**PROIECT**

**SISTEME DE GESTIUNE A BAZELOR DE DATE**

Restaurant

Moisii Lucian 241

**CUPRINS**

1. Prezentarea bazei de date
2. Diagrama entitate-relație (ERD)
3. Diagrama conceptuală
4. Implementarea tabelelor în Oracle
5. Inserarea datelor în tabele
6. Subprogram stocat care utilizează două tipuri de colecții
7. Subprogram stocat care utilizează doua tipuri de cursoare
8. Subprogram stocat de tip funcț ie care să utilizeze într-o singură comandă SQL 3 tabele definite
9. Subprogram stocat de tip procedură care să utilizeze într-o singură comandă de tip SQL 5 dintre tabelele definite
10. Definiți un trigger de tip LMD la nivel de comandă
11. Definiți un trigger de tip LMD la nivel de linie
12. Definiți un trigger de tip LDD

**1. Prezentarea bazei de date**

Acest proiect cuprinde proiectarea unei baze de date și rezolvarea unor probleme pe aceasta.

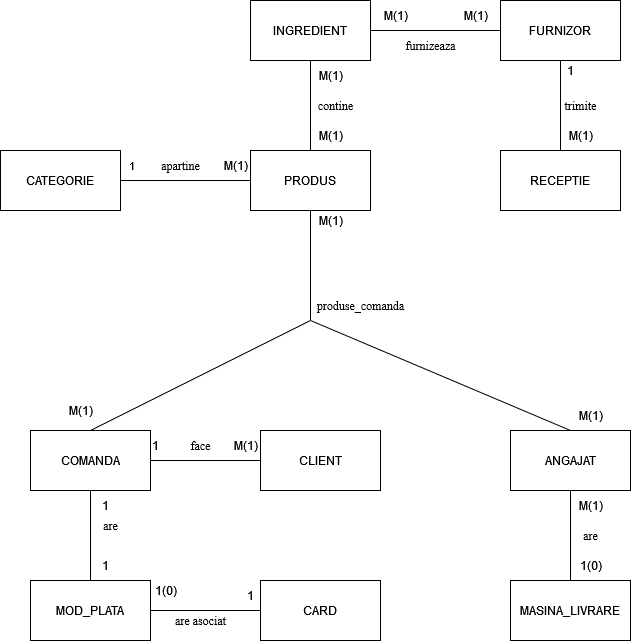
Tema aleasă pentru această bază de date are în centrul său gestiunea unui restaurant, în cadrul căreia sunt reținute informații despre personalul, comenzile, clienții, meniul, furnizorii și livrarea produselor ce aparțin acestei afaceri.

Restaurantul are o locație fizică unde se pot efectua comenzi ce vor urma a fi servite în locație, la pachet sau livrate. De asemenea, restaurantul dispune și de o platformă online prin cadrul căreia clienții pot da comandă.

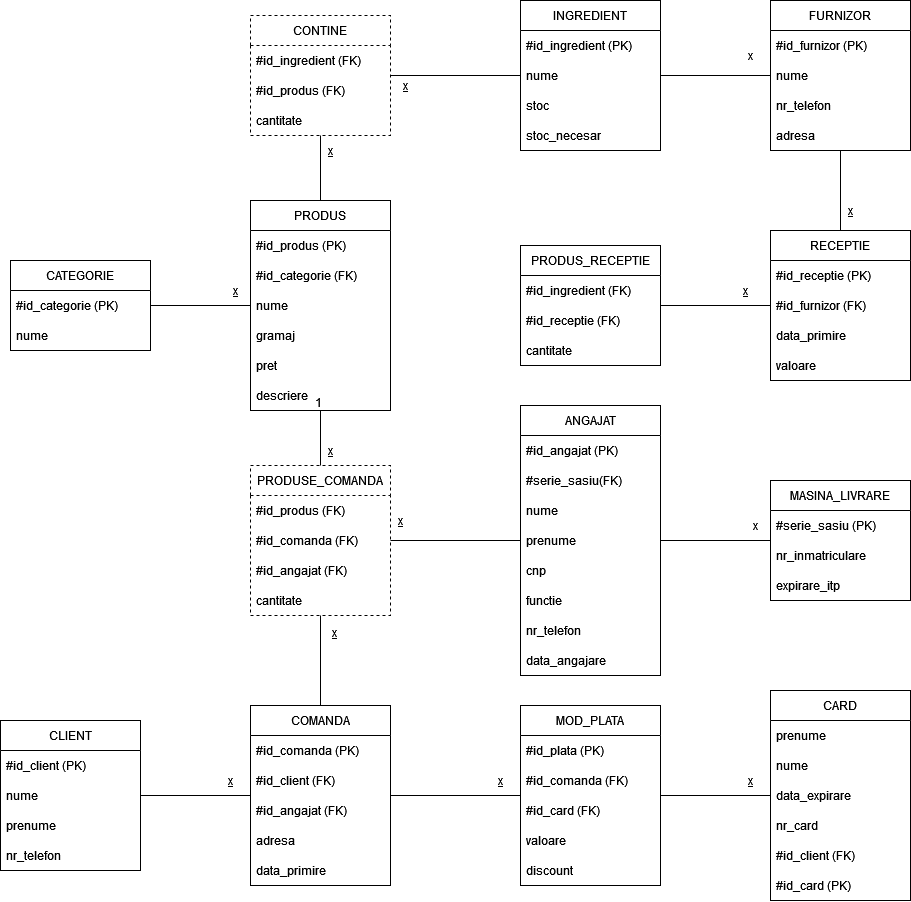
Personalul restaurantului este format din șef (sau șefi daca sunt mai multe persoane care dețin împreuna restaurantul), manageri, bucătari-șefi, bucătari, ajutor de bucătar, pizzeri, ajutor de pizzeri, barmani, ajutor de barman, ospătari și livratori.

Furnizorii restaurantului sunt diferite firme locale sau lanțuri mari de magazinee prin care acesta se aprovizioneaza.

**2. Diagrama entitate-relație (ERD)**

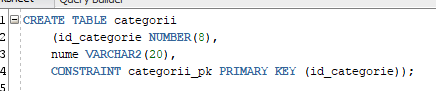
****

**3. Diagrama conceptuală**

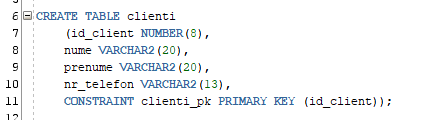
****

**4. Implementarea tabelelor in SQL**

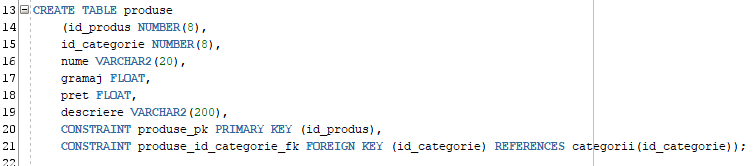
1. Crearea tabelului CATEGORII:



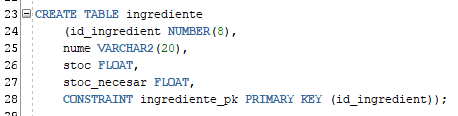
1. Crearea tabelului CLIENTI:



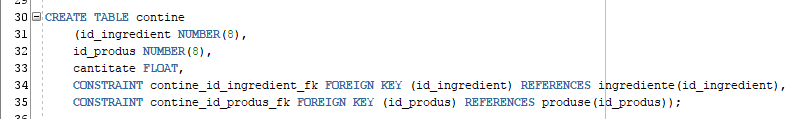
1. Crearea tabelului PRODUSE:



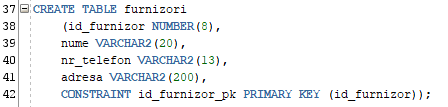
1. Crearea tabelului INGREDIENTE:



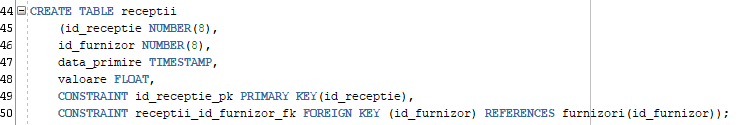
1. Crearea tabelului CONTINE:



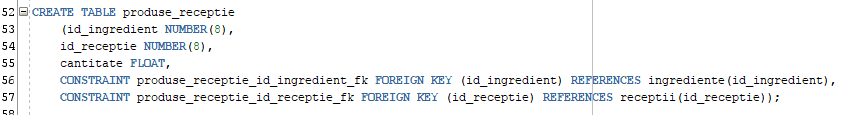
1. Crearea tabelului FURNIZORI:



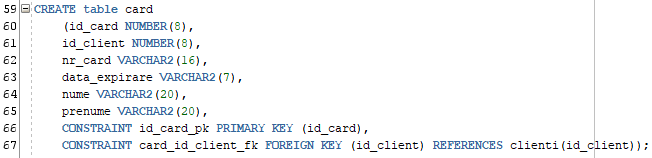
1. Crearea tabelului RECEPTII:



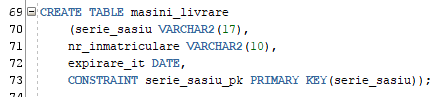
1. Crearea tabelului PRODUSE\_RECEPTIE:



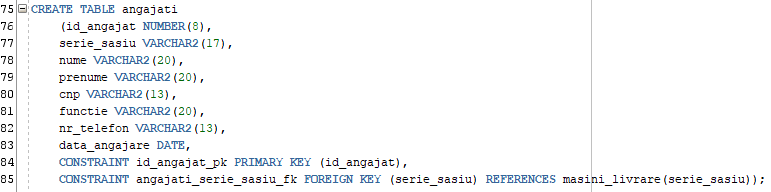
1. Crearea tabelului CARD:



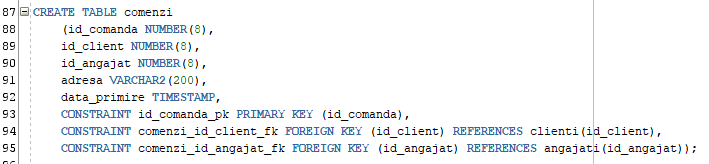
1. Crearea tabelului MASINI\_LIVRARE:



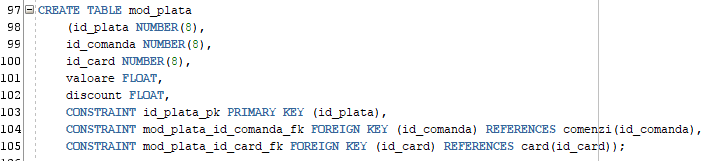
1. Crearea tabelului ANGAJATI:



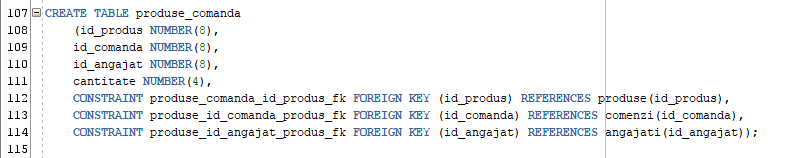
1. Crearea tabelului COMENZI:



1. Crearea tabelului MOD\_PLATA:

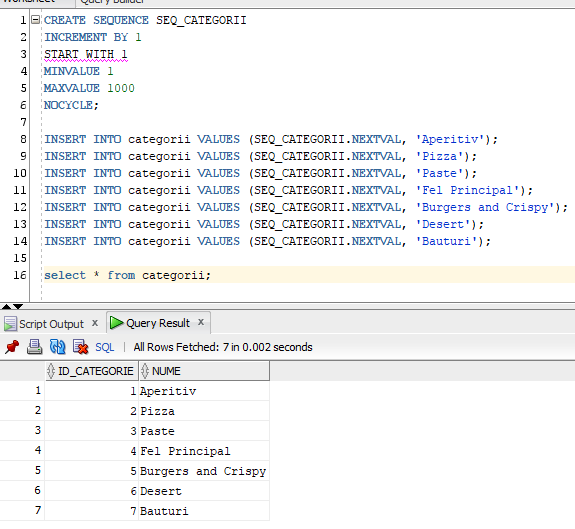


1. Crearea tabelului PRODUSE\_COMANDA:

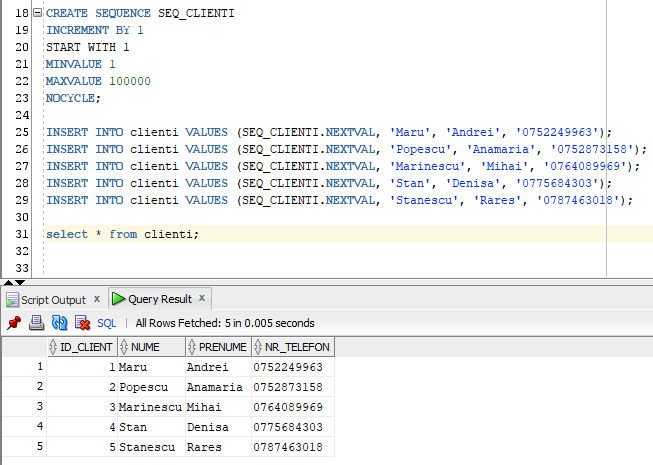


**5. Inserarea datelor în tabele**

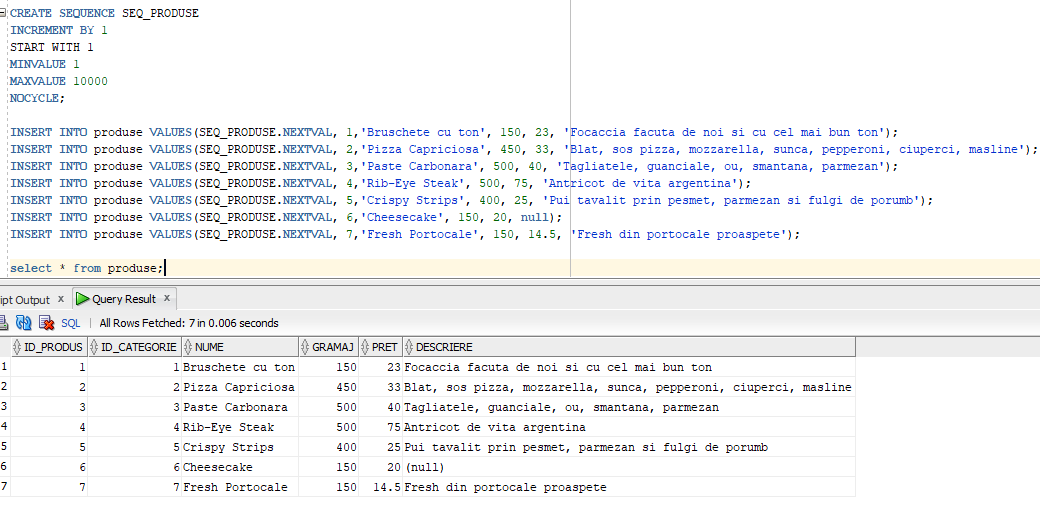
1. Inserarea datelor în tabelul CATEGORII:

****

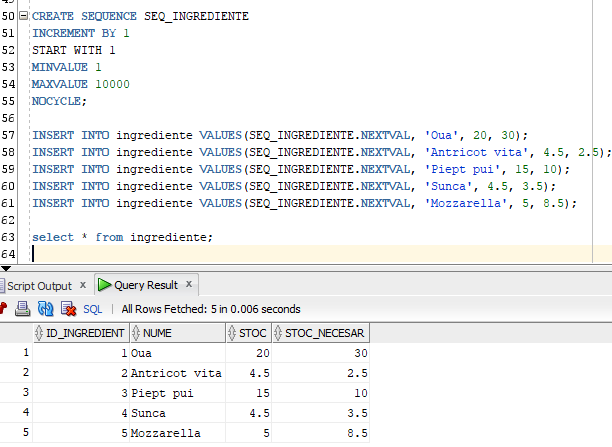
2. Inserarea datelor în tabelul CLIENTI:

****

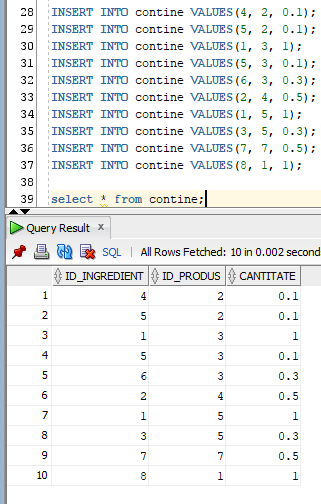
3. Inserarea datelor în tabelul PRODUSE:

****

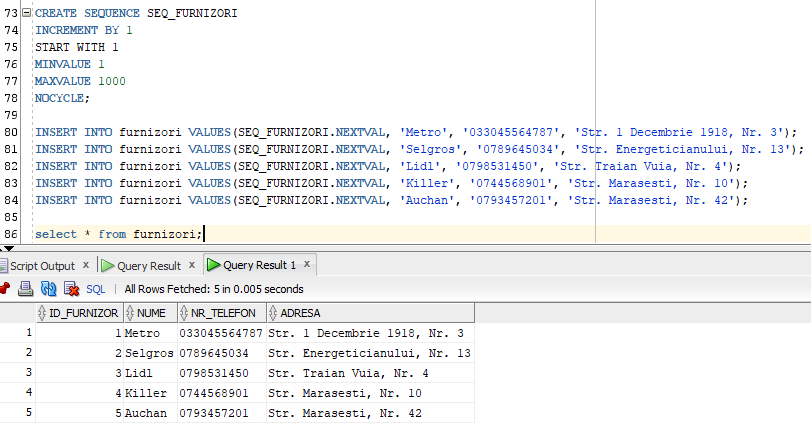
4. Inserarea datelor în tabelul INGREDIENTE:

****

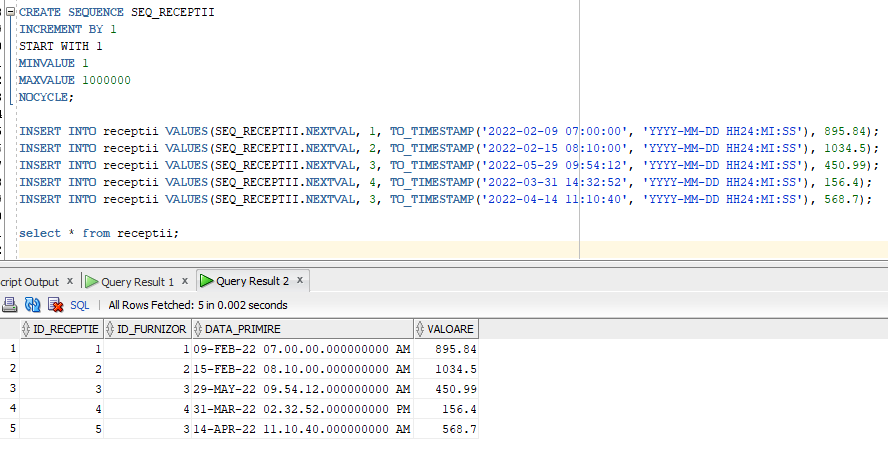
5. Inserarea datelor în tabelul CONTINE:

****

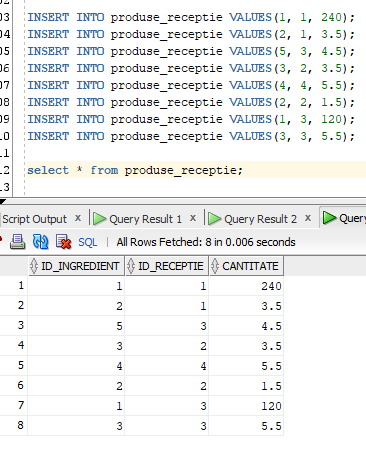
6. Inserarea datelor în tabelul FURNIZORI:

****

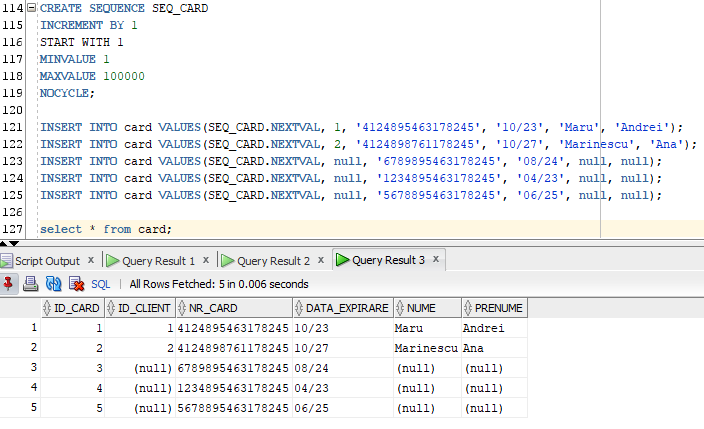
7. Inserarea datelor în tabelul RECEPTII:

****

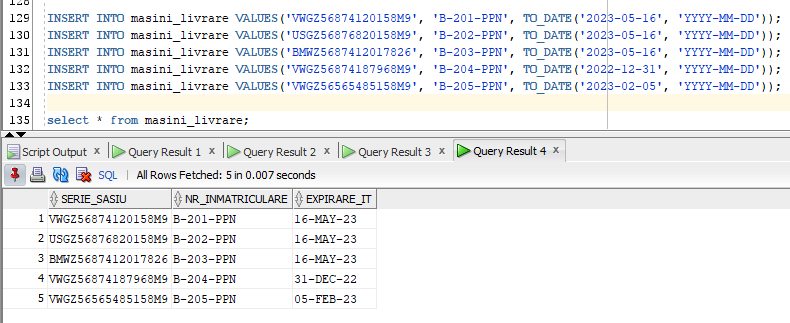
8. Inserarea datelor în tabelul PRODUSE\_RECEPTIE:

****

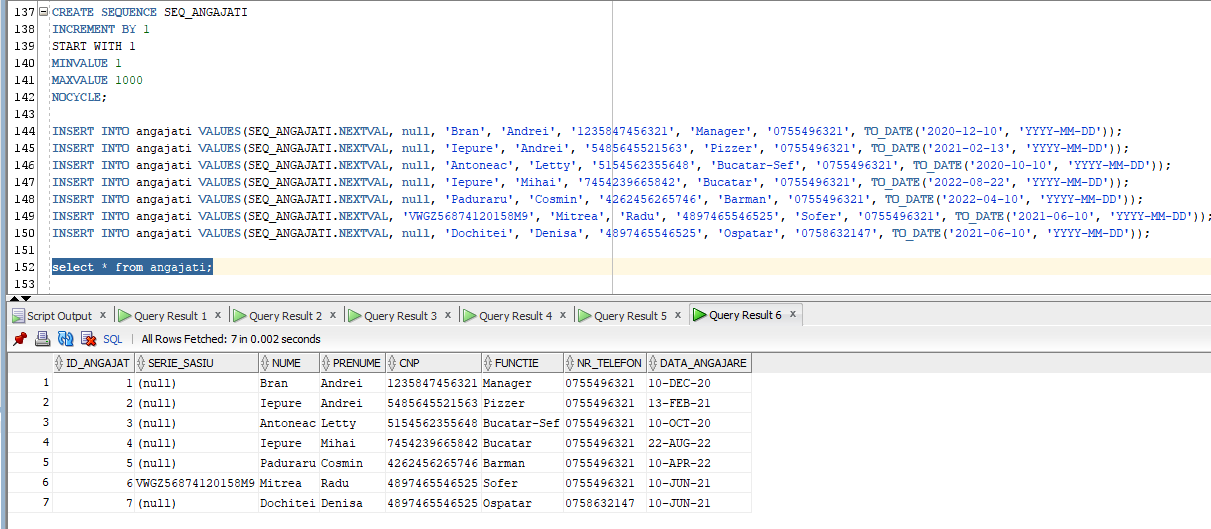
9. Inserarea datelor în tabelul CARD:

****

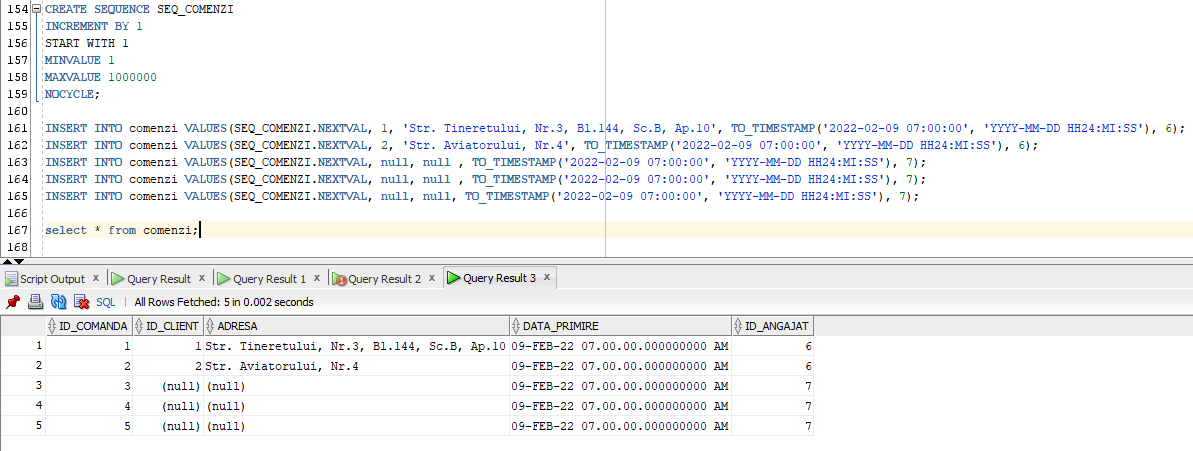
10. Inserarea datelor în tabelul MASINI\_LIVRARE:

****

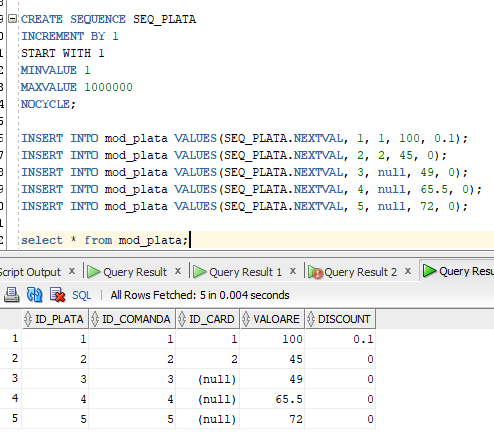
11. Inserarea datelor în tabelul ANGAJATI:

****

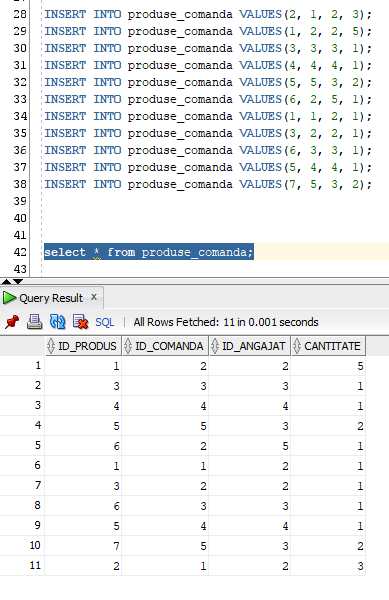
12. Inserarea datelor în tabelul COMENZI:

****

13. Inserarea datelor în tabelul MOD\_PLATA:

****

14. Inserarea datelor în tabelul PRODUSE\_COMANDA:

****

**6. Subprogram stocat care utilizează două tipuri de colecții**

*Scrieti o procedura ce afișează pentru fiecare categorie produsele din acea categorie, iar pentru fiecare produs ingredientele sale.*

CREATE OR REPLACE PROCEDURE ex\_6

IS

TYPE produs IS RECORD (id\_produs produse.id\_produs%TYPE, nume produse.nume%TYPE);

TYPE vec\_categorii IS TABLE OF categorii%ROWTYPE INDEX BY PLS\_INTEGER;

TYPE vec\_produse IS TABLE OF produs;

TYPE vec\_contine IS VARRAY(20) OF (ingrediente.id\_ingredient%TYPE);

v\_produse vec\_produse := vec\_produse();

v\_categorii vec\_categorii;

v\_contine vec\_contine := vec\_contine();

j INTEGER;

v\_nume ingrediente.nume%TYPE;

a NUMBER(4);

BEGIN

SELECT \* BULK COLLECT INTO v\_categorii FROM categorii;

FOR i IN v\_categorii.FIRST..v\_categorii.LAST LOOP

SELECT id\_produs, nume BULK COLLECT INTO v\_produse FROM produse WHERE id\_categorie = v\_categorii(i).id\_categorie;

DBMS\_OUTPUT.PUT\_LINE('Categoria ' || v\_categorii(i).nume);

j := v\_produse.FIRST;

WHILE j <= v\_produse.LAST LOOP

DBMS\_OUTPUT.PUT(v\_produse(j).nume || ' : ');

SELECT COUNT(\*) INTO a FROM contine WHERE id\_produs = v\_produse(j).id\_produs;

IF a > 0 THEN

a := a - 1;

SELECT id\_ingredient BULK COLLECT INTO v\_contine FROM contine WHERE v\_produse(j).id\_produs = id\_produs;

