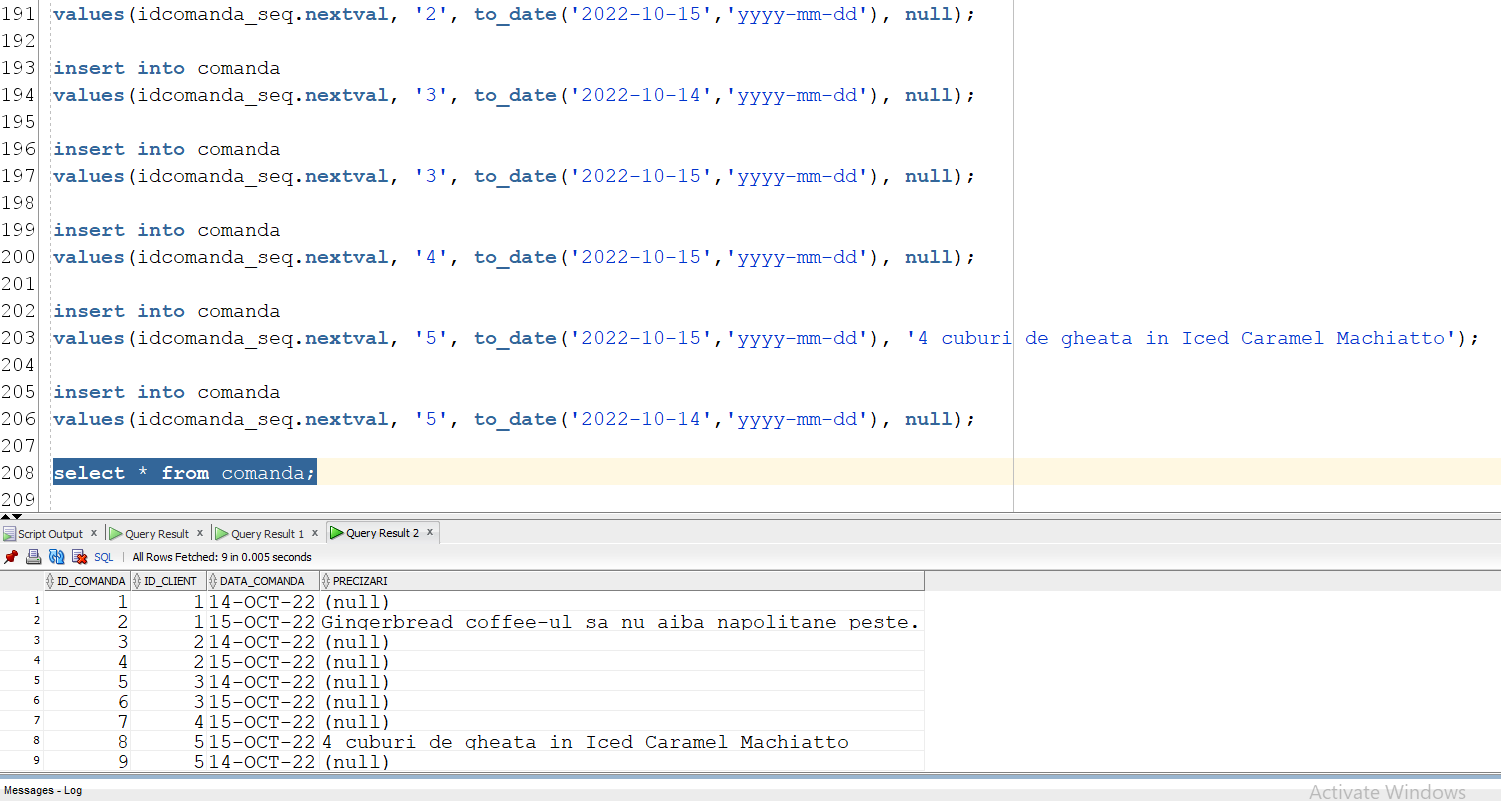
**insert** **into** inventar\_ingrediente **values**  
(1, 100000);  
   
**insert** **into** inventar\_ingrediente **values**  
(2, 50000);  
   
**insert** **into** inventar\_ingrediente **values**  
(3, 4000);  
   
**insert** **into** inventar\_ingrediente **values**  
(4, 4000);  
   
**insert** **into** inventar\_ingrediente **values**  
(5, 4000);  
   
**insert** **into** inventar\_ingrediente **values**  
(6, 2000);  
   
**insert** **into** inventar\_ingrediente **values**  
(7, 4000);  
   
**insert** **into** inventar\_ingrediente **values**  
(8, 1000);  
   
**insert** **into** inventar\_ingrediente **values**  
(9, 100);



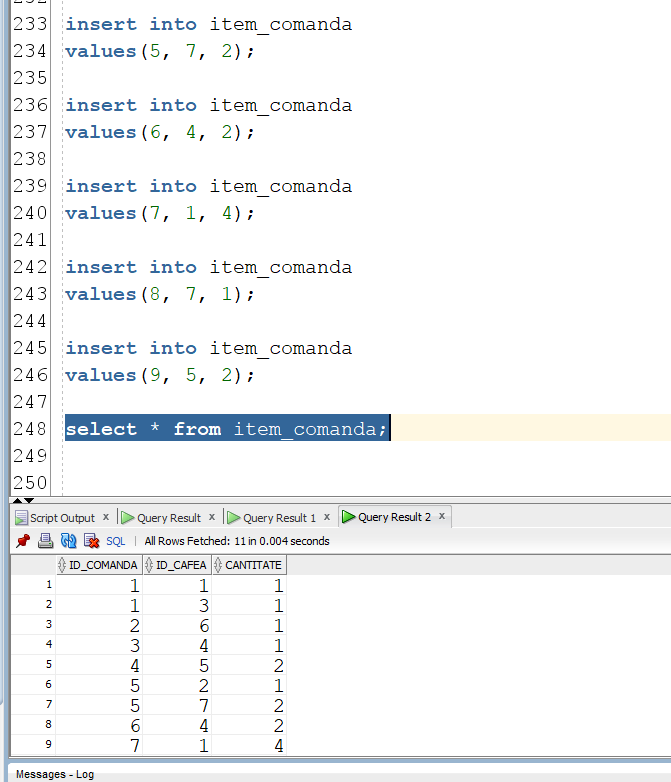
**insert** **into** **client**  
**values (**1, 'Smith', 'Maximilian', 'SmithMax@yahoo.com');  
   
**insert** **into** **client**  
**values (**2, 'Holt', 'Scarlet', 'HoltSc@yahoo.com');  
   
**insert** **into** **client**  
**values (**3, 'Munoz', 'Lottie', 'MunozLott@yahoo.com');  
   
**insert** **into** **client**  
**values (**4, 'Jane', 'Alexander', 'JaneAlex@yahoo.com');  
   
**insert** **into** **client**  
**values (**5, 'Phelps', 'Jason', '[JasonPhel@yahoo.com](mailto:JasonPhel@yahoo.com)');



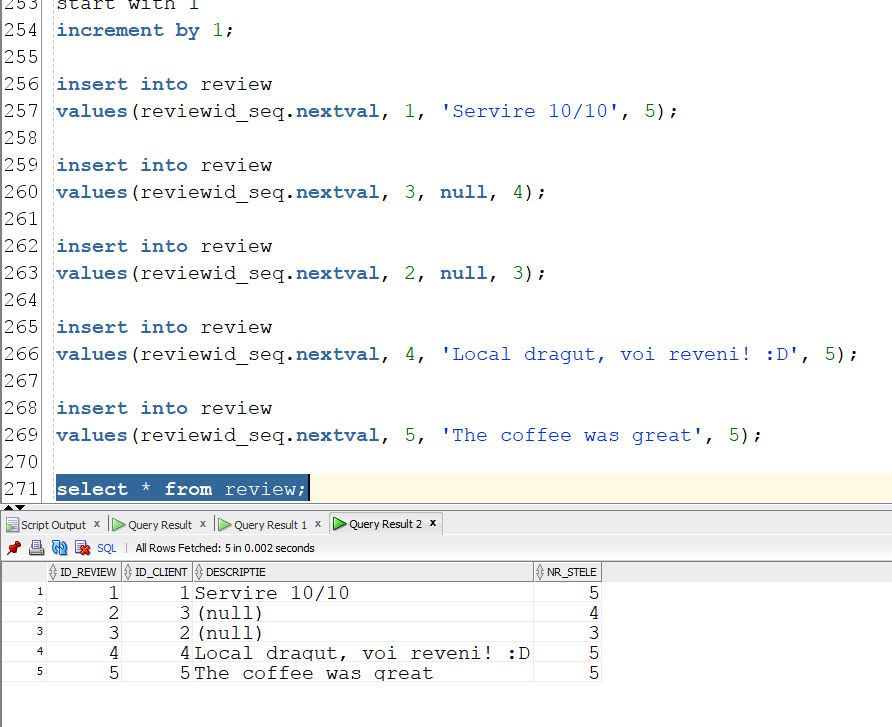
**create** **sequence** idcomanda\_seq    
**start** **with** 1  
**increment** **by** 1;  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '1', **to\_date**('2022-10-14', 'yyyy-mm-dd'), null);  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '1', **to\_date**('2022-10-15', 'yyyy-mm-dd'), 'Gingerbread coffee-ul sa nu aiba napolitane peste.');  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '2', **to\_date**('2022-10-14', 'yyyy-mm-dd'), null);  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '2', **to\_date**('2022-10-15','yyyy-mm-dd'), null);  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '3', **to\_date**('2022-10-14','yyyy-mm-dd'), null);  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '3', **to\_date**('2022-10-15','yyyy-mm-dd'), null);  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '4', **to\_date**('2022-10-15','yyyy-mm-dd'), null);  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '5', **to\_date**('2022-10-15','yyyy-mm-dd'), '4 cuburi de gheata in Iced Caramel Machiatto');  
   
**insert** **into** comanda  
**values**(idcomanda\_seq.nextval, '5', **to\_date**('2022-10-14','yyyy-mm-dd'), null);



insert into item\_comanda  
values(1, 1, 1);  
   
insert into item\_comanda  
values(1, 3, 1);  
   
insert into item\_comanda  
values(2, 6, 1);  
   
insert into item\_comanda  
values(3, 4, 1);  
   
insert into item\_comanda  
values(4, 5, 2);  
   
insert into item\_comanda  
values(5, 2, 1);  
   
insert into item\_comanda  
values(5, 7, 2);  
   
insert into item\_comanda  
values(6, 4, 2);  
   
insert into item\_comanda  
values(7, 1, 4);  
   
insert into item\_comanda  
values(8, 7, 1);  
   
insert into item\_comanda  
values(9, 5, 2);

