**Gestiunea unui lanț de restaurante**

Proiect

Sisteme de Gestiune a Bazelor de Date

Jăhăleanu Vlad-Gabriel

Grupa 232

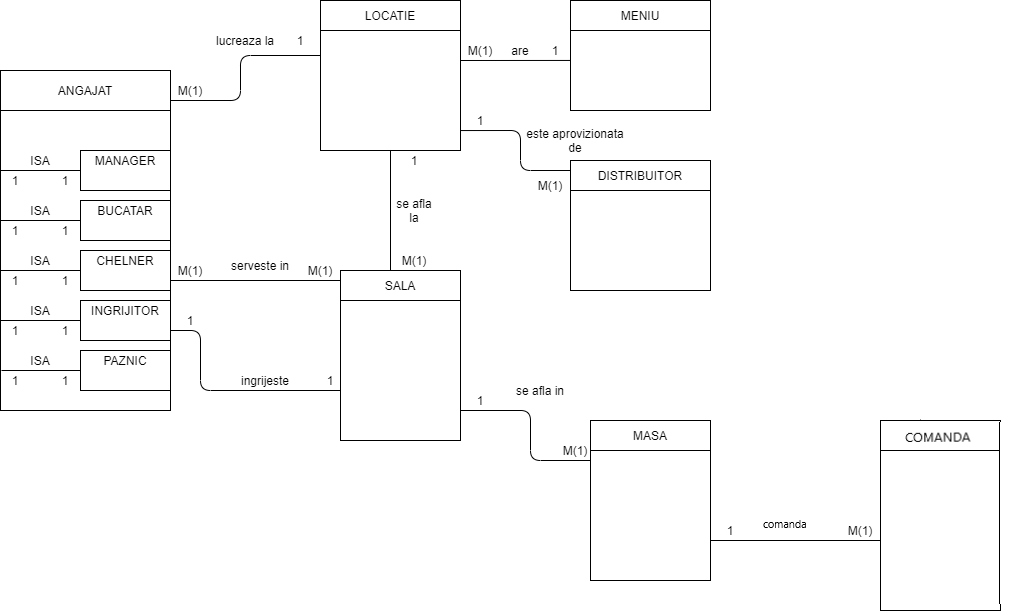
**1.Prezentarea bazei de date si utilitatea acesteia**

Modelul are scopul de a digitaliza functionarea unui lant de restaurante care apartine aceleiasi francize si, implicit, facilitand accesul la datele angajatilor, locatiilor si comenzilor.

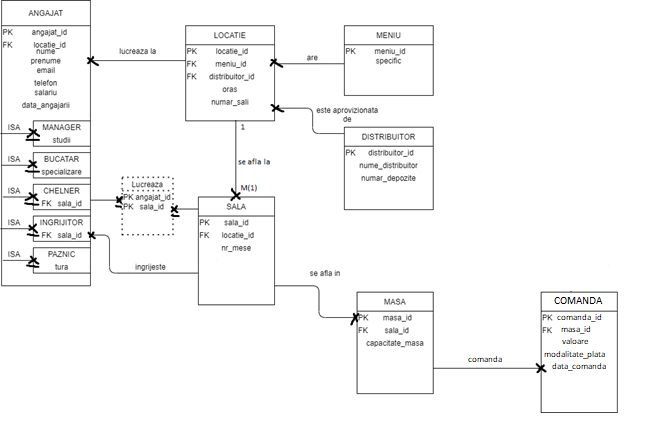
Am considerat locatiile ca fiind unice in fiecare oras.

O locatie poate avea mai multe sali, iar fiecare sala poate avea mai multe mese. La o masa se pot face mai multe comenzi, dar o comanda poate apartine unei singure mese.

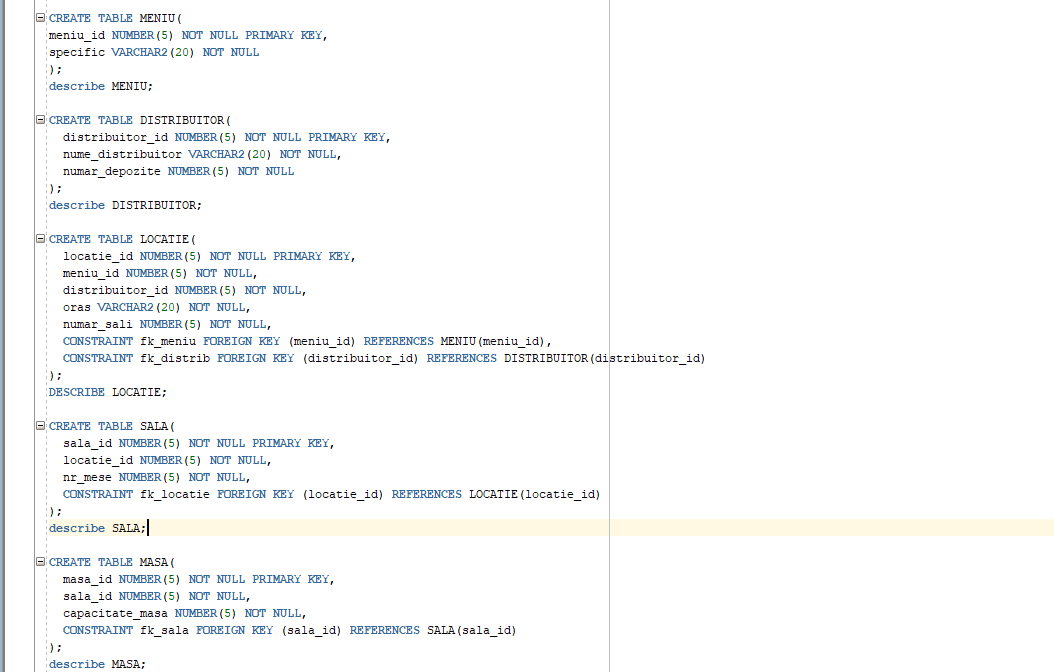
**2.Diagrama entitate-relatie**

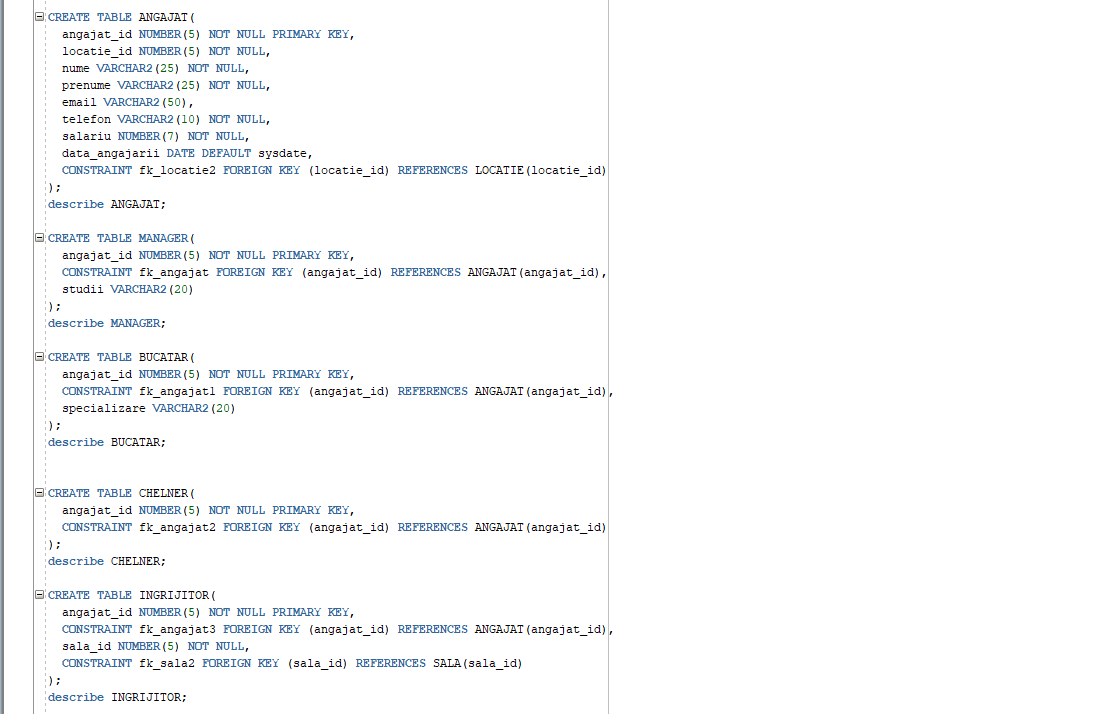


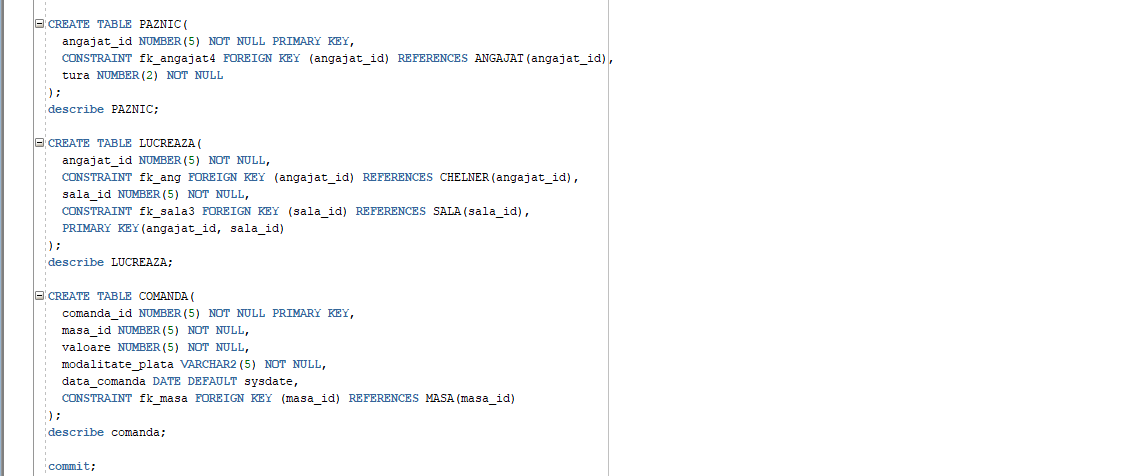
**3.Diagrama conceptuala**



**4.Implementarea in Oracle**







CREATE TABLE MENIU(

meniu\_id NUMBER(5) NOT NULL PRIMARY KEY,

specific VARCHAR2(20) NOT NULL

);

describe MENIU;

