Gestiunea unui salon

Badescu George-Gabriel

Grupa 241

Gestionarea unui salon

Aplicatia este una utila si moderna. Baza de date prezinta un sistem in care sunt inregistrati clientii cat si programarile acestora, angajatii si serviciile pe care acestia le presteaza. Pe langa istoricul programarilor, aplicatia va gestiona si cardurile de fidelitate ale clientilor, fiecare avand un cod propriu pe care acumuleaza puncta pe care le vor putea folosi pentru a beneficia de anumite oferte. Aplicatia pune la dispozitie si gestionarea unui stoc de produse deoarece salonul nostrum se ocupa si de comercializarea produselor cosmetic, iar aplicatia va retine in baza de date fiecare tranzactie efectuata.

Diagrama ERD:

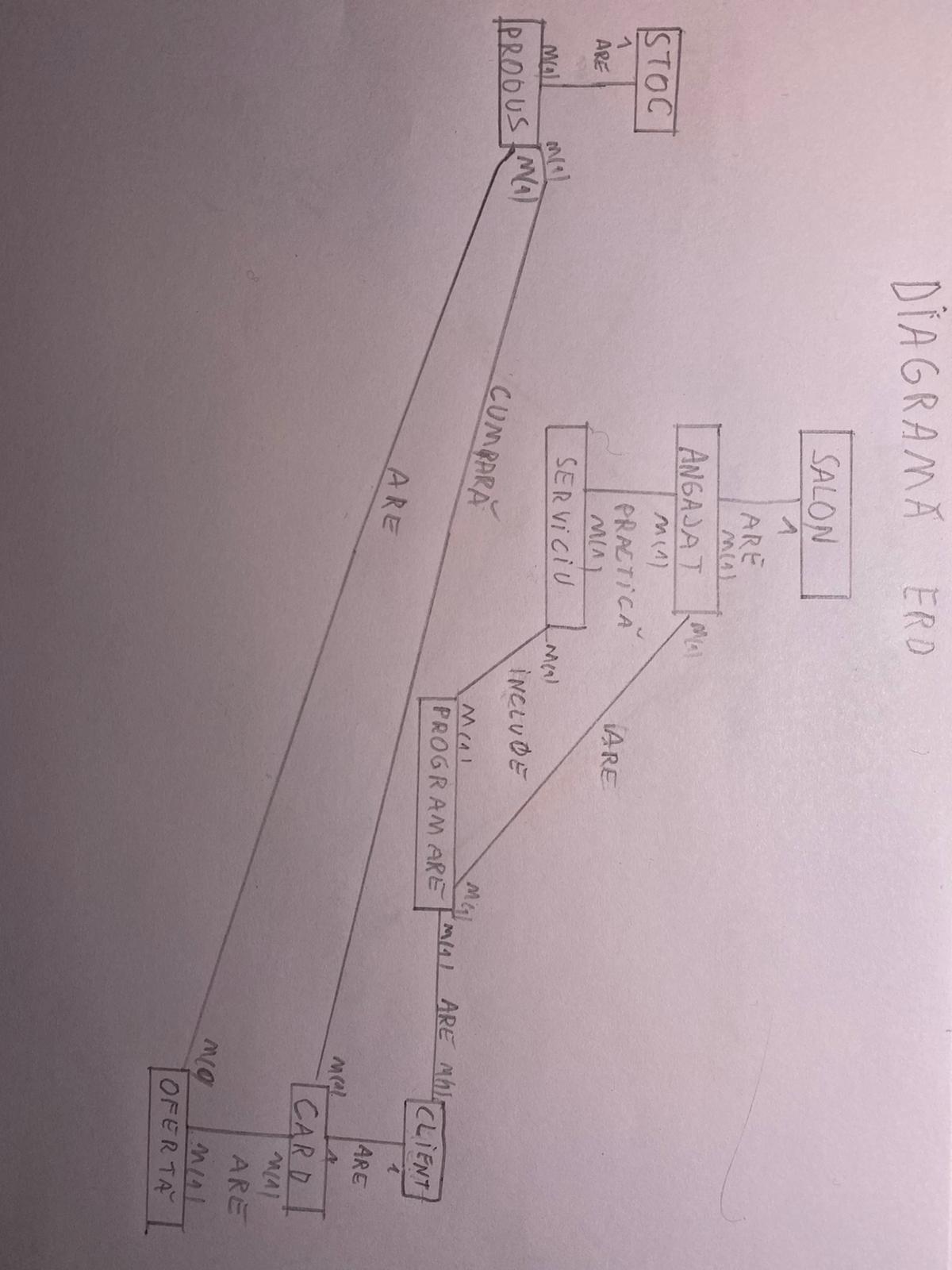
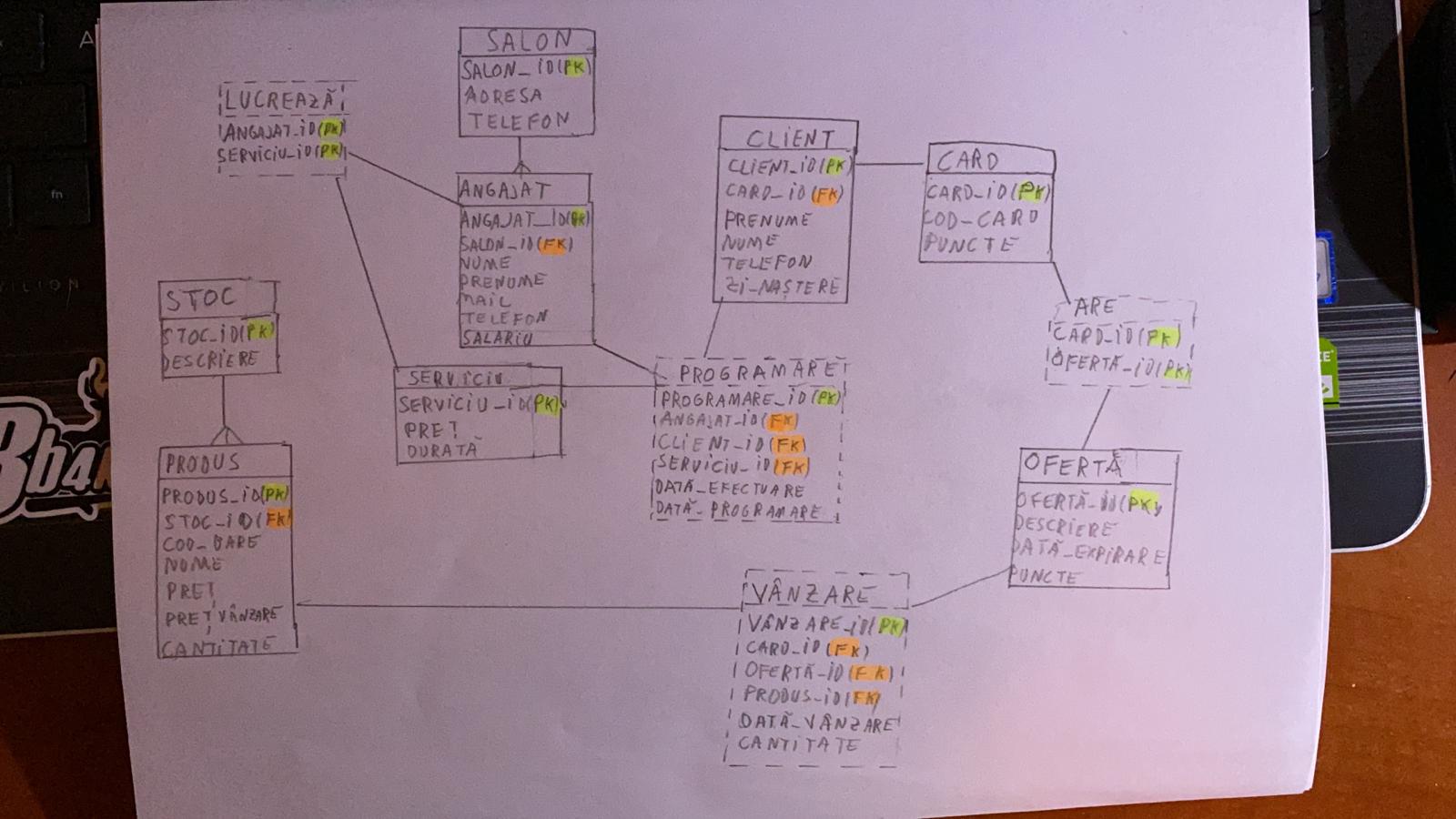


Diagrama conceptuala:



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

--==========================================================================================================

CREATE TABLE SALON

(

SALON\_ID NUMBER(4) NOT NULL

, NUME VARCHAR2(30) NOT NULL

, ADRESA VARCHAR2(200) NOT NULL

, TELEFON VARCHAR2(20) NOT NULL

, CONSTRAINT SALON\_PK PRIMARY KEY

(

SALON\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE ANGAJAT

(

ANGAJAT\_ID NUMBER NOT NULL

, SALON\_ID NUMBER NOT NULL

, NUME VARCHAR2(30) NOT NULL

, PRENUME VARCHAR2(30) NOT NULL

, EMAIL VARCHAR2(30) NOT NULL

, TELEFON VARCHAR2(20) NOT NULL

, SALARIU FLOAT

, CONSTRAINT TABLE1\_PK PRIMARY KEY

(

ANGAJAT\_ID

)

ENABLE

);

--==================================================================================================================

CREATE TABLE SERVICIU

(

SERVICIU\_ID VARCHAR2(30) NOT NULL

, PRET NUMBER NOT NULL

, DURATA NUMBER NOT NULL

, CONSTRAINT SERVICIU\_PK PRIMARY KEY

(

SERVICIU\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE LUCREAZA

(

ANGAJAT\_ID VARCHAR2(20) NOT NULL

, SERVICIU\_ID VARCHAR2(20) NOT NULL

, CONSTRAINT LUCREAZA\_PK PRIMARY KEY

(

ANGAJAT\_ID

, SERVICIU\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE STOC

(

STOC\_ID NUMBER NOT NULL

, DESCRIERE VARCHAR2(300)

, CONSTRAINT Stoc\_PK PRIMARY KEY

(

STOC\_ID

)

USING INDEX

(

CREATE UNIQUE INDEX Stoc\_PK1 ON STOC (STOC\_ID ASC)

)

ENABLE

);

--==========================================================================================================

CREATE TABLE PRODUS

(

PRODUS\_ID NUMBER NOT NULL

, STOC\_ID NUMBER NOT NULL

, COD\_BARE VARCHAR2(40) NOT NULL

, NUME VARCHAR2(40) NOT NULL

, PRET NUMBER NOT NULL

, PRET\_VANZARE NUMBER NOT NULL

,CANTITATE NUMBER NOT NULL

, CONSTRAINT PRODUS\_PK PRIMARY KEY

(

PRODUS\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE PROGRAMARE

(

PROGRAMARE\_ID NUMBER NOT NULL

, ANGAJAT\_ID NUMBER NOT NULL

, CLIENT\_ID NUMBER NOT NULL

, SERVICIU\_ID VARCHAR2(30) NOT NULL

, DATA\_EFECTUARE DATE NOT NULL

, DATA\_PROGRAMARE DATE NOT NULL

, CONSTRAINT PROGRAMARE\_PK PRIMARY KEY

(

PROGRAMARE\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE CLIENT

(

CLIENT\_ID NUMBER NOT NULL

, CARD\_ID NUMBER NOT NULL

, PRENUME VARCHAR2(20)

, NUME VARCHAR2(20)

, TELEFON VARCHAR2(20)

, MAIL VARCHAR2(20)

, ZI\_NASTERE VARCHAR2(20)

, CONSTRAINT CLIENT\_PK PRIMARY KEY

(

CLIENT\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE CARD

(

CARD\_ID NUMBER NOT NULL

, COD\_CARD VARCHAR2(20) NOT NULL UNIQUE

, PUNCTE NUMBER

, CONSTRAINT CARD\_PK PRIMARY KEY

(

CARD\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE OFERTA

(

OFERTA\_ID NUMBER NOT NULL

, DESCRIERE VARCHAR2(200)

, DATA\_EXPIRARE DATE

, PUNCTE NUMBER

, CONSTRAINT OFERTA\_PK PRIMARY KEY

(

OFERTA\_ID

)

ENABLE

);

--==========================================================================================================

CREATE TABLE ARE

(

CARD\_ID NUMBER NOT NULL

, OFERTA\_ID NUMBER NOT NULL

, CONSTRAINT ARE\_PK PRIMARY KEY

(

CARD\_ID

, OFERTA\_ID

)

USING INDEX

(

CREATE UNIQUE INDEX ARE\_PK1 ON ARE (CARD\_ID ASC, OFERTA\_ID ASC)

)

ENABLE

);

--==========================================================================================================

CREATE TABLE VANZARE

(

VANZARE\_ID NUMBER NOT NULL

, CARD\_ID NUMBER NOT NULL

, OFERTA\_ID NUMBER

, PRODUS\_ID NUMBER NOT NULL

, NUMAR\_PRODUSE NUMBER NOT NULL

, DATA\_VANZARE DATE NOT NULL

,CANTITATE

, CONSTRAINT VANZARE\_PK PRIMARY KEY

(

VANZARE\_ID

)

ENABLE

);

--=============================================================================================

ALTER TABLE ANGAJAT ADD CONSTRAINT ANGAJAT\_FK1 FOREIGN KEY (SALON\_ID) REFERENCES SALON(SALON\_ID);

alter table PROGRAMARE add constraint programare\_fk foreign key (ANGAJAT\_ID) references ANGAJAT (ANGAJAT\_ID);

alter table PROGRAMARE add constraint programare\_fk1 foreign key(CLIENT\_ID) references CLIENT (CLIENT\_ID);

alter table PROGRAMARE add constraint programare\_fk2 foreign key(SERVICIU\_ID) references SERVICIU (SERVICIU\_ID);

alter table CLIENT add constraint client\_fk foreign key(CARD\_ID) references CARD(CARD\_ID);

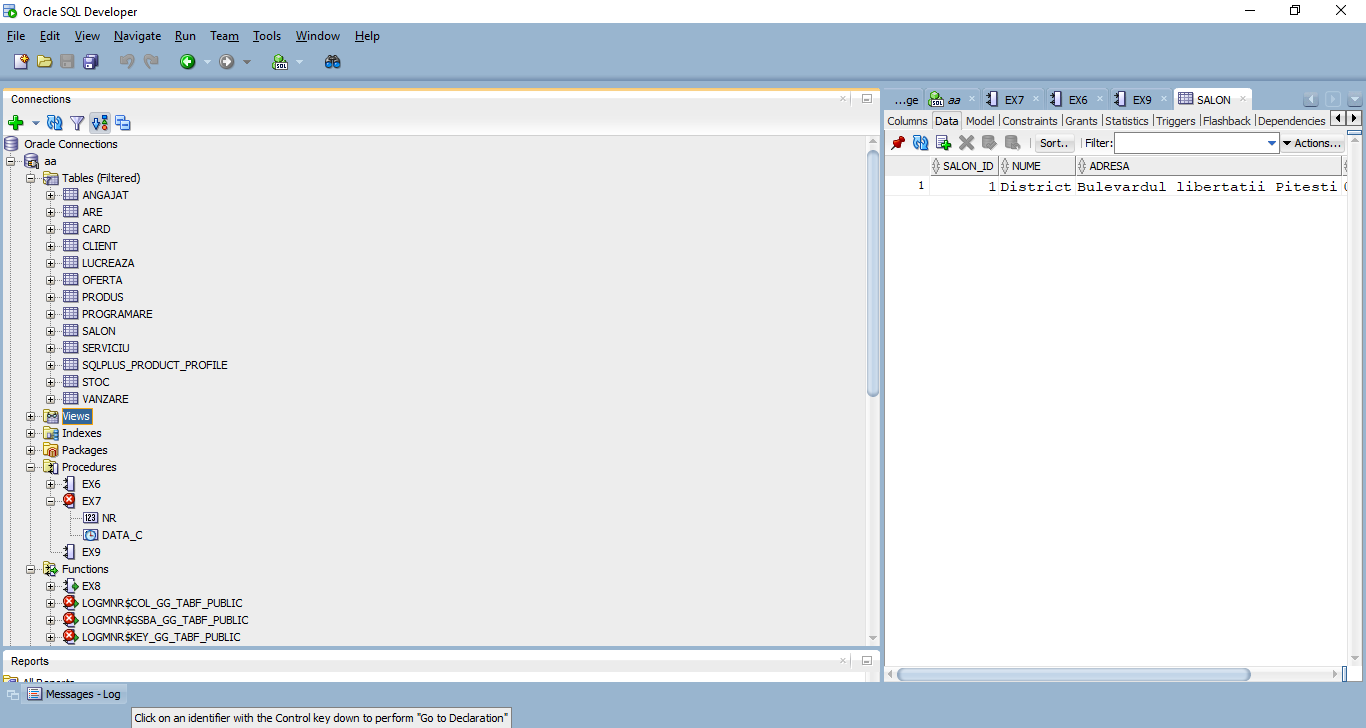
alter table VANZARE add constraint vanzare\_fk foreign key(CARD\_ID) references CARD (CARD\_ID);

alter table VANZARE add constraint vanzare\_fk1 foreign key(OFERTA\_ID) references OFERTA(OFERTA\_ID);

alter table VANZARE add constraint vanzare\_fk2 foreign key(PRODUS\_ID) references PRODUS(PRODUS\_ID);

ALTER TABLE PRODUS ADD CONSTRAINT PRODUS\_FK1 FOREIGN KEY (STOC\_ID) REFERENCES STOC(STOC\_ID);

--==============================================================================================================================



5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

Insert into salon(salon\_id,nume,adresa,telefon)

values (1,'District','Bulevardul libertatii Pitesti','0755234123');

insert into angajat(angajat\_id,salon\_id,nume,prenume,email,telefon,salariu)

values(2,1,'David','Albert','dalbert@gmail.com','0745623431',650);

insert into angajat(angajat\_id,salon\_id,nume,prenume,email,telefon,salariu)

values(3,1,'Badescu','Gabriel','gabi\_badescu@email.com','0745632344',750);

insert into angajat(angajat\_id,salon\_id,nume,prenume,email,telefon,salariu)

values(4,1,'Dabu','Alexandru','dalex@gmail.com','0753263431',900);

insert into angajat(angajat\_id,salon\_id,nume,prenume,email,telefon,salariu)

values(5,1,'Daineanu','Denis','ddenis@gmail.com','0751112331',1200);

insert into serviciu(serviciu\_id,pret,durata)

values('tuns',25,30);

insert into serviciu(serviciu\_id,pret,durata)

values('barbierit',10,15);

insert into serviciu(serviciu\_id,pret,durata)

values('vopsit',30,40);

insert into serviciu(serviciu\_id,pret,durata)

values('aranjat',10,10);

insert into lucreaza(angajat\_id,serviciu\_id)

values(4,'tuns');

insert into lucreaza(angajat\_id,serviciu\_id)

values(4,'barbierit');

insert into lucreaza(angajat\_id,serviciu\_id)

values(4,'vopsit');

insert into lucreaza(angajat\_id,serviciu\_id)

values(5,'tuns');

insert into lucreaza(angajat\_id,serviciu\_id)

values(3,'aranjat');

insert into lucreaza(angajat\_id,serviciu\_id)

values(3,'vopsit');

insert into lucreaza(angajat\_id,serviciu\_id)

values(2,'tuns');

insert into lucreaza(angajat\_id,serviciu\_id)

values(2,'aranjat');

insert into lucreaza(angajat\_id,serviciu\_id)

values(5,'aranjat');

insert into lucreaza(angajat\_id,serviciu\_id)

values(5,'barbierit');

insert into stoc(stoc\_id,descriere)

values(6,'Stocul de produse al salonului District');

insert into produs(produs\_id,stoc\_id,cod\_bare,nume,pret,pret\_vanzare,cantitate)

values(7,6,'90002135','fixativ volum','30','50','100');

insert into produs(produs\_id,stoc\_id,cod\_bare,nume,pret,pret\_vanzare,cantitate)

values(8,6,'80004574','ceara','20','40','120');

insert into produs(produs\_id,stoc\_id,cod\_bare,nume,pret,pret\_vanzare,cantitate)

values(9,6,'12345768','sampon uscat','20','60','1');

insert into produs(produs\_id,stoc\_id,cod\_bare,nume,pret,pret\_vanzare,cantitate)

values(10,6,'36329508','gel','10','25','50');

insert into card(card\_id,cod\_card,puncte)

values(43,'24513',12);

insert into card(card\_id,cod\_card,puncte)

values(44,'11114',4);

insert into card(card\_id,cod\_card,puncte)

values(45,'33345',0);

insert into card(card\_id,cod\_card,puncte)

values(46,'12045',5);

insert into card(card\_id,cod\_card,puncte)

values(47,'00452',3);

insert into oferta(oferta\_id,descriere,data\_expirare,puncte)

values(11,'oferta valabila cu ocazia zilei de nastere',null,1);

insert into oferta(oferta\_id,descriere,data\_expirare,puncte)

values(12,'oferta de 20% la tuns',to\_date('17-12-2021', 'dd-mm-yyyy'),3);

insert into oferta(oferta\_id,descriere,data\_expirare,puncte)

values(13,'vopsit 1+1 gratis',to\_date('02-06-2021', 'dd-mm-yyyy'),10);

--are ramane gol momentan

insert into client(client\_id,card\_id,prenume,nume,telefon,mail,zi\_nastere)

values(14,47,'Tudor','Bara','0723467890',null,to\_date('02-01-1999', 'dd-mm-yyyy'));

insert into client(client\_id,card\_id,prenume,nume,telefon,mail,zi\_nastere)

values(15,46,'Marian','Bigan','0723527953',null,to\_date('12-11-2002', 'dd-mm-yyyy'));

insert into client(client\_id,card\_id,prenume,nume,telefon,mail,zi\_nastere)

values(16,43,'Dragos','Mihu','0721237890','dmihu@gmail.com',to\_date('12-04-1992', 'dd-mm-yyyy'));

insert into client(client\_id,card\_id,prenume,nume,telefon,mail,zi\_nastere)

values(17,45,'Andrei','Neacsu','0736325689','nandrei@gmail.com',to\_date('17-10-1999', 'dd-mm-yyyy'));

insert into client(client\_id,card\_id,prenume,nume,telefon,mail,zi\_nastere)

values(38,44,'Alex','Stan','0721485932','salex@gmail.com',to\_date('17-08-1994', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(18,5,14,'barbierit',to\_date('03-01-2021', 'dd-mm-yyyy'),to\_date('12-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(19,4,16,'tuns',to\_date('02-01-2021', 'dd-mm-yyyy'),to\_date('15-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(39,4,38,'tuns',to\_date('01-01-2021', 'dd-mm-yyyy'),to\_date('06-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(40,4,15,'barbierit',to\_date('12-02-2020', 'dd-mm-yyyy'),to\_date('14-03-2020', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(41,4,15,'vopsit',to\_date('06-12-2020', 'dd-mm-yyyy'),to\_date('23-12-2020', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(20,3,16,'vopsit',to\_date('01-05-2021', 'dd-mm-yyyy'),to\_date('12-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(21,4,17,'barbierit',to\_date('04-01-2021', 'dd-mm-yyyy'),to\_date('16-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(22,5,15,'barbierit',to\_date('11-01-2021', 'dd-mm-yyyy'),to\_date('14-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(23,2,17,'tuns',to\_date('11-12-2020', 'dd-mm-yyyy'),to\_date('02-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(24,2,17,'tuns',to\_date('11-12-2020', 'dd-mm-yyyy'),to\_date('12-12-2020', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(25,3,16,'aranjat',to\_date('01-05-2021', 'dd-mm-yyyy'),to\_date('12-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(26,3,15,'vopsit',to\_date('11-11-2020', 'dd-mm-yyyy'),to\_date('18-11-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(27,4,14,'barbierit',to\_date('01-01-2021', 'dd-mm-yyyy'),to\_date('10-01-2021', 'dd-mm-yyyy'));

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(42,5,17,'tuns',to\_date('11-01-2020', 'dd-mm-yyyy'),to\_date('12-01-2020', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(28,46,null,9,2,to\_date('03-01-2021', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(38,46,null,7,1,to\_date('02-01-2021', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(29,45,null,10,8,to\_date('12-12-2020', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(30,45,12,8,1,to\_date('03-01-2021', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(31,47,null,7,10,to\_date('10-12-2020', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare,cantitate)

values(32,47,null,8,1,to\_date('10-11-2020', 'dd-mm-yyyy'),2);

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(33,46,13,7,3,to\_date('04-12-2020', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare,cantitate)

values(34,46,11,8,1,to\_date('16-11-2020', 'dd-mm-yyyy'),4);

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare,cantitate)

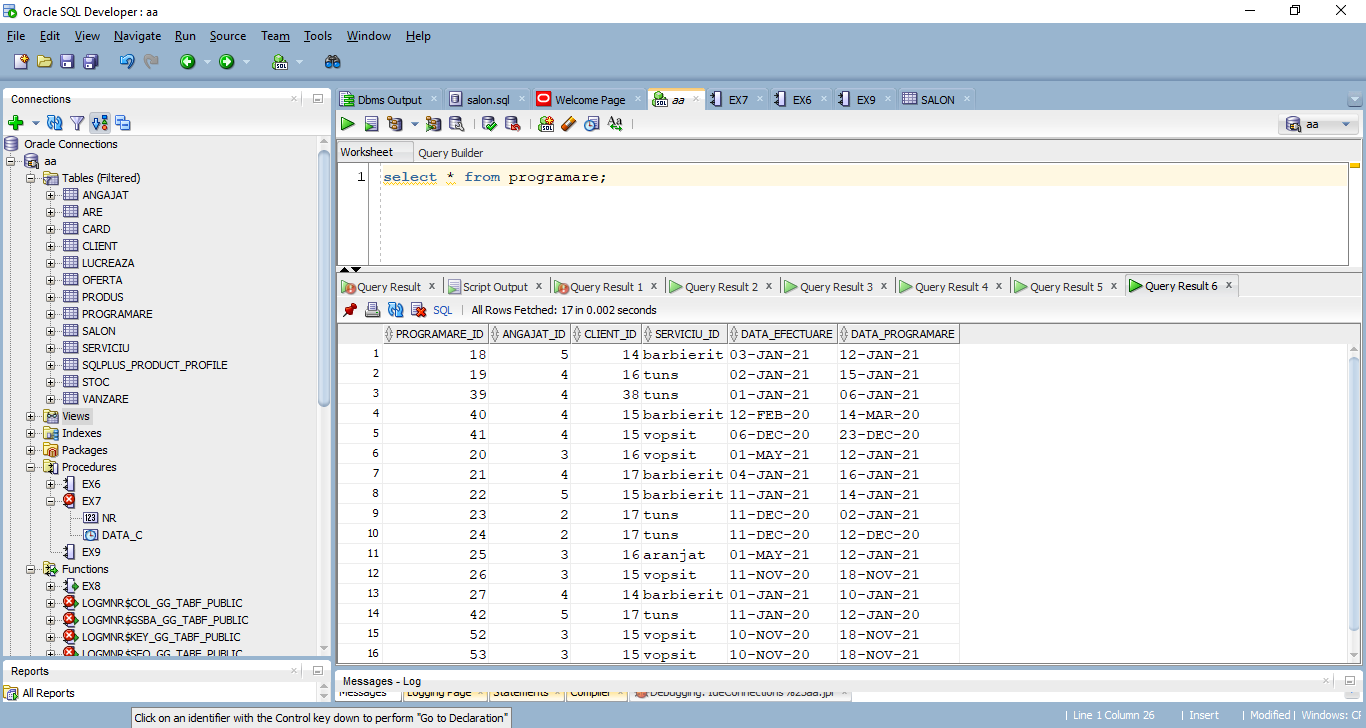
values(35,45,null,10,1,to\_date('13-10-2020', 'dd-mm-yyyy'),3);

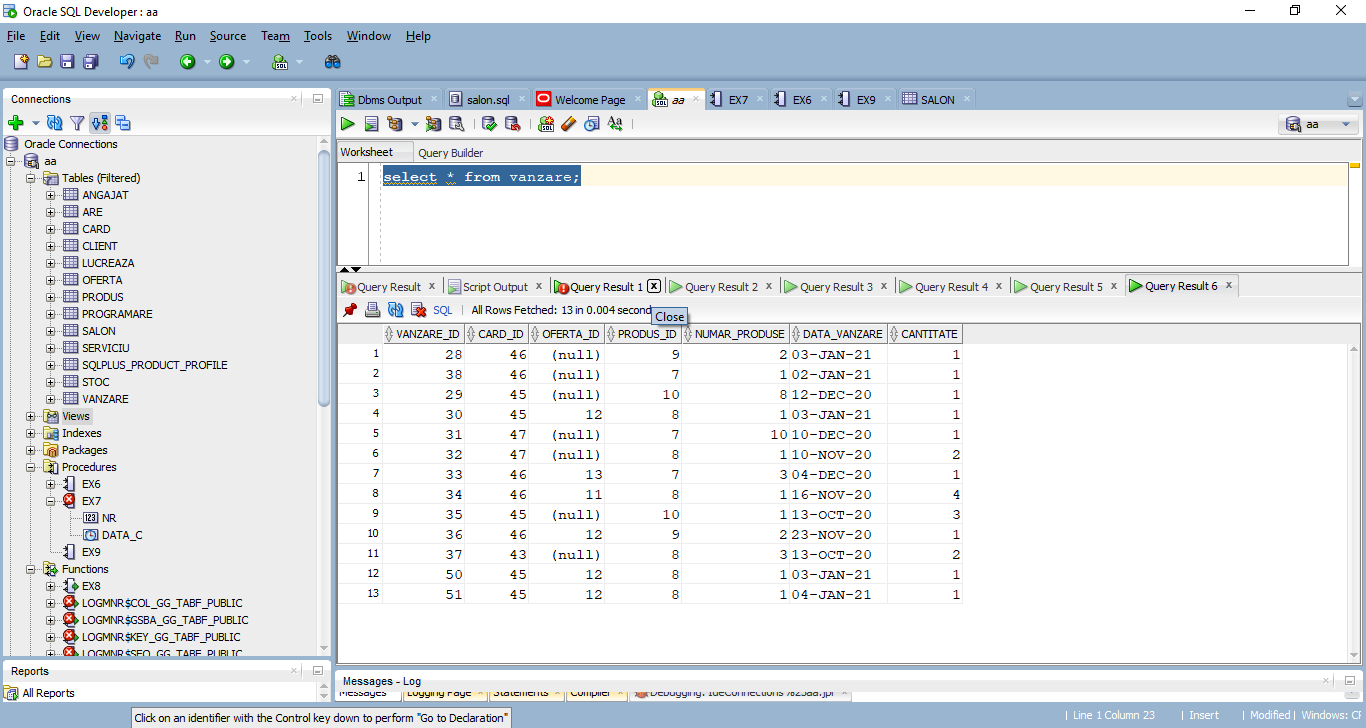
insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(36,46,12,9,2,to\_date('23-11-2020', 'dd-mm-yyyy'));

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare,cantitate)

values(37,43,null,8,3,to\_date('13-10-2020', 'dd-mm-yyyy'),2);





6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

--ex6

--sa se afiseze lista de clienti a ficarui angajat pentru un interval de timp dat , iar daca acesta

--a avut un numar de clienti dat el va primi o marire de 5% a salariului

CREATE OR REPLACE PROCEDURE ex6 (nr in number ,

data\_c in date ) IS

TYPE rec IS RECORD (nume\_angajat angajat.nume%type,prenume\_angajat angajat.prenume%type, nr\_clienti number,sal number);

type v1 is varray(10) of rec;

vector v1;

sal\_nou float:=0;

begin

select a.nume,a.prenume,count(client\_id) nr, a.salariu salariu

BULK COLLECT INTO vector

from angajat a join programare p using (angajat\_id) join client c using(client\_id)

where p.data\_programare<to\_date('12-01-2021', 'dd-mm-yyyy')

group by a.nume,a.prenume, a.salariu

order by a.nume;

for i in 1..vector.count loop

if vector(i).nr\_clienti>=nr then

update angajat

set salariu=salariu+5/100\*salariu

where nume=vector(i).nume\_angajat and prenume=vector(i).prenume\_angajat;

select salariu

into sal\_nou

from angajat

where nume=vector(i).nume\_angajat and prenume=vector(i).prenume\_angajat;

DBMS\_OUTPUT.PUT\_LINE ('Angajatul '||vector(i).nume\_angajat ||' '||

vector(i).prenume\_angajat|| ' a avut '|| vector(i).nr\_clienti||

' de pe data de '|| data\_c|| ' pana in prezent de accea a primit o marire de salariu de la ' ||vector(i).sal||

' la '||sal\_nou);

else

DBMS\_OUTPUT.PUT\_LINE ('Angajatul '||vector(i).nume\_angajat ||' '||

vector(i).prenume\_angajat|| ' a avut '|| vector(i).nr\_clienti||

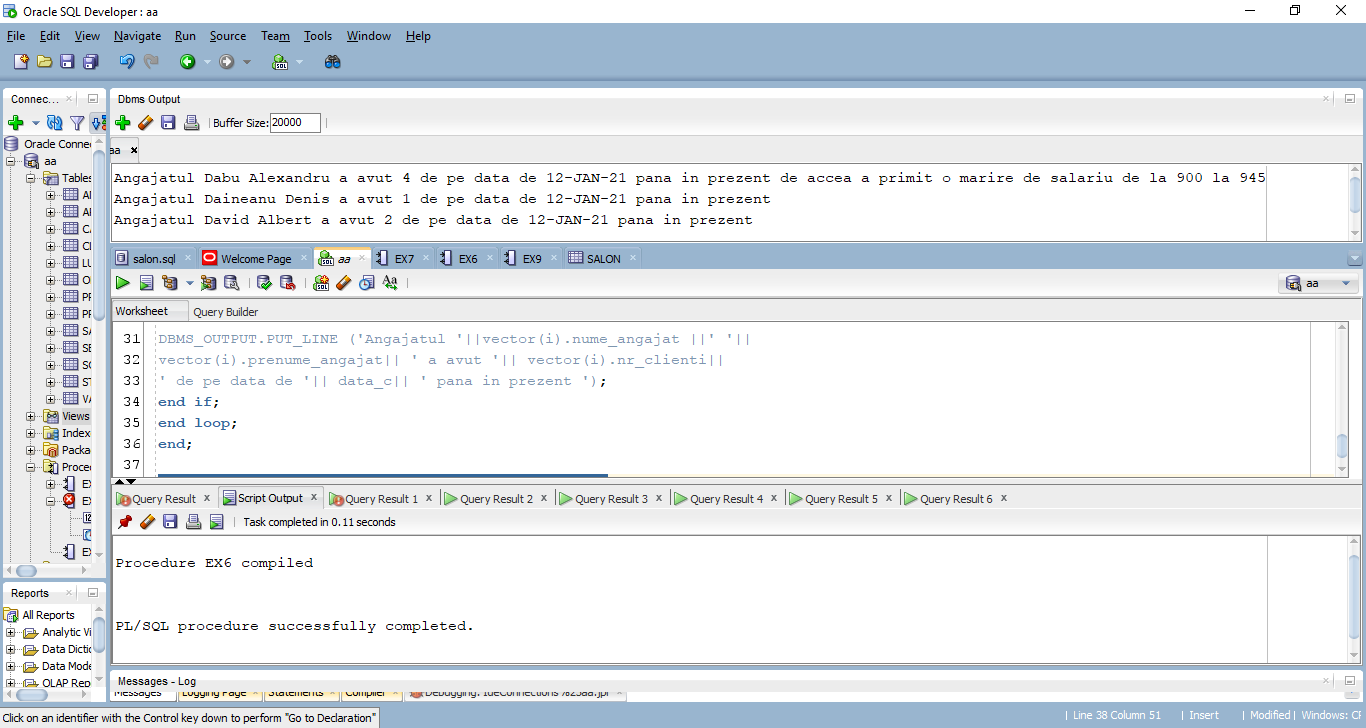
' de pe data de '|| data\_c|| ' pana in prezent ');

end if;

end loop;

end;

execute ex6(3,to\_date('12-01-2021','dd-mm-yyyy'));



7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

--sa se afiseze numele si prenumle top 3 cei mai fideli clienti si de cate ori a venit fiecare

--(daca 2 clienti au venit de un numar egal de ori, amandoi vor ocupa aceeasi pozatiein top)

CREATE OR REPLACE PROCEDURE ex7 IS

ind number(2) :=1;

last\_val number:=-1;

vizite number:=0;

nume\_1 client.nume%type;

prenume\_1 client.prenume%type;

cursor c is select count(\*) nr ,nume, prenume

from programare join client using (client\_id)

group by nume,prenume

order by nr desc;

BEGIN

open c;

loop

fetch c into vizite, nume\_1, prenume\_1;

exit when c%notfound or ind=4;

if last\_val=-1 and vizite!=null then

DBMS\_OUTPUT.PUT\_LINE('Locul '||ind|| ' care a avut '|| vizite|| ' vizite este ocupat de ' );

DBMS\_OUTPUT.PUT\_LINE(nume\_1||' '|| prenume\_1);

else

if last\_val=vizite then

DBMS\_OUTPUT.PUT\_LINE(nume\_1||' '|| prenume\_1);

ind:=ind-1;

else

DBMS\_OUTPUT.PUT\_LINE('Locul '||ind|| ' care a avut '|| vizite|| ' vizite este ocupat de ' );

DBMS\_OUTPUT.PUT\_LINE(nume\_1||' '|| prenume\_1);

end if;

end if;

last\_val:=vizite;

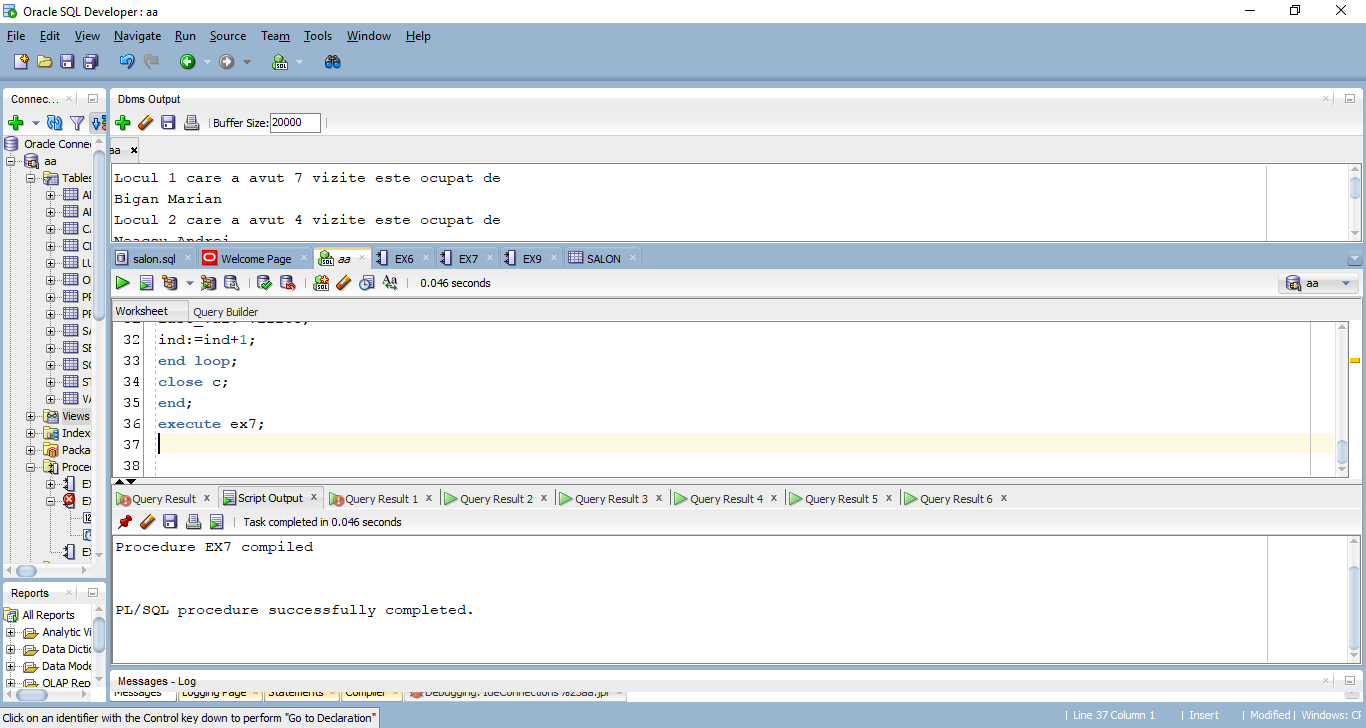
ind:=ind+1;

end loop;

close c;

end;

execute ex7;



8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--ex8

--sa se afiseze profitul obtinut in urma vanzarilor fara oferte si sase afiseze codurile cardurilor pentru produsul

--care a adus profitul cel mai mare

CREATE OR REPLACE function ex8 return number IS

profit number(3):=0;

ultim varchar2(50):='.';

ultim\_2 varchar2(50):='.';

profit\_max number:=0;

profit\_tot number :=0;

excep EXCEPTION;

type car is varray(20) of varchar2(50);

list\_car car;

begin

for i in(select p.pret p\_original,p.pret\_vanzare p\_vanzare,p.nume nume,v.cantitate cantitate ,

count(v.vanzare\_id) nr\_c,c.cod\_card cod

from produs p join vanzare v using(produs\_id) join card c using (card\_id)

group by p.pret,p.pret\_vanzare,p.nume,v.cantitate,c.cod\_card

order by nume) loop

profit\_tot:=profit\_tot+(i.p\_vanzare-i.p\_original)\*i.cantitate\*i.nr\_c;

if(ultim='.')then

ultim:=i.nume;

profit\_max:=(i.p\_vanzare-i.p\_original)\*i.cantitate\*i.nr\_c;

ultim\_2:=i.nume;

profit:=profit\_max;

end if;

if(ultim=i.nume)then

profit:=profit+(i.p\_vanzare-i.p\_original)\*i.cantitate\*i.nr\_c;

if profit>profit\_max then

profit\_max:=profit;

ultim\_2:=i.nume;

end if;

else

profit:=(i.p\_vanzare-i.p\_original)\*i.cantitate\*i.nr\_c;

ultim:=i.nume;

end if;

end loop;

DBMS\_OUTPUT.PUT\_LINE('Profitul obtinut in urma produselor ce nu au oferta este '|| profit\_tot);

DBMS\_OUTPUT.PUT\_LINE('Cel mai profitabil produs a fost '|| ultim\_2|| ' Cumparat de ');

select distinct c.cod\_card cod1

bulk collect into list\_car

from vanzare v join card c using (card\_id) join produs p using(produs\_id)

where nume=ultim\_2;

if list\_car.count() = 0 then raise excep;

end if;

for i in 1..list\_car.count() loop

DBMS\_OUTPUT.PUT\_LINE(list\_car(i));

end loop;

return 0;

EXCEPTION

when excep then

return -1;

end;

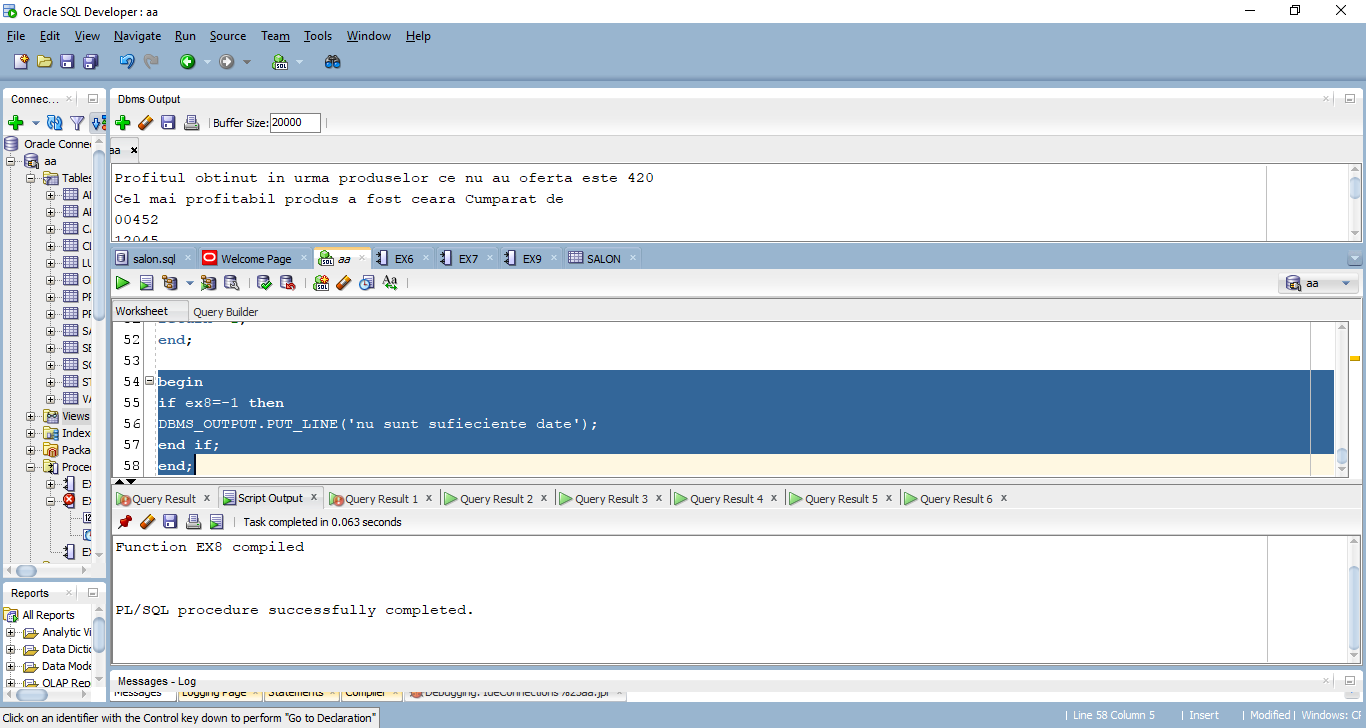
begin

if ex8=-1 then

DBMS\_OUTPUT.PUT\_LINE('nu sunt sufieciente date');

end if;

end;



9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--ex9

--sa se afisee fiecare angajat cu cel mai fidel client, suma totala cheltuita de acesta la angajatul

--respectiv si care este media punctelor de pe cardurile clientiolor

create or replace NONEDITIONABLE PROCEDURE ex9 IS

ultim varchar2(50):='.';

ok number:=0;

pct number:=0;

begin

for i in (select a.nume nume\_a,c.nume nume\_c,sum(s.pret) suma,count(c.nume) nr ,ca.puncte puncte,ca.cod\_card cod

from programare p join client c using(client\_id) join serviciu s using(serviciu\_id) join angajat a using(angajat\_id)

join card ca using (card\_id)

group by a.nume,c.nume ,ca.puncte,ca.cod\_card

order by 1,4 desc) loop

pct:=i.puncte;

if(ultim='.')then

ultim:=i.nume\_a;

end if;

if(ultim!=i.nume\_a or (ultim=i.nume\_a and ok=0)) then

update card

set puncte=puncte+2

where cod\_card=i.cod;

DBMS\_OUTPUT.PUT\_LINE('Clientul '||i.nume\_c|| ' a apelat cel mai des la angajatul '|| i.nume\_a|| '(de '||i.nr

||'ori'||'iar acesta a primit drept recompensa 2 puncte pe cardul sau de fidelitate. El are acum '||i.puncte||

' iar inainte avea '|| pct);

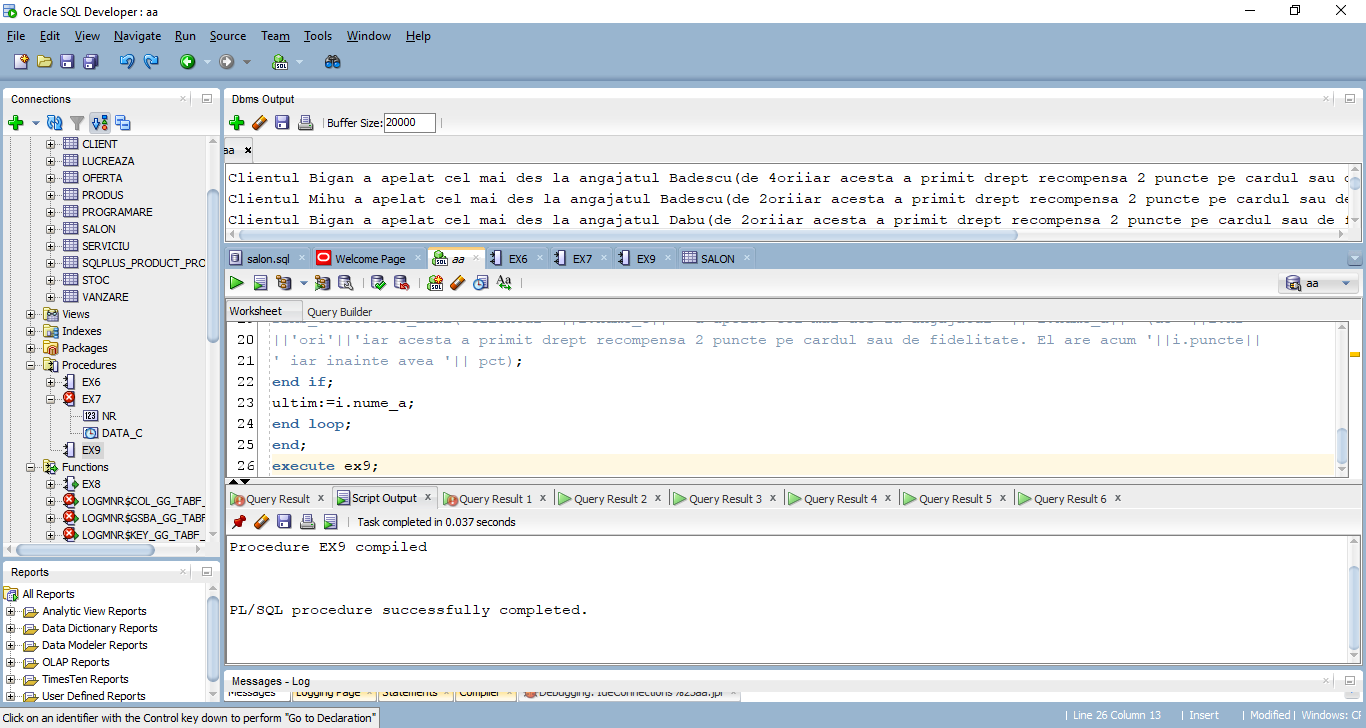
end if;

ultim:=i.nume\_a;

end loop;

end;

execute ex9;



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

--ex11 Sa se defineasca un trigger care nu permite operatiile de insert,update delete in programare in intervalul

--20:00-08:00

CREATE OR REPLACE TRIGGER ex\_10

BEFORE INSERT OR UPDATE OR DELETE ON programare

BEGIN

IF (TO\_CHAR(SYSDATE,'HH24') not BETWEEN 20 AND 08)

THEN

RAISE\_APPLICATION\_ERROR(-20001,'programarea nu poate fi facuta deoarece salonul este inchis la ora respectiva');

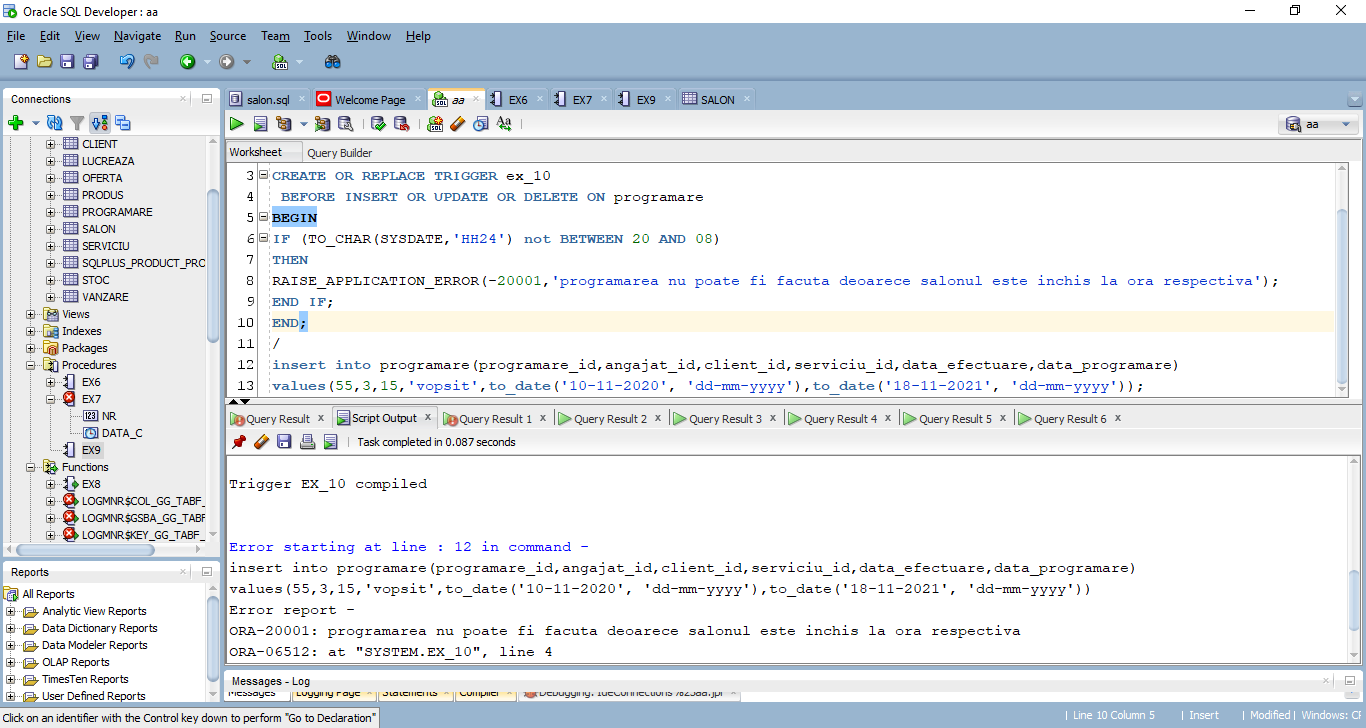
END IF;

END;

/

insert into programare(programare\_id,angajat\_id,client\_id,serviciu\_id,data\_efectuare,data\_programare)

values(55,3,15,'vopsit',to\_date('10-11-2020', 'dd-mm-yyyy'),to\_date('18-11-2021', 'dd-mm-yyyy'));



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

--11 Definiti un trigger care in momentul inserarii unei vanzari, daca are oferta , se vor lua punctele de pe cardul

--clientului care a cumparat produsul

CREATE OR REPLACE PROCEDURE modifica\_pct

(ofert number,

car number)

is

ofe number:=0;

BEGIN

select puncte

into ofe

from oferta

where oferta\_id=ofert;

UPDATE card

SET puncte = puncte-ofe

WHERE card\_id =car;

END;

CREATE OR REPLACE TRIGGER ex11

AFTER INSERT ON vanzare

for each row

BEGIN

if inserting then

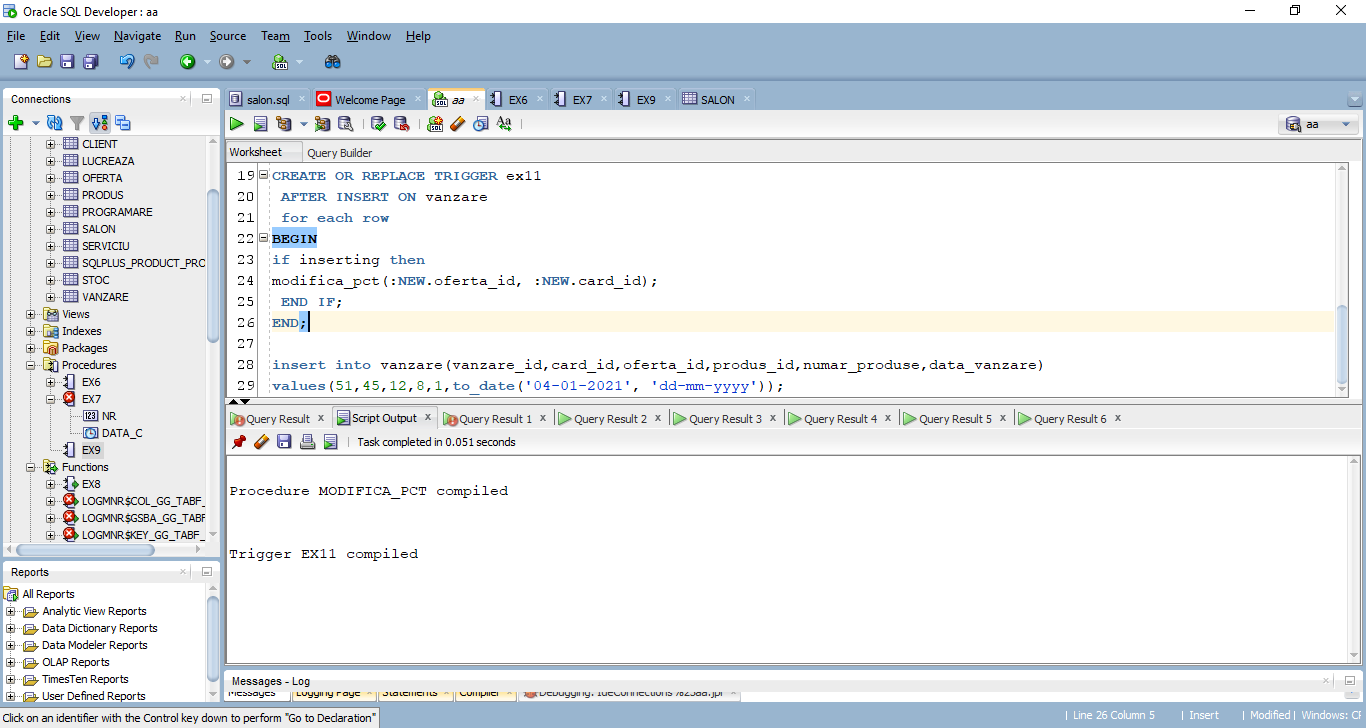
modifica\_pct(:NEW.oferta\_id, :NEW.card\_id);

END IF;

END;

insert into vanzare(vanzare\_id,card\_id,oferta\_id,produs\_id,numar\_produse,data\_vanzare)

values(51,45,12,8,1,to\_date('04-01-2021', 'dd-mm-yyyy'));



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

--ex12

--Definiți un declanșator care să introducă date în acest tabel după ce utilizatorul a folosit o

--comandă LDD

CREATE TABLE ex12

(nume\_bd VARCHAR2(50),

user\_logat VARCHAR2(30),

eveniment VARCHAR2(20),

tip\_obiect\_referit VARCHAR2(30),

nume\_obiect\_referit VARCHAR2(30),

data TIMESTAMP(3));

CREATE OR REPLACE TRIGGER ex12

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO audit\_user

VALUES (SYS.DATABASE\_NAME, SYS.LOGIN\_USER,

SYS.SYSEVENT, SYS.DICTIONARY\_OBJ\_TYPE,

SYS.DICTIONARY\_OBJ\_NAME, SYSTIMESTAMP(3));

END;

/

CREATE TABLE tabel (coloana\_1 number(2));

ALTER TABLE tabel ADD (coloana\_2 number(2));

INSERT INTO tabel VALUES (1,2);

CREATE INDEX ind\_tabel ON tabel(coloana\_1);

SELECT \* FROM audit\_user;