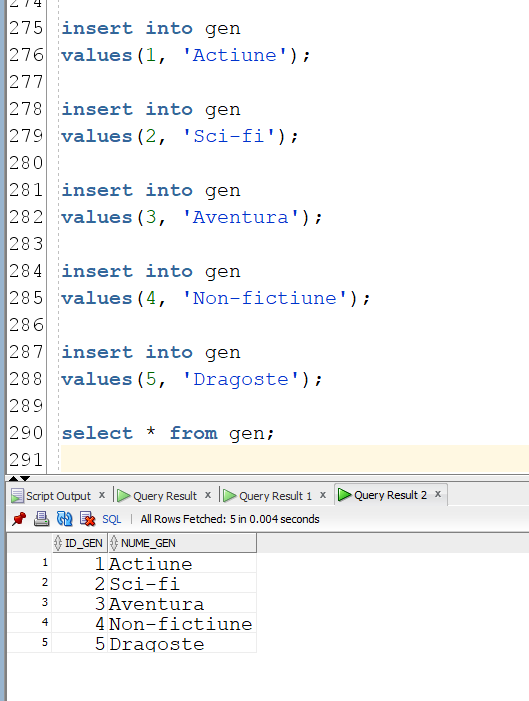


**create** **sequence** reviewid\_seq  
**start** **with** 1  
**increment** **by** 1;  
   
**insert** **into** review  
**values**(reviewid\_seq.nextval, 1, 'Servire 10/10', 5);  
   
**insert** **into** review  
**values**(reviewid\_seq.nextval, 3, null, 4);  
   
**insert** **into** review  
**values**(reviewid\_seq.nextval, 2, null, 3);  
   
**insert** **into** review  
**values**(reviewid\_seq.nextval, 4, 'Local dragut, voi reveni! :D', 5);  
   
**insert** **into** review  
**values**(reviewid\_seq.nextval, 5, 'The coffee was great', 5);

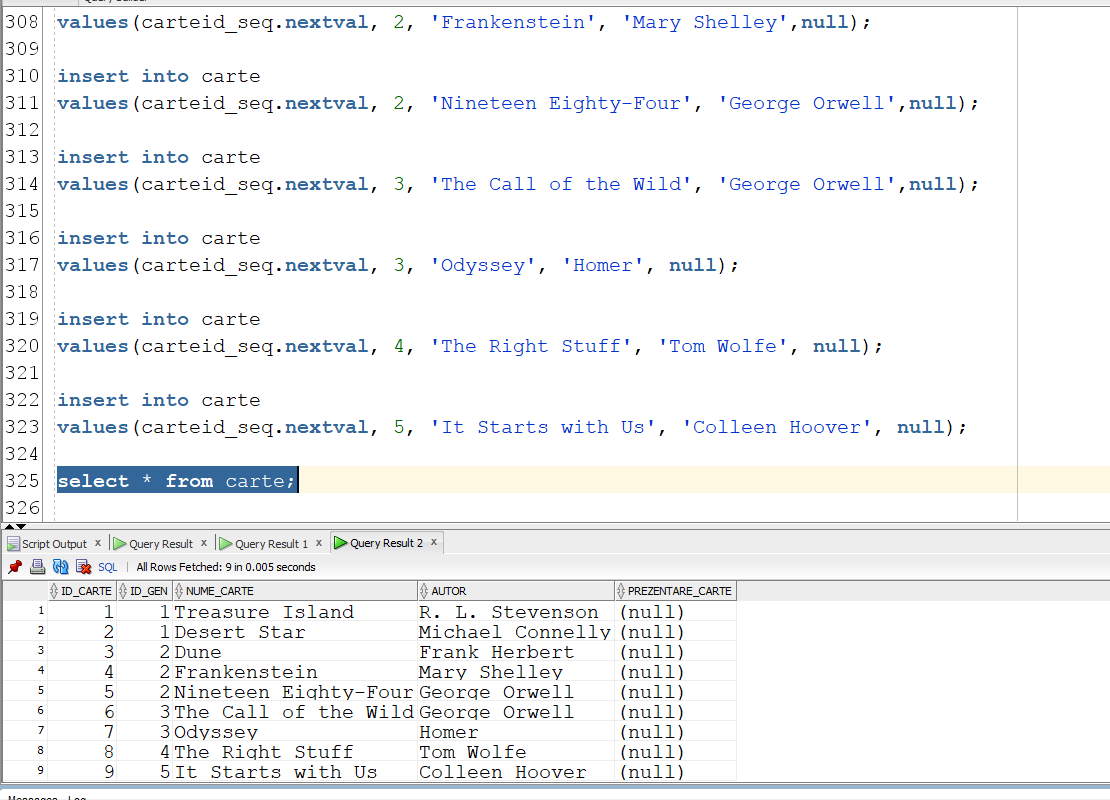


**insert** **into** gen  
**values**(1, 'Actiune');

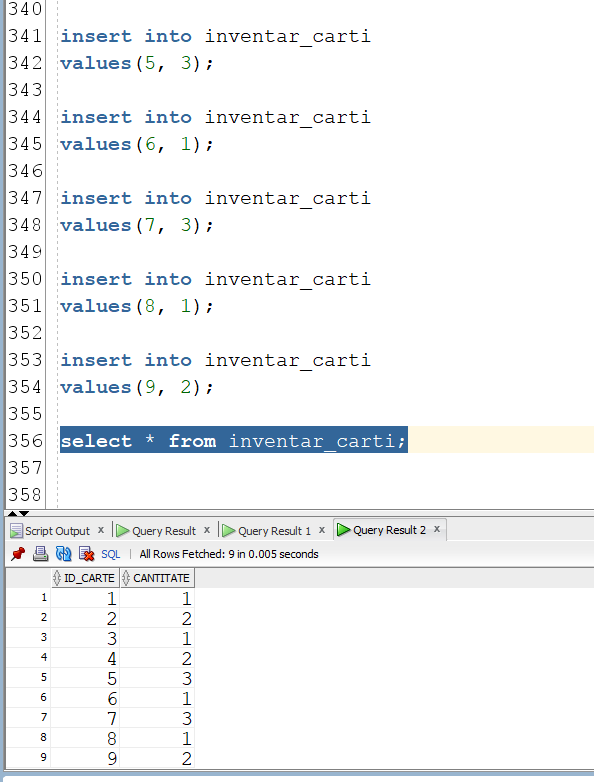
**insert** **into** gen  
**values**(2, 'Sci-fi');  
   
**insert** **into** gen  
**values**(3, 'Aventura');  
   
**insert** **into** gen  
**values**(4, 'Non-fictiune');  
   
**insert** **into** gen  
**values**(5, 'Dragoste');



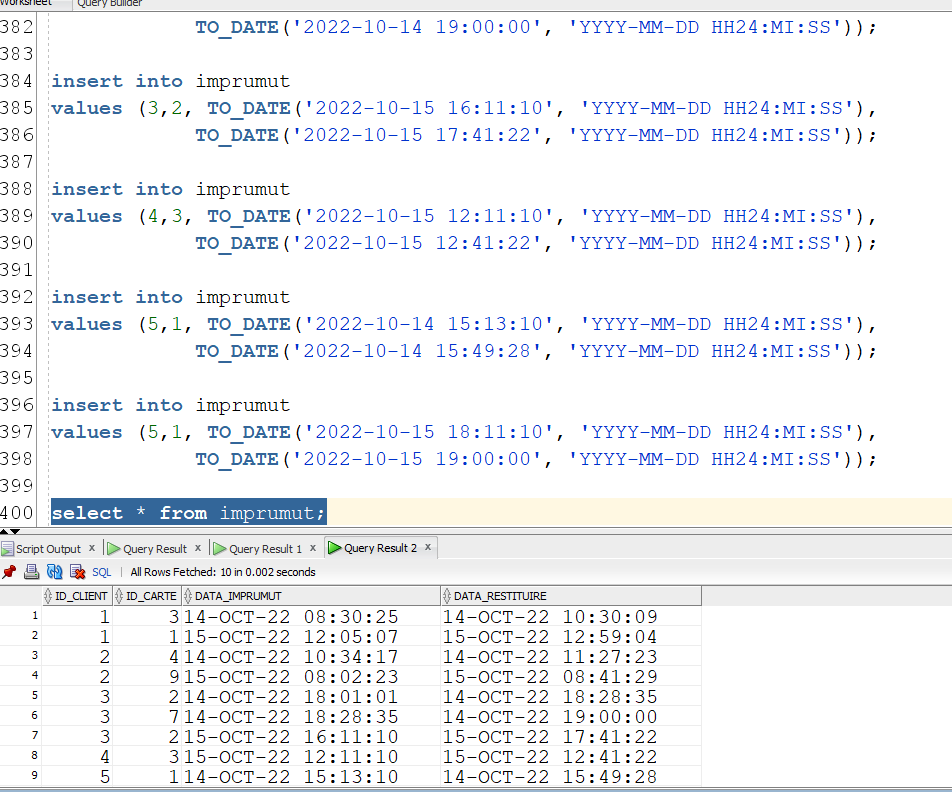
**create** **sequence** carteid\_seq  
**start** **with** 1  
**increment** **by** 1;  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 1, 'Treasure Island', 'R. L. Stevenson',null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 1, 'Desert Star', 'Michael Connelly',null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 2, 'Dune', 'Frank Herbert',null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 2, 'Frankenstein', 'Mary Shelley',null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 2, 'Nineteen Eighty-Four', 'George Orwell',null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 3, 'The Call of the Wild', 'George Orwell',null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 3, 'Odyssey', 'Homer', null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 4, 'The Right Stuff', 'Tom Wolfe', null);  
   
**insert** **into** carte  
**values**(carteid\_seq.nextval, 5, 'It Starts with Us', 'Colleen Hoover', null);



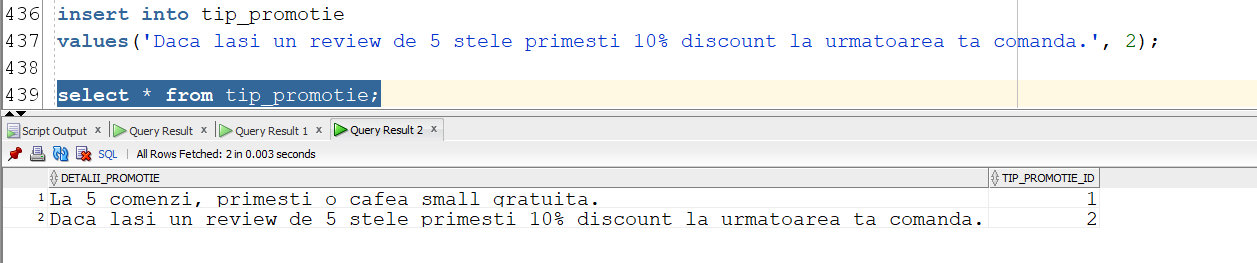
**insert** **into** inventar\_carti  
**values**(1, 1);  
   
**insert** **into** inventar\_carti  
**values**(2, 2);  
   
**insert** **into** inventar\_carti  
**values**(3, 1);  
   
**insert** **into** inventar\_carti  
**values**(4, 2);  
   
**insert** **into** inventar\_carti  
**values**(5, 3);  
   
**insert** **into** inventar\_carti  
**values**(6, 1);  
   
**insert** **into** inventar\_carti  
**values**(7, 3);  
   
**insert** **into** inventar\_carti  
**values**(8, 1);  
   
**insert** **into** inventar\_carti  
**values**(9, 2);



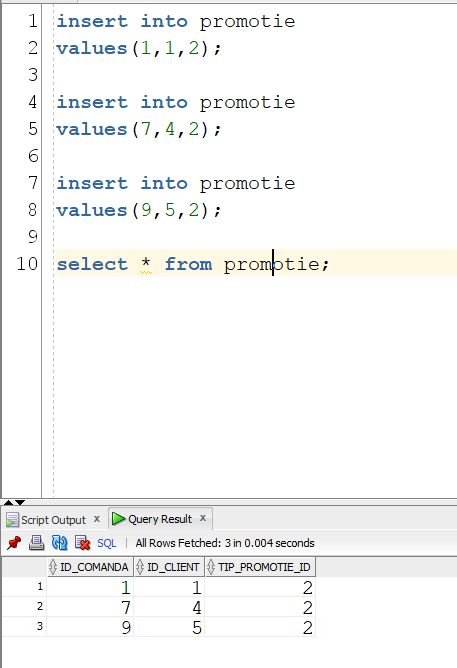
**insert** **into** imprumut  
**values** (1,3, **TO\_DATE**('2022-10-14 08:30:25', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-14 10:30:09', 'YYYY-MM-DD HH24:MI:SS'));  
              
**insert** **into** imprumut  
**values** (1,1, **TO\_DATE**('2022-10-15 12:05:07', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-15 12:59:04', 'YYYY-MM-DD HH24:MI:SS'));  
              
**insert** **into** imprumut  
**values** (2,4, **TO\_DATE**('2022-10-14 10:34:17', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-14 11:27:23', 'YYYY-MM-DD HH24:MI:SS'));  
              
**insert** **into** imprumut  
**values** (2,9, **TO\_DATE**('2022-10-15 8:02:23', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-15 8:41:29', 'YYYY-MM-DD HH24:MI:SS'));  
   
**insert** **into** imprumut  
**values** (3,2, **TO\_DATE**('2022-10-14 18:01:01', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-14 18:28:35', 'YYYY-MM-DD HH24:MI:SS'));  
              
**insert** **into** imprumut  
**values** (3,7, **TO\_DATE**('2022-10-14 18:28:35', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-14 19:00:00', 'YYYY-MM-DD HH24:MI:SS'));  
              
**insert** **into** imprumut  
**values** (3,2, **TO\_DATE**('2022-10-15 16:11:10', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-15 17:41:22', 'YYYY-MM-DD HH24:MI:SS'));  
   
**insert** **into** imprumut  
**values** (4,3, **TO\_DATE**('2022-10-15 12:11:10', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-15 12:41:22', 'YYYY-MM-DD HH24:MI:SS'));  
   
**insert** **into** imprumut  
**values** (5,1, **TO\_DATE**('2022-10-14 15:13:10', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-14 15:49:28', 'YYYY-MM-DD HH24:MI:SS'));  
   
**insert** **into** imprumut  
**values** (5,1, **TO\_DATE**('2022-10-15 18:11:10', 'YYYY-MM-DD HH24:MI:SS'),  
            **TO\_DATE**('2022-10-15 19:00:00', 'YYYY-MM-DD HH24:MI:SS'));



**insert** **into** tip\_promotie  
**values**('La 5 comenzi, primesti o cafea small gratuita.', 1);   
   
**insert** **into** tip\_promotie  
**values**('Daca lasi un review de 5 stele primesti 10% discount la urmatoarea ta comanda.', 2);

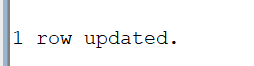


insert into promotie  
values(1,1,2);  
   
insert into promotie  
values(7,4,2);  
   
insert into promotie  
values(9,5,2);



Modificare pt exercitiul 9:

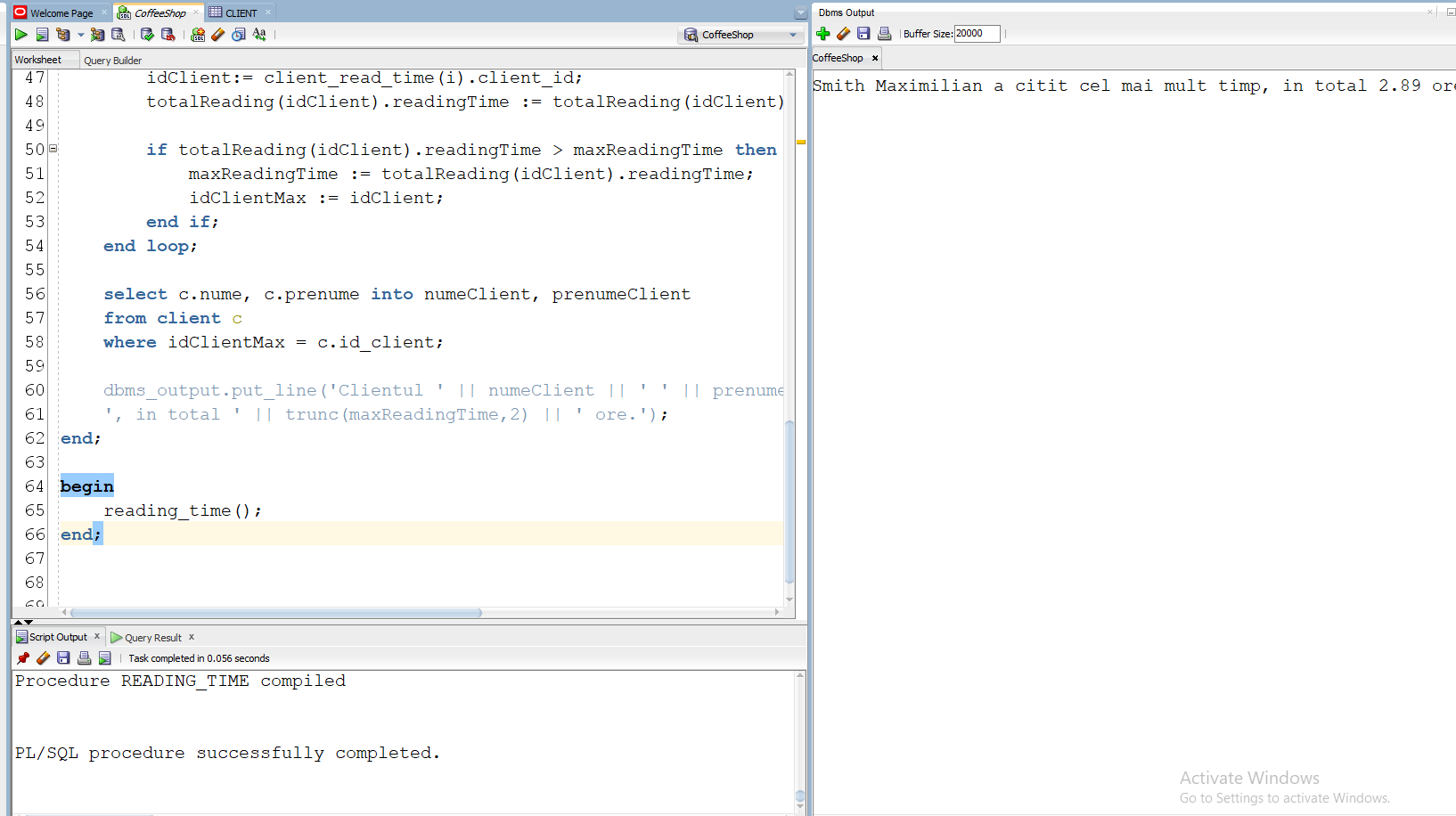
**update** **client**  
    **set** **nume** = 'Jane'  
    where nume = 'Holt';



6.