FOR k in v\_contine.FIRST..v\_contine.LAST LOOP

SELECT nume INTO v\_nume FROM ingrediente WHERE id\_ingredient = v\_contine(k);

DBMS\_OUTPUT.PUT(v\_nume);

IF a > 0 THEN

DBMS\_OUTPUT.PUT(', ');

a := a - 1;

END IF;

END LOOP;

ELSE

DBMS\_OUTPUT.PUT('Nu exista inca ingrediente introduse pentru acest produs');

END IF;

j := v\_produse.NEXT(j);

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

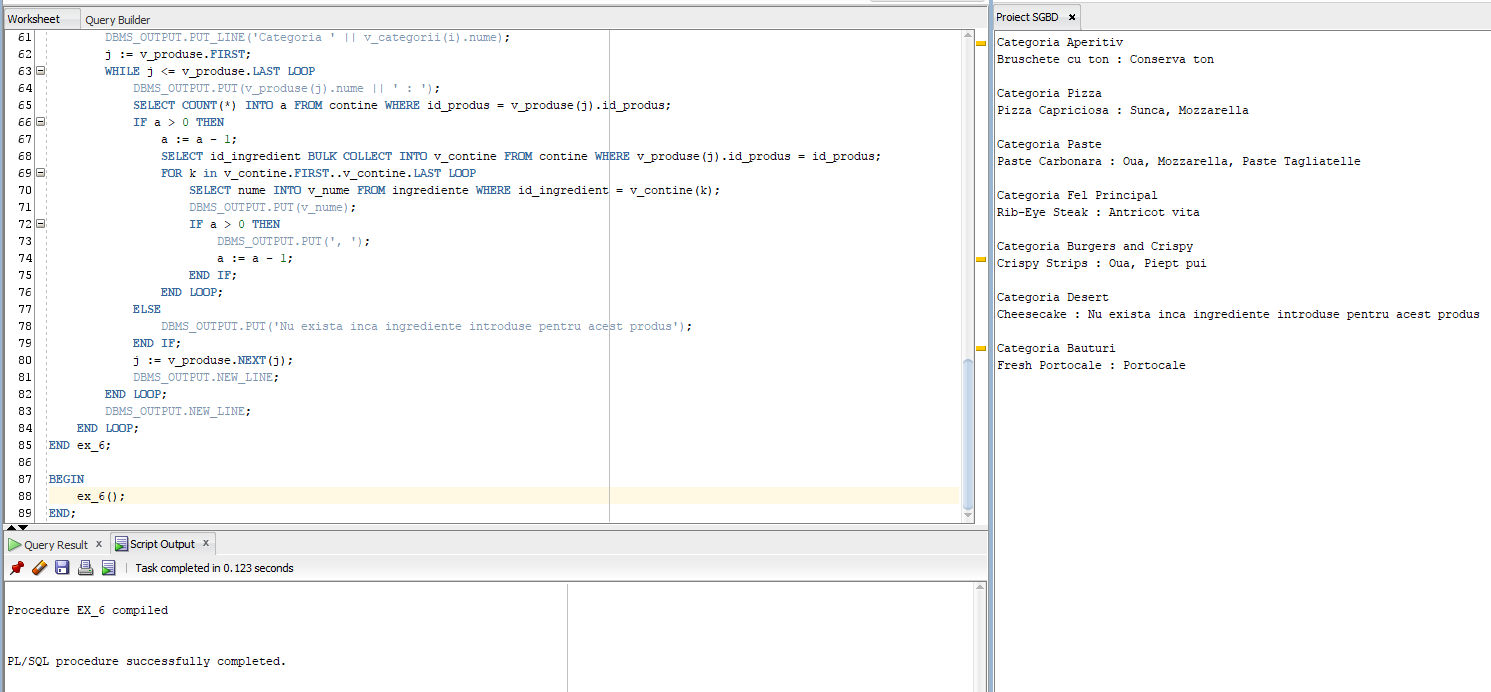
END LOOP;

END ex\_6;

BEGIN

ex\_6();

END;



**7. Subprogram stocat care utilizează doua tipuri de cursoare**

*Să se scrie o procedură ce primește ca parametri numele și prenumele unui client și afișeaza data fiecărei comenzi precum și produsele comandate.*

CREATE OR REPLACE PROCEDURE ex\_7

(p\_nume clienti.nume%TYPE,

p\_prenume clienti.prenume%TYPE)

IS

g\_id\_comanda comenzi.id\_comanda%TYPE;

g\_data\_primire comenzi.data\_primire%TYPE;

g\_nume\_produs produse.nume%TYPE;

g\_index NUMBER(4) := 1;

g\_comma BOOLEAN;

CURSOR c\_comenzi\_client

IS

SELECT id\_comanda, data\_primire FROM comenzi WHERE id\_client = (SELECT id\_client

FROM clienti

WHERE nume = p\_nume AND prenume = p\_prenume);

CURSOR c\_produse\_comanda (v\_id\_comanda comenzi.id\_comanda%TYPE)

IS

SELECT id\_produs FROM produse\_comanda WHERE id\_comanda = v\_id\_comanda;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Clientul cu numele ' || p\_nume || ' ' || p\_prenume || ' a dat comenzile:');

--cursor clasic

OPEN c\_comenzi\_client;

LOOP

FETCH c\_comenzi\_client INTO g\_id\_comanda, g\_data\_primire;

EXIT WHEN c\_comenzi\_client%NOTFOUND;

DBMS\_OUTPUT.PUT(g\_index || '. Comanda de la data de ' || TO\_CHAR(g\_data\_primire, 'YYYY-MM-DD HH24:MI:SS') || ' contine: ');

g\_comma := true;

--ciclu cursor cu parametru

FOR i IN c\_produse\_comanda(g\_id\_comanda) LOOP

SELECT nume INTO g\_nume\_produs FROM produse WHERE id\_produs = i.id\_produs;

IF g\_comma THEN

DBMS\_OUTPUT.PUT(g\_nume\_produs);

g\_comma := false;

ELSE

DBMS\_OUTPUT.PUT(', ' || g\_nume\_produs);

END IF;

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

g\_index := g\_index + 1;

END LOOP;

END ex\_7;