CREATE TABLE DISTRIBUITOR(

distribuitor\_id NUMBER(5) NOT NULL PRIMARY KEY,

nume\_distribuitor VARCHAR2(20) NOT NULL,

numar\_depozite NUMBER(5) NOT NULL

);

describe DISTRIBUITOR;

CREATE TABLE LOCATIE(

locatie\_id NUMBER(5) NOT NULL PRIMARY KEY,

meniu\_id NUMBER(5) NOT NULL,

distribuitor\_id NUMBER(5) NOT NULL,

oras VARCHAR2(20) NOT NULL,

numar\_sali NUMBER(5) NOT NULL,

CONSTRAINT fk\_meniu FOREIGN KEY (meniu\_id) REFERENCES MENIU(meniu\_id),

CONSTRAINT fk\_distrib FOREIGN KEY (distribuitor\_id) REFERENCES DISTRIBUITOR(distribuitor\_id)

);

DESCRIBE LOCATIE;

CREATE TABLE SALA(

sala\_id NUMBER(5) NOT NULL PRIMARY KEY,

locatie\_id NUMBER(5) NOT NULL,

nr\_mese NUMBER(5) NOT NULL,

CONSTRAINT fk\_locatie FOREIGN KEY (locatie\_id) REFERENCES LOCATIE(locatie\_id)

);

describe SALA;

CREATE TABLE MASA(

masa\_id NUMBER(5) NOT NULL PRIMARY KEY,

sala\_id NUMBER(5) NOT NULL,

capacitate\_masa NUMBER(5) NOT NULL,

CONSTRAINT fk\_sala FOREIGN KEY (sala\_id) REFERENCES SALA(sala\_id)

);

describe MASA;

CREATE TABLE ANGAJAT(

angajat\_id NUMBER(5) NOT NULL PRIMARY KEY,

locatie\_id NUMBER(5) NOT NULL,

nume VARCHAR2(25) NOT NULL,

prenume VARCHAR2(25) NOT NULL,

email VARCHAR2(50),

telefon VARCHAR2(10) NOT NULL,

salariu NUMBER(7) NOT NULL,

data\_angajarii DATE DEFAULT sysdate,

CONSTRAINT fk\_locatie2 FOREIGN KEY (locatie\_id) REFERENCES LOCATIE(locatie\_id)

);

describe ANGAJAT;

CREATE TABLE MANAGER(

angajat\_id NUMBER(5) NOT NULL PRIMARY KEY,

CONSTRAINT fk\_angajat FOREIGN KEY (angajat\_id) REFERENCES ANGAJAT(angajat\_id),

studii VARCHAR2(20)

);

describe MANAGER;

CREATE TABLE BUCATAR(

angajat\_id NUMBER(5) NOT NULL PRIMARY KEY,

CONSTRAINT fk\_angajat1 FOREIGN KEY (angajat\_id) REFERENCES ANGAJAT(angajat\_id),

specializare VARCHAR2(20)

);

describe BUCATAR;

CREATE TABLE CHELNER(

angajat\_id NUMBER(5) NOT NULL PRIMARY KEY,

CONSTRAINT fk\_angajat2 FOREIGN KEY (angajat\_id) REFERENCES ANGAJAT(angajat\_id)

);

describe CHELNER;

CREATE TABLE INGRIJITOR(

angajat\_id NUMBER(5) NOT NULL PRIMARY KEY,

CONSTRAINT fk\_angajat3 FOREIGN KEY (angajat\_id) REFERENCES ANGAJAT(angajat\_id),

sala\_id NUMBER(5) NOT NULL,

CONSTRAINT fk\_sala2 FOREIGN KEY (sala\_id) REFERENCES SALA(sala\_id)

);

describe INGRIJITOR;

CREATE TABLE PAZNIC(

angajat\_id NUMBER(5) NOT NULL PRIMARY KEY,

CONSTRAINT fk\_angajat4 FOREIGN KEY (angajat\_id) REFERENCES ANGAJAT(angajat\_id),

tura NUMBER(2) NOT NULL

);

describe PAZNIC;

CREATE TABLE LUCREAZA(

angajat\_id NUMBER(5) NOT NULL,

CONSTRAINT fk\_ang FOREIGN KEY (angajat\_id) REFERENCES CHELNER(angajat\_id),

sala\_id NUMBER(5) NOT NULL,

CONSTRAINT fk\_sala3 FOREIGN KEY (sala\_id) REFERENCES SALA(sala\_id),

PRIMARY KEY(angajat\_id, sala\_id)

);

describe LUCREAZA;

CREATE TABLE COMANDA(

comanda\_id NUMBER(5) NOT NULL PRIMARY KEY,

masa\_id NUMBER(5) NOT NULL,

valoare NUMBER(5) NOT NULL,

modalitate\_plata VARCHAR2(5) NOT NULL,

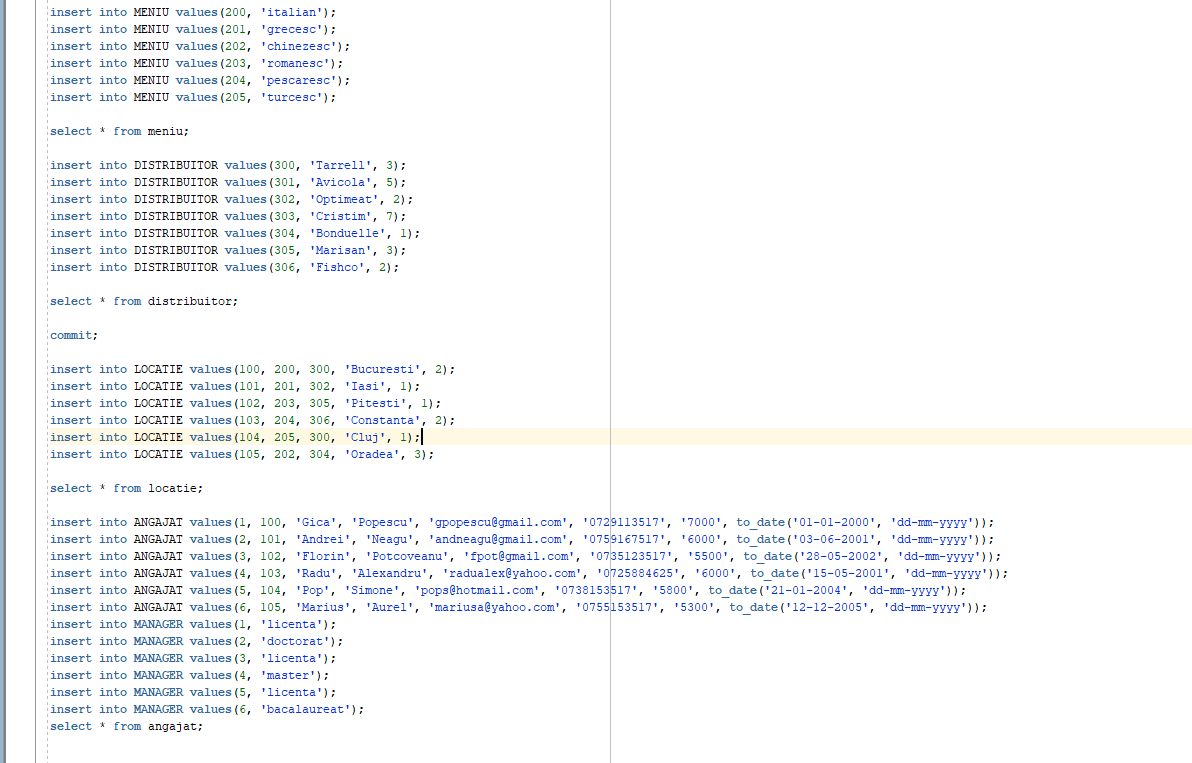
data\_comanda DATE DEFAULT sysdate,

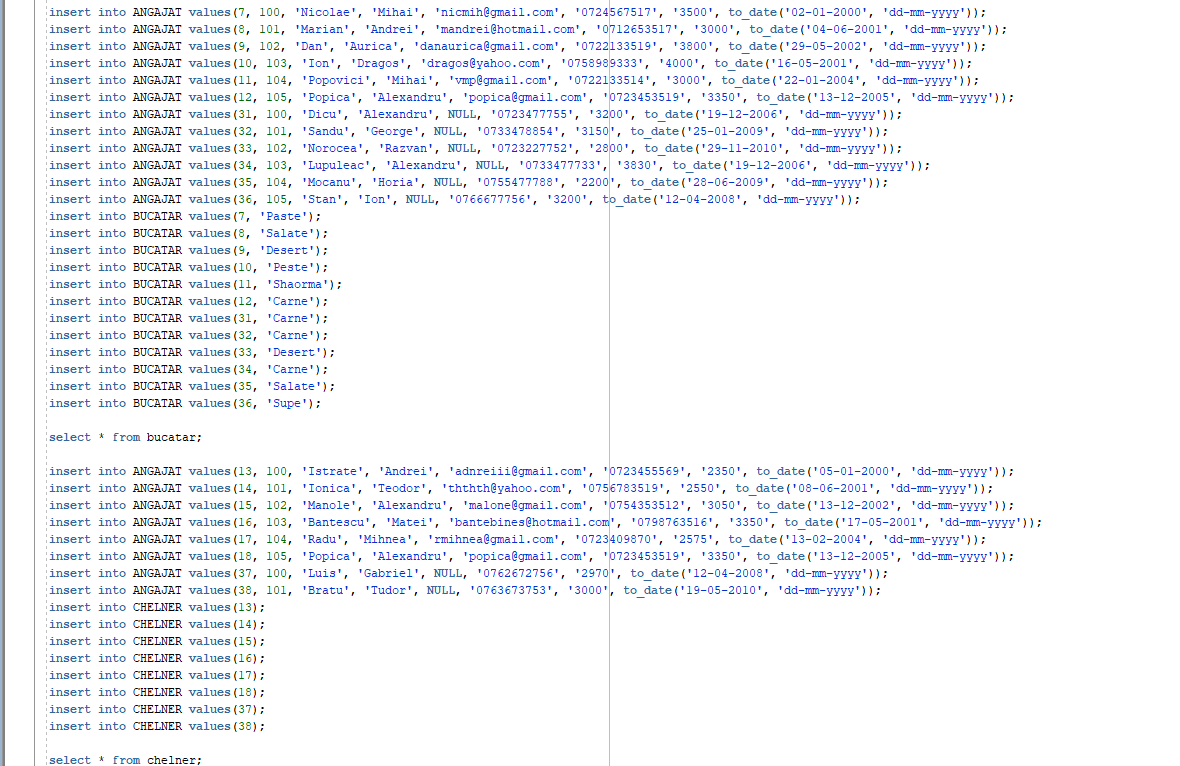
CONSTRAINT fk\_masa FOREIGN KEY (masa\_id) REFERENCES MASA(masa\_id)

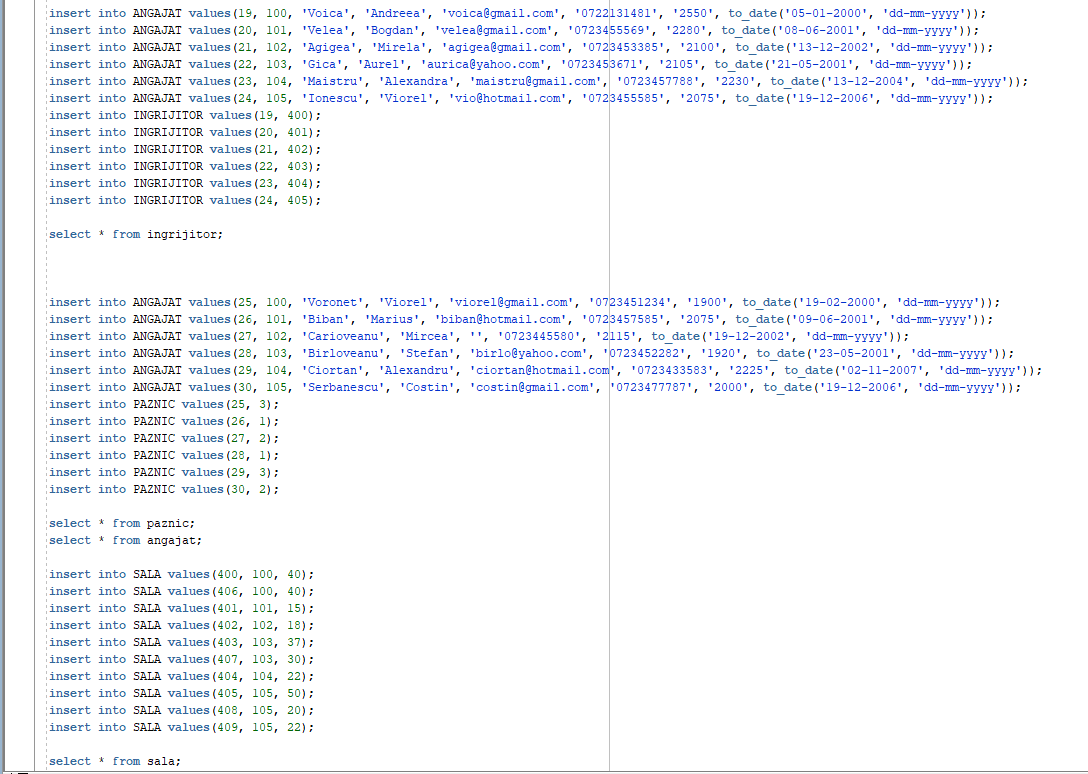
);

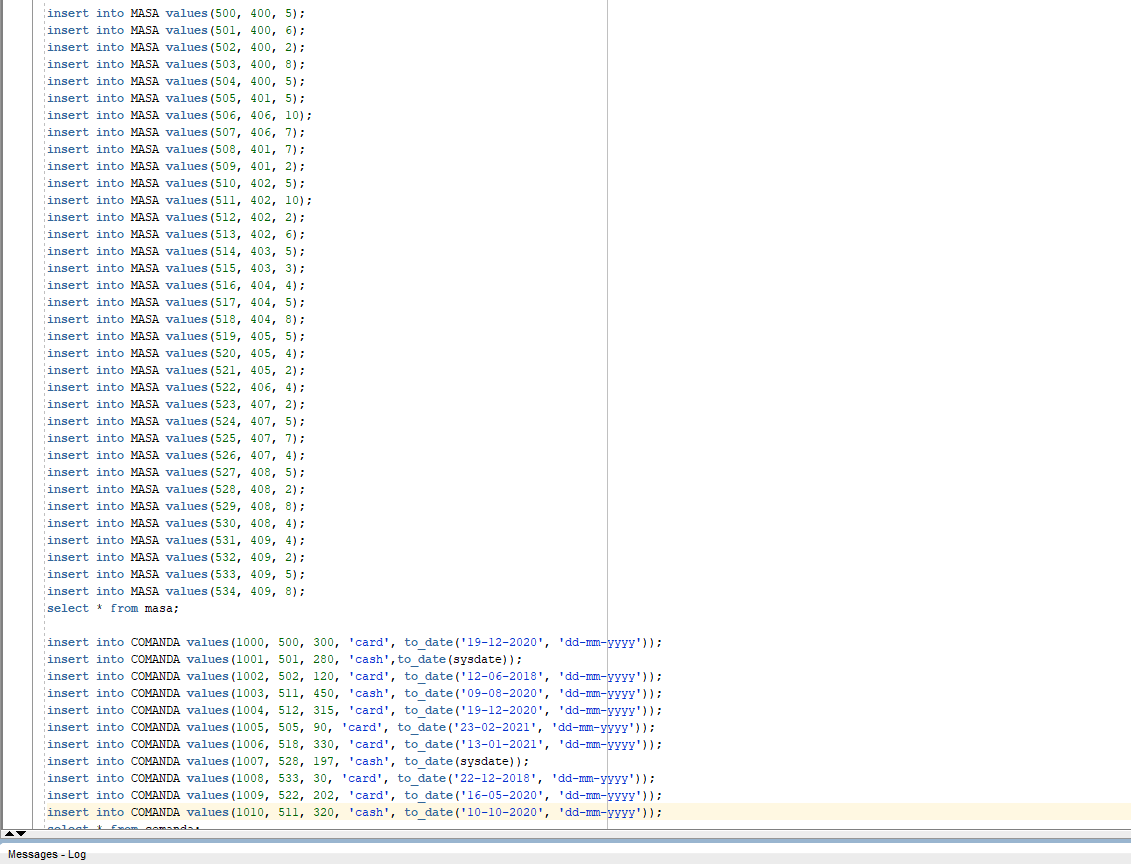
describe comanda;

**5.Inserari**









insert into MENIU values(200, 'italian');

insert into MENIU values(201, 'grecesc');

insert into MENIU values(202, 'chinezesc');

insert into MENIU values(203, 'romanesc');

insert into MENIU values(204, 'pescaresc');

insert into MENIU values(205, 'turcesc');

select \* from meniu;

insert into DISTRIBUITOR values(300, 'Tarrell', 3);

insert into DISTRIBUITOR values(301, 'Avicola', 5);

insert into DISTRIBUITOR values(302, 'Optimeat', 2);

insert into DISTRIBUITOR values(303, 'Cristim', 7);

insert into DISTRIBUITOR values(304, 'Bonduelle', 1);

insert into DISTRIBUITOR values(305, 'Marisan', 3);

insert into DISTRIBUITOR values(306, 'Fishco', 2);

select \* from distribuitor;

commit;

insert into LOCATIE values(100, 200, 300, 'Bucuresti', 2);

insert into LOCATIE values(101, 201, 302, 'Iasi', 1);

insert into LOCATIE values(102, 203, 305, 'Pitesti', 1);

insert into LOCATIE values(103, 204, 306, 'Constanta', 2);

insert into LOCATIE values(104, 205, 300, 'Cluj', 1);

insert into LOCATIE values(105, 202, 304, 'Oradea', 3);

select \* from locatie;

insert into ANGAJAT values(1, 100, 'Gica', 'Popescu', 'gpopescu@gmail.com', '0729113517', '7000', to\_date('01-01-2000', 'dd-mm-yyyy'));

insert into ANGAJAT values(2, 101, 'Andrei', 'Neagu', 'andneagu@gmail.com', '0759167517', '6000', to\_date('03-06-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(3, 102, 'Florin', 'Potcoveanu', 'fpot@gmail.com', '0735123517', '5500', to\_date('28-05-2002', 'dd-mm-yyyy'));

insert into ANGAJAT values(4, 103, 'Radu', 'Alexandru', 'radualex@yahoo.com', '0725884625', '6000', to\_date('15-05-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(5, 104, 'Pop', 'Simone', 'pops@hotmail.com', '0738153517', '5800', to\_date('21-01-2004', 'dd-mm-yyyy'));

insert into ANGAJAT values(6, 105, 'Marius', 'Aurel', 'mariusa@yahoo.com', '0755153517', '5300', to\_date('12-12-2005', 'dd-mm-yyyy'));

insert into MANAGER values(1, 'licenta');

insert into MANAGER values(2, 'doctorat');

insert into MANAGER values(3, 'licenta');

insert into MANAGER values(4, 'master');

insert into MANAGER values(5, 'licenta');

insert into MANAGER values(6, 'bacalaureat');

select \* from angajat;

select \* from manager;

insert into ANGAJAT values(7, 100, 'Nicolae', 'Mihai', 'nicmih@gmail.com', '0724567517', '3500', to\_date('02-01-2000', 'dd-mm-yyyy'));

insert into ANGAJAT values(8, 101, 'Marian', 'Andrei', 'mandrei@hotmail.com', '0712653517', '3000', to\_date('04-06-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(9, 102, 'Dan', 'Aurica', 'danaurica@gmail.com', '0722133519', '3800', to\_date('29-05-2002', 'dd-mm-yyyy'));

insert into ANGAJAT values(10, 103, 'Ion', 'Dragos', 'dragos@yahoo.com', '0758989333', '4000', to\_date('16-05-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(11, 104, 'Popovici', 'Mihai', 'vmp@gmail.com', '0722133514', '3000', to\_date('22-01-2004', 'dd-mm-yyyy'));

insert into ANGAJAT values(12, 105, 'Popica', 'Alexandru', 'popica@gmail.com', '0723453519', '3350', to\_date('13-12-2005', 'dd-mm-yyyy'));

insert into ANGAJAT values(31, 100, 'Dicu', 'Alexandru', NULL, '0723477755', '3200', to\_date('19-12-2006', 'dd-mm-yyyy'));

insert into ANGAJAT values(32, 101, 'Sandu', 'George', NULL, '0733478854', '3150', to\_date('25-01-2009', 'dd-mm-yyyy'));

insert into ANGAJAT values(33, 102, 'Norocea', 'Razvan', NULL, '0723227752', '2800', to\_date('29-11-2010', 'dd-mm-yyyy'));

insert into ANGAJAT values(34, 103, 'Lupuleac', 'Alexandru', NULL, '0733477733', '3830', to\_date('19-12-2006', 'dd-mm-yyyy'));

insert into ANGAJAT values(35, 104, 'Mocanu', 'Horia', NULL, '0755477788', '2200', to\_date('28-06-2009', 'dd-mm-yyyy'));

insert into ANGAJAT values(36, 105, 'Stan', 'Ion', NULL, '0766677756', '3200', to\_date('12-04-2008', 'dd-mm-yyyy'));

insert into BUCATAR values(7, 'Paste');

insert into BUCATAR values(8, 'Salate');

insert into BUCATAR values(9, 'Desert');

insert into BUCATAR values(10, 'Peste');

insert into BUCATAR values(11, 'Shaorma');

insert into BUCATAR values(12, 'Carne');

insert into BUCATAR values(31, 'Carne');

insert into BUCATAR values(32, 'Carne');

insert into BUCATAR values(33, 'Desert');

insert into BUCATAR values(34, 'Carne');

insert into BUCATAR values(35, 'Salate');

insert into BUCATAR values(36, 'Supe');

select \* from bucatar;

insert into ANGAJAT values(13, 100, 'Istrate', 'Andrei', 'adnreiii@gmail.com', '0723455569', '2350', to\_date('05-01-2000', 'dd-mm-yyyy'));

insert into ANGAJAT values(14, 101, 'Ionica', 'Teodor', 'ththth@yahoo.com', '0756783519', '2550', to\_date('08-06-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(15, 102, 'Manole', 'Alexandru', 'malone@gmail.com', '0754353512', '3050', to\_date('13-12-2002', 'dd-mm-yyyy'));

insert into ANGAJAT values(16, 103, 'Bantescu', 'Matei', 'bantebines@hotmail.com', '0798763516', '3350', to\_date('17-05-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(17, 104, 'Radu', 'Mihnea', 'rmihnea@gmail.com', '0723409870', '2575', to\_date('13-02-2004', 'dd-mm-yyyy'));

insert into ANGAJAT values(18, 105, 'Popica', 'Alexandru', 'popica@gmail.com', '0723453519', '3350', to\_date('13-12-2005', 'dd-mm-yyyy'));

insert into ANGAJAT values(37, 100, 'Luis', 'Gabriel', NULL, '0762672756', '2970', to\_date('12-04-2008', 'dd-mm-yyyy'));

insert into ANGAJAT values(38, 101, 'Bratu', 'Tudor', NULL, '0763673753', '3000', to\_date('19-05-2010', 'dd-mm-yyyy'));

insert into CHELNER values(13);

insert into CHELNER values(14);

insert into CHELNER values(15);

insert into CHELNER values(16);

insert into CHELNER values(17);

insert into CHELNER values(18);

insert into CHELNER values(37);

insert into CHELNER values(38);

select \* from chelner;

insert into ANGAJAT values(19, 100, 'Voica', 'Andreea', 'voica@gmail.com', '0722131481', '2550', to\_date('05-01-2000', 'dd-mm-yyyy'));

insert into ANGAJAT values(20, 101, 'Velea', 'Bogdan', 'velea@gmail.com', '0723455569', '2280', to\_date('08-06-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(21, 102, 'Agigea', 'Mirela', 'agigea@gmail.com', '0723453385', '2100', to\_date('13-12-2002', 'dd-mm-yyyy'));

insert into ANGAJAT values(22, 103, 'Gica', 'Aurel', 'aurica@yahoo.com', '0723453671', '2105', to\_date('21-05-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(23, 104, 'Maistru', 'Alexandra', 'maistru@gmail.com', '0723457788', '2230', to\_date('13-12-2004', 'dd-mm-yyyy'));

insert into ANGAJAT values(24, 105, 'Ionescu', 'Viorel', 'vio@hotmail.com', '0723455585', '2075', to\_date('19-12-2006', 'dd-mm-yyyy'));

insert into INGRIJITOR values(19, 400);

insert into INGRIJITOR values(20, 401);

insert into INGRIJITOR values(21, 402);

insert into INGRIJITOR values(22, 403);

insert into INGRIJITOR values(23, 404);

insert into INGRIJITOR values(24, 405);

select \* from ingrijitor;

insert into ANGAJAT values(25, 100, 'Voronet', 'Viorel', 'viorel@gmail.com', '0723451234', '1900', to\_date('19-02-2000', 'dd-mm-yyyy'));

insert into ANGAJAT values(26, 101, 'Biban', 'Marius', 'biban@hotmail.com', '0723457585', '2075', to\_date('09-06-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(27, 102, 'Carioveanu', 'Mircea', '', '0723445580', '2115', to\_date('19-12-2002', 'dd-mm-yyyy'));

insert into ANGAJAT values(28, 103, 'Birloveanu', 'Stefan', 'birlo@yahoo.com', '0723452282', '1920', to\_date('23-05-2001', 'dd-mm-yyyy'));

insert into ANGAJAT values(29, 104, 'Ciortan', 'Alexandru', 'ciortan@hotmail.com', '0723433583', '2225', to\_date('02-11-2007', 'dd-mm-yyyy'));

insert into ANGAJAT values(30, 105, 'Serbanescu', 'Costin', 'costin@gmail.com', '0723477787', '2000', to\_date('19-12-2006', 'dd-mm-yyyy'));

insert into PAZNIC values(25, 3);

insert into PAZNIC values(26, 1);

insert into PAZNIC values(27, 2);

insert into PAZNIC values(28, 1);

insert into PAZNIC values(29, 3);

insert into PAZNIC values(30, 2);

select \* from paznic;

select \* from angajat;

insert into SALA values(400, 100, 40);

insert into SALA values(406, 100, 40);

insert into SALA values(401, 101, 15);

insert into SALA values(402, 102, 18);

insert into SALA values(403, 103, 37);

insert into SALA values(407, 103, 30);

insert into SALA values(404, 104, 22);

insert into SALA values(405, 105, 50);

insert into SALA values(408, 105, 20);

insert into SALA values(409, 105, 22);

select \* from sala;

insert into MASA values(500, 400, 5);

insert into MASA values(501, 400, 6);

insert into MASA values(502, 400, 2);

insert into MASA values(503, 400, 8);

insert into MASA values(504, 400, 5);

insert into MASA values(505, 401, 5);

insert into MASA values(506, 406, 10);

insert into MASA values(507, 406, 7);

insert into MASA values(508, 401, 7);

insert into MASA values(509, 401, 2);

insert into MASA values(510, 402, 5);

insert into MASA values(511, 402, 10);

insert into MASA values(512, 402, 2);

insert into MASA values(513, 402, 6);

insert into MASA values(514, 403, 5);

insert into MASA values(515, 403, 3);

insert into MASA values(516, 404, 4);

insert into MASA values(517, 404, 5);

insert into MASA values(518, 404, 8);

insert into MASA values(519, 405, 5);

insert into MASA values(520, 405, 4);

insert into MASA values(521, 405, 2);

insert into MASA values(522, 406, 4);

insert into MASA values(523, 407, 2);

insert into MASA values(524, 407, 5);

insert into MASA values(525, 407, 7);

insert into MASA values(526, 407, 4);

insert into MASA values(527, 408, 5);

insert into MASA values(528, 408, 2);

insert into MASA values(529, 408, 8);

insert into MASA values(530, 408, 4);

insert into MASA values(531, 409, 4);

insert into MASA values(532, 409, 2);

insert into MASA values(533, 409, 5);

insert into MASA values(534, 409, 8);

select \* from masa;

insert into COMANDA values(1000, 500, 300, 'card', to\_date('19-12-2020', 'dd-mm-yyyy'));

insert into COMANDA values(1001, 501, 280, 'cash',to\_date(sysdate));

insert into COMANDA values(1002, 502, 120, 'card', to\_date('12-06-2018', 'dd-mm-yyyy'));

insert into COMANDA values(1003, 511, 450, 'cash', to\_date('09-08-2020', 'dd-mm-yyyy'));

insert into COMANDA values(1004, 512, 315, 'card', to\_date('19-12-2020', 'dd-mm-yyyy'));

insert into COMANDA values(1005, 505, 90, 'card', to\_date('23-02-2021', 'dd-mm-yyyy'));

insert into COMANDA values(1006, 518, 330, 'card', to\_date('13-01-2021', 'dd-mm-yyyy'));

insert into COMANDA values(1007, 528, 197, 'cash', to\_date(sysdate));

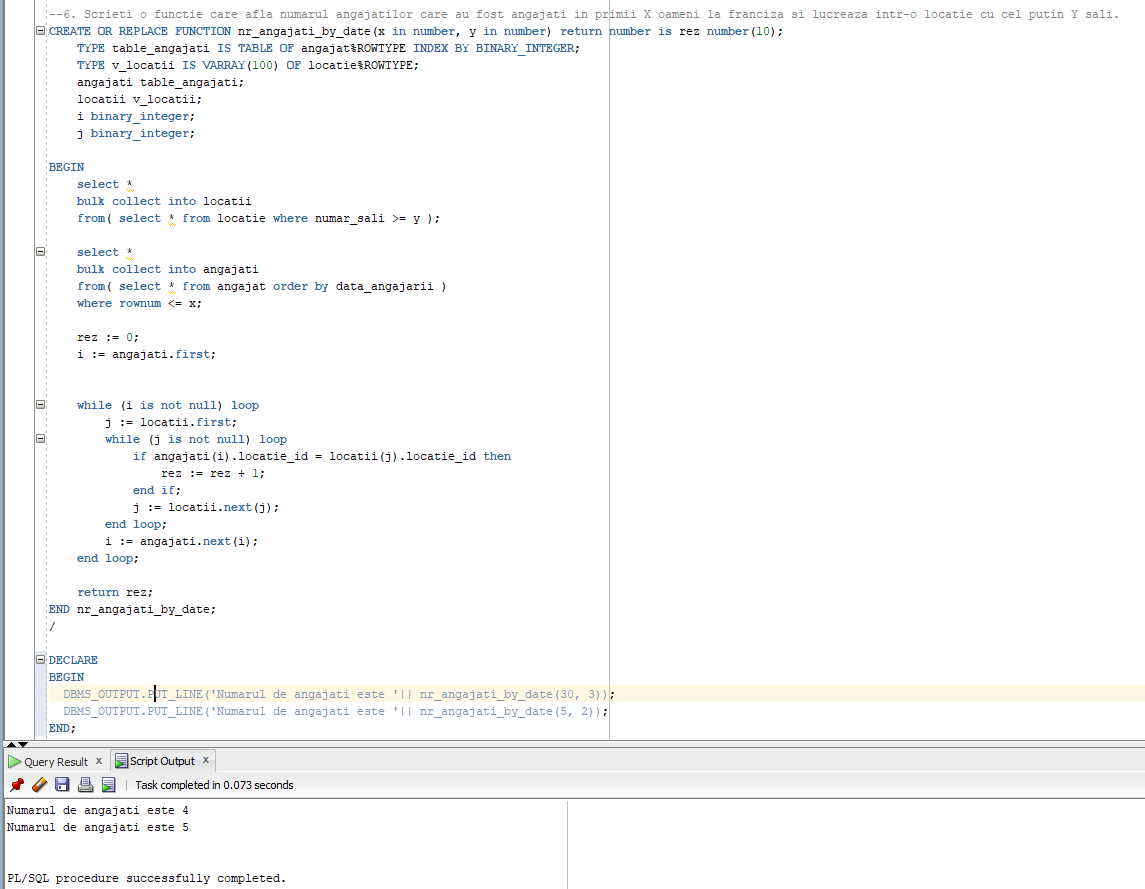
insert into COMANDA values(1008, 533, 30, 'card', to\_date('22-12-2018', 'dd-mm-yyyy'));

insert into COMANDA values(1009, 522, 202, 'card', to\_date('16-05-2020', 'dd-mm-yyyy'));

insert into COMANDA values(1010, 511, 320, 'cash', to\_date('10-10-2020', 'dd-mm-yyyy'));

select \* from comanda;

**6.** **Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze două tipuri de colecție studiate. Apelați subprogramul.**



--6. Scrieti o functie care afla numarul angajatilor care au fost angajati in primii X oameni la franciza si lucreaza intr-o locatie cu cel putin Y sali.

CREATE OR REPLACE FUNCTION nr\_angajati\_by\_date(x in number, y in number) return number is rez number(10);

TYPE table\_angajati IS TABLE OF angajat%ROWTYPE INDEX BY BINARY\_INTEGER;

TYPE v\_locatii IS VARRAY(100) OF locatie%ROWTYPE;

angajati table\_angajati;

locatii v\_locatii;

i binary\_integer;

j binary\_integer;

BEGIN

select \*

bulk collect into locatii

from( select \* from locatie where numar\_sali >= y );

select \*

bulk collect into angajati

from( select \* from angajat order by data\_angajarii )

where rownum <= x;

rez := 0;

i := angajati.first;

while (i is not null) loop

j := locatii.first;

while (j is not null) loop

if angajati(i).locatie\_id = locatii(j).locatie\_id then

rez := rez + 1;

end if;

j := locatii.next(j);

end loop;

i := angajati.next(i);

end loop;

return rez;

END nr\_angajati\_by\_date;

/

DECLARE

BEGIN

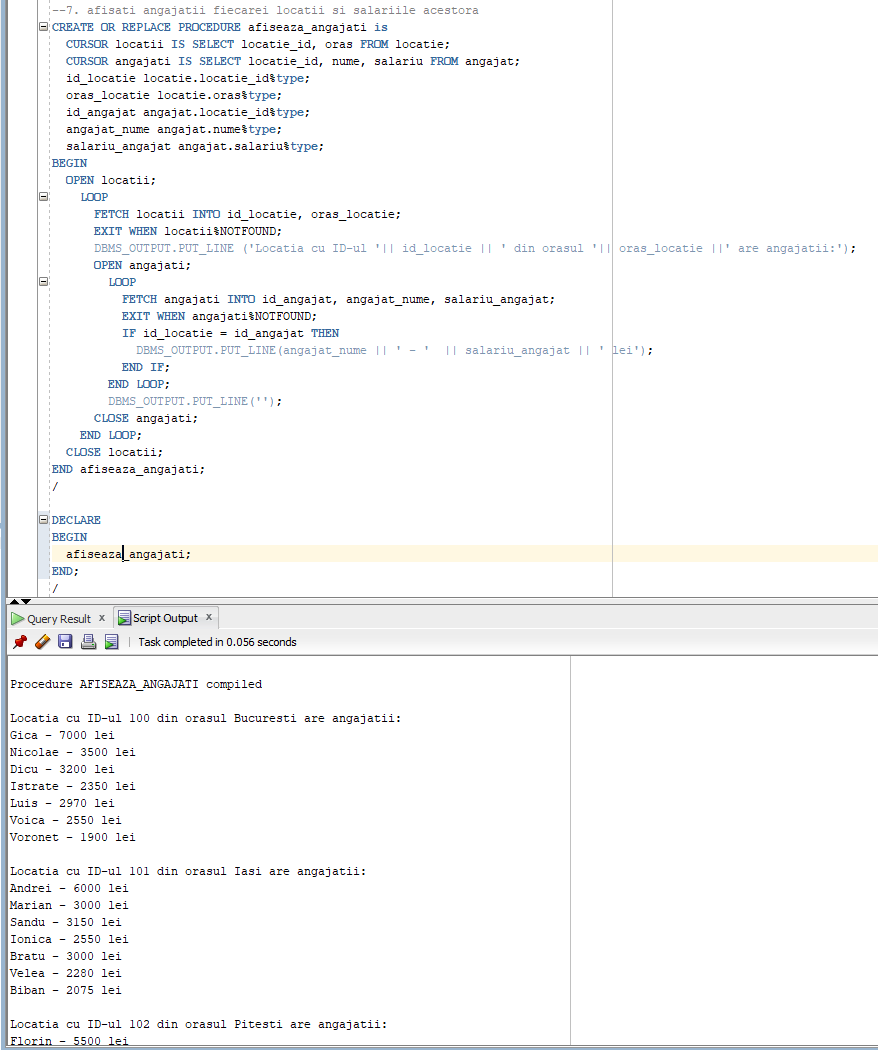
DBMS\_OUTPUT.PUT\_LINE('Numarul de angajati este '|| nr\_angajati\_by\_date(30, 3));

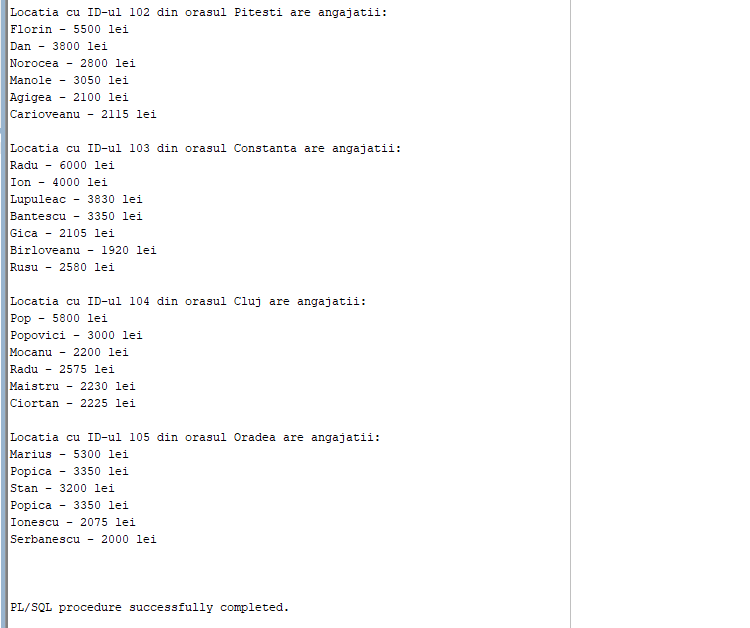
DBMS\_OUTPUT.PUT\_LINE('Numarul de angajati este '|| nr\_angajati\_by\_date(5, 2));

END;

/

**7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul**



****

--7. afisati angajatii fiecarei locatii si salariile acestora

CREATE OR REPLACE PROCEDURE afiseaza\_angajati is

CURSOR locatii IS SELECT locatie\_id, oras FROM locatie;

CURSOR angajati IS SELECT locatie\_id, nume, salariu FROM angajat;

id\_locatie locatie.locatie\_id%type;

oras\_locatie locatie.oras%type;

id\_angajat angajat.locatie\_id%type;

angajat\_nume angajat.nume%type;

salariu\_angajat angajat.salariu%type;

BEGIN

OPEN locatii;

LOOP

FETCH locatii INTO id\_locatie, oras\_locatie;

EXIT WHEN locatii%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE ('Locatia cu ID-ul '|| id\_locatie || ' din orasul '|| oras\_locatie ||' are angajatii:');

OPEN angajati;

LOOP

FETCH angajati INTO id\_angajat, angajat\_nume, salariu\_angajat;

EXIT WHEN angajati%NOTFOUND;

IF id\_locatie = id\_angajat THEN

DBMS\_OUTPUT.PUT\_LINE(angajat\_nume || ' - ' || salariu\_angajat || ' lei');

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('');

CLOSE angajati;

END LOOP;

CLOSE locatii;

END afiseaza\_angajati;

/

DECLARE

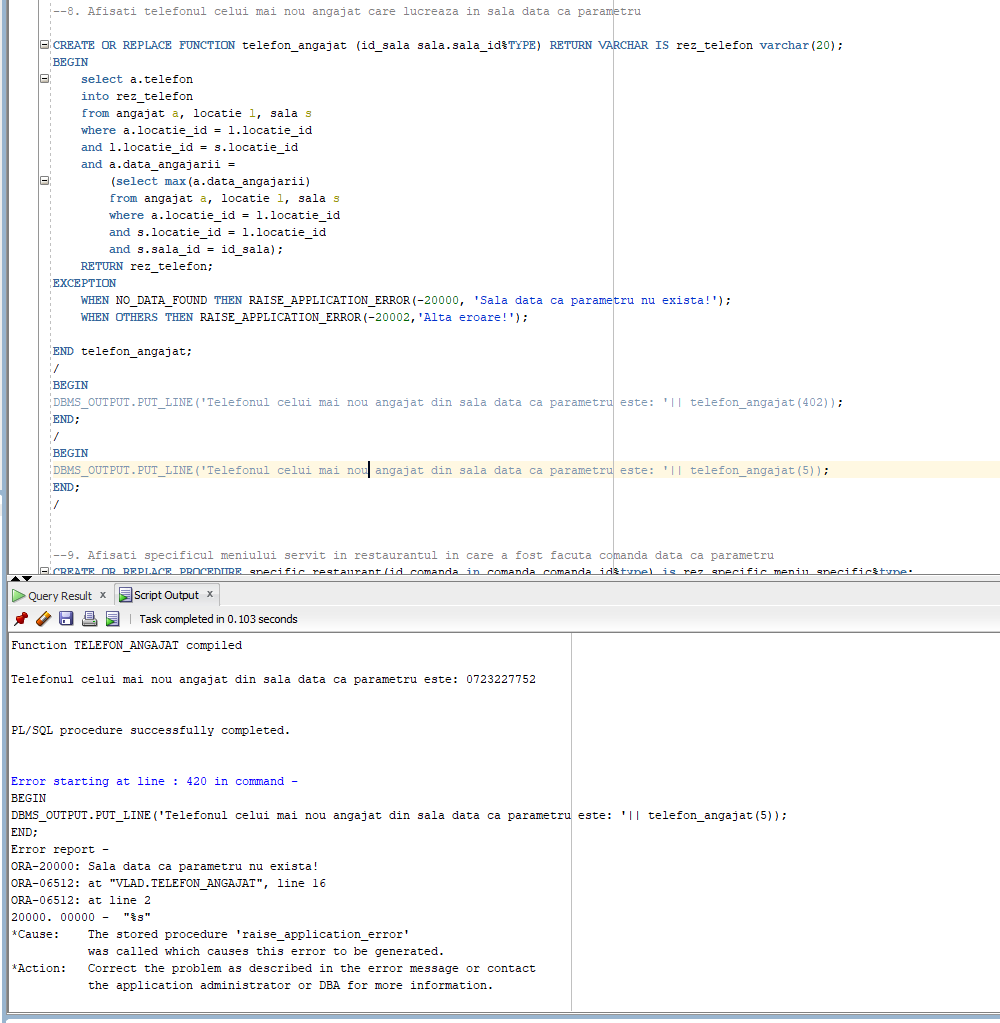
BEGIN

afiseaza\_angajati;

END;

/

**8.** **Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite**



--8. Afisati telefonul celui mai nou angajat care lucreaza in sala data ca parametru

CREATE OR REPLACE FUNCTION telefon\_angajat (id\_sala sala.sala\_id%TYPE) RETURN VARCHAR IS rez\_telefon varchar(20);

BEGIN

select a.telefon

into rez\_telefon

from angajat a, locatie l, sala s

where a.locatie\_id = l.locatie\_id

and l.locatie\_id = s.locatie\_id

and a.data\_angajarii =

(select max(a.data\_angajarii)

from angajat a, locatie l, sala s

where a.locatie\_id = l.locatie\_id

and s.locatie\_id = l.locatie\_id

and s.sala\_id = id\_sala);

RETURN rez\_telefon;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RAISE\_APPLICATION\_ERROR(-20000, 'Sala data ca parametru nu exista!');

WHEN OTHERS THEN RAISE\_APPLICATION\_ERROR(-20002,'Alta eroare!');

END telefon\_angajat;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Telefonul celui mai nou angajat din sala data ca parametru este: '|| telefon\_angajat(402));

END;

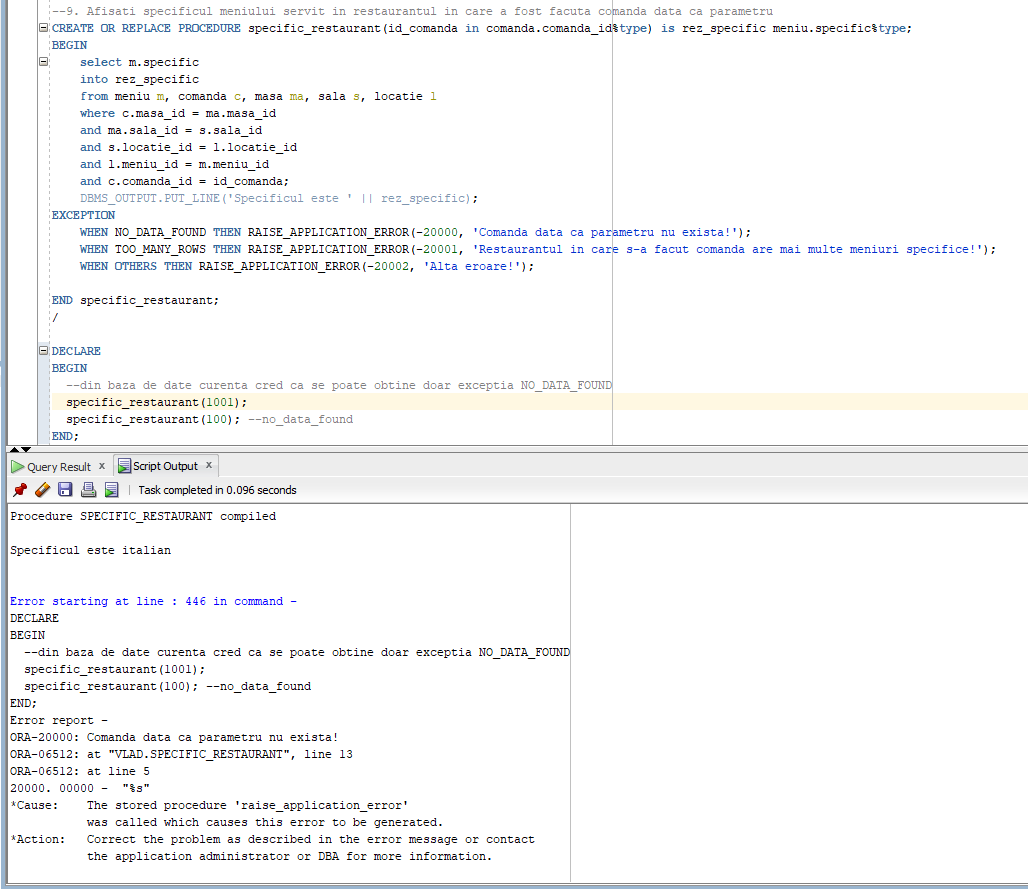
/

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Telefonul celui mai nou angajat din sala data ca parametru este: '|| telefon\_angajat(5));

END;

**9.** **Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite**



--9. Afisati specificul meniului servit in restaurantul in care a fost facuta comanda data ca parametru

CREATE OR REPLACE PROCEDURE specific\_restaurant(id\_comanda in comanda.comanda\_id%type) is rez\_specific meniu.specific%type;

BEGIN

select m.specific

into rez\_specific

from meniu m, comanda c, masa ma, sala s, locatie l

where c.masa\_id = ma.masa\_id

and ma.sala\_id = s.sala\_id

and s.locatie\_id = l.locatie\_id

and l.meniu\_id = m.meniu\_id

and c.comanda\_id = id\_comanda;

DBMS\_OUTPUT.PUT\_LINE('Specificul este ' || rez\_specific);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RAISE\_APPLICATION\_ERROR(-20000, 'Comanda data ca parametru nu exista!');

WHEN TOO\_MANY\_ROWS THEN RAISE\_APPLICATION\_ERROR(-20001, 'Restaurantul in care s-a facut comanda are mai multe meniuri specifice!');

WHEN OTHERS THEN RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare!');

END specific\_restaurant;

/

DECLARE

BEGIN

--din baza de date curenta cred ca se poate obtine doar exceptia NO\_DATA\_FOUND

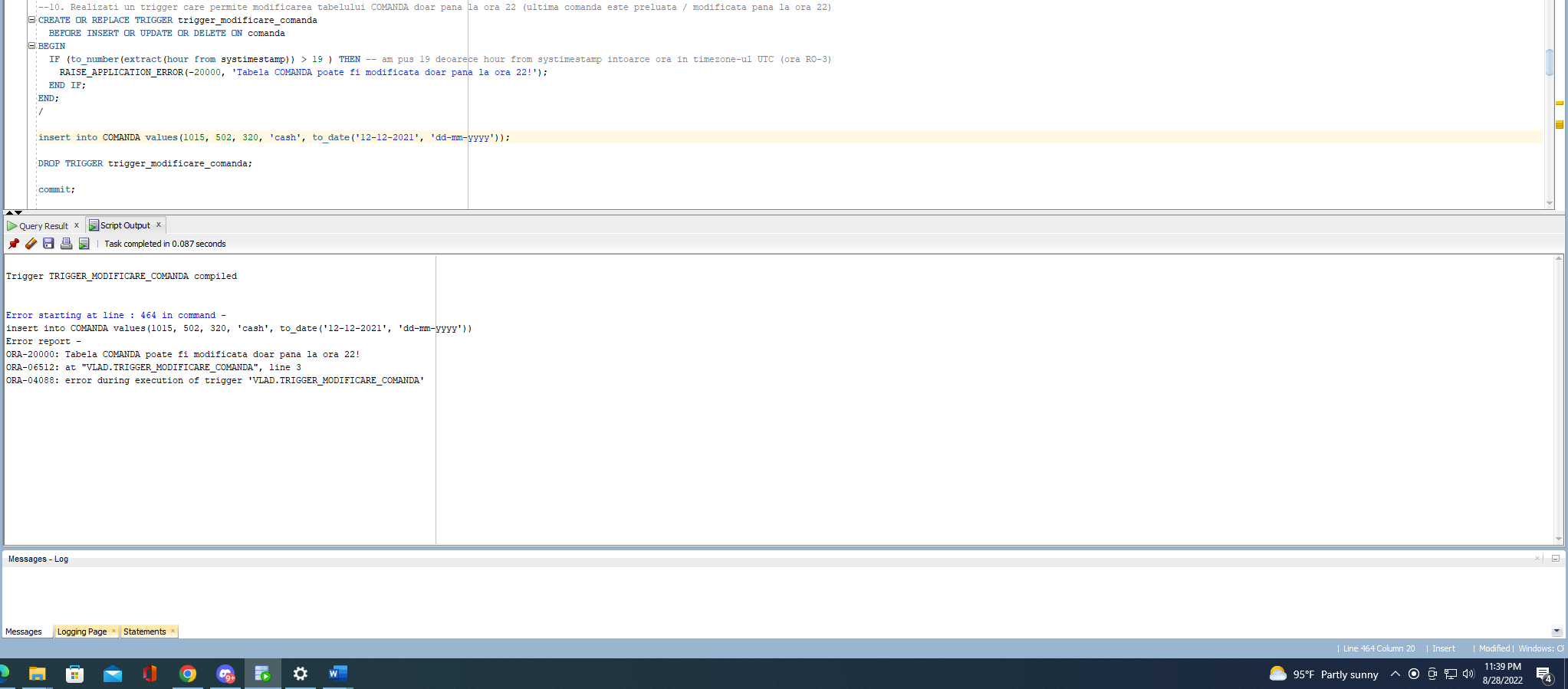
specific\_restaurant(1001);

specific\_restaurant(100); --no\_data\_found

END;

/

**10.** **Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.**



--10. Realizati un trigger care permite modificarea tabelului COMANDA doar pana la ora 22 (ultima comanda este preluata / modificata pana la ora 22)

CREATE OR REPLACE TRIGGER trigger\_modificare\_comanda

BEFORE INSERT OR UPDATE OR DELETE ON comanda

BEGIN

IF (to\_number(extract(hour from systimestamp)) > 19 ) THEN -- am pus 19 deoarece hour from systimestamp intoarce ora in timezone-ul UTC (ora RO-3)

RAISE\_APPLICATION\_ERROR(-20000, 'Tabela COMANDA poate fi modificata doar pana la ora 22!');

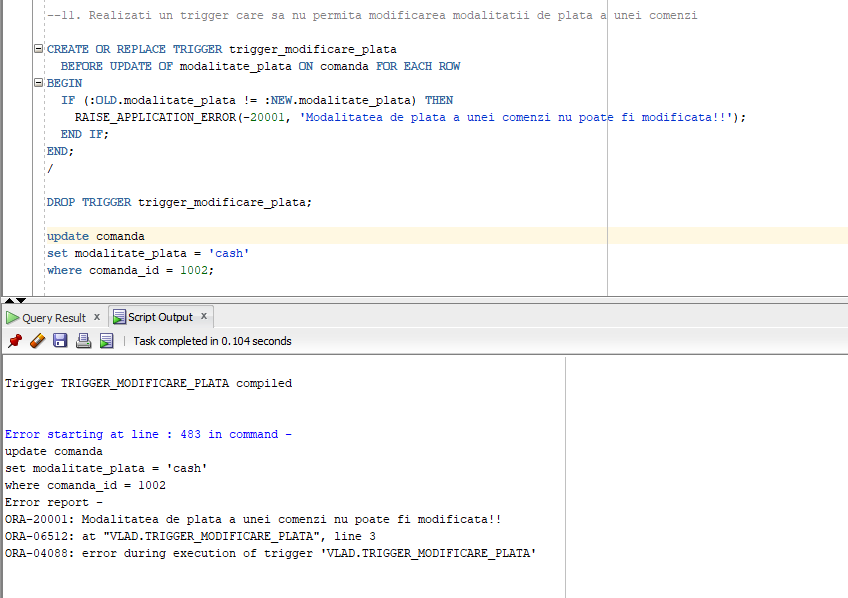
END IF;

END;

/

insert into COMANDA values(1015, 502, 320, 'cash', to\_date('12-12-2021', 'dd-mm-yyyy'));

**11.** **Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.**



--11. Realizati un trigger care sa nu permita modificarea modalitatii de plata a unei comenzi

CREATE OR REPLACE TRIGGER trigger\_modificare\_plata

BEFORE UPDATE OF modalitate\_plata ON comanda FOR EACH ROW

BEGIN

IF (:OLD.modalitate\_plata != :NEW.modalitate\_plata) THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Modalitatea de plata a unei comenzi nu poate fi modificata!!');

END IF;

END;

/

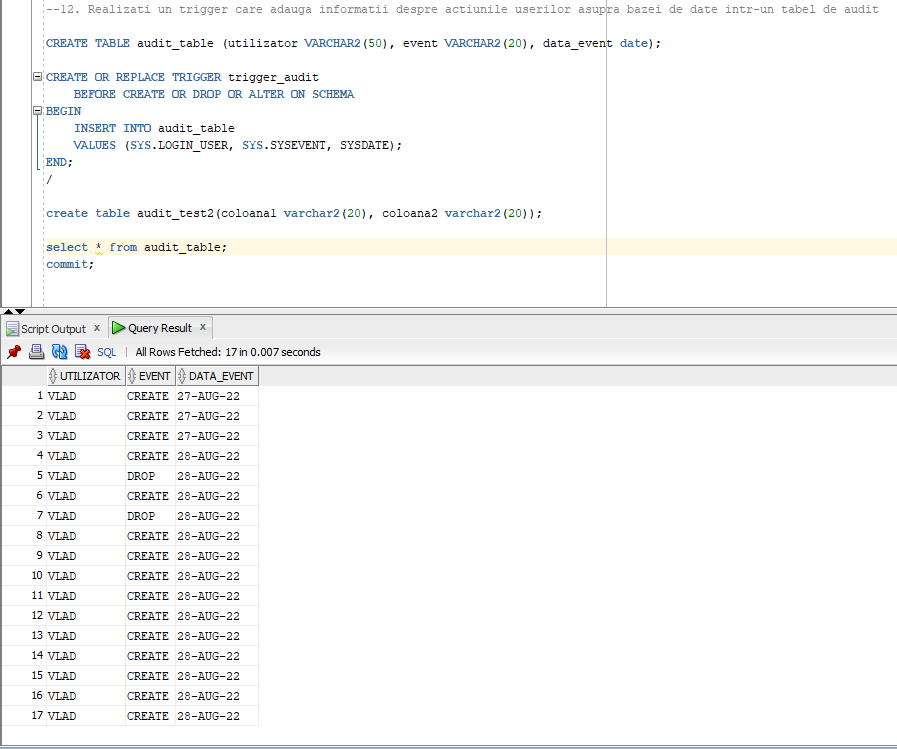
DROP TRIGGER trigger\_modificare\_plata;

update comanda

set modalitate\_plata = 'cash'

where comanda\_id = 1002;

**12.** **Definiți un trigger de tip LDD. Declanșați trigger-ul.**

****

--12. Realizati un trigger care adauga informatii despre actiunile userilor asupra bazei de date intr-un tabel de audit

CREATE TABLE audit\_table (utilizator VARCHAR2(50), event VARCHAR2(20), data\_event date);

CREATE OR REPLACE TRIGGER trigger\_audit

BEFORE CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO audit\_table

VALUES (SYS.LOGIN\_USER, SYS.SYSEVENT, SYSDATE);

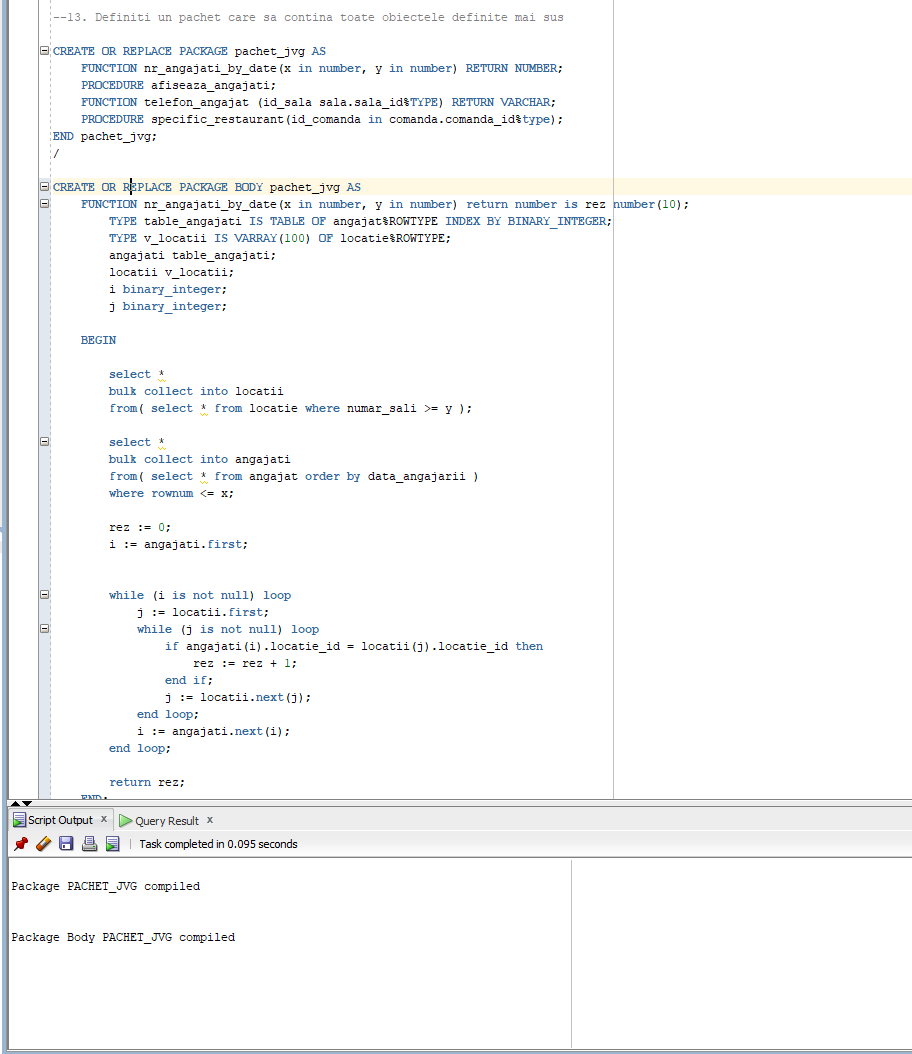
END;

/

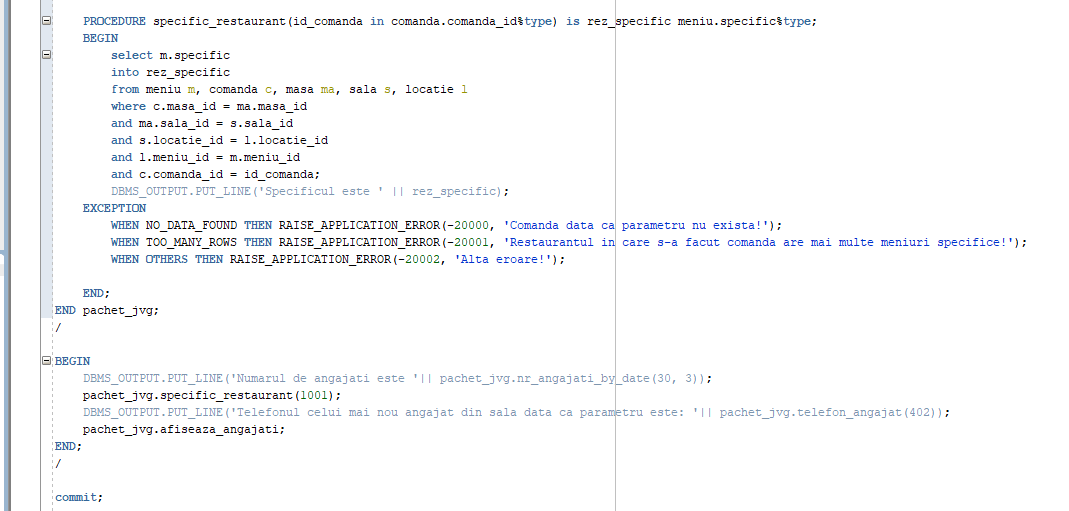
create table audit\_test2(coloana1 varchar2(20), coloana2 varchar2(20));

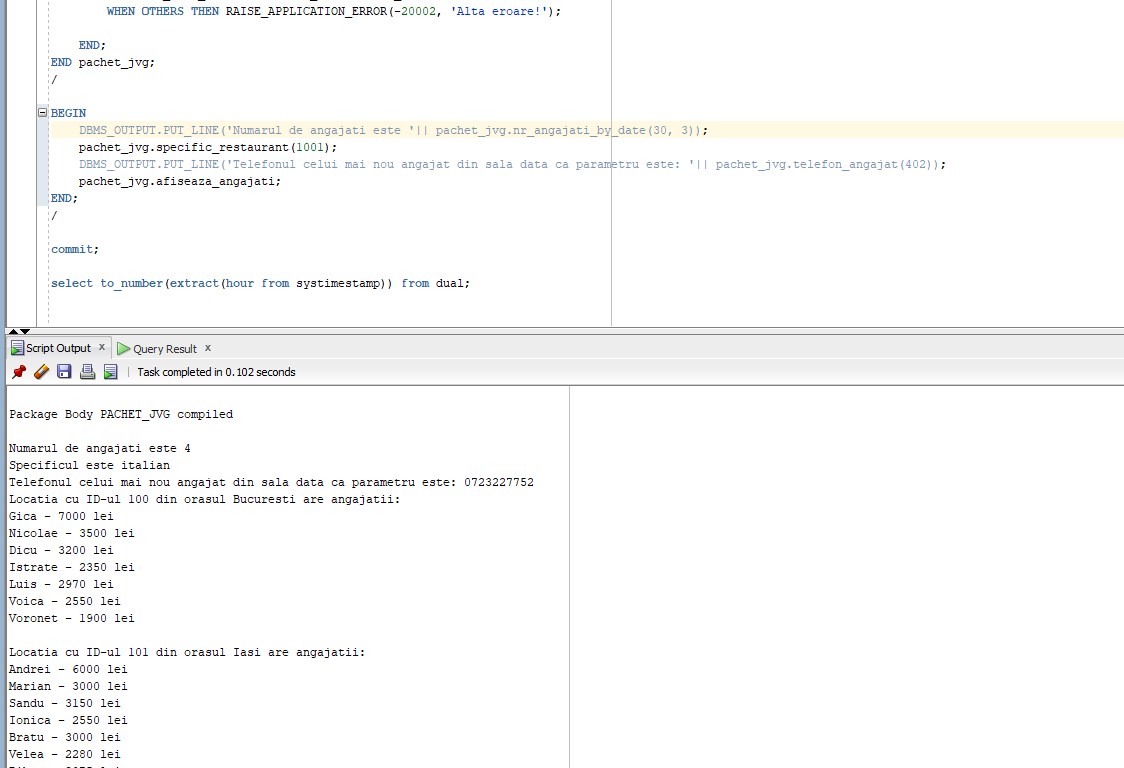
select \* from audit\_table;

**13.** **Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.**

****







--13. Definiti un pachet care sa contina toate obiectele definite mai sus

CREATE OR REPLACE PACKAGE pachet\_jvg AS

FUNCTION nr\_angajati\_by\_date(x in number, y in number) RETURN NUMBER;

PROCEDURE afiseaza\_angajati;

FUNCTION telefon\_angajat (id\_sala sala.sala\_id%TYPE) RETURN VARCHAR;

PROCEDURE specific\_restaurant(id\_comanda in comanda.comanda\_id%type);

END pachet\_jvg;

/

CREATE OR REPLACE PACKAGE BODY pachet\_jvg AS

FUNCTION nr\_angajati\_by\_date(x in number, y in number) return number is rez number(10);

TYPE table\_angajati IS TABLE OF angajat%ROWTYPE INDEX BY BINARY\_INTEGER;

TYPE v\_locatii IS VARRAY(100) OF locatie%ROWTYPE;

angajati table\_angajati;

locatii v\_locatii;

i binary\_integer;

j binary\_integer;

BEGIN

select \*

bulk collect into locatii

from( select \* from locatie where numar\_sali >= y );

select \*

bulk collect into angajati

from( select \* from angajat order by data\_angajarii )

where rownum <= x;

rez := 0;

i := angajati.first;

while (i is not null) loop

j := locatii.first;

while (j is not null) loop

if angajati(i).locatie\_id = locatii(j).locatie\_id then

rez := rez + 1;

end if;

j := locatii.next(j);

end loop;

i := angajati.next(i);

end loop;

return rez;

END;

--

PROCEDURE afiseaza\_angajati is

CURSOR locatii IS SELECT locatie\_id, oras FROM locatie;

CURSOR angajati IS SELECT locatie\_id, nume, salariu FROM angajat;

id\_locatie locatie.locatie\_id%type;

oras\_locatie locatie.oras%type;

id\_angajat angajat.locatie\_id%type;

nume\_angajat angajat.nume%type;

salariu\_angajat angajat.salariu%type;

BEGIN

OPEN locatii;

LOOP

FETCH locatii INTO id\_locatie, oras\_locatie;

EXIT WHEN locatii%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE ('Locatia cu ID-ul '|| id\_locatie || ' din orasul '|| oras\_locatie ||' are angajatii:');

OPEN angajati;

LOOP

FETCH angajati INTO id\_angajat, nume\_angajat, salariu\_angajat;

EXIT WHEN angajati%NOTFOUND;

IF id\_locatie = id\_angajat THEN

DBMS\_OUTPUT.PUT\_LINE(nume\_angajat || ' - ' || salariu\_angajat || ' lei');

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('');

CLOSE angajati;

END LOOP;

CLOSE locatii;

END;

--

FUNCTION telefon\_angajat (id\_sala sala.sala\_id%TYPE) RETURN VARCHAR IS rez\_telefon varchar(20);

BEGIN

select a.telefon

into rez\_telefon

from angajat a, locatie l, sala s

where a.locatie\_id = l.locatie\_id

and l.locatie\_id = s.locatie\_id

and a.data\_angajarii =

(select max(a.data\_angajarii)

from angajat a, locatie l, sala s

where a.locatie\_id = l.locatie\_id

and s.locatie\_id = l.locatie\_id

and s.sala\_id = id\_sala);

RETURN rez\_telefon;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RAISE\_APPLICATION\_ERROR(-20000, 'Sala data ca parametru nu exista!');

WHEN OTHERS THEN RAISE\_APPLICATION\_ERROR(-20002,'Alta eroare!');

END;

--

PROCEDURE specific\_restaurant(id\_comanda in comanda.comanda\_id%type) is rez\_specific meniu.specific%type;

BEGIN

select m.specific

into rez\_specific

from meniu m, comanda c, masa ma, sala s, locatie l

where c.masa\_id = ma.masa\_id

and ma.sala\_id = s.sala\_id

and s.locatie\_id = l.locatie\_id

and l.meniu\_id = m.meniu\_id

and c.comanda\_id = id\_comanda;

DBMS\_OUTPUT.PUT\_LINE('Specificul este ' || rez\_specific);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RAISE\_APPLICATION\_ERROR(-20000, 'Comanda data ca parametru nu exista!');

WHEN TOO\_MANY\_ROWS THEN RAISE\_APPLICATION\_ERROR(-20001, 'Restaurantul in care s-a facut comanda are mai multe meniuri specifice!');

WHEN OTHERS THEN RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare!');

END;

END pachet\_jvg;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Numarul de angajati este '|| pachet\_jvg.nr\_angajati\_by\_date(30, 3));

pachet\_jvg.specific\_restaurant(1001);

DBMS\_OUTPUT.PUT\_LINE('Telefonul celui mai nou angajat din sala data ca parametru este: '|| pachet\_jvg.telefon\_angajat(402));

pachet\_jvg.afiseaza\_angajati;

END;

/