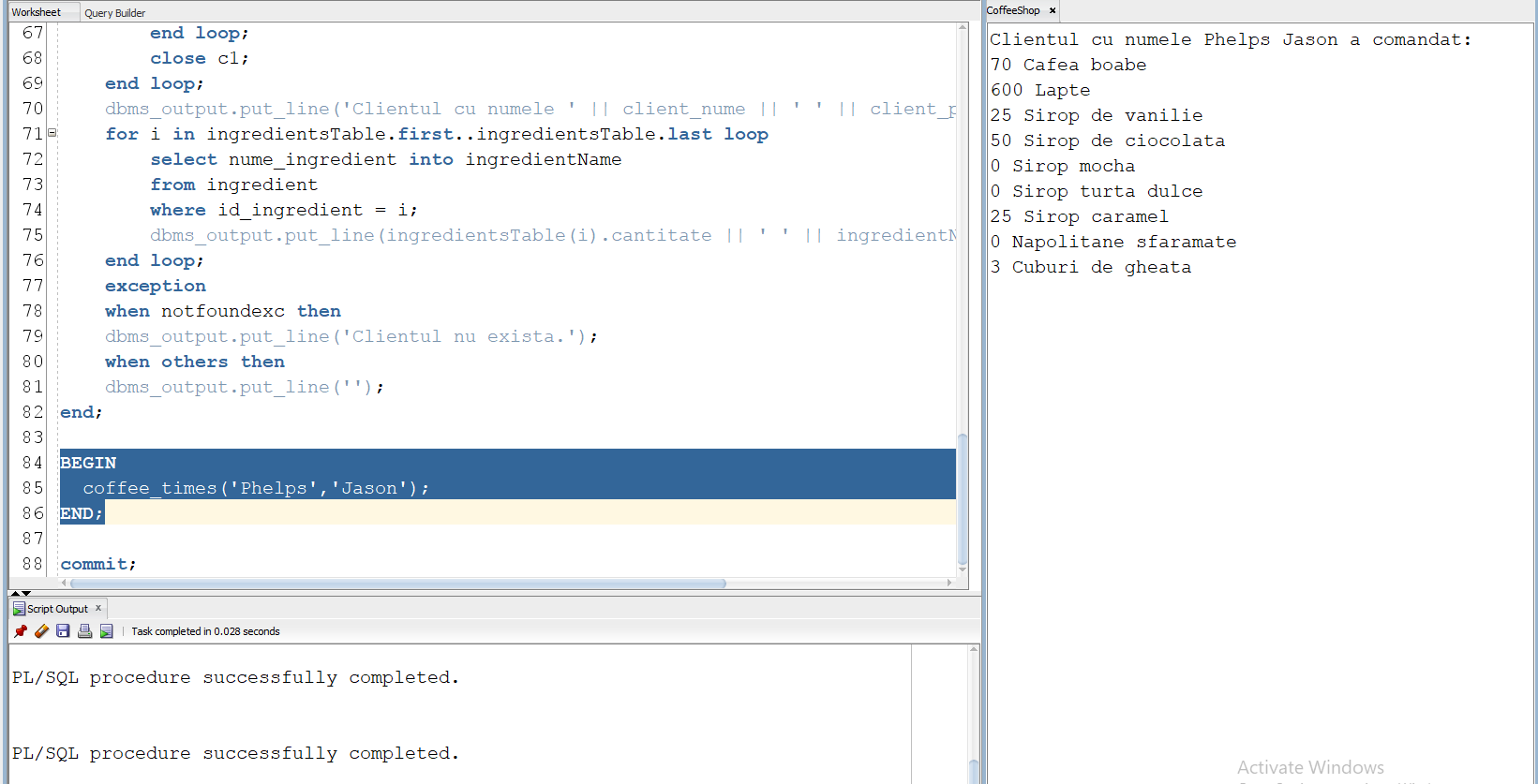
-- Sa se implementeze o procedura care afiseaza clientul care **a** citit cel mai mult timp  
-- si numarul de ore pe care l-**a** citit.  
-- Idee: inregistrez fiecare imprumut intr-un tabel imbricat cu   
-- **in** care retin durata timpului de citit si id-**ul** clientului **in** cauza.  
-- Adun timpurile de citit retinute **in** tabelul imbricat intr-un tablou indexat.  
-- Intr-un **if** clasic determin clientul care **a** citit cel mai mult.

**create** **or replace** **procedure** reading\_time  
**is**  
    **type** c\_r **is** record  
    (  
    client\_id client.id\_client%type,  
    readingTime float --in ore  
    );  
    **type** hours\_read **is** **table** **of** c\_r;  --tablou imbricat  
    client\_read\_time hours\_read;  
    **type** tr **is** **table** **of** c\_r **index** **by** pls\_integer; --tablou indexat   
    totalReading tr;  
    maxReadingTime FLOAT := 0;  
    idClientMax number(8);  
    numeClient client.nume%type;  
    prenumeClient client.prenume%type;  
    idClient client.id\_client%type;  
**begin**  
    **select** id\_client, (data\_restituire - data\_imprumut)\*24  
    bulk collect **into** client\_read\_time  
    **from** imprumut;  
      
    **for** i **in** client\_read\_time.first..client\_read\_time.last **loop**  
        idClient:= client\_read\_time(i).client\_id;  
        totalReading(idClient).client\_id := client\_read\_time(i).client\_id;  
        totalReading(idClient).readingTime := 0;  
    **end** **loop**;  
   
    **for** i **in** client\_read\_time.first..client\_read\_time.last **loop**  
        idClient:= client\_read\_time(i).client\_id;  
        totalReading(idClient).readingTime := totalReading(idClient).readingTime + client\_read\_time(i).readingTime;  
          
        **if** totalReading(idClient).readingTime > maxReadingTime **then**  
            maxReadingTime := totalReading(idClient).readingTime;  
            idClientMax := idClient;  
        **end** **if**;  
    **end** **loop**;  
      
    **select** c.nume, c.prenume **into** numeClient, prenumeClient  
    **from** client c  
    **where** idClientMax = c.id\_client;  
      
    dbms\_output.put\_line('Clientul ' || numeClient || ' ' || prenumeClient ||' a citit cel mai mult timp' ||  
    ', in total ' || trunc(maxReadingTime,2) || ' ore.');  
**end**;  
   
**begin**  
    reading\_time();  
**end**;



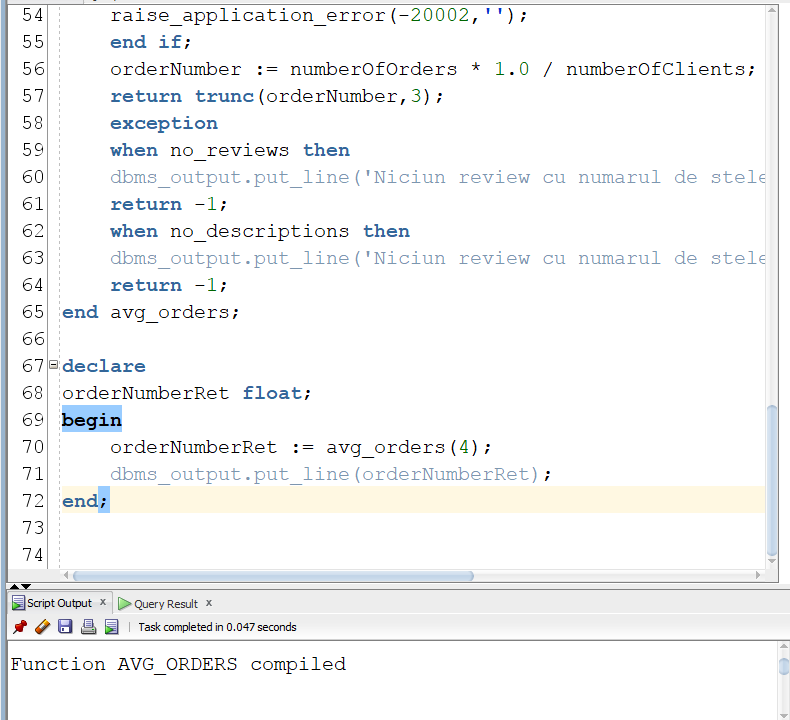
7.

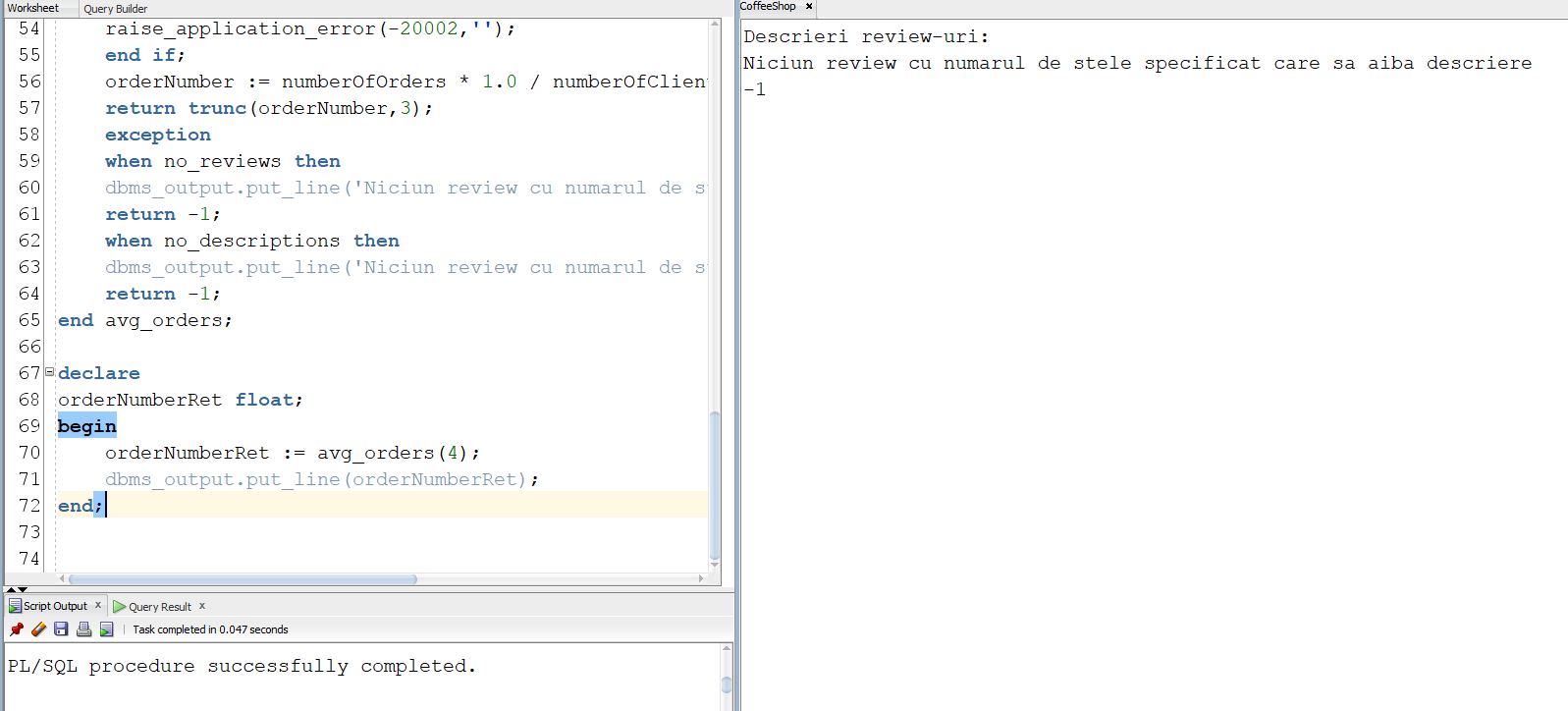
- --7.  
--Pentru un client al carui nume e dat de la tastatura,  
--sa se afiseze, totalul de ingrediente care a fost folosit  
--in prepararea comenzilor lui, de-a lungul istoricului sau  
--de comenzi  
**create** **or replace** **procedure** coffee\_times  
(client\_nume client.nume%type,  
client\_prenume client.prenume%type)  
**is**   
    **cursor** c **is**   
    **select** id\_comanda **from** comanda  
    **where** id\_client = (**select** id\_client **from** client   
    **where** upper(nume) = upper(client\_nume)   
    **and** upper(prenume) = upper(client\_prenume));  
      