DECLARE

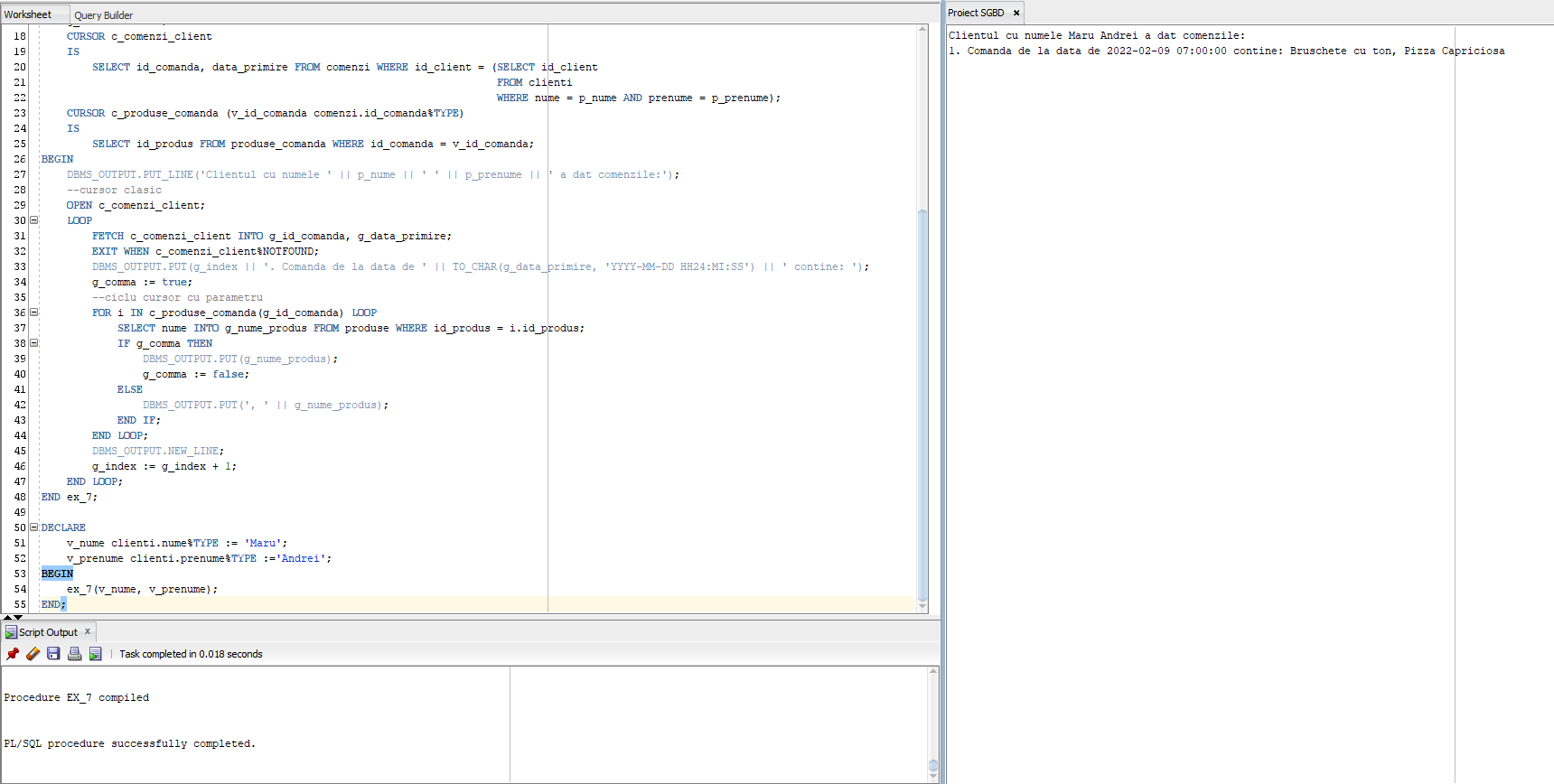
v\_nume clienti.nume%TYPE := 'Maru';

v\_prenume clienti.prenume%TYPE :='Andrei';

BEGIN

ex\_7(v\_nume, v\_prenume);

END;

****

**8. Subprogram stocat de tip funcție care să utilizeze într-o singură comandă SQL 3 tabele definite**

*Să se scrie o funcție care primește ca parametri numele, prenumele și data unei comenzi si returneaza valoarea comenzii. Tratați excepțiile necesare.*

CREATE OR REPLACE FUNCTION ex\_8 (f\_nume clienti.nume%TYPE,

f\_prenume clienti.prenume%TYPE,

f\_data comenzi.data\_primire%TYPE)

RETURN NUMBER

IS

TYPE price\_table\_datatype IS TABLE OF NUMBER(8, 2) INDEX BY PLS\_INTEGER;

price\_table price\_table\_datatype;

pret\_total NUMBER(8,2) := 0;

nr NUMBER(4);

v\_id\_client clienti.id\_client%TYPE;

zero\_clients\_exc EXCEPTION;

multiple\_clients\_exc EXCEPTION;

zero\_orders\_exc EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO nr FROM clienti WHERE nume = f\_nume AND prenume = f\_prenume;

IF nr = 0 THEN

RAISE zero\_clients\_exc;

ELSIF nr > 1 THEN

RAISE multiple\_clients\_exc;

END IF;

SELECT id\_client INTO v\_id\_client FROM clienti WHERE UPPER(nume) = UPPER(f\_nume) AND UPPER(prenume) = UPPER(f\_prenume);

SELECT COUNT(\*) INTO nr FROM comenzi WHERE id\_client = v\_id\_client AND data\_primire = f\_data;

IF nr = 0 THEN

RAISE zero\_orders\_exc;

END IF;

SELECT pc.cantitate \* p.pret BULK COLLECT INTO price\_table FROM produse\_comanda pc

JOIN comenzi c

ON c.id\_comanda = pc.id\_comanda

JOIN produse p

ON p.id\_produs = pc.id\_produs

WHERE c.id\_client = v\_id\_client AND c.data\_primire = f\_data;

FOR i IN price\_table.FIRST..price\_table.LAST LOOP

pret\_total := pret\_total + price\_table(i);

END LOOP;

RETURN NVL(pret\_total, 0);

EXCEPTION

WHEN zero\_clients\_exc THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Nu exista nici un client cu numele dat');

WHEN multiple\_clients\_exc THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Exista mai multi clienti cu numele dat');

WHEN zero\_orders\_exc THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Nu exista nici o comanda data de acest client la data respectiva');

WHEN value\_error THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Nu exista nici un produs in comanda data');

END ex\_8;

DECLARE

v\_nume clienti.nume%TYPE := 'Maru';

v\_prenume clienti.prenume%TYPE := 'Andrei';

v\_data comenzi.data\_primire%TYPE := '09-FEB-22 07.00.00.000000000 AM';

-- exemplu pentru eroarea -20001

-- v\_nume clienti.nume%TYPE := 'Maru';

-- v\_prenume clienti.prenume%TYPE := 'Marius';

-- v\_data comenzi.data\_primire%TYPE := '09-FEB-22 07.00.00.000000000 AM';

---- exemplu pentru eroarea -20002

-- v\_nume clienti.nume%TYPE := 'Stanescu';

-- v\_prenume clienti.prenume%TYPE := 'Rares';

-- v\_data comenzi.data\_primire%TYPE := '09-FEB-22 07.00.00.000000000 AM';

---- exemplu pentru eroarea -20003

-- v\_nume clienti.nume%TYPE := 'Maru';

-- v\_prenume clienti.prenume%TYPE := 'Andrei';

-- v\_data comenzi.data\_primire%TYPE := '09-FEB-22 08.20.00.000000000 AM';

---- exemplu pentru eroarea -20004

-- v\_nume clienti.nume%TYPE := 'Stan';