    **cursor** c1(id\_com comanda.id\_comanda%type) **is**   
    **select** id\_cafea, cantitate **from** item\_comanda  
    **where** id\_comanda = id\_com;  
      
    **cursor** c2(idcaf cafea.id\_cafea%type)  
    **is** **select** id\_ingredient, cantitate  
    **from** contine\_ingrediente **where** id\_cafea = idcaf;  
      
    cafeaCantitate c1%rowtype;  
    ingredientCantitate c2%rowtype;  
      
    **type** ingred **is** record  
    (  
    id\_ing ingredient.id\_ingredient%type,  
    cantitate contine\_ingrediente.cantitate%type  
    );  
    **type** ingTab **is** **table** **of** ingred;  
    ingredientsTable ingTab := ingTab();  
    nringred number(4);  
    ingredientName ingredient.nume\_ingredient%type;  
    nrclienti number(4);  
    notfoundexc **exception**;  
**begin**  
    **select** count(id\_client) **into** nrclienti **from** client   
    **where** upper(nume) = upper(client\_nume)   
    **and** upper(prenume) = upper(client\_prenume);  
      
    **if** nrclienti = 0 **then**  
        **raise** notfoundexc;   
    **end** **if**;  
   
    **select** count(\*) **into** nringred   
    **from** ingredient;  
    ingredientsTable.extend(nringred+1);  
    **for** i **in** 1..nringred **loop**  
        ingredientsTable(i).id\_ing := i;  
        ingredientsTable(i).cantitate := 0;  
    **end** **loop**;  
    **for** comanda\_entry **in** c **loop** --ciclu cursor  
        **open** c1(comanda\_entry.id\_comanda);  
        **loop**  
            **fetch** c1 **into** cafeaCantitate; --cursor clasic  
            **exit** **when** c1%NOTFOUND;  
                **for** ingredientCantitate **in** c2(cafeaCantitate.id\_cafea) **loop**  
                    --acum am: in comanda\_entry Id comanda  
                    --in cafeaCantitate id\_cafea si cantitatea cafelei  
                    --in ingredientCantitate id\_ingredient si cantitatea ingredientului  
                    ingredientsTable(ingredientCantitate.id\_ingredient).cantitate :=   
                    ingredientsTable(ingredientCantitate.id\_ingredient).cantitate +   
                    cafeaCantitate.cantitate \* ingredientCantitate.cantitate;  
                **end** **loop**;  
        **end** **loop**;  
        **close** c1;  
    **end** **loop**;  
    dbms\_output.put\_line('Clientul cu numele ' || client\_nume || ' ' || client\_prenume || ' a comandat:');  
    **for** i **in** ingredientsTable.first..ingredientsTable.last **loop**  
        **select** nume\_ingredient **into** ingredientName   
        **from** ingredient  
        **where** id\_ingredient = i;  
        dbms\_output.put\_line(ingredientsTable(i).cantitate || ' ' || ingredientName);  
    **end** **loop**;  
    **exception**  
    **when** notfoundexc **then**  
    dbms\_output.put\_line('Clientul nu exista.');  
    **when** others **then**  
    dbms\_output.put\_line('');  
**end**;  
   
**BEGIN**  
  coffee\_times('Phelps','Jason');  
**END**;

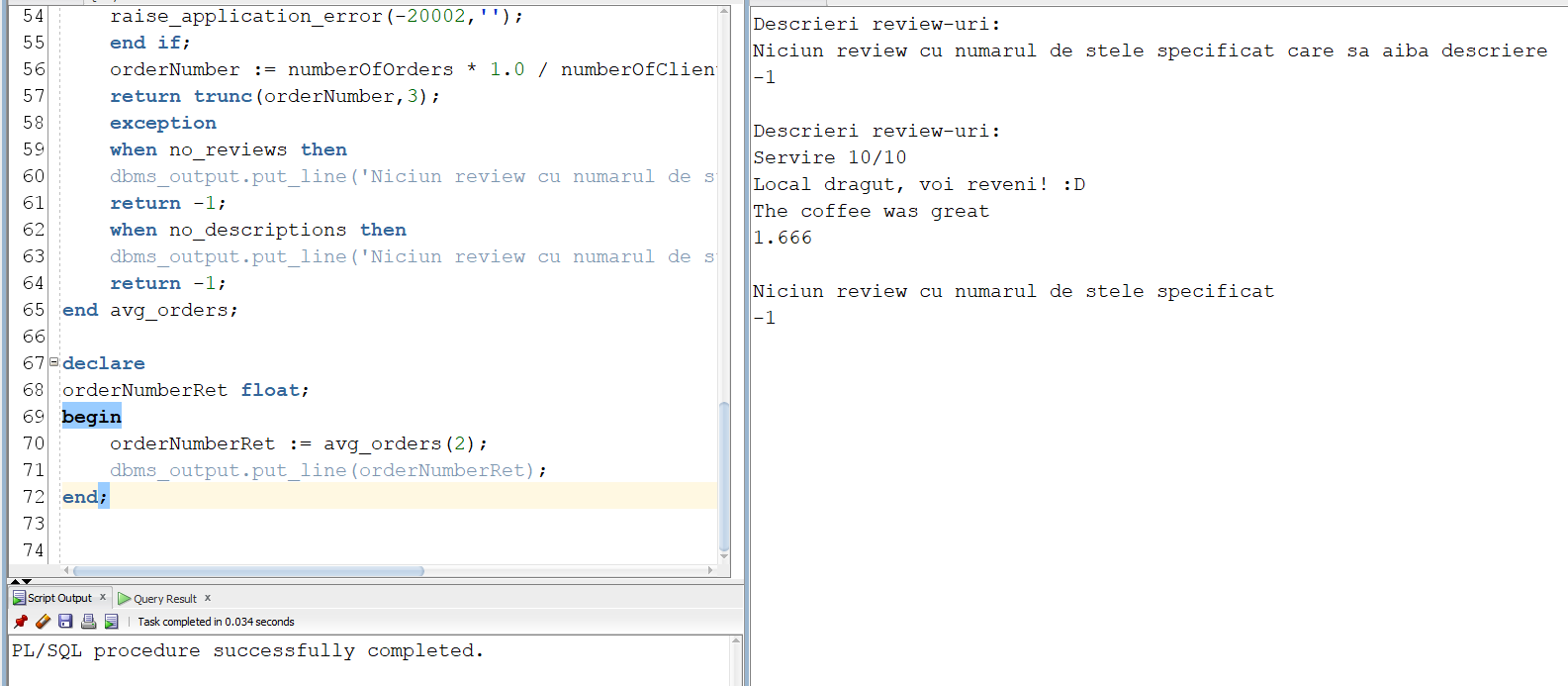
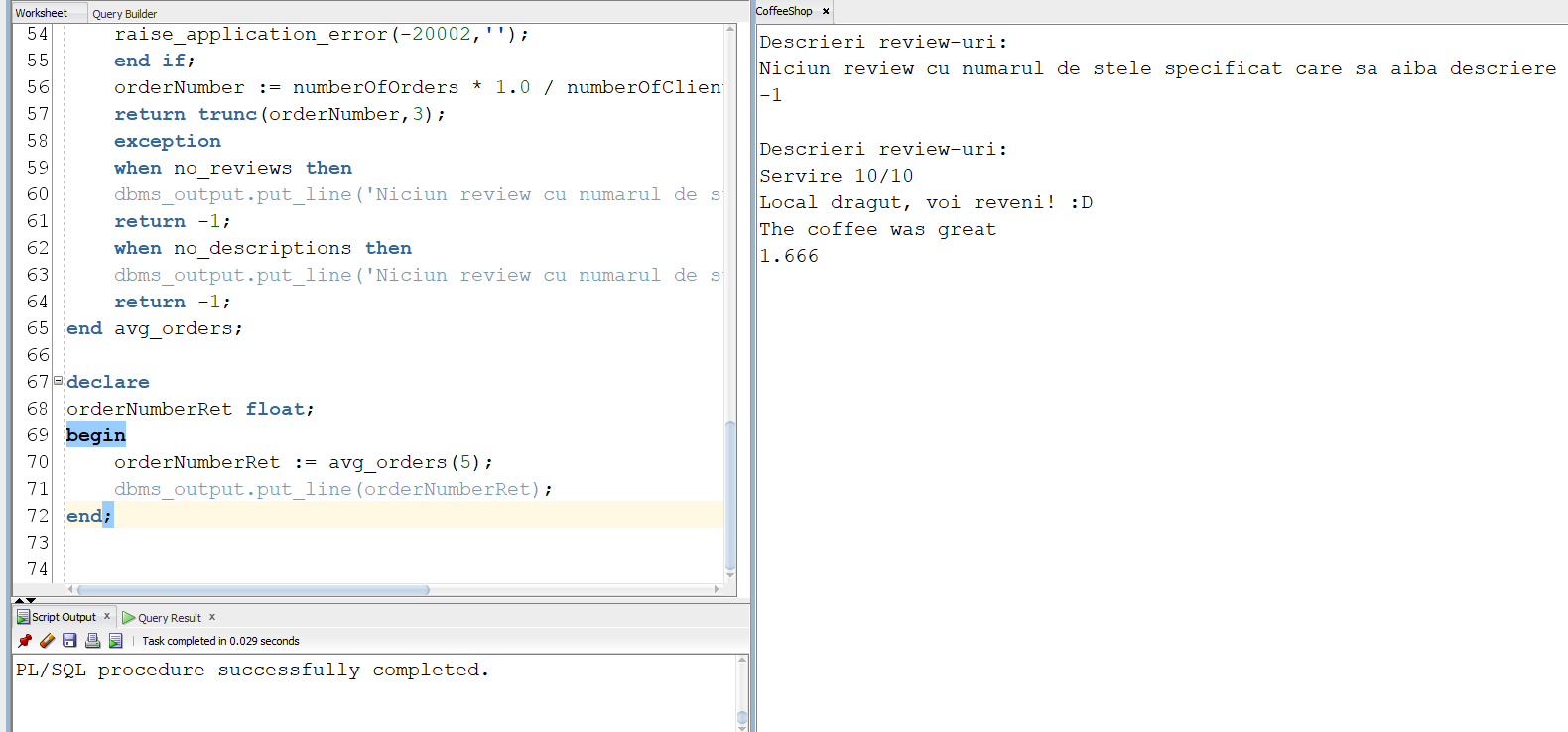


8  
  
8  
  
--Sa se scrie o functie care returneaza numarul mediu  
--de comenzi facute de un client care a lasat un review  
--de X stele (x fiind parametru dat), si afiseaza descriptiile review-urilor **in** cauza (daca   
--nu exista cel putin o descriere, se va apela o exceptie).

**create** **or replace** **function** avg\_orders(stele review.nr\_stele%type)  
**return** float **is**   
no\_reviews **EXCEPTION**;  
PRAGMA exception\_init(no\_reviews, -20001);  
   
no\_descriptions **EXCEPTION**;  
PRAGMA exception\_init(no\_descriptions, -20002);  
   
orderNumber float;  
**type** descriptionAndCount **is** record  
(  
id\_client client.id\_client%type,  
countOfOrders number(6),  
review\_description review.descriptie%type  
);  
**type** descCount **is** **table** **of** descriptionAndCount **index** **by** pls\_integer;  
reviewTable descCount;   
numberOfOrders number(4) := 0;  
numberOfClients number(4) :=0;  
numberOfDescriptions number(4) :=0;  
**begin**  
    **select** count(\*) **into** numberOfClients  
    **from** review  
    **where** nr\_stele = stele;  
      
    **if** numberOfClients =0 **then**  
        raise\_application\_error(-20001, '');  
    **end** **if**;  
    numberOfClients := 0;  
    **select** max(c.id\_client), count(com.id\_comanda), max(r.descriptie) bulk collect **into** reviewTable  
    **from** review r   
    **join** client c   
    **on** r.id\_client = c.id\_client  
    **join** comanda com   
    **on** com.id\_client = c.id\_client  
    **where** r.nr\_stele = stele  
    **group** **by** c.id\_client;  
    dbms\_output.put\_line('Descrieri review-uri:');  
    **for** i **in** reviewTable.first..reviewTable.last **loop**  
       numberOfOrders := numberOfOrders + reviewTable(i).countOfOrders;  
       numberOfClients := numberOfClients +1;  
       **if** reviewTable(i).review\_description **is** **not** **null** **then**  
       dbms\_output.put\_line(reviewTable(i).review\_description);  
       numberOfDescriptions := numberOfDescriptions + 1;  
       **end** **if**;  
    **end** **loop**;  
    **if** numberOfDescriptions =0 **then**  
    raise\_application\_error(-20002,'');  
    **end** **if**;  
    orderNumber := numberOfOrders \* 1.0 / numberOfClients;  
    **return** trunc(orderNumber,3);  
    **exception**  
    **when** no\_reviews **then**  
    dbms\_output.put\_line('Niciun review cu numarul de stele specificat');  
    **return** -1;  
    **when** no\_descriptions **then**  
    dbms\_output.put\_line('Niciun review cu numarul de stele specificat care sa aiba descriere');  
    **return** -1;  
**end** avg\_orders;

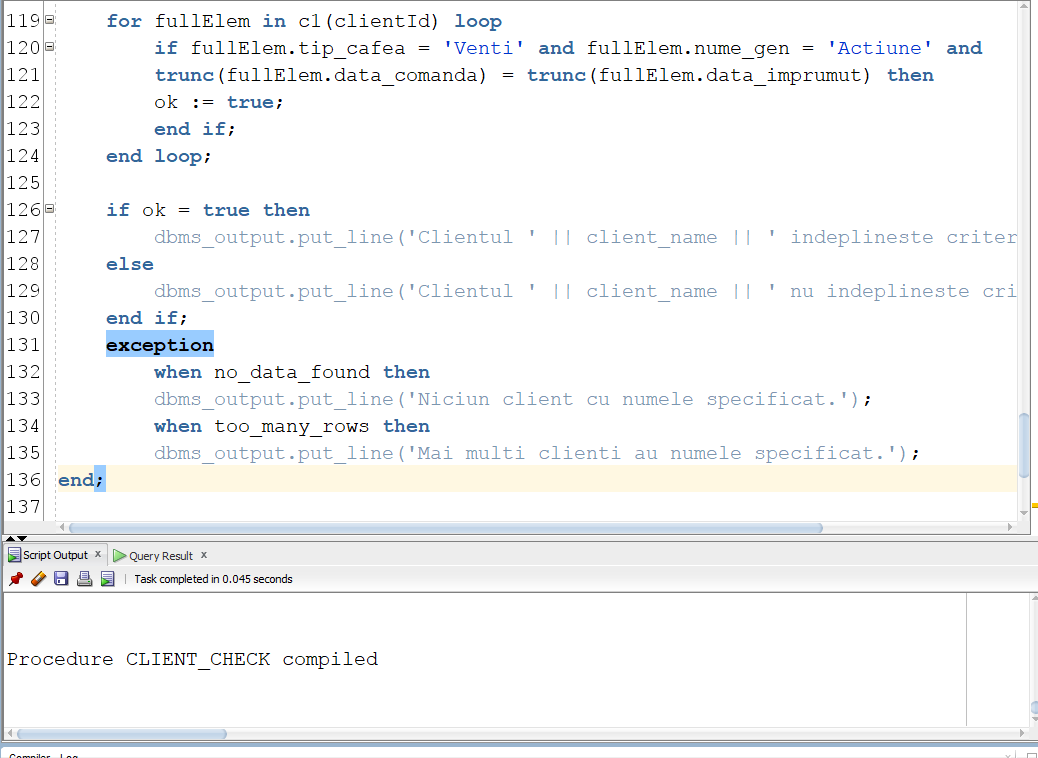


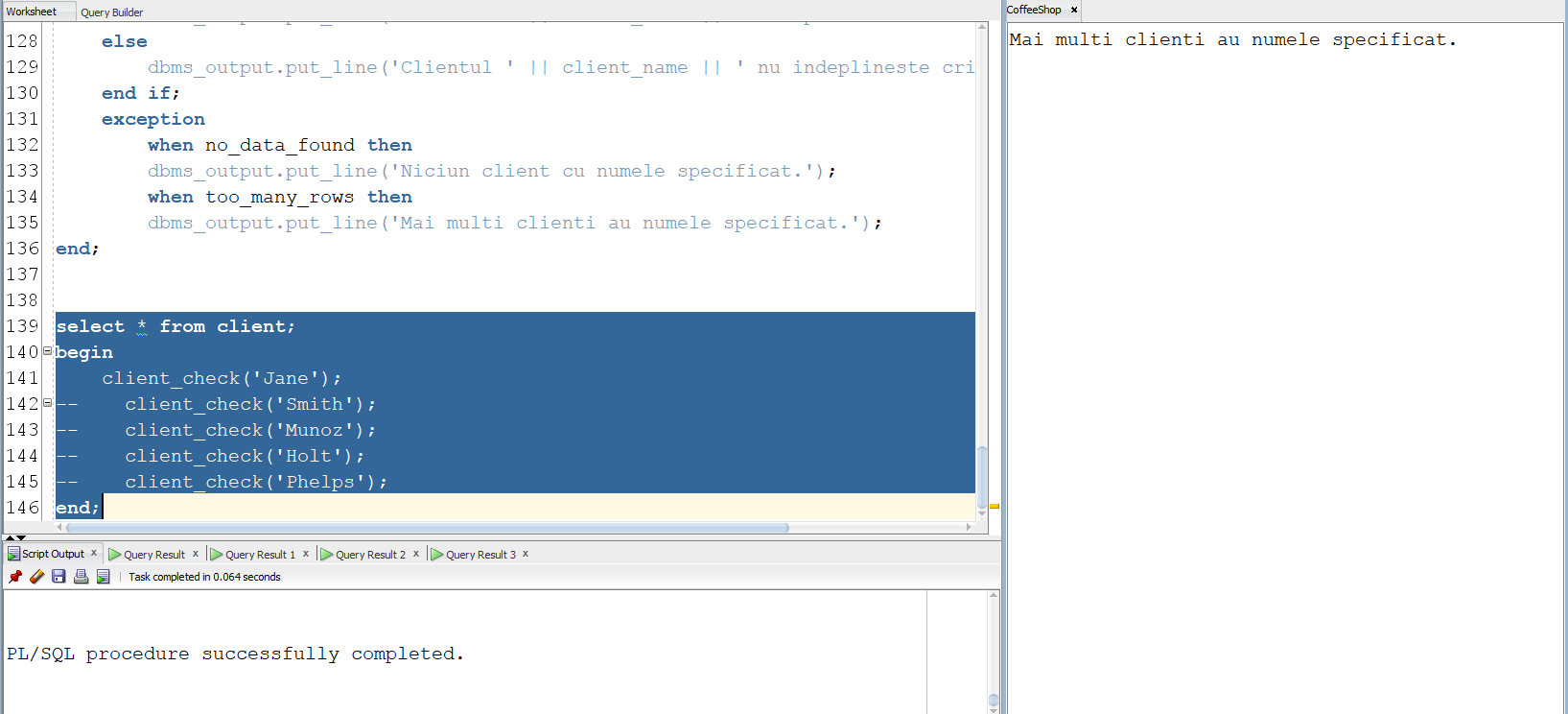


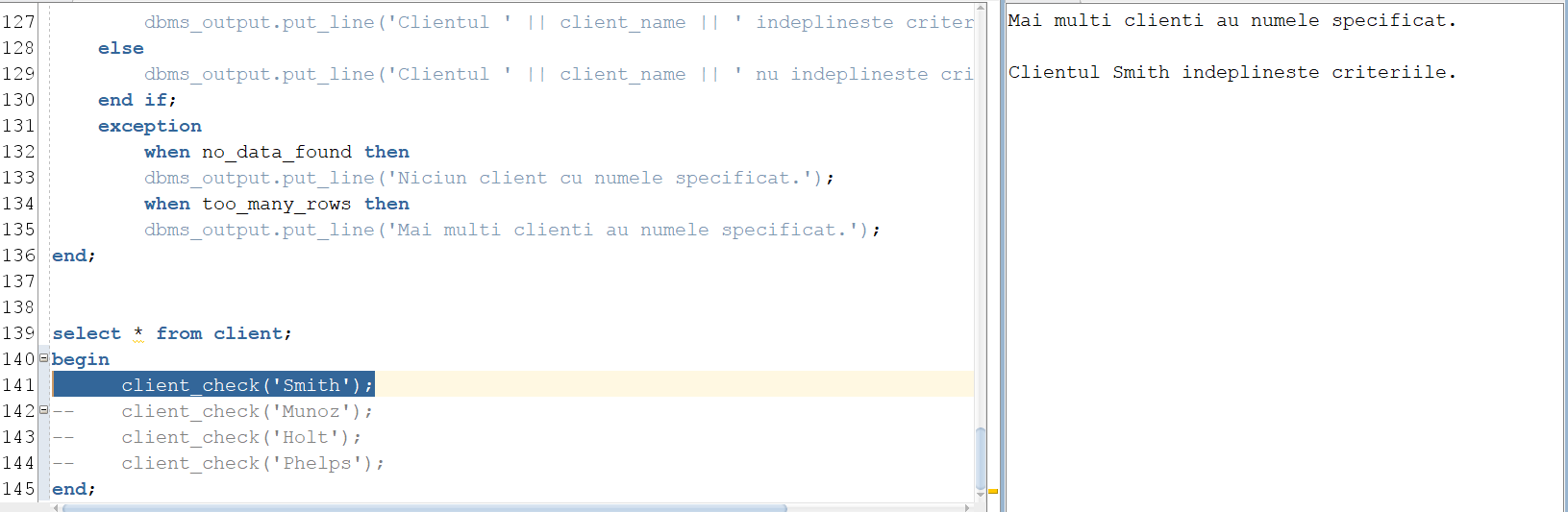


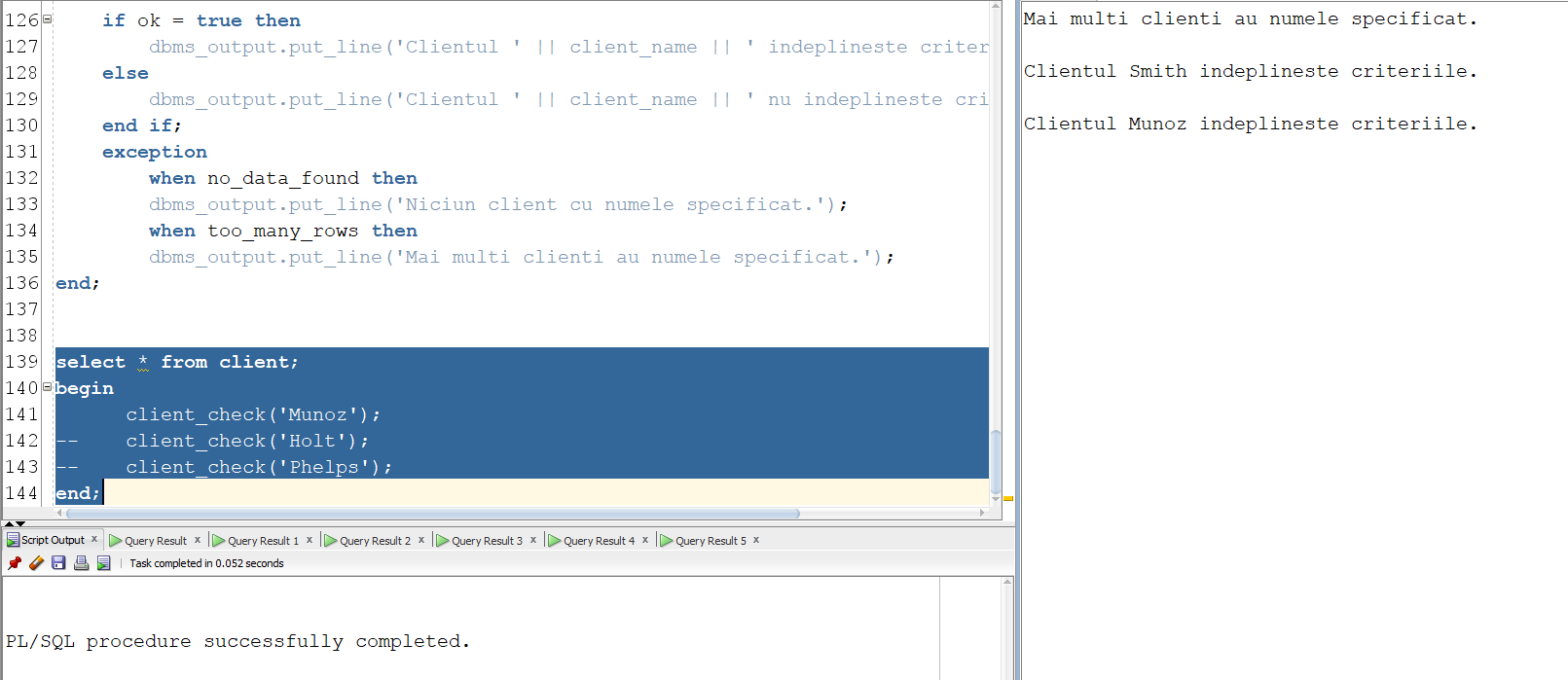
9.   
--9. Pentru un client cu numele dat de la tastatura,  
-- sa se verifice daca clientul a comandat o cafea venti si  
-- a imprumutat o carte din categoria actiune in cadrul aceleasi vizite.

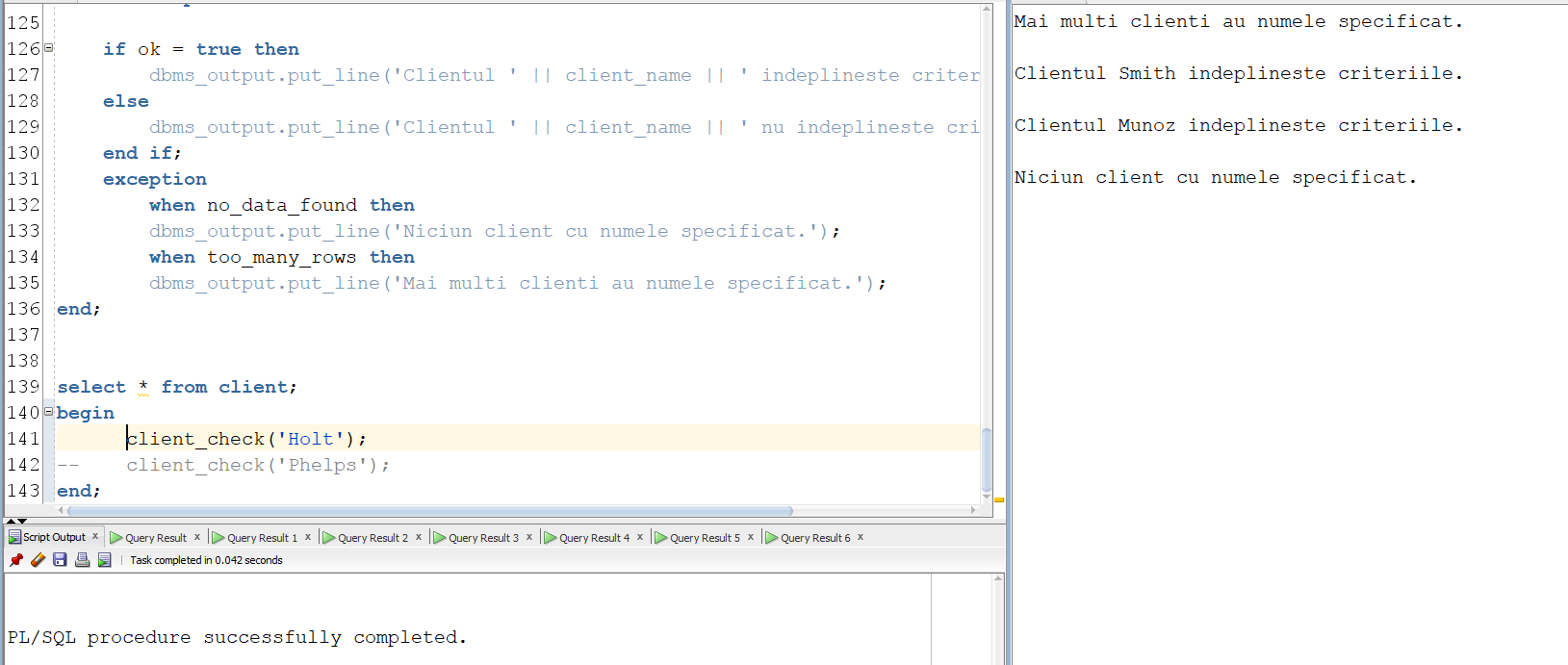
**create** **or replace** **procedure** client\_check  
(client\_name client.nume%type)  
**is**  
    **cursor** c1(cid client.id\_client%type) **is**   
    **select** c.id\_client, caf.tip\_cafea, g.nume\_gen, com.data\_comanda, imp.data\_imprumut  
    **from** client c   
    **join** comanda com   
    **on** com.id\_client = c.id\_client  
    **join** item\_comanda ic  
    **on** ic.id\_comanda = com.id\_comanda  
    **join** cafea caf  
    **on** caf.id\_cafea = ic.id\_cafea  
    **join** imprumut imp   
    **on** imp.id\_client = c.id\_client  
    **join** carte car   
    **on** car.id\_carte = imp.id\_carte  
    **join** gen g  
    **on** g.id\_gen = car.id\_gen  
    **where** id\_client = cid;  
    clientId client.id\_client%type;   
    ok boolean := **false**;  
**begin**  
    **select** id\_client **into** clientId  
    **from** client c  
    **where** upper(c.nume) = upper(client\_name);  
      
    **for** fullElem **in** c1(clientId) **loop**  
        **if** fullElem.tip\_cafea = 'Venti' **and** fullElem.nume\_gen = 'Actiune' **and**   
        trunc(fullElem.data\_comanda) = trunc(fullElem.data\_imprumut) **then**  
        ok := **true**;  
        **end** **if**;  
    **end** **loop**;  
      
    **if** ok = **true** **then**  
        dbms\_output.put\_line('Clientul ' || client\_name || ' indeplineste criteriile.');  
    **else**  
        dbms\_output.put\_line('Clientul ' || client\_name || ' nu indeplineste criteriile.');  
    **end** **if**;  
    **exception**  
        **when** no\_data\_found **then**  
        dbms\_output.put\_line('Niciun client cu numele specificat.');  
        **when** too\_many\_rows **then**  
        dbms\_output.put\_line('Mai multi clienti au numele specificat.');  
**end**;

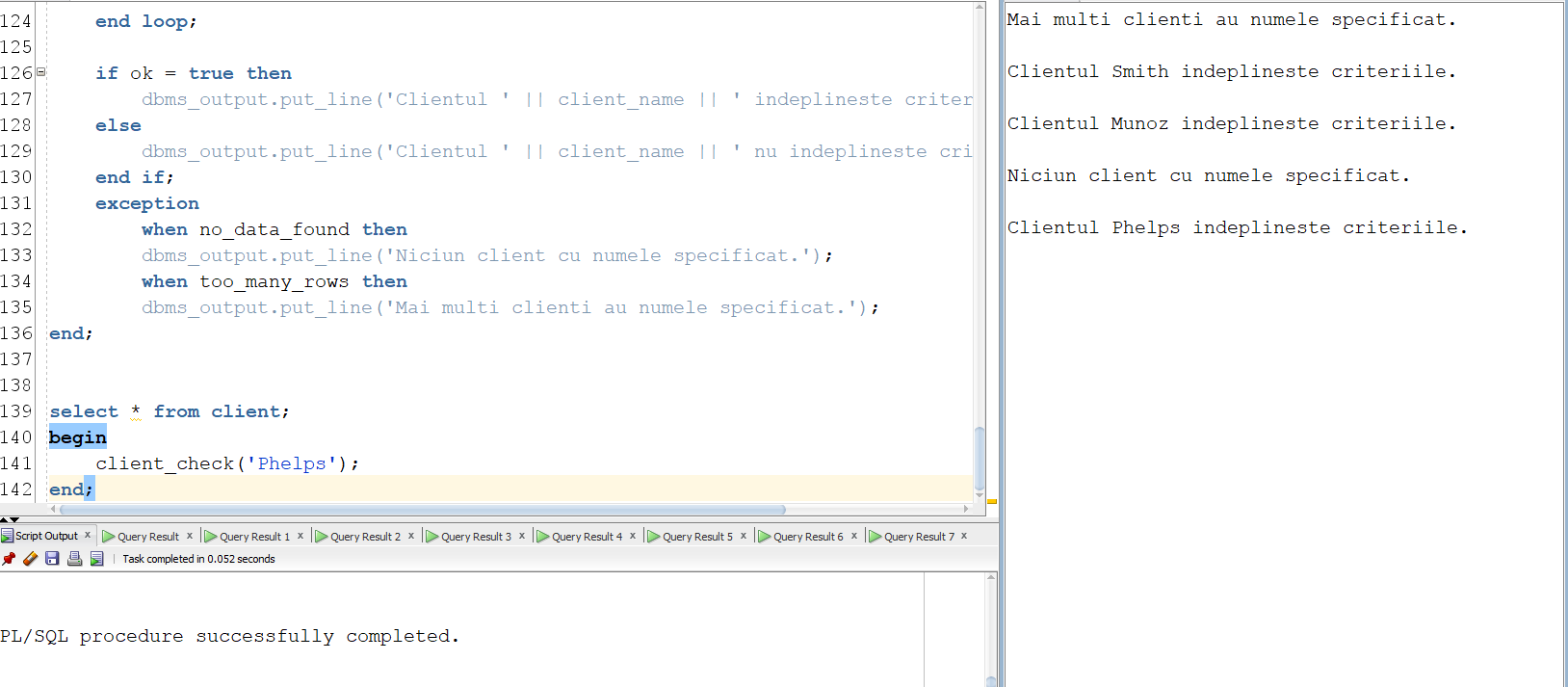






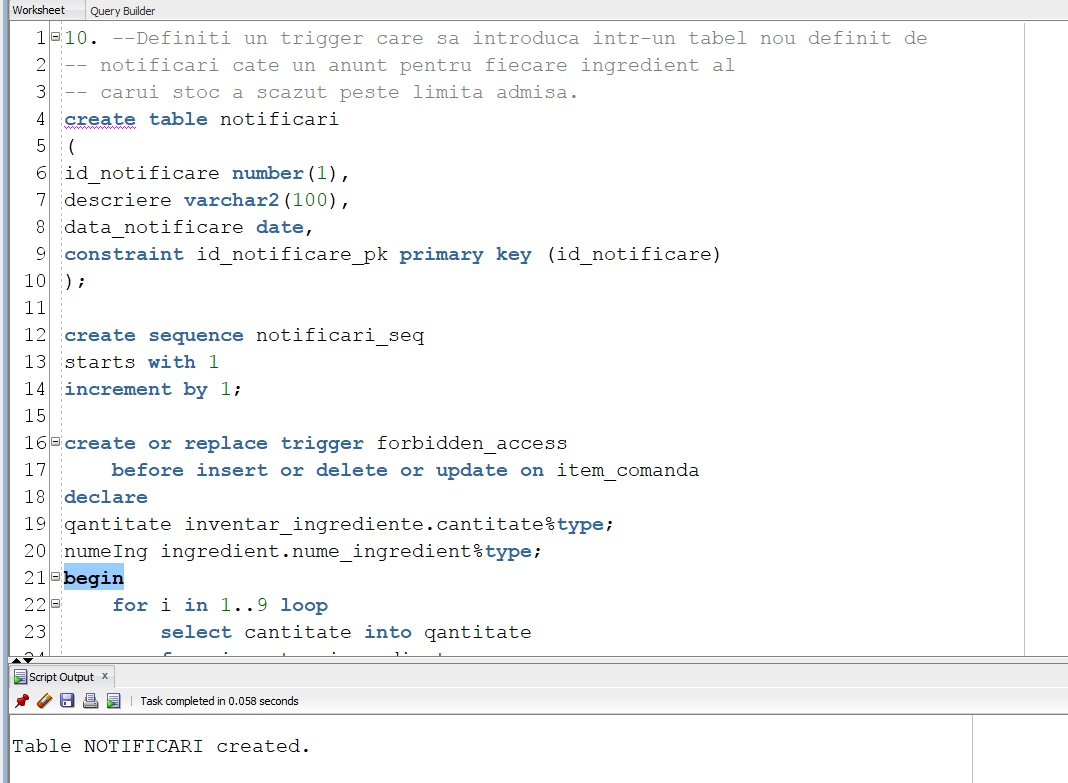


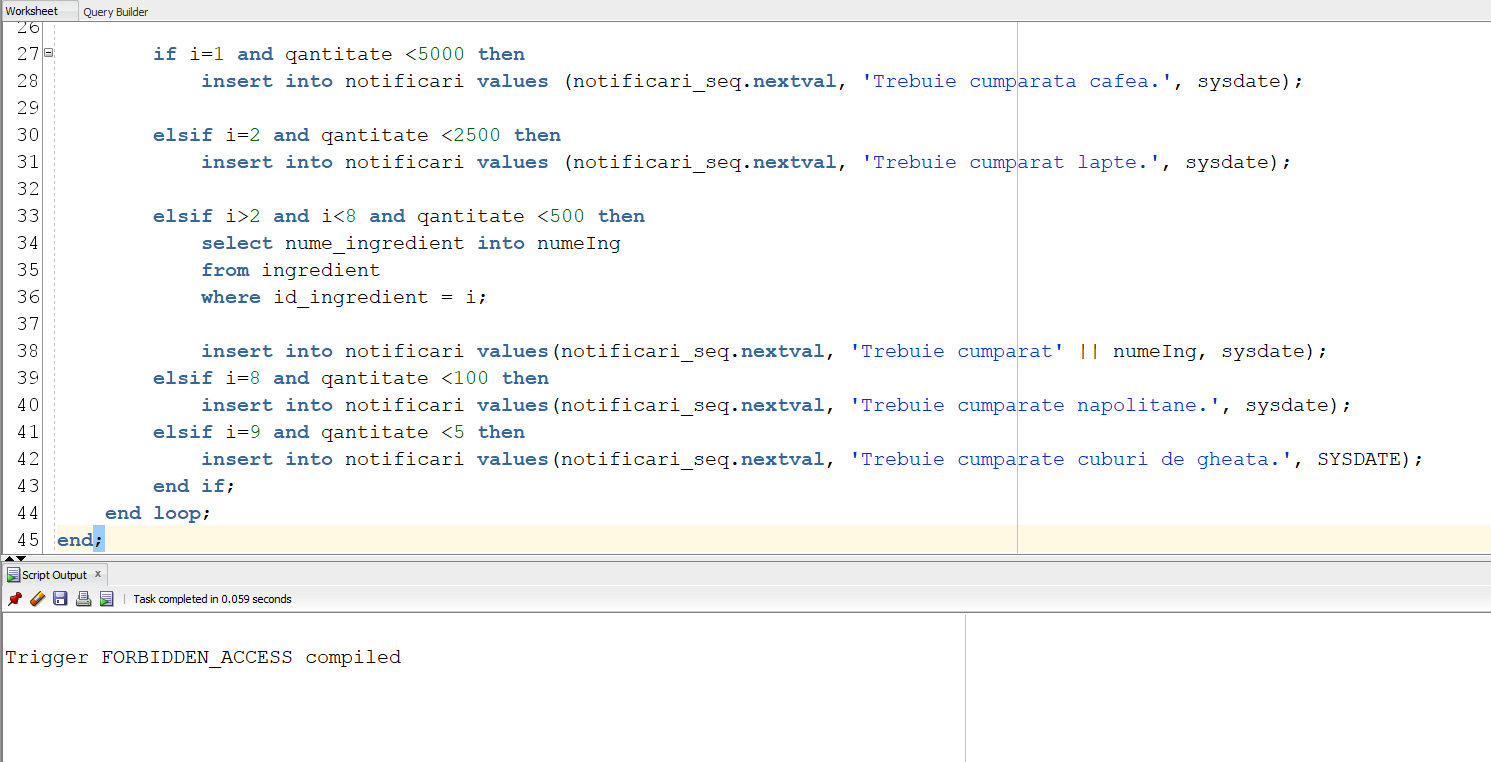


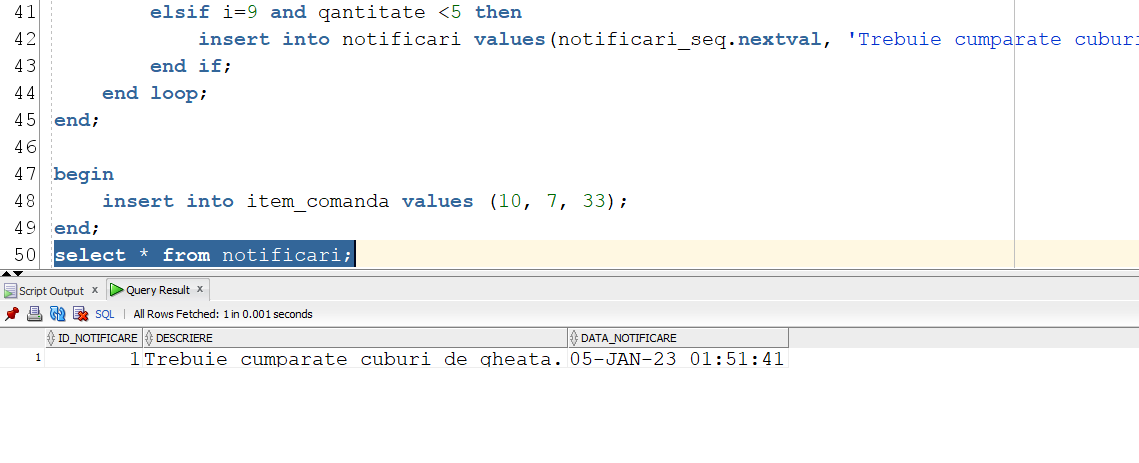


10.

10. --Definiti un trigger care sa introduca intr-un tabel nou definit de  
-- notificari cate un anunt pentru fiecare ingredient al  
-- carui stoc a scazut peste limita admisa.  
**create** **table** notificari   
(  
id\_notificare number(3),  
descriere varchar2(100),  
data\_notificare date,  
**constraint** id\_notificare\_pk **primary key** (id\_notificare)  
);  
   
**create** **sequence** notificari\_seq  
**start** **with** 1  
**increment** **by** 1;  
   
**create** **or replace** **trigger** forbidden\_access  
    **after** **insert** **on** item\_comanda  
**declare**  
qantitate inventar\_ingrediente.cantitate\_disponibila%type;  
numeIng ingredient.nume\_ingredient%type;  
**begin**  
    **for** i **in** 1..9 **loop**   
        **select** cantitate\_disponibila **into** qantitate  
        **from** inventar\_ingrediente  
        **where** id\_ingredient = i;  
          
        **if** i=1 **and** qantitate <5000 **then**  
            **insert** **into** notificari **values** (notificari\_seq.nextval, 'Trebuie cumparata cafea.', sysdate);  
              
        **elsif** i=2 **and** qantitate <2500 **then**  
            **insert** **into** notificari **values** (notificari\_seq.nextval, 'Trebuie cumparat lapte.', sysdate);  
              
        **elsif** i>2 **and** i<8 **and** qantitate <500 **then**  
            **select** nume\_ingredient **into** numeIng  
            **from** ingredient  
            **where** id\_ingredient = i;  
              
            **insert** **into** notificari **values**(notificari\_seq.nextval, 'Trebuie cumparat' || numeIng, sysdate);  
        **elsif** i=8 **and** qantitate <100 **then**  
            **insert** **into** notificari **values**(notificari\_seq.nextval, 'Trebuie cumparate napolitane.', sysdate);  
        **elsif** i=9 **and** qantitate <5 **then**  
            **insert** **into** notificari **values**(notificari\_seq.nextval, 'Trebuie cumparate cuburi de gheata.', SYSDATE);  
        **end** **if**;  
    **end** **loop**;  
**end**;  
   
**begin**  
    **insert** **into** item\_comanda **values** (10, 7, 33);  
**end**;  
**select** \* **from** notificari;  
**commit**;

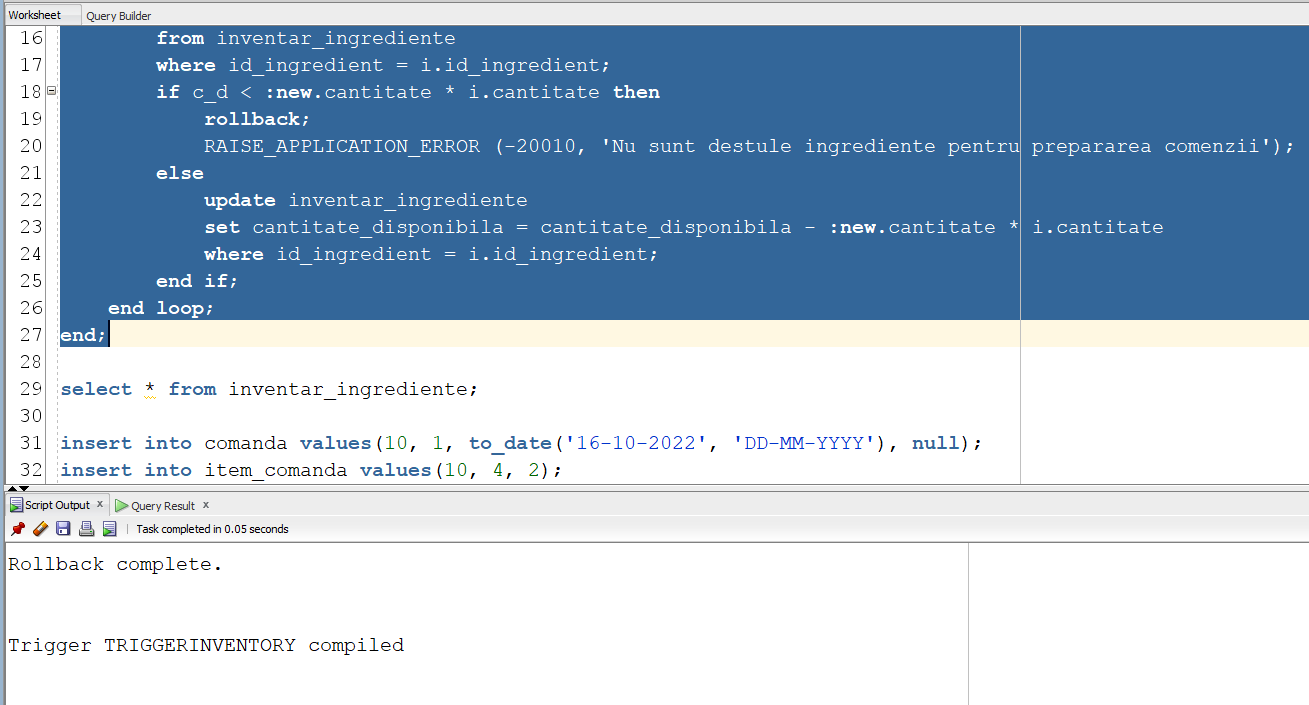


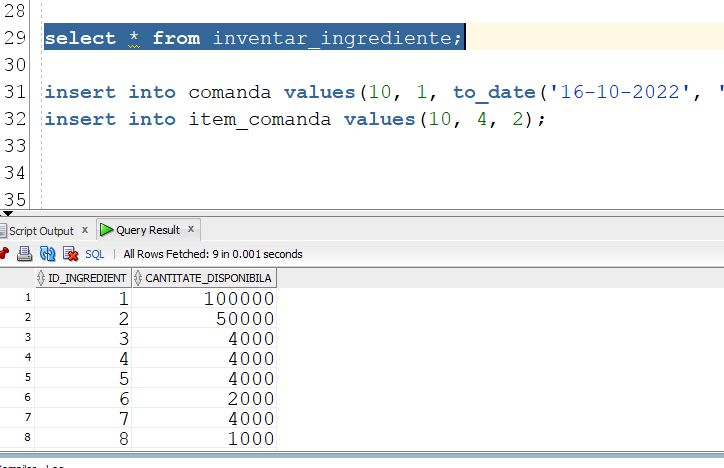


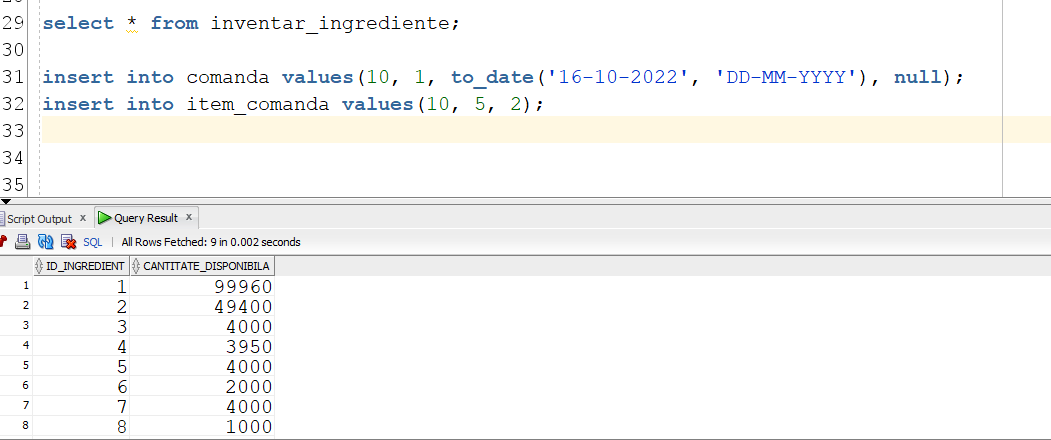
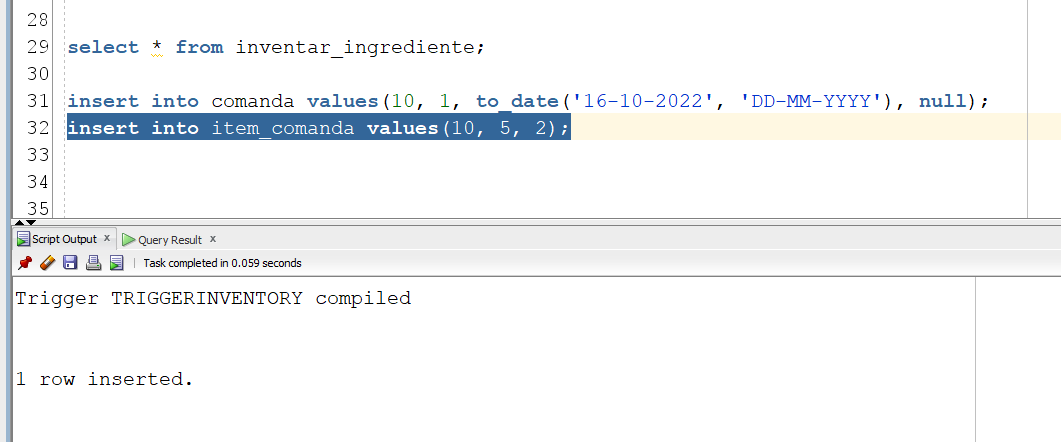


11.  
-- Creati un trigger care, odata introdusa o cafea ca membra a unei comenzi,  
-- scade din inventarul ingredientelor pe care le contine cantitatile   
-- respective.  
   
**create** **or replace** **trigger** triggerInventory  
    **after** **insert** **on** item\_comanda  
    **for** **each** **row**  
**declare**  
    **cursor** c(id\_caf cafea.id\_cafea%type) **is** **select**   
    id\_ingredient, cantitate **from** contine\_ingrediente ci  
    **where** ci.id\_cafea = id\_caf;   
    c\_d inventar\_ingrediente.cantitate\_disponibila%type;  
**begin**  
    **for** i **in** c(:new.id\_cafea) **loop**  
        **select** cantitate\_disponibila **into** c\_d  
        **from** inventar\_ingrediente  
        **where** id\_ingredient = i.id\_ingredient;  
        **if** c\_d < :new.cantitate \* i.cantitate **then**  
            **rollback**;  
            RAISE\_APPLICATION\_ERROR (-20010, 'Nu sunt destule ingrediente pentru prepararea comenzii');  
        **else**  
            **update** inventar\_ingrediente  
            **set** cantitate\_disponibila = cantitate\_disponibila - :new.cantitate \* i.cantitate  
            **where** id\_ingredient = i.id\_ingredient;  
        **end** **if**;  
    **end** **loop**;  
**end**;

**select** \* **from** inventar\_ingrediente;  
   
**insert** **into** comanda **values**(10, 1, **to\_date**('16-10-2022', 'DD-MM-YYYY'), null);  
**insert** **into** item\_comanda **values**(10, 5, 2);

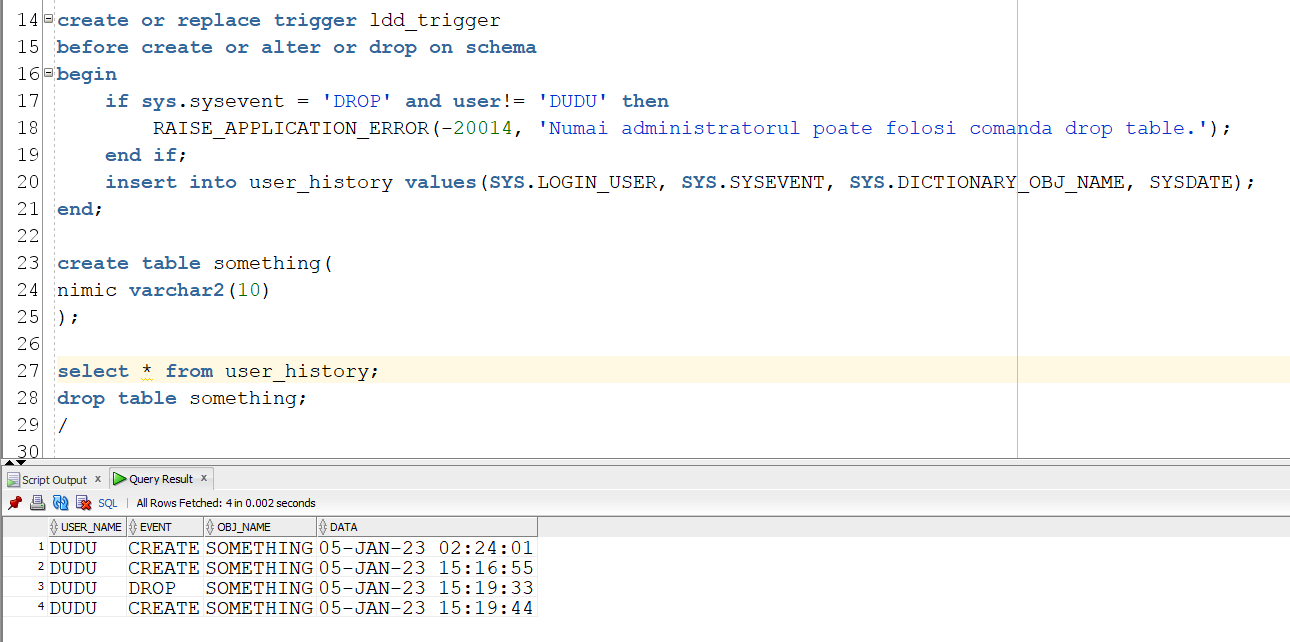






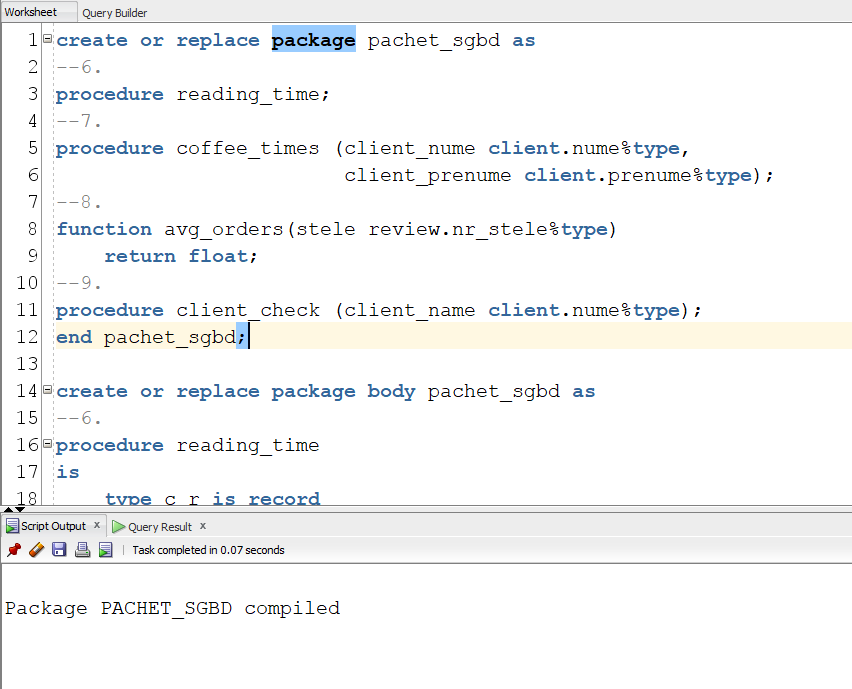
12.  
   
--12.

--Definiti un trigger de tip LDD care sa retina intr-un tabel auxiliar istoricul  
--comenzilor de tip LDD apelate de catre un user. Totodata, interziceti stergerea  
--tabelelor de catre oricare alt utilizator decat adminul.  
**create** **table** user\_history  
(  
user\_name varchar2(20),  
**event**      varchar2(20),    
obj\_name varchar2(20),  
**data**      date       
);  
   
**create** **or** **replace** **trigger** ldd\_trigger  
**before** **create** **or** **alter** **or** **drop** **on** **schema**  
**begin**  
    **if** sys.sysevent = 'DROP' **and** **user**!= 'DUDU' **then**  
        RAISE\_APPLICATION\_ERROR(-20014, 'Numai administratorul poate folosi comanda drop table.');  
    **end** **if**;  
    **insert** **into** user\_history **values**(SYS.LOGIN\_USER, SYS.SYSEVENT, SYS.DICTIONARY\_OBJ\_NAME, **SYSDATE**);  
**end**;  
   
**create** **table** something(  
nimic varchar2(10)  
);  
   
**select** \* **from** user\_history;  
**drop** **table** something;  
/



13.

**create** **or replace** package pachet\_sgbd **as**  
--6.  
**procedure** reading\_time;  
--7.  
**procedure** coffee\_times (client\_nume client.nume%type,   
                        client\_prenume client.prenume%type);  
--8.  
**function** avg\_orders(stele review.nr\_stele%type)   
    **return** float;  
--9.  
**procedure** client\_check (client\_name client.nume%type);  
**end** pachet\_sgbd;



**create** **or replace** package body pachet\_sgbd **as**  
--6.  
**procedure** reading\_time  
**is**  
    **type** c\_r **is** record  
    (  
    client\_id client.id\_client%type,  
    readingTime float --in ore  
    );  
    **type** hours\_read **is** **table** **of** c\_r;  --tablou imbricat  
    client\_read\_time hours\_read;  
    **type** tr **is** **table** **of** c\_r **index** **by** pls\_integer; --tablou indexat   
    totalReading tr;  
    maxReadingTime FLOAT := 0;  
    idClientMax number(8);  
    numeClient client.nume%type;  
    prenumeClient client.prenume%type;  
    idClient client.id\_client%type;  
**begin**  
    **select** id\_client, (data\_restituire - data\_imprumut)\*24  
    bulk collect **into** client\_read\_time  
    **from** imprumut;  
      
    **for** i **in** client\_read\_time.first..client\_read\_time.last **loop**  
        idClient:= client\_read\_time(i).client\_id;  
        totalReading(idClient).client\_id := client\_read\_time(i).client\_id;  
        totalReading(idClient).readingTime := 0;  
    **end** **loop**;  
   
    **for** i **in** client\_read\_time.first..client\_read\_time.last **loop**  
        idClient:= client\_read\_time(i).client\_id;  
        totalReading(idClient).readingTime := totalReading(idClient).readingTime + client\_read\_time(i).readingTime;  
          
        **if** totalReading(idClient).readingTime > maxReadingTime **then**  
            maxReadingTime := totalReading(idClient).readingTime;  
            idClientMax := idClient;  
        **end** **if**;  
    **end** **loop**;  
      
    **select** c.nume, c.prenume **into** numeClient, prenumeClient  
    **from** client c  
    **where** idClientMax = c.id\_client;  
      
    dbms\_output.put\_line('Clientul ' || numeClient || ' ' || prenumeClient ||' a citit cel mai mult timp' ||  
    ', in total ' || trunc(maxReadingTime,2) || ' ore.');  
**end**;  
--7.  
**procedure** coffee\_times  
(client\_nume client.nume%type,  
client\_prenume client.prenume%type)  
**is**   
    **cursor** c **is**   
    **select** id\_comanda **from** comanda  
    **where** id\_client = (**select** id\_client **from** client   
    **where** upper(nume) = upper(client\_nume)   
    **and** upper(prenume) = upper(client\_prenume));  
      
    **cursor** c1(id\_com comanda.id\_comanda%type) **is**   
    **select** id\_cafea, cantitate **from** item\_comanda  
    **where** id\_comanda = id\_com;  
      
    **cursor** c2(idcaf cafea.id\_cafea%type)  
    **is** **select** id\_ingredient, cantitate  
    **from** contine\_ingrediente **where** id\_cafea = idcaf;  
      
    cafeaCantitate c1%rowtype;  
    ingredientCantitate c2%rowtype;  
      
    **type** ingred **is** record  
    (  
    id\_ing ingredient.id\_ingredient%type,  
    cantitate contine\_ingrediente.cantitate%type  
    );  
    **type** ingTab **is** **table** **of** ingred;  
    ingredientsTable ingTab := ingTab();  
    nringred number(4);  
    ingredientName ingredient.nume\_ingredient%type;  
    nrclienti number(4);  
    notfoundexc **exception**;  
**begin**  
    **select** count(id\_client) **into** nrclienti **from** client   
    **where** upper(nume) = upper(client\_nume)   
    **and** upper(prenume) = upper(client\_prenume);  
      
    **if** nrclienti = 0 **then**  
        **raise** notfoundexc;   
    **end** **if**;  
   
    **select** count(\*) **into** nringred   
    **from** ingredient;  
    ingredientsTable.extend(nringred+1);  
    **for** i **in** 1..nringred **loop**  
        ingredientsTable(i).id\_ing := i;  
        ingredientsTable(i).cantitate := 0;  
    **end** **loop**;  
    **for** comanda\_entry **in** c **loop** --ciclu cursor  
        **open** c1(comanda\_entry.id\_comanda);  
        **loop**  
            **fetch** c1 **into** cafeaCantitate; --cursor clasic  
            **exit** **when** c1%NOTFOUND;  
                **for** ingredientCantitate **in** c2(cafeaCantitate.id\_cafea) **loop**  
                    --acum am: in comanda\_entry Id comanda  
                    --in cafeaCantitate id\_cafea si cantitatea cafelei  
                    --in ingredientCantitate id\_ingredient si cantitatea ingredientului  
                    ingredientsTable(ingredientCantitate.id\_ingredient).cantitate :=   
                    ingredientsTable(ingredientCantitate.id\_ingredient).cantitate +   
                    cafeaCantitate.cantitate \* ingredientCantitate.cantitate;  
                **end** **loop**;  
        **end** **loop**;  
        **close** c1;  
    **end** **loop**;  
    dbms\_output.put\_line('Clientul cu numele ' || client\_nume || ' ' || client\_prenume || ' a comandat:');  
    **for** i **in** ingredientsTable.first..ingredientsTable.last **loop**  
        **select** nume\_ingredient **into** ingredientName   
        **from** ingredient  
        **where** id\_ingredient = i;  
        dbms\_output.put\_line(ingredientsTable(i).cantitate || ' ' || ingredientName);  
    **end** **loop**;  
    **exception**  
    **when** notfoundexc **then**  
    dbms\_output.put\_line('Clientul nu exista.');  
    **when** others **then**  
    dbms\_output.put\_line('');  
**end**;  
   
--8.  
**function** avg\_orders(stele review.nr\_stele%type)  
**return** float **is**   
no\_reviews **EXCEPTION**;  
PRAGMA exception\_init(no\_reviews, -20001);  
   
no\_descriptions **EXCEPTION**;  
PRAGMA exception\_init(no\_descriptions, -20002);  
   
orderNumber float;  
**type** descriptionAndCount **is** record  
(  
id\_client client.id\_client%type,  
countOfOrders number(6),  
review\_description review.descriptie%type  
);  
**type** descCount **is** **table** **of** descriptionAndCount **index** **by** pls\_integer;  
reviewTable descCount;   
numberOfOrders number(4) := 0;  
numberOfClients number(4) :=0;  
numberOfDescriptions number(4) :=0;  
**begin**  
    **select** count(\*) **into** numberOfClients  
    **from** review  
    **where** nr\_stele = stele;  
      
    **if** numberOfClients =0 **then**  
        raise\_application\_error(-20001, '');  
    **end** **if**;  
    numberOfClients := 0;  
    **select** max(c.id\_client), count(com.id\_comanda), max(r.descriptie) bulk collect **into** reviewTable  
    **from** review r   
    **join** client c   
    **on** r.id\_client = c.id\_client  
    **join** comanda com   
    **on** com.id\_client = c.id\_client  
    **where** r.nr\_stele = stele  
    **group** **by** c.id\_client;  
    dbms\_output.put\_line('Descrieri review-uri:');  
    **for** i **in** reviewTable.first..reviewTable.last **loop**  
       numberOfOrders := numberOfOrders + reviewTable(i).countOfOrders;  
       numberOfClients := numberOfClients +1;  
       **if** reviewTable(i).review\_description **is** **not** **null** **then**  
       dbms\_output.put\_line(reviewTable(i).review\_description);  
       numberOfDescriptions := numberOfDescriptions + 1;  
       **end** **if**;  
    **end** **loop**;  
    **if** numberOfDescriptions =0 **then**  
    raise\_application\_error(-20002,'');  
    **end** **if**;  
    orderNumber := numberOfOrders \* 1.0 / numberOfClients;  
    **return** trunc(orderNumber,3);  
    **exception**  
    **when** no\_reviews **then**  
    dbms\_output.put\_line('Niciun review cu numarul de stele specificat');  
    **return** -1;  
    **when** no\_descriptions **then**  
    dbms\_output.put\_line('Niciun review cu numarul de stele specificat care sa aiba descriere');  
    **return** -1;  
**end** avg\_orders;  
   
--9.  
**procedure** client\_check  
(client\_name client.nume%type)  
**is**  
    **cursor** c1(cid client.id\_client%type) **is**   
    **select** c.id\_client, caf.tip\_cafea, g.nume\_gen, com.data\_comanda, imp.data\_imprumut  
    **from** client c   
    **join** comanda com   
    **on** com.id\_client = c.id\_client  
    **join** item\_comanda ic  
    **on** ic.id\_comanda = com.id\_comanda  
    **join** cafea caf  
    **on** caf.id\_cafea = ic.id\_cafea  
    **join** imprumut imp   
    **on** imp.id\_client = c.id\_client  
    **join** carte car   
    **on** car.id\_carte = imp.id\_carte  
    **join** gen g  
    **on** g.id\_gen = car.id\_gen  
    **where** id\_client = cid;  
    clientId client.id\_client%type;   
    ok boolean := **false**;  
**begin**  
    **select** id\_client **into** clientId  
    **from** client c  
    **where** upper(c.nume) = upper(client\_name);  
      
    **for** fullElem **in** c1(clientId) **loop**  
        **if** fullElem.tip\_cafea = 'Venti' **and** fullElem.nume\_gen = 'Actiune' **and**   
        trunc(fullElem.data\_comanda) = trunc(fullElem.data\_imprumut) **then**  
        ok := **true**;  
        **end** **if**;  
    **end** **loop**;  
      
    **if** ok = **true** **then**  
        dbms\_output.put\_line('Clientul ' || client\_name || ' indeplineste criteriile.');  
    **else**  
        dbms\_output.put\_line('Clientul ' || client\_name || ' nu indeplineste criteriile.');  
    **end** **if**;  
    **exception**  
        **when** no\_data\_found **then**  
        dbms\_output.put\_line('Niciun client cu numele specificat.');  
        **when** too\_many\_rows **then**  
        dbms\_output.put\_line('Mai multi clienti au numele specificat.');  
**end**;  
**end** pachet\_sgbd;