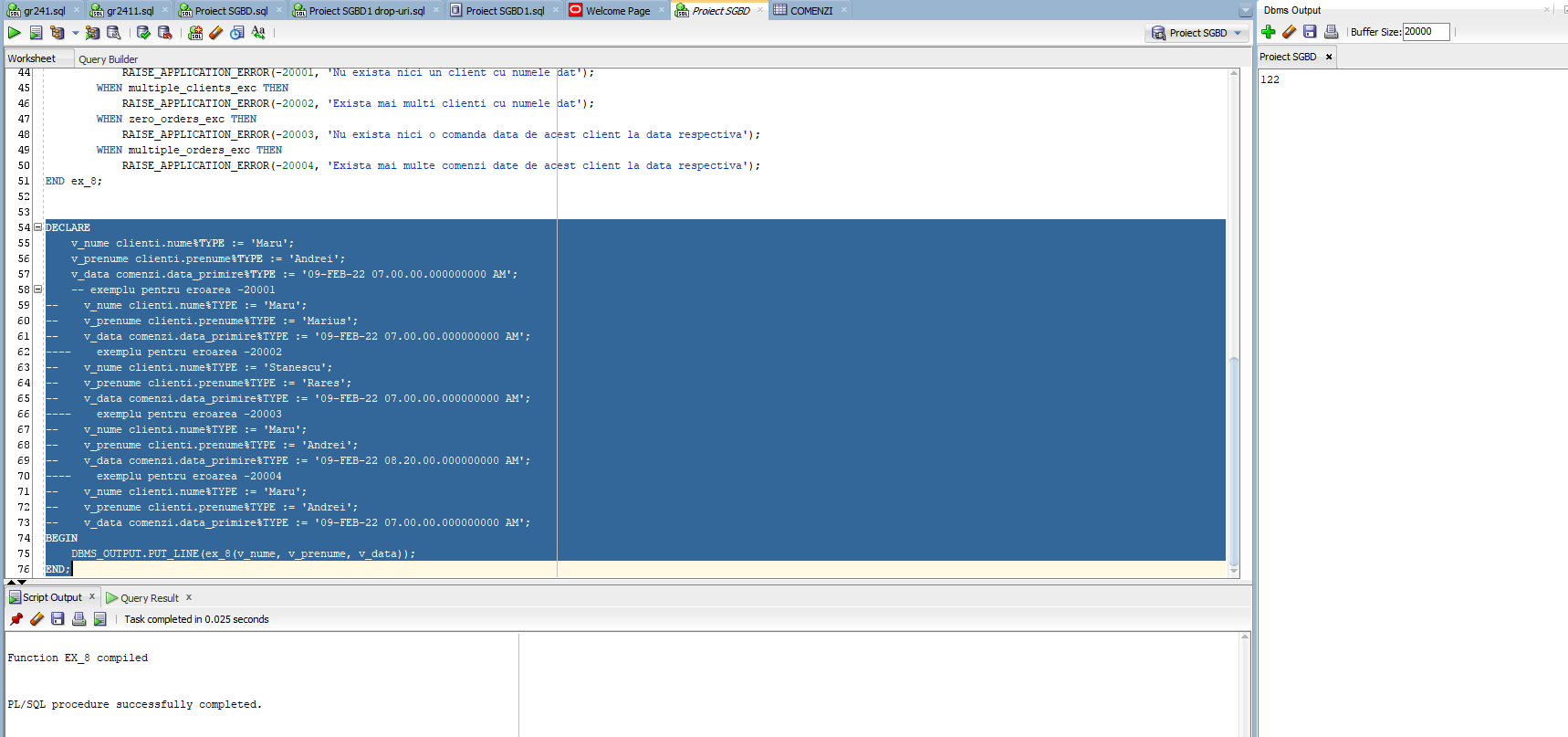
-- v\_prenume clienti.prenume%TYPE := 'Denisa';

-- v\_data comenzi.data\_primire%TYPE := '09-FEB-22 07.00.00.000000000 AM';

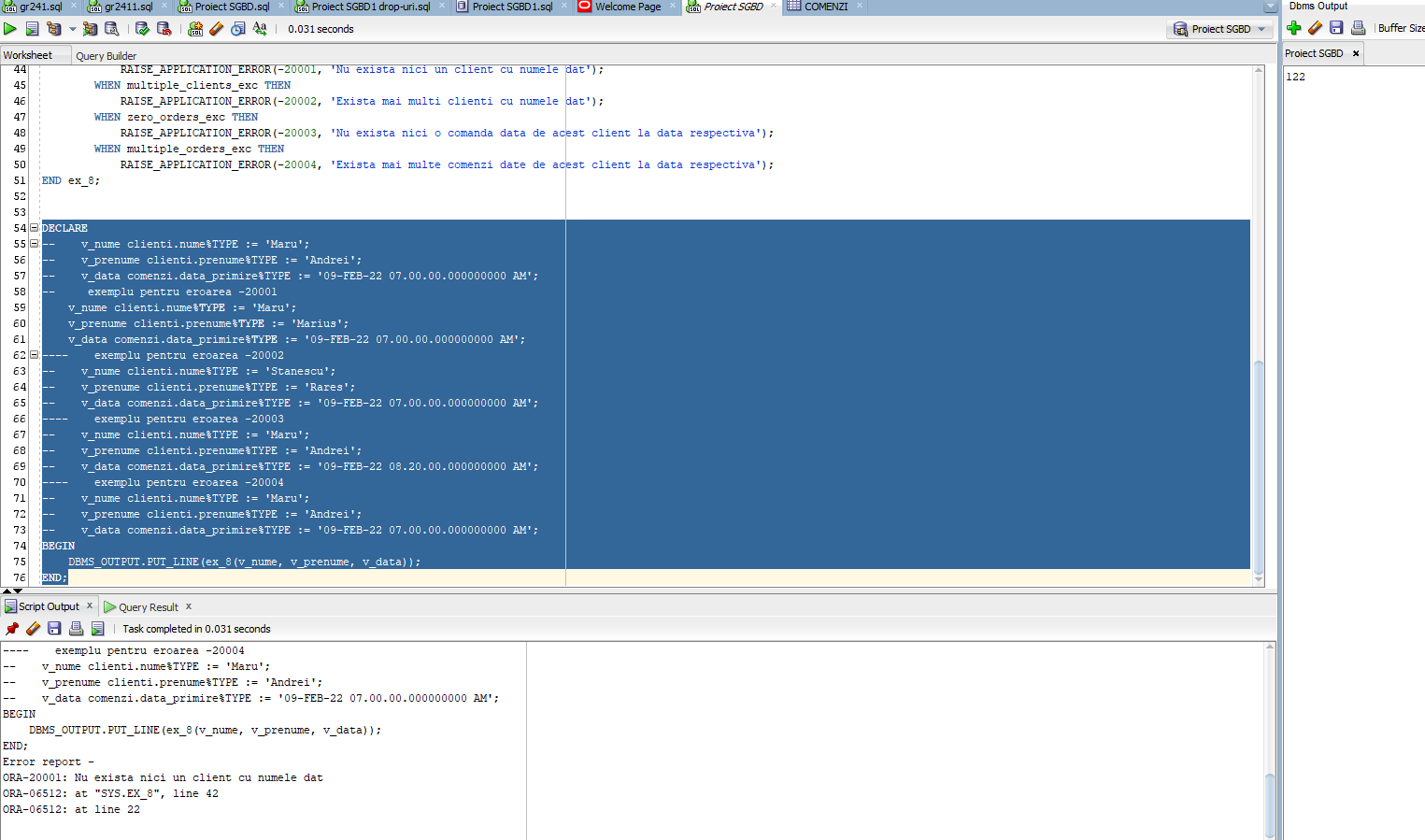
BEGIN

DBMS\_OUTPUT.PUT\_LINE(ex\_8(v\_nume, v\_prenume, v\_data));

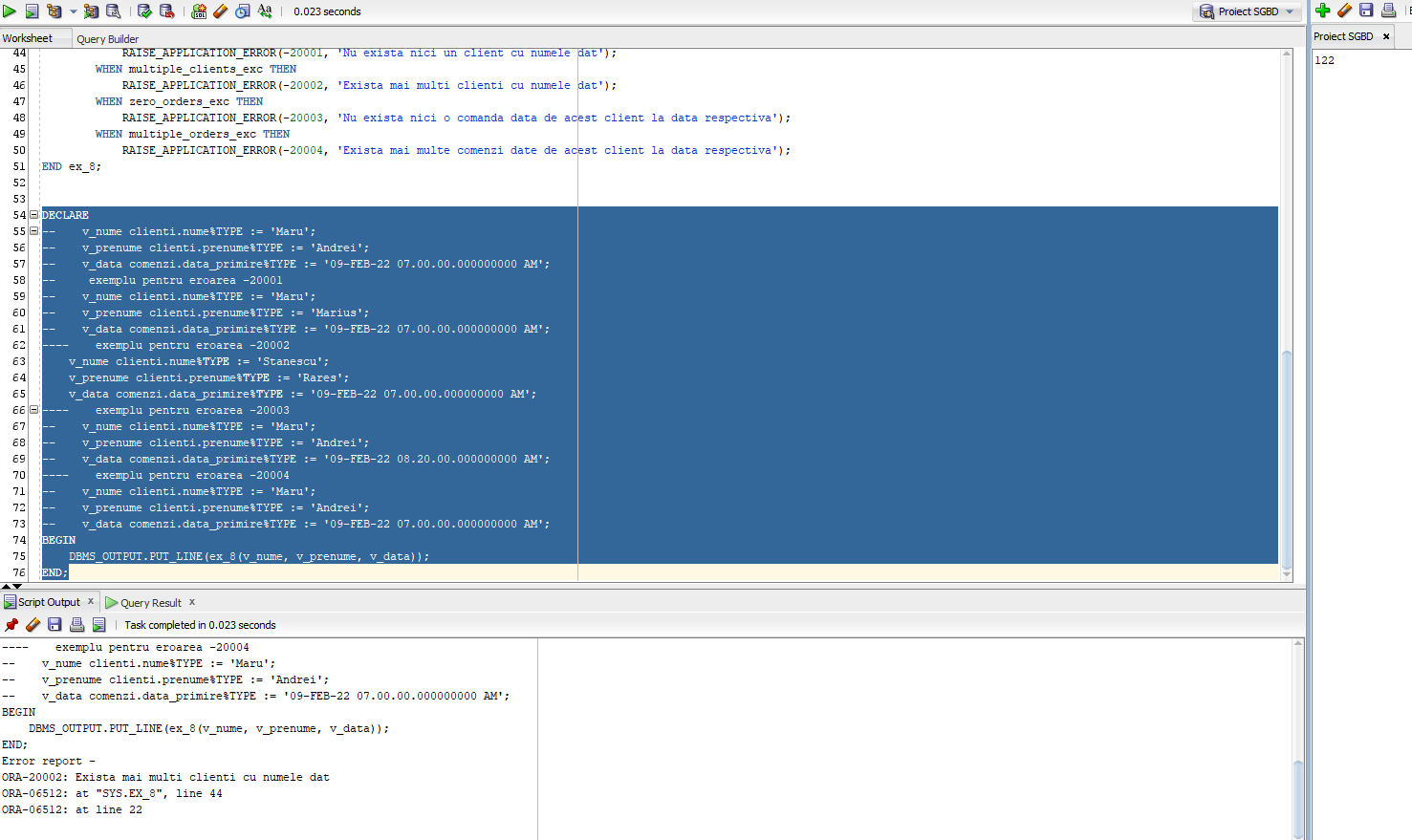
END;



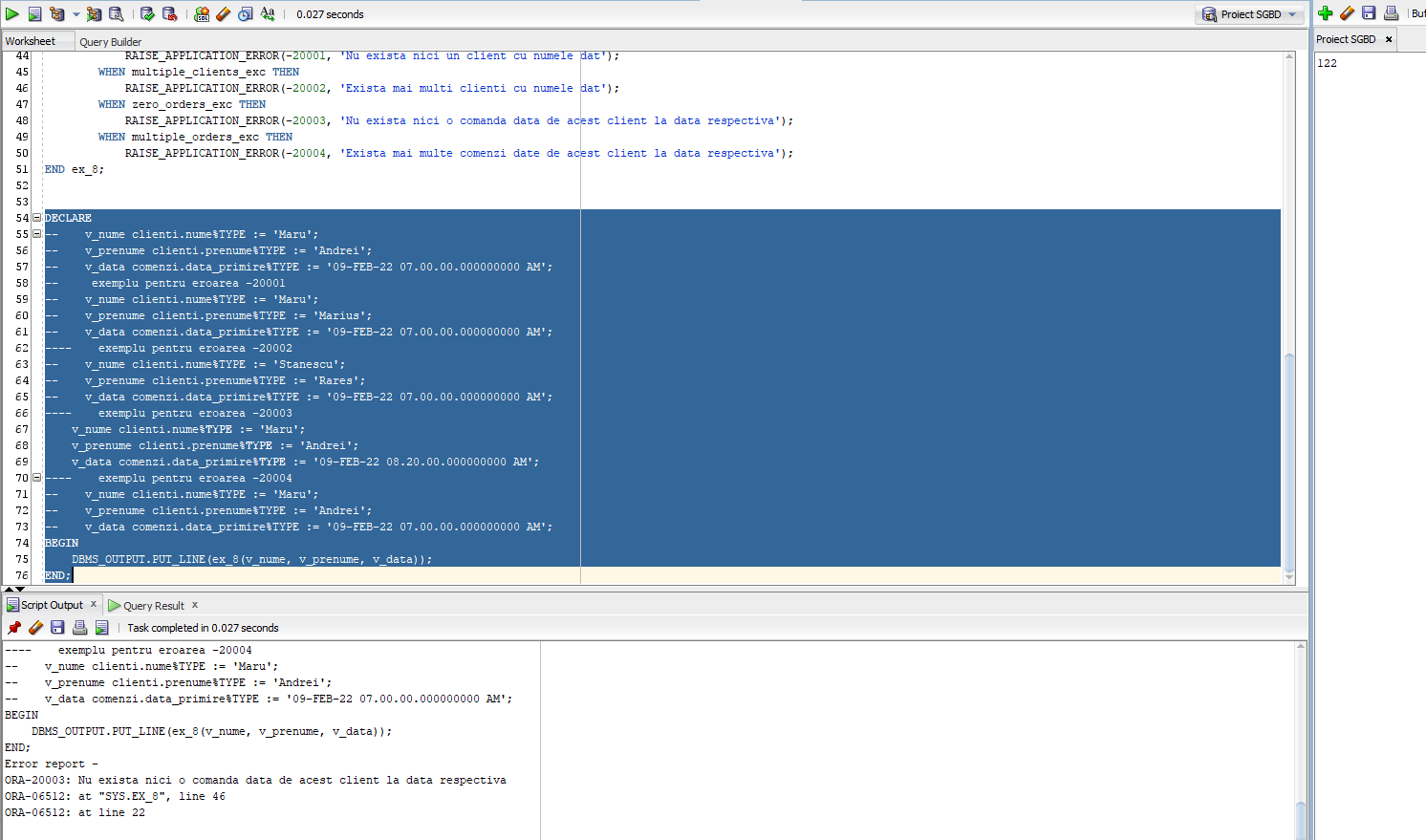
Exemplu pentru cazul în care nu există nici un client cu numele dat:



Exemplu pentru cazul în care există mai mulți clienți cu același nume:



Exemplu pentru cazul în care clientul nu a dat nici o comandă la ora specificată:



Exemplu pentru cazul în care nu există nici un produs în comanda respectivă:

