Gestiunea unui magazin de inghetata

Proiect sisteme de baze de date

Rosca Maria Alexandra

Grupa 405

a) Scurta prezentare

Cati dintre noi nu au intrat intr-un magazin de inghetata si privind vitrinele colorate, simtind aerul rece si mirosul fructat al sutelor de sortimente, vazand zambetele luminate ale oamenilor, ne-am gandit in sinea noastra „Da! As putea sa fac asta!”.

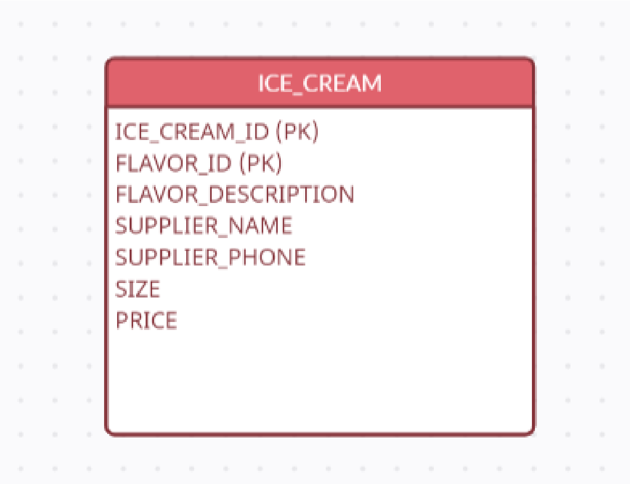
Ei bine, de la vis la realitate nu este un pas atat de mare. Pe langa finantatare pentru deschiderea unei astfel de afaceri, alegerea locatiei, alegerea producatorilor si alte astfel de „detalii”, avem nevoie, desigur, ca in cazul oricarui magazin, de o structura organizatorica bine pusa la punct in ceea ce priveste angajatii, clientii, comenzile si, bineinteles, INGHETATA!

In cele ce urmeaza voi prezenta un exemplu de baza de date ce poate fi folosita pentru o astfel de afacere, pornind de la idei simple si prelucrandu-le in functie de nevoile pe care le are magazinul nostru.

b) Schema conceptuala

Vom incepe cu cateva necesitati de baza.

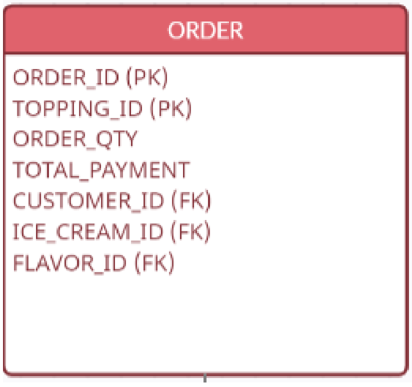
Pentru ce obiecte avem nevoie sa tinem evidenta? Desigur, sortimentele de inghetata! Avem nevoie de un cod pentru fiecare sortiment si desigur fiecare dintre acestea poate avea mai multe arome (vanilie, ciocolata, capsuni) asa ca cele doua chei primare vor fi codul si aroma.

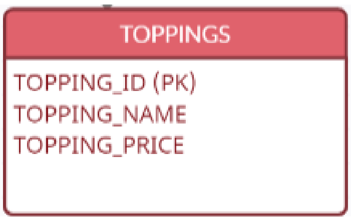


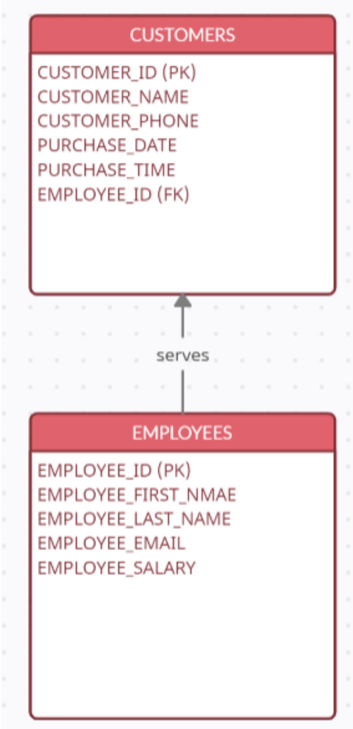
ICE_CREAM

ICE_CREAM_ID (PK)
FLAVOR_ID (PK)
FLAVOR_DESCRIPTION
SUPPLIER_NAME
SUPPLIER_PHONE
SIZE
PRICE

Avem nevoie si de un tabel ce va tine evidenta comenzilor (cod de comanda, cantitate, un tip de inghetata, aroma, valoarea acesteia). Pentru a tine evidenta persoanei ce a dat comanda vom introduce si un cod de client. Pentru o comanda de inghetata putem avea mai multe toppinguri, deci vom introduce si cheia TOPPING_ID. Desigur, vom tine evidenta toppingurilor intr-un tabel separat.



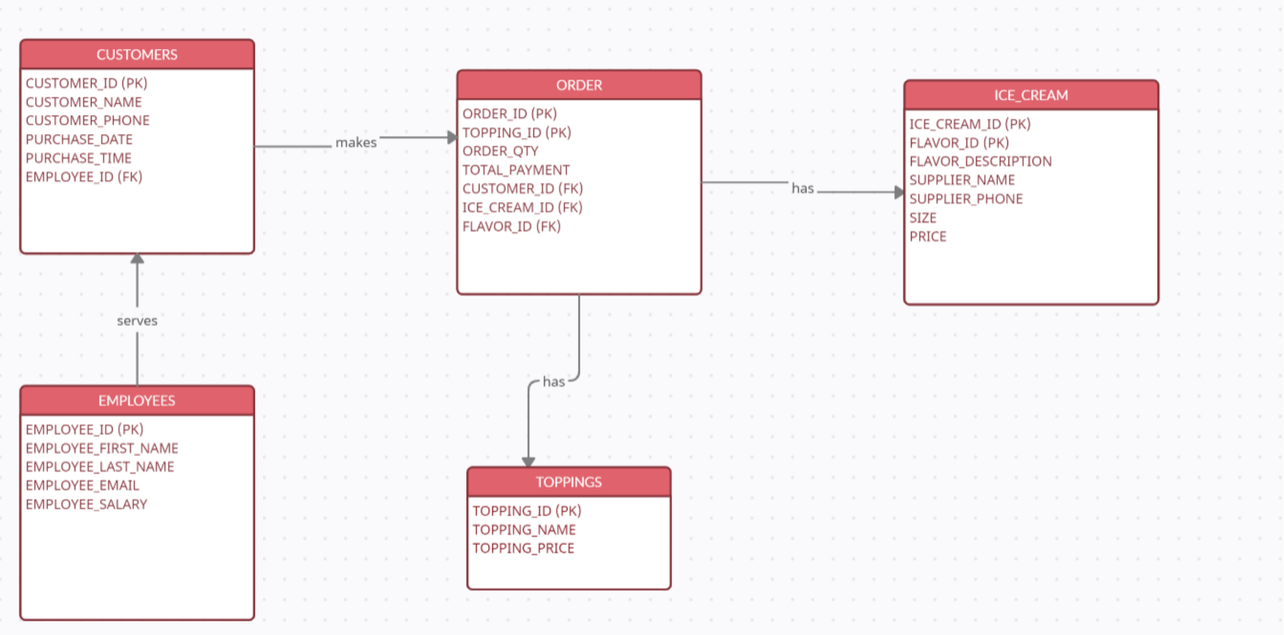




Vom tine, desigur si evidenta clientilor si a angajatilor. Angajatii vor servi diversi clienti, deci putem de la inceput sa definim relatia dintre acestia astfel, folosind cheia externa cod_angajat.

Despre angajati vom avea nevoie de diferite date printre care si salariul.

Dupa ce am definit entitatile, vom defini si relatiile dintre acestea si diagrama ERD. O prima schema conceptuala rezultata este:



CUSTOMERS
CUSTOMER_ID (PK)
CUSTOMER_NAME
CUSTOMER_PHONE
PURCHASE_DATE
PURCHASE_TIME
EMPLOYEE_ID (FK)

ORDER
ORDER_ID (PK)
TOPPING_ID (PK)
ORDER_QTY
TOTAL_PAYMENT
CUSTOMER_ID (FK)
ICE_CREAM_ID (FK)
FLAVOR_ID (FK)

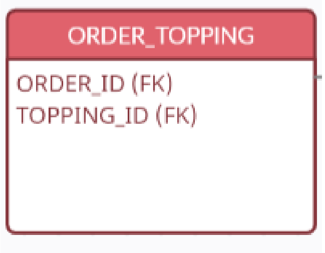
ICE_CREAM
ICE_CREAM_ID (PK)
FLAVOR_ID (PK)
FLAVOR_DESCRIPTION
SUPPLIER_NAME
SUPPLIER_PHONE
SIZE
PRICE

EMPLOYEES
EMPLOYEE_ID (PK)
EMPLOYEE_FIRST_NAME
EMPLOYEE_LAST_NAME
EMPLOYEE_EMAIL
EMPLOYEE_SALARY

TOPPINGS
TOPPING_ID (PK)
TOPPING_NAME
TOPPING_PRICE

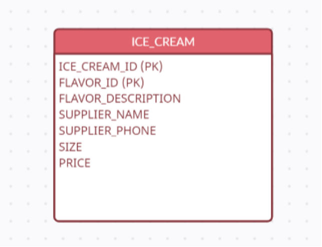
c) Rafinarea schemei conceptuale

Dupa cum observam mai sus, entitatea ORDER are ca si cheie primara coloanele ORDER_ID si TOPPING_ID. Acest lucru este cauzat de faptul ca un client poate comanda o inghetata cu toppinguri diferite. Pentru o mai buna structurare a bazei de date, vom folosi un tabel de legatura intre entitatea ORDER si entitatea TOPPINGS ce va contine cate o referinta catre fiecare dintre cele doua tabele.

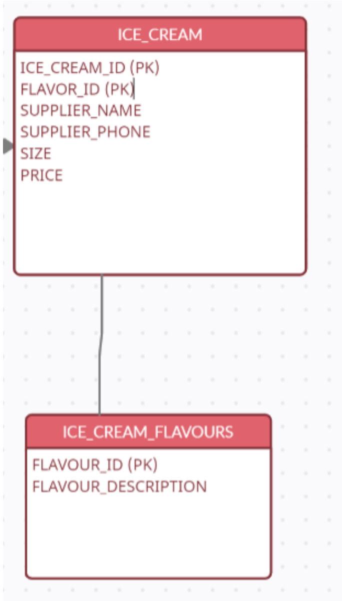


ORDER_TOPPING
ORDER_ID (FK)
TOPPING_ID (FK)

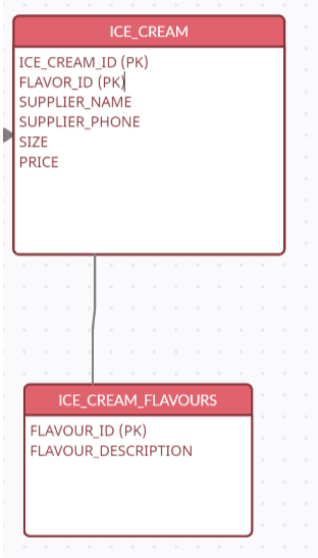
Pentru entitatea ICE_CREAM, observam ca o prima problema ca pentru un anumit tip de inghetata putem avea mai multe arome, iar descrierea acesteia se afla tot in cadrul tabelului definit, desi ea depinde doar de coloana FLAVOR_ID. O solutie simpla este sa cream un nou tabel ce va tine evidenta aromelor si descrierilor acestora, iar in tabelul principal vom mentine doar referinta catre aceasta pentru a nu il incarca inutil (forma FN1 -> FN2).



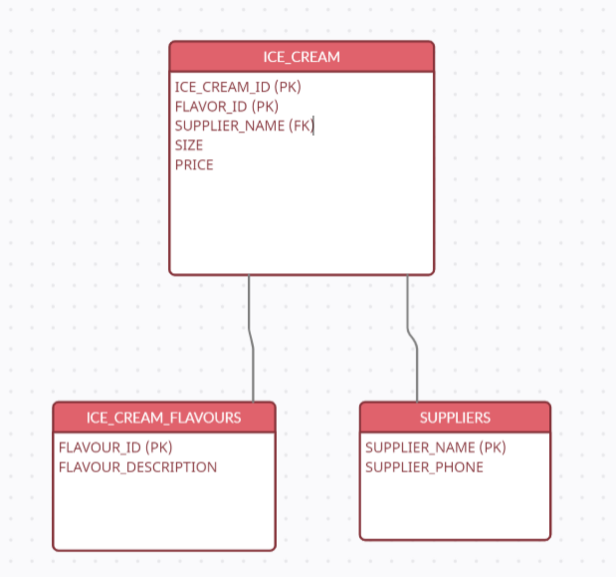
→



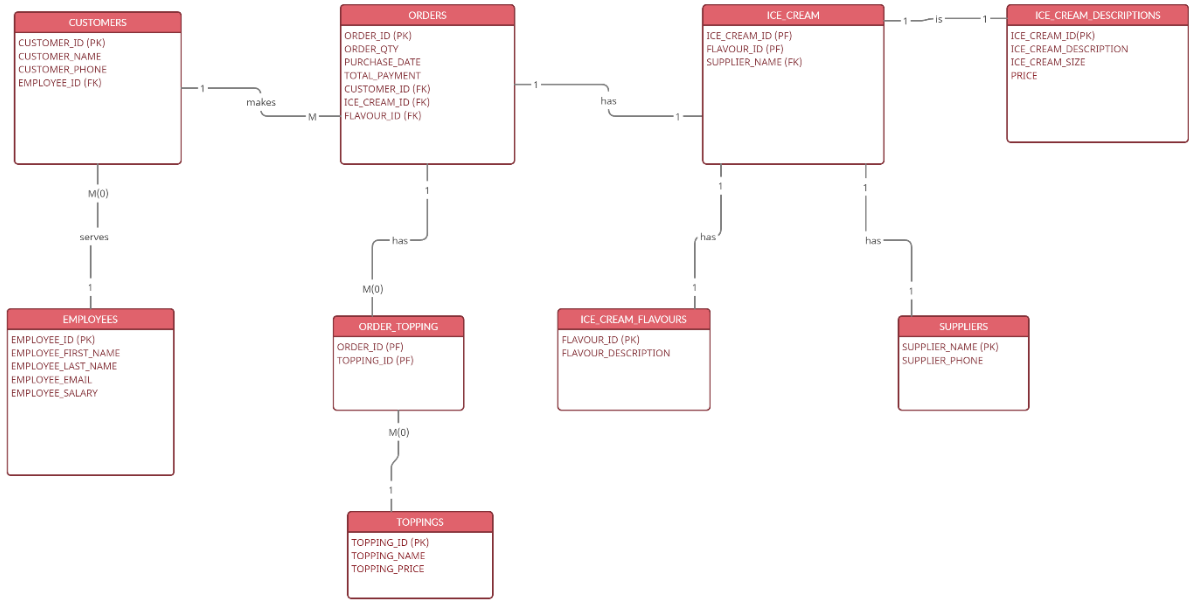
O alta posibila problema este faptul ca tabelul ICE_CREAM mentine si datele despre producatorul/distribuitorul inghetatei (SUPPLIER_NAME, SUPPLIER_PHONE). Desi cele doua coloane nu fac parte din cheia primara, ele mentin date inutile in tabelul principal (SUPPLIER_PHONE). Acest lucru poate fi evitat creand un tabel adiacent care sa mentina datele despre producator, stergand coloana SUPPLIER_PHONE si transformand coloana SUPPLIER_NAME in cheie externa. (forma FN2 -> FN3)



→



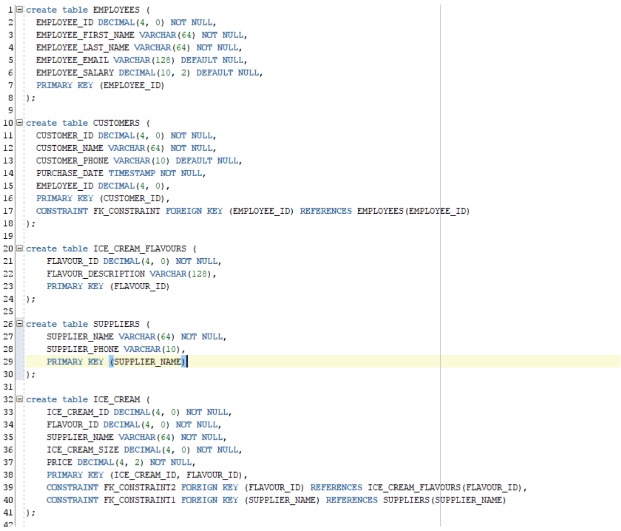
Dupa ce am terminat de rafinat modelul, am adaugat cheile primare, respectiv externe si am evidentiat relatiile dintre entitati, diagrama ERD devine:



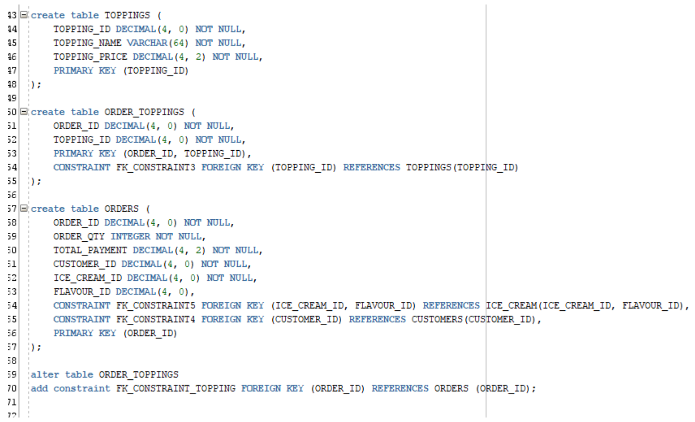
d) Implementare

Dupa ce am reusit sa intocmim o diagrama ERD cat mai corecta, vom incepe implementarea efectiva a acesteia.

Crearea tabelelor si adaugarea constrangerilor:

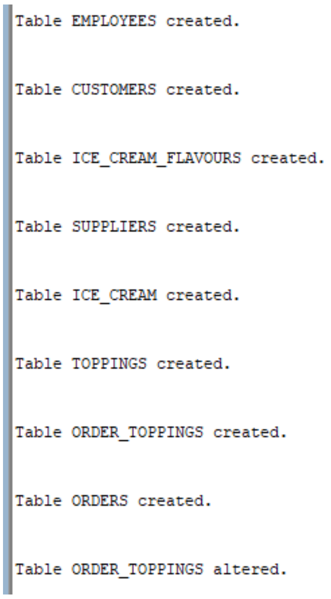


```sql
create table EMPLOYEES (
  EMPLOYEE_ID DECIMAL(4, 0) NOT NULL,
  EMPLOYEE_FIRST_NAME VARCHAR(64) NOT NULL,
  EMPLOYEE_LAST_NAME VARCHAR(64) NOT NULL,
  EMPLOYEE_EMAIL VARCHAR(128) DEFAULT NULL,
  EMPLOYEE_SALARY DECIMAL(10, 2) DEFAULT NULL,
  PRIMARY KEY (EMPLOYEE_ID)
);

create table CUSTOMERS (
  CUSTOMER_ID DECIMAL(4, 0) NOT NULL,
  CUSTOMER_NAME VARCHAR(64) NOT NULL,
  CUSTOMER_PHONE VARCHAR(10) DEFAULT NULL,
  PURCHASE_DATE TIMESTAMP NOT NULL,
  EMPLOYEE_ID DECIMAL(4, 0),
  PRIMARY KEY (CUSTOMER_ID),
  CONSTRAINT FK_CONSTRAINT FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEES(EMPLOYEE_ID)
);

create table ICE_CREAM_FLAVOURS (
    FLAVOUR_ID DECIMAL(4, 0) NOT NULL,
    FLAVOUR_DESCRIPTION VARCHAR(128),
    PRIMARY KEY (FLAVOUR_ID)
);

create table SUPPLIERS (
    SUPPLIER_NAME VARCHAR(64) NOT NULL,
    SUPPLIER_PHONE VARCHAR(10),
    PRIMARY KEY (SUPPLIER_NAME)
);

create table ICE_CREAM (
    ICE_CREAM_ID DECIMAL(4, 0) NOT NULL,
    FLAVOUR_ID DECIMAL(4, 0) NOT NULL,
    SUPPLIER_NAME VARCHAR(64) NOT NULL,
    ICE_CREAM_SIZE DECIMAL(4, 0) NOT NULL,
    PRICE DECIMAL(4, 2) NOT NULL,
    PRIMARY KEY (ICE_CREAM_ID, FLAVOUR_ID),
    CONSTRAINT FK_CONSTRAINT2 FOREIGN KEY (FLAVOUR_ID) REFERENCES ICE_CREAM_FLAVOURS(FLAVOUR_ID),
    CONSTRAINT FK_CONSTRAINT1 FOREIGN KEY (SUPPLIER_NAME) REFERENCES SUPPLIERS(SUPPLIER_NAME)
);
```



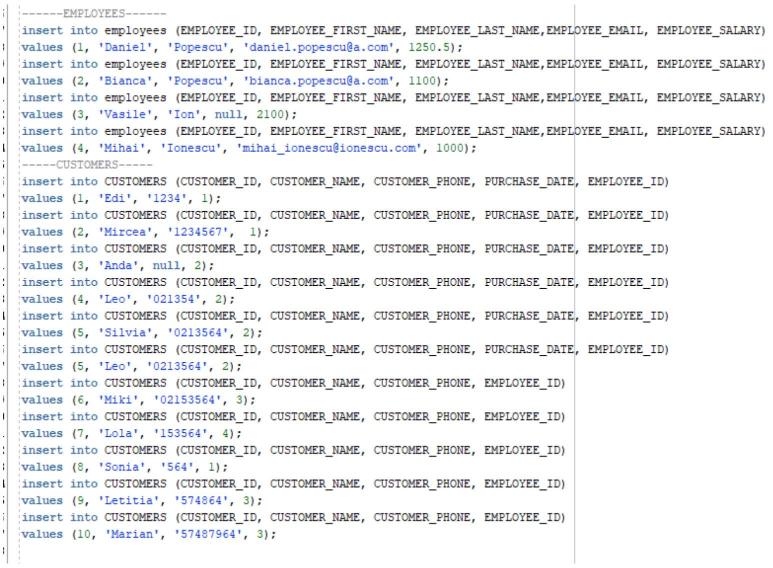
```sql
create table TOPPINGS (
    TOPPING_ID DECIMAL(4, 0) NOT NULL,
    TOPPING_NAME VARCHAR(64) NOT NULL,
    TOPPING_PRICE DECIMAL(4, 2) NOT NULL,
    PRIMARY KEY (TOPPING_ID)
);

create table ORDER_TOPPINGS (
    ORDER_ID DECIMAL(4, 0) NOT NULL,
    TOPPING_ID DECIMAL(4, 0) NOT NULL,
    PRIMARY KEY (ORDER_ID, TOPPING_ID),
    CONSTRAINT FK_CONSTRAINT3 FOREIGN KEY (TOPPING_ID) REFERENCES TOPPINGS(TOPPING_ID)
);

create table ORDERS (
    ORDER_ID DECIMAL(4, 0) NOT NULL,
    ORDER_QTY INTEGER NOT NULL,
    TOTAL_PAYMENT DECIMAL(4, 2) NOT NULL,
    CUSTOMER_ID DECIMAL(4, 0) NOT NULL,
    ICE_CREAM_ID DECIMAL(4, 0) NOT NULL,
    FLAVOUR_ID DECIMAL(4, 0),
    CONSTRAINT FK_CONSTRAINT5 FOREIGN KEY (ICE_CREAM_ID, FLAVOUR_ID) REFERENCES ICE_CREAM(ICE_CREAM_ID, FLAVOUR_ID),
    CONSTRAINT FK_CONSTRAINT4 FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS(CUSTOMER_ID),
    PRIMARY KEY (ORDER_ID)
);

alter table ORDER_TOPPINGS
add constraint FK_CONSTRAINT_TOPPING FOREIGN KEY (ORDER_ID) REFERENCES ORDERS (ORDER_ID);
```

Rezultat:

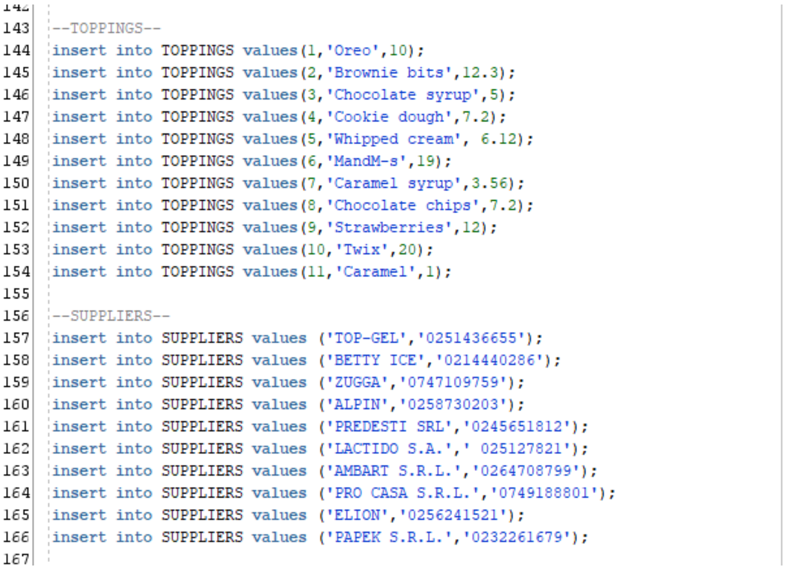


```
Table EMPLOYEES created.


Table CUSTOMERS created.


Table ICE_CREAM_FLAVOURS created.


Table SUPPLIERS created.


Table ICE_CREAM created.


Table TOPPINGS created.


Table ORDER_TOPPINGS created.


Table ORDERS created.


Table ORDER_TOPPINGS altered.
```

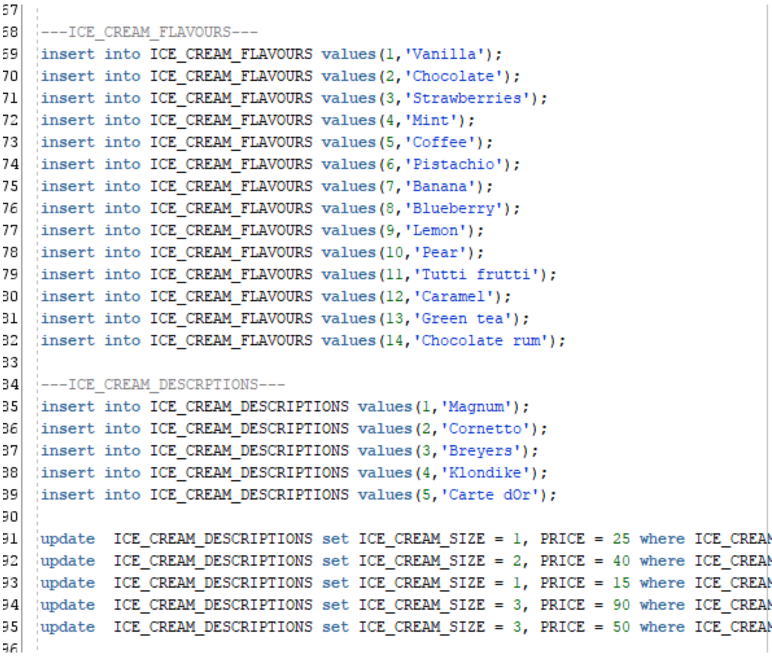
Inserarea datelor:



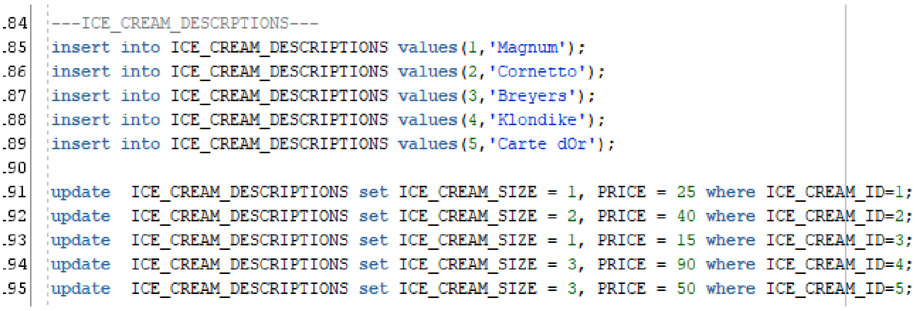
```
------EMPLOYEES------
insert into employees (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME,EMPLOYEE_EMAIL, EMPLOYEE_SALARY)
values (1, 'Daniel', 'Popescu', 'daniel.popescu@a.com', 1250.5);
insert into employees (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME,EMPLOYEE_EMAIL, EMPLOYEE_SALARY)
values (2, 'Bianca', 'Popescu', 'bianca.popescu@a.com', 1100);
insert into employees (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME,EMPLOYEE_EMAIL, EMPLOYEE_SALARY)
values (3, 'Vasile', 'Ion', null, 2100);
insert into employees (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME,EMPLOYEE_EMAIL, EMPLOYEE_SALARY)
values (4, 'Mihai', 'Ionescu', 'mihai_ionescu@ionescu.com', 1000);
-----CUSTOMERS-----
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, PURCHASE_DATE, EMPLOYEE_ID)
values (1, 'Edi', '1234', 1);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, PURCHASE_DATE, EMPLOYEE_ID)
values (2, 'Mircea', '1234567',  1);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, PURCHASE_DATE, EMPLOYEE_ID)
values (3, 'Anda', null, 2);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, PURCHASE_DATE, EMPLOYEE_ID)
values (4, 'Leo', '021354', 2);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, PURCHASE_DATE, EMPLOYEE_ID)
values (5, 'Silvia', '0213564', 2);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, PURCHASE_DATE, EMPLOYEE_ID)
values (5, 'Leo', '0213564', 2);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, EMPLOYEE_ID)
values (6, 'Miki', '02153564', 3);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, EMPLOYEE_ID)
values (7, 'Lola', '153564', 4);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, EMPLOYEE_ID)
values (8, 'Sonia', '564', 1);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, EMPLOYEE_ID)
values (9, 'Letitia', '574864', 3);
insert into CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, EMPLOYEE_ID)
values (10, 'Marian', '57487964', 3);
```



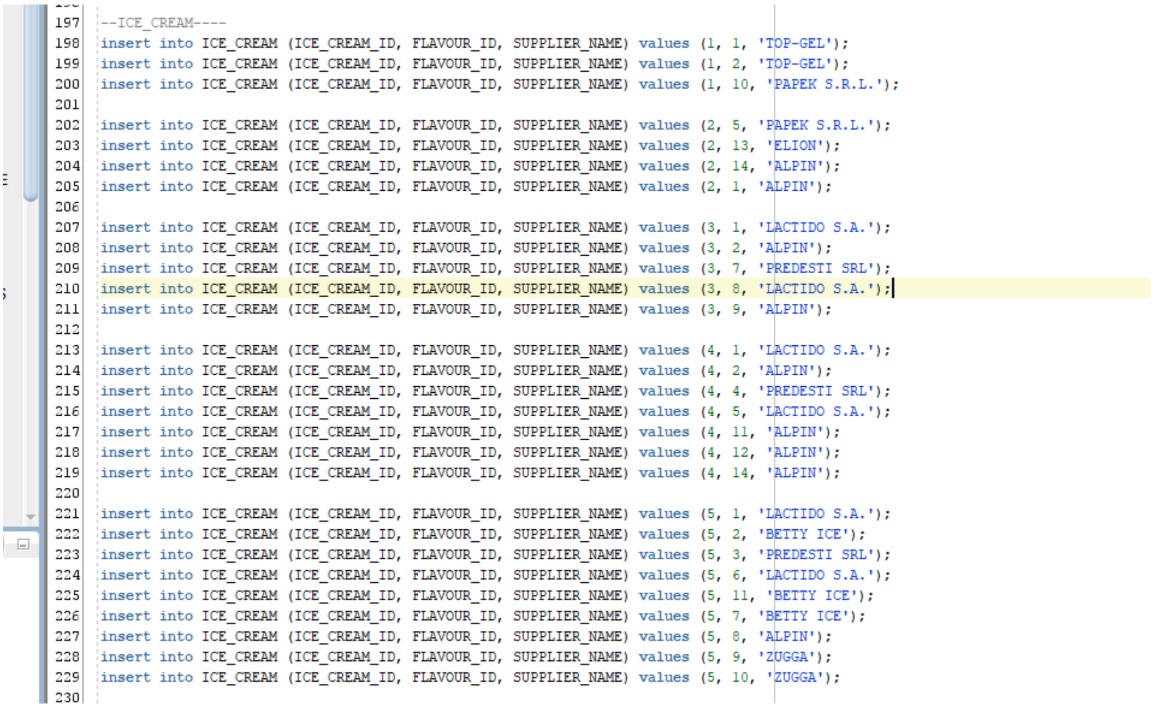
```

--TOPPINGS--
insert into TOPPINGS values(1,'Oreo',10);
insert into TOPPINGS values(2,'Brownie bits',12.3);
insert into TOPPINGS values(3,'Chocolate syrup',5);
insert into TOPPINGS values(4,'Cookie dough',7.2);
insert into TOPPINGS values(5,'Whipped cream', 6.12);
insert into TOPPINGS values(6,'MandM-s',19);
insert into TOPPINGS values(7,'Caramel syrup',3.56);
insert into TOPPINGS values(8,'Chocolate chips',7.2);
insert into TOPPINGS values(9,'Strawberries',12);
insert into TOPPINGS values(10,'Twix',20);
insert into TOPPINGS values(11,'Caramel',1);

--SUPPLIERS--
insert into SUPPLIERS values ('TOP-GEL','0251436655');
insert into SUPPLIERS values ('BETTY ICE','0214440286');
insert into SUPPLIERS values ('ZUGGA','0747109759');
insert into SUPPLIERS values ('ALPIN','0258730203');
insert into SUPPLIERS values ('PREDESTI SRL','0245651812');
insert into SUPPLIERS values ('LACTIDO S.A.',' 025127821');
insert into SUPPLIERS values ('AMBART S.R.L.','0264708799');
insert into SUPPLIERS values ('PRO CASA S.R.L.','0749188801');
insert into SUPPLIERS values ('ELION','0256241521');
insert into SUPPLIERS values ('PAPEK S.R.L.','0232261679');

```



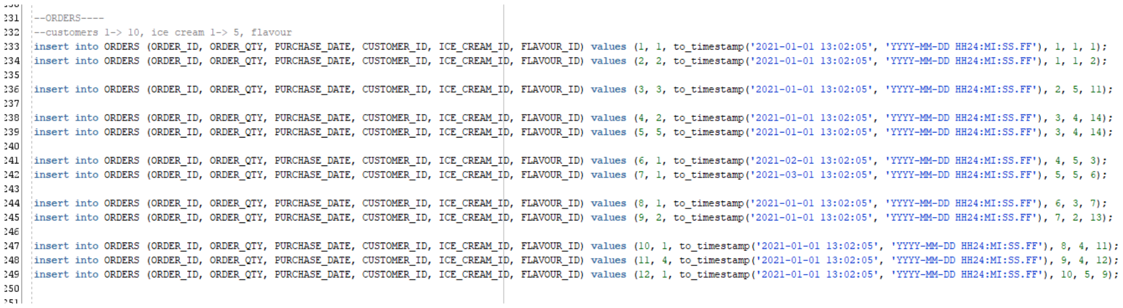
```
---ICE_CREAM_FLAVOURS---
insert into ICE_CREAM_FLAVOURS values(1,'Vanilla');
insert into ICE_CREAM_FLAVOURS values(2,'Chocolate');
insert into ICE_CREAM_FLAVOURS values(3,'Strawberries');
insert into ICE_CREAM_FLAVOURS values(4,'Mint');
insert into ICE_CREAM_FLAVOURS values(5,'Coffee');
insert into ICE_CREAM_FLAVOURS values(6,'Pistachio');
insert into ICE_CREAM_FLAVOURS values(7,'Banana');
insert into ICE_CREAM_FLAVOURS values(8,'Blueberry');
insert into ICE_CREAM_FLAVOURS values(9,'Lemon');
insert into ICE_CREAM_FLAVOURS values(10,'Pear');
insert into ICE_CREAM_FLAVOURS values(11,'Tutti frutti');
insert into ICE_CREAM_FLAVOURS values(12,'Caramel');
insert into ICE_CREAM_FLAVOURS values(13,'Green tea');
insert into ICE_CREAM_FLAVOURS values(14,'Chocolate rum');

---ICE_CREAM_DESCRPTIONS---
insert into ICE_CREAM_DESCRIPTIONS values(1,'Magnum');
insert into ICE_CREAM_DESCRIPTIONS values(2,'Cornetto');
insert into ICE_CREAM_DESCRIPTIONS values(3,'Breyers');
insert into ICE_CREAM_DESCRIPTIONS values(4,'Klondike');
insert into ICE_CREAM_DESCRIPTIONS values(5,'Carte dOr');

update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 1, PRICE = 25 where ICE_CREAM_ID=1;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 2, PRICE = 40 where ICE_CREAM_ID=2;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 1, PRICE = 15 where ICE_CREAM_ID=3;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 3, PRICE = 90 where ICE_CREAM_ID=4;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 3, PRICE = 50 where ICE_CREAM_ID=5;
```



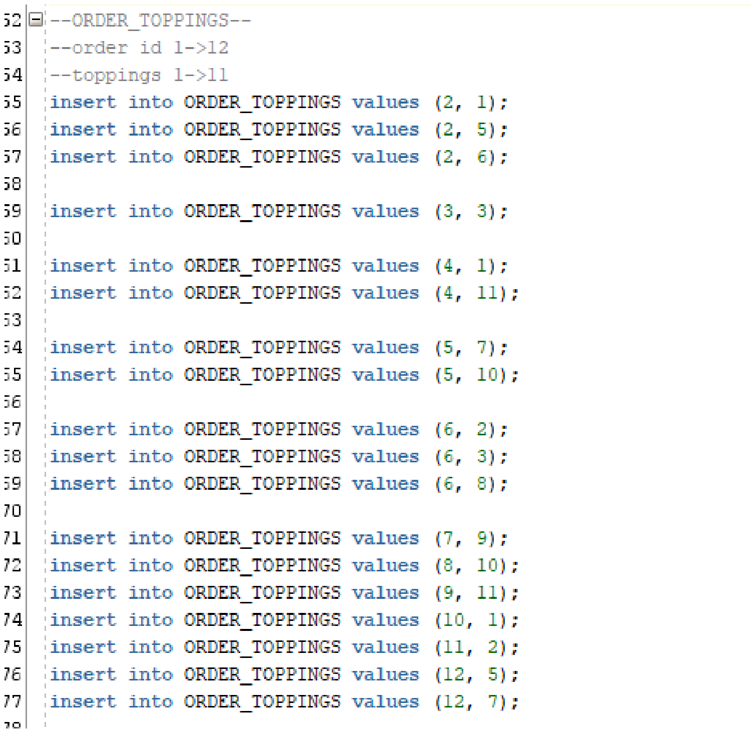
```
---ICE_CREAM_DESCRPTIONS---
insert into ICE_CREAM_DESCRIPTIONS values(1,'Magnum');
insert into ICE_CREAM_DESCRIPTIONS values(2,'Cornetto');
insert into ICE_CREAM_DESCRIPTIONS values(3,'Breyers');
insert into ICE_CREAM_DESCRIPTIONS values(4,'Klondike');
insert into ICE_CREAM_DESCRIPTIONS values(5,'Carte dOr');

update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 1, PRICE = 25 where ICE_CREAM_ID=1;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 2, PRICE = 40 where ICE_CREAM_ID=2;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 1, PRICE = 15 where ICE_CREAM_ID=3;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 3, PRICE = 90 where ICE_CREAM_ID=4;
update  ICE_CREAM_DESCRIPTIONS set ICE_CREAM_SIZE = 3, PRICE = 50 where ICE_CREAM_ID=5;
```



```
--ICE_CREAM----
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (1, 1, 'TOP-GEL');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (1, 2, 'TOP-GEL');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (1, 10, 'PAPEK S.R.L.');

insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (2, 5, 'PAPEK S.R.L.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (2, 13, 'ELION');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (2, 14, 'ALPIN');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (2, 1, 'ALPIN');

insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (3, 1, 'LACTIDO S.A.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (3, 2, 'ALPIN');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (3, 7, 'PREDESTI SRL');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (3, 8, 'LACTIDO S.A.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (3, 9, 'ALPIN');

insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 1, 'LACTIDO S.A.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 2, 'ALPIN');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 4, 'PREDESTI SRL');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 5, 'LACTIDO S.A.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 11, 'ALPIN');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 12, 'ALPIN');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (4, 14, 'ALPIN');

insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 1, 'LACTIDO S.A.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 2, 'BETTY ICE');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 3, 'PREDESTI SRL');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 6, 'LACTIDO S.A.');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 11, 'BETTY ICE');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 7, 'BETTY ICE');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 8, 'ALPIN');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 9, 'ZUGGA');
insert into ICE_CREAM (ICE_CREAM_ID, FLAVOUR_ID, SUPPLIER_NAME) values (5, 10, 'ZUGGA');
```



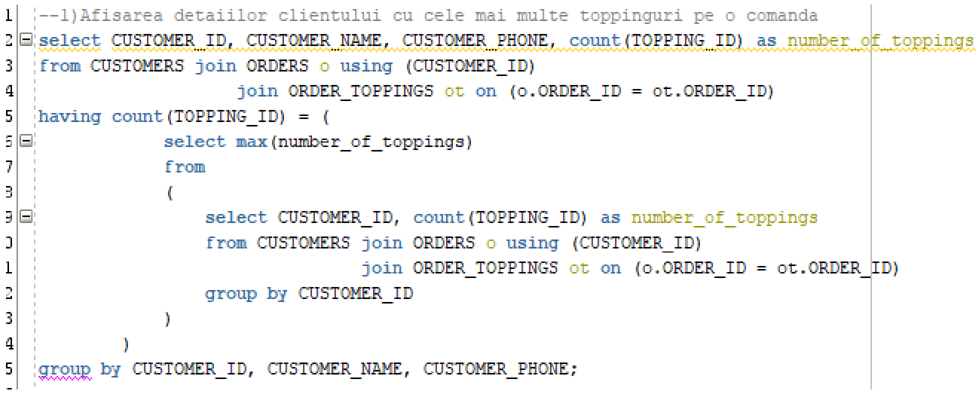
```sql
--ORDERS----
--customers 1-> 10, ice cream 1-> 5, flavour
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (1, 1, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 1, 1, 1);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (2, 2, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 1, 1, 2);

insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (3, 3, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 2, 5, 11);

insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (4, 2, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 3, 4, 14);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (5, 5, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 3, 4, 14);

insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (6, 1, to_timestamp('2021-02-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 4, 5, 3);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (7, 1, to_timestamp('2021-03-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 5, 5, 6);

insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (8, 1, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 6, 3, 7);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (9, 2, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 7, 2, 13);

insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (10, 1, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 8, 4, 11);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (11, 4, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 9, 4, 12);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (12, 1, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 10, 5, 9);
```



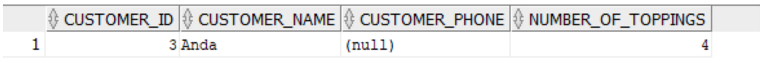
```sql
--ORDER_TOPPINGS--
--order id 1->12
--toppings 1->11
insert into ORDER_TOPPINGS values (2, 1);
insert into ORDER_TOPPINGS values (2, 5);
insert into ORDER_TOPPINGS values (2, 6);

insert into ORDER_TOPPINGS values (3, 3);

insert into ORDER_TOPPINGS values (4, 1);
insert into ORDER_TOPPINGS values (4, 11);

insert into ORDER_TOPPINGS values (5, 7);
insert into ORDER_TOPPINGS values (5, 10);

insert into ORDER_TOPPINGS values (6, 2);
insert into ORDER_TOPPINGS values (6, 3);
insert into ORDER_TOPPINGS values (6, 8);

insert into ORDER_TOPPINGS values (7, 9);
insert into ORDER_TOPPINGS values (8, 10);
insert into ORDER_TOPPINGS values (9, 11);
insert into ORDER_TOPPINGS values (10, 1);
insert into ORDER_TOPPINGS values (11, 2);
insert into ORDER_TOPPINGS values (12, 5);
insert into ORDER_TOPPINGS values (12, 7);
```

e) Interogari

1.

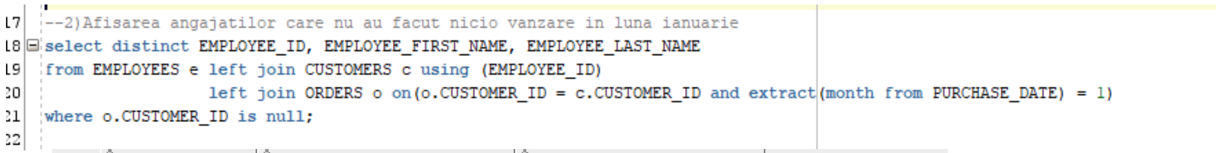


```sql
--1)Afisarea detaiilor clientului cu cele mai multe toppinguri pe o comanda
select CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE, count(TOPPING_ID) as number_of_toppings
from CUSTOMERS join ORDERS o using (CUSTOMER_ID)
                    join ORDER_TOPPINGS ot on (o.ORDER_ID = ot.ORDER_ID)
having count(TOPPING_ID) = (
            select max(number_of_toppings)
            from
            (
                select CUSTOMER_ID, count(TOPPING_ID) as number_of_toppings
                from CUSTOMERS join ORDERS o using (CUSTOMER_ID)
                               join ORDER_TOPPINGS ot on (o.ORDER_ID = ot.ORDER_ID)
                group by CUSTOMER_ID
            )
        )
group by CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_PHONE;
```

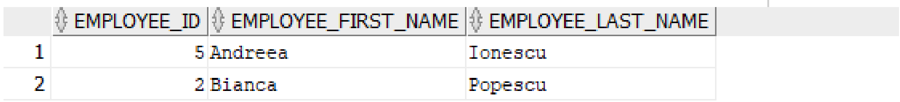


	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_PHONE	NUMBER_OF_TOPPINGS
1	3	Anda	(null)	4

2.

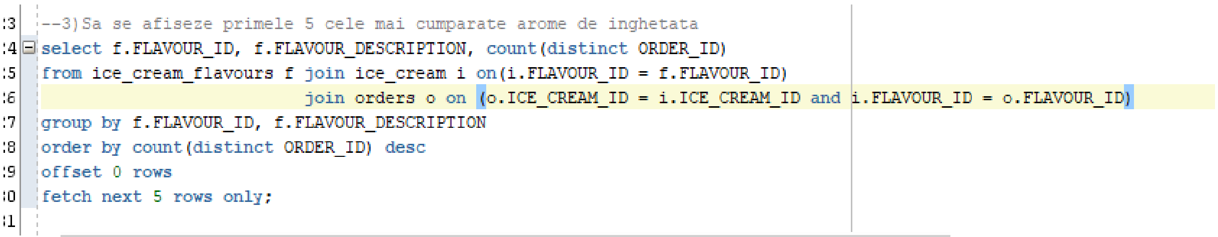


```
--2)Afisarea angajatilor care nu au facut nicio vanzare in luna ianuarie
select distinct EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME
from EMPLOYEES e left join CUSTOMERS c using (EMPLOYEE_ID)
                 left join ORDERS o on(o.CUSTOMER_ID = c.CUSTOMER_ID and extract(month from PURCHASE_DATE) = 1)
where o.CUSTOMER_ID is null;
```

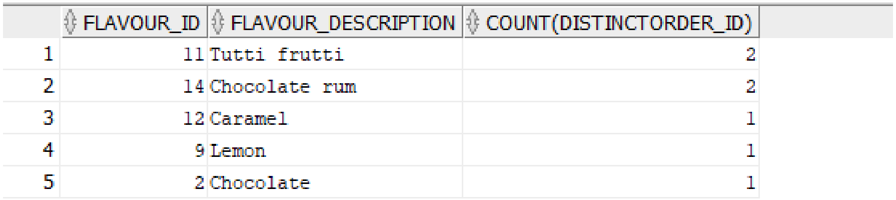


	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME
1	5	Andreea	Ionescu
2	2	Bianca	Popescu

3.

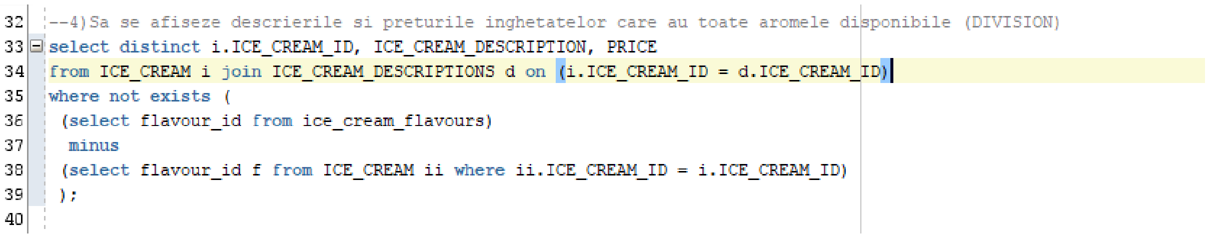


```
--3)Sa se afiseze primele 5 cele mai cumparate arome de inghetata
select f.FLAVOUR_ID, f.FLAVOUR_DESCRIPTION, count(distinct ORDER_ID)
from ice_cream_flavours f join ice_cream i on(i.FLAVOUR_ID = f.FLAVOUR_ID)
                          join orders o on (o.ICE_CREAM_ID = i.ICE_CREAM_ID and i.FLAVOUR_ID = o.FLAVOUR_ID)
group by f.FLAVOUR_ID, f.FLAVOUR_DESCRIPTION
order by count(distinct ORDER_ID) desc
offset 0 rows
fetch next 5 rows only;
```

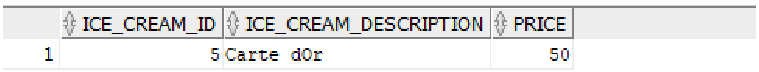


	FLAVOUR_ID	FLAVOUR_DESCRIPTION	COUNT(DISTINCTORDER_ID)
1	11	Tutti frutti	2
2	14	Chocolate rum	2
3	12	Caramel	1
4	9	Lemon	1
5	2	Chocolate	1

4.

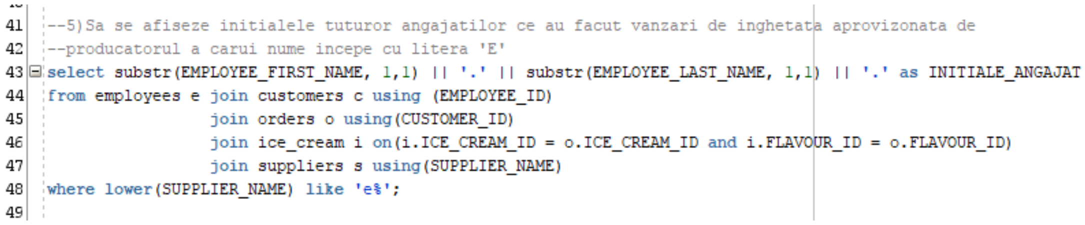


```
--4)Sa se afiseze descrierile si preturile inghetatelor care au toate aromele disponibile (DIVISION)
select distinct i.ICE_CREAM_ID, ICE_CREAM_DESCRIPTION, PRICE
from ICE_CREAM i join ICE_CREAM_DESCRIPTIONS d on (i.ICE_CREAM_ID = d.ICE_CREAM_ID)
where not exists (
 (select flavour_id from ice_cream_flavours)
  minus
 (select flavour_id f from ICE_CREAM ii where ii.ICE_CREAM_ID = i.ICE_CREAM_ID)
 );
```

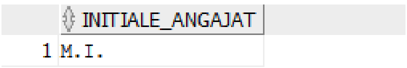


	ICE_CREAM_ID	ICE_CREAM_DESCRIPTION	PRICE
1	5	Carte dOr	50

5.

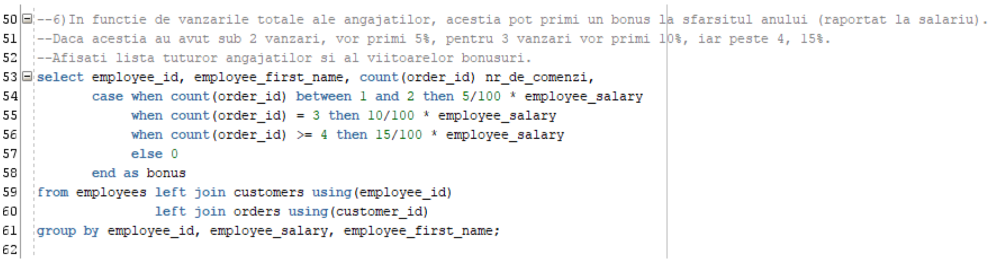


```
--5)Sa se afiseze initialele tuturor angajatilor ce au facut vanzari de inghetata aprovizonata de
--producatorul a carui nume incepe cu litera 'E'
select substr(EMPLOYEE_FIRST_NAME, 1,1) || '.' || substr(EMPLOYEE_LAST_NAME, 1,1) || '.' as INITIALE_ANGAJAT
from employees e join customers c using (EMPLOYEE_ID)
                 join orders o using(CUSTOMER_ID)
                 join ice_cream i on(i.ICE_CREAM_ID = o.ICE_CREAM_ID and i.FLAVOUR_ID = o.FLAVOUR_ID)
                 join suppliers s using(SUPPLIER_NAME)
where lower(SUPPLIER_NAME) like 'e%';
```

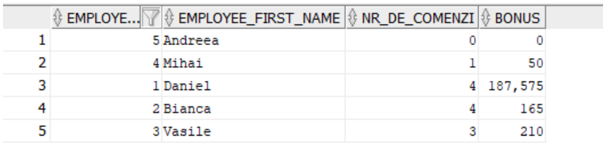


	INITIALE_ANGAJAT
1	M.I.

6.

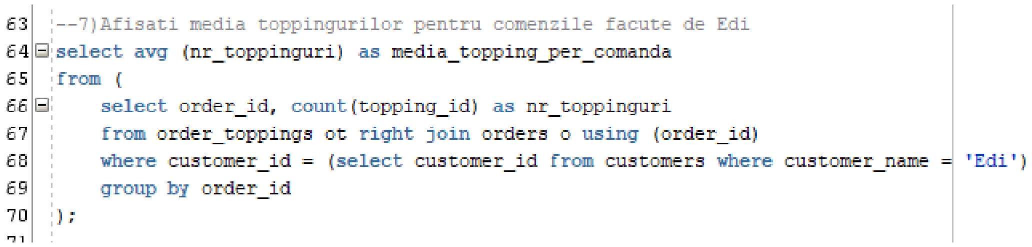


```
--6)In functie de vanzarile totale ale angajatilor, acestia pot primi un bonus la sfarsitul anului (raportat la salariu).
--Daca acestia au avut sub 2 vanzari, vor primi 5%, pentru 3 vanzari vor primi 10%, iar peste 4, 15%.
--Afisati lista tuturor angajatilor si al viitoarelor bonusuri.
select employee_id, employee_first_name, count(order_id) nr_de_comenzi,
       case when count(order_id) between 1 and 2 then 5/100 * employee_salary
            when count(order_id) = 3 then 10/100 * employee_salary
            when count(order_id) >= 4 then 15/100 * employee_salary
            else 0
       end as bonus
from employees left join customers using(employee_id)
               left join orders using(customer_id)
group by employee_id, employee_salary, employee_first_name;
```

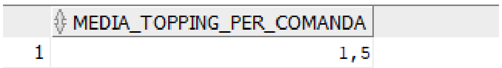


	EMPLOYE...	EMPLOYEE_FIRST_NAME	NR_DE_COMENZI	BONUS
1	5	Andreea	0	0
2	4	Mihai	1	50
3	1	Daniel	4	187,575
4	2	Bianca	4	165
5	3	Vasile	3	210

7.

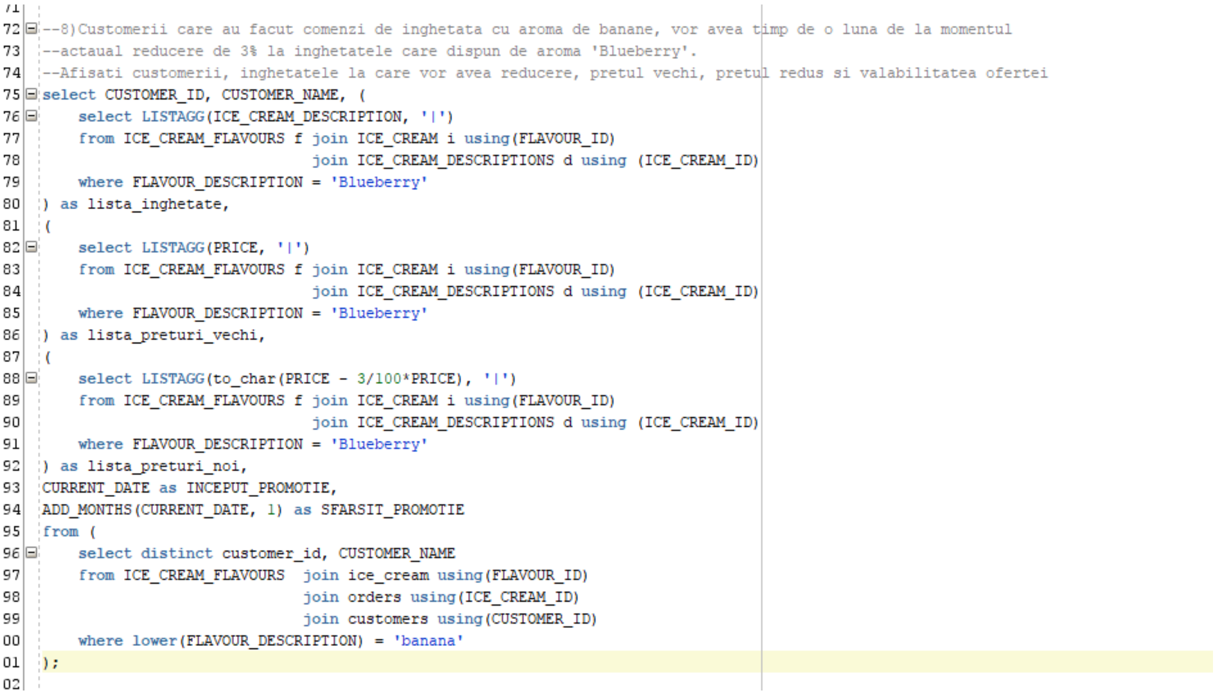


```
--7)Afisati media toppingurilor pentru comenzile facute de Edi
select avg (nr_toppinguri) as media_topping_per_comanda
from (
    select order_id, count(topping_id) as nr_toppinguri
    from order_toppings ot right join orders o using (order_id)
    where customer_id = (select customer_id from customers where customer_name = 'Edi')
    group by order_id
);
```

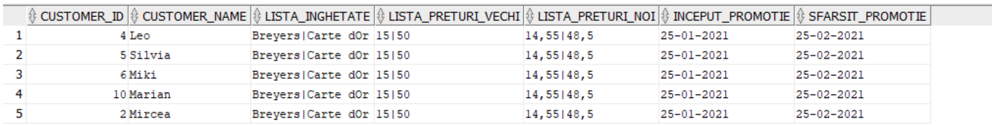


	MEDIA_TOPPING_PER_COMANDA
1	1,5

8.

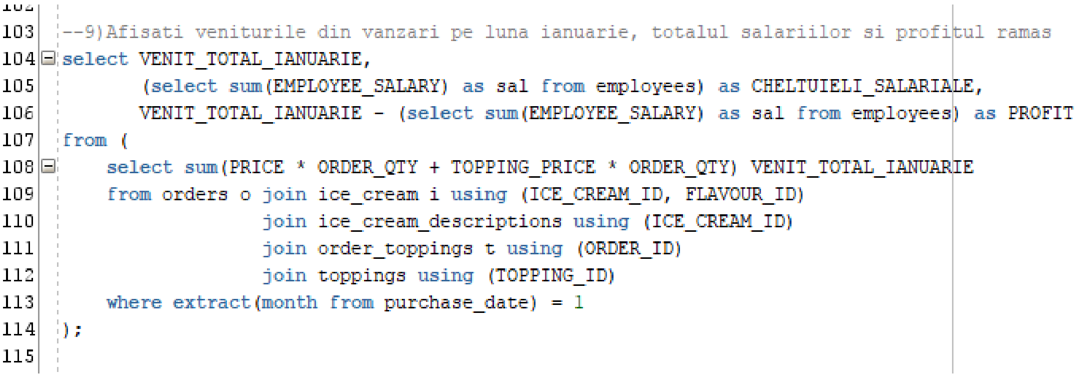


```
--8)Customerii care au facut comenzi de inghetata cu aroma de banane, vor avea timp de o luna de la momentul
--actaual reducere de 3% la inghetatele care dispun de aroma 'Blueberry'.
--Afisati customerii, inghetatele la care vor avea reducere, pretul vechi, pretul redus si valabilitatea ofertei
select CUSTOMER_ID, CUSTOMER_NAME, (
    select LISTAGG(ICE_CREAM_DESCRIPTION, '|')
    from ICE_CREAM_FLAVOURS f join ICE_CREAM i using(FLAVOUR_ID)
                             join ICE_CREAM_DESCRIPTIONS d using (ICE_CREAM_ID)
    where FLAVOUR_DESCRIPTION = 'Blueberry'
) as lista_inghetate,
(
    select LISTAGG(PRICE, '|')
    from ICE_CREAM_FLAVOURS f join ICE_CREAM i using(FLAVOUR_ID)
                             join ICE_CREAM_DESCRIPTIONS d using (ICE_CREAM_ID)
    where FLAVOUR_DESCRIPTION = 'Blueberry'
) as lista_preturi_vechi,
(
    select LISTAGG(to_char(PRICE - 3/100*PRICE), '|')
    from ICE_CREAM_FLAVOURS f join ICE_CREAM i using(FLAVOUR_ID)
                             join ICE_CREAM_DESCRIPTIONS d using (ICE_CREAM_ID)
    where FLAVOUR_DESCRIPTION = 'Blueberry'
) as lista_preturi_noi,
CURRENT_DATE as INCEPUT_PROMOTIE,
ADD_MONTHS(CURRENT_DATE, 1) as SFARSIT_PROMOTIE
from (
    select distinct customer_id, CUSTOMER_NAME
    from ICE_CREAM_FLAVOURS  join ice_cream using(FLAVOUR_ID)
                            join orders using(ICE_CREAM_ID)
                            join customers using(CUSTOMER_ID)
    where lower(FLAVOUR_DESCRIPTION) = 'banana'
);
```

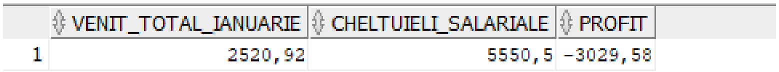


	CUSTOMER_ID	CUSTOMER_NAME	LISTA_INGHETATE	LISTA_PRETURI_VECHI	LISTA_PRETURI_NOI	INCEPUT_PROMOTIE	SFARSIT_PROMOTIE
1	4	Leo	Breyers|Carte dOr	15|50	14,55|48,5	25-01-2021	25-02-2021
2	5	Silvia	Breyers|Carte dOr	15|50	14,55|48,5	25-01-2021	25-02-2021
3	6	Miki	Breyers|Carte dOr	15|50	14,55|48,5	25-01-2021	25-02-2021
4	10	Marian	Breyers|Carte dOr	15|50	14,55|48,5	25-01-2021	25-02-2021
5	2	Mircea	Breyers|Carte dOr	15|50	14,55|48,5	25-01-2021	25-02-2021

9.

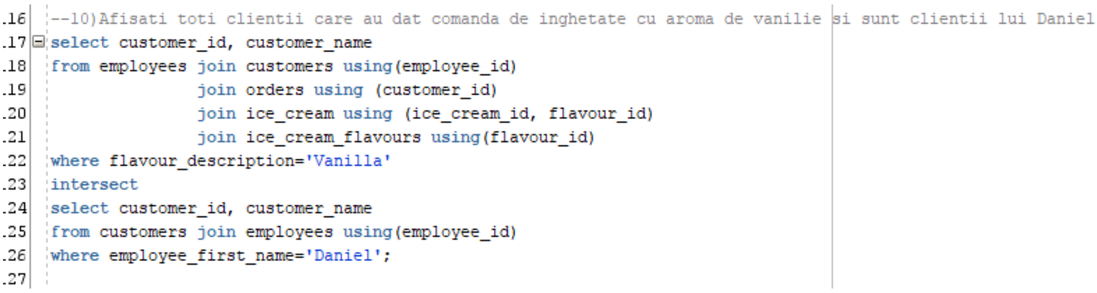


```
--9)Afisati veniturile din vanzari pe luna ianuarie, totalul salariilor si profitul ramas
select VENIT_TOTAL_IANUARIE,
       (select sum(EMPLOYEE_SALARY) as sal from employees) as CHELTUIELI_SALARIALE,
       VENIT_TOTAL_IANUARIE - (select sum(EMPLOYEE_SALARY) as sal from employees) as PROFIT
from (
    select sum(PRICE * ORDER_QTY + TOPPING_PRICE * ORDER_QTY) VENIT_TOTAL_IANUARIE
    from orders o join ice_cream i using (ICE_CREAM_ID, FLAVOUR_ID)
                  join ice_cream_descriptions using (ICE_CREAM_ID)
                  join order_toppings t using (ORDER_ID)
                  join toppings using (TOPPING_ID)
    where extract(month from purchase_date) = 1
);
```

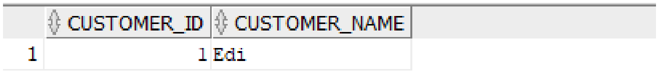


	VENIT_TOTAL_IANUARIE	CHELTUIELI_SALARIALE	PROFIT
1	2520,92	5550,5	-3029,58

10.

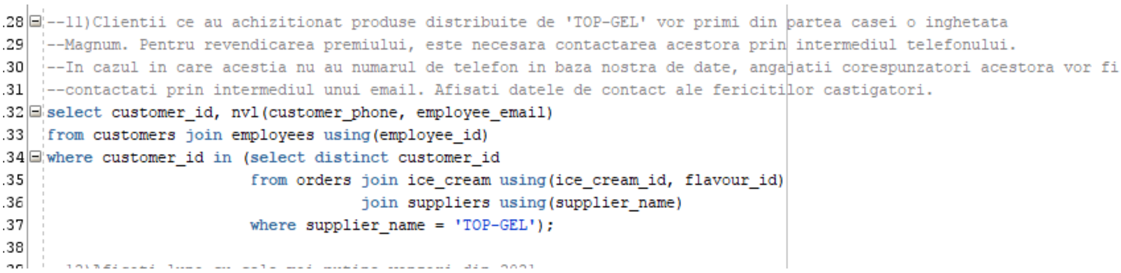


```
--10)Afisati toti clientii care au dat comanda de inghetate cu aroma de vanilie si sunt clientii lui Daniel
select customer_id, customer_name
from employees join customers using(employee_id)
               join orders using (customer_id)
               join ice_cream using (ice_cream_id, flavour_id)
               join ice_cream_flavours using(flavour_id)
where flavour_description='Vanilla'
intersect
select customer_id, customer_name
from customers join employees using(employee_id)
where employee_first_name='Daniel';
```

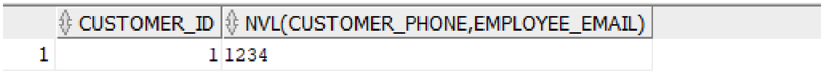


	CUSTOMER_ID	CUSTOMER_NAME
1	1	Edi

11.

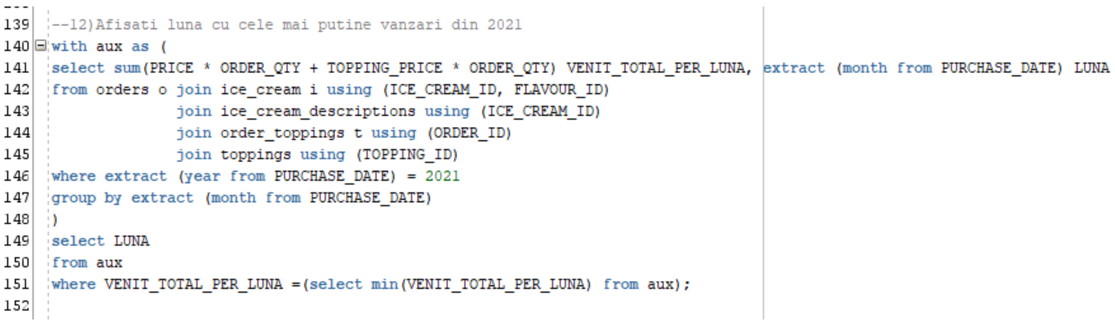


```
--11)Clientii ce au achizitionat produse distribuite de 'TOP-GEL' vor primi din partea casei o inghetata
--Magnum. Pentru revendicarea premiului, este necesara contactarea acestora prin intermediul telefonului.
--In cazul in care acestia nu au numarul de telefon in baza nostra de date, angajatii corespunzatori acestora vor fi
--contactati prin intermediul unui email. Afisati datele de contact ale fericitilor castigatori.
select customer_id, nvl(customer_phone, employee_email)
from customers join employees using(employee_id)
where customer_id in (select distinct customer_id
                      from orders join ice_cream using(ice_cream_id, flavour_id)
                                  join suppliers using(supplier_name)
                      where supplier_name = 'TOP-GEL');
```

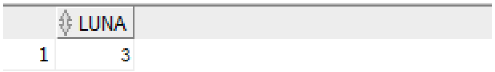


	CUSTOMER_ID	NVL(CUSTOMER_PHONE,EMPLOYEE_EMAIL)
1	1	1234

12.

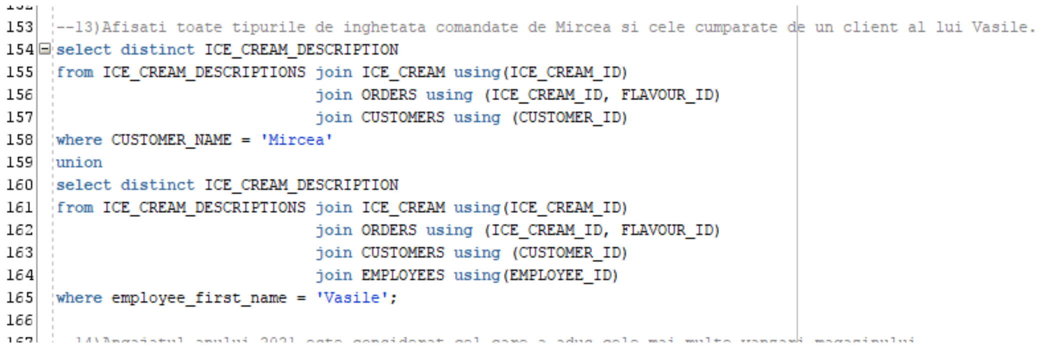


```sql
--12)Afisati luna cu cele mai putine vanzari din 2021
with aux as (
select sum(PRICE * ORDER_QTY + TOPPING_PRICE * ORDER_QTY) VENIT_TOTAL_PER_LUNA, extract (month from PURCHASE_DATE) LUNA
from orders o join ice_cream i using (ICE_CREAM_ID, FLAVOUR_ID)
                join ice_cream_descriptions using (ICE_CREAM_ID)
                join order_toppings t using (ORDER_ID)
                join toppings using (TOPPING_ID)
where extract (year from PURCHASE_DATE) = 2021
group by extract (month from PURCHASE_DATE)
)
select LUNA
from aux
where VENIT_TOTAL_PER_LUNA =(select min(VENIT_TOTAL_PER_LUNA) from aux);
```

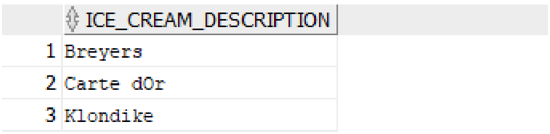


	LUNA
1	3

13.

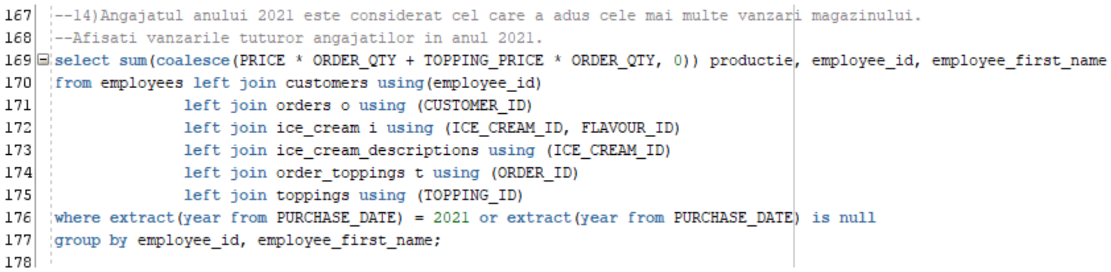


```sql
--13)Afisati toate tipurile de inghetata comandate de Mircea si cele cumparate de un client al lui Vasile.
select distinct ICE_CREAM_DESCRIPTION
from ICE_CREAM_DESCRIPTIONS join ICE_CREAM using(ICE_CREAM_ID)
                            join ORDERS using (ICE_CREAM_ID, FLAVOUR_ID)
                            join CUSTOMERS using (CUSTOMER_ID)
where CUSTOMER_NAME = 'Mircea'
union
select distinct ICE_CREAM_DESCRIPTION
from ICE_CREAM_DESCRIPTIONS join ICE_CREAM using(ICE_CREAM_ID)
                            join ORDERS using (ICE_CREAM_ID, FLAVOUR_ID)
                            join CUSTOMERS using (CUSTOMER_ID)
                            join EMPLOYEES using(EMPLOYEE_ID)
where employee_first_name = 'Vasile';
```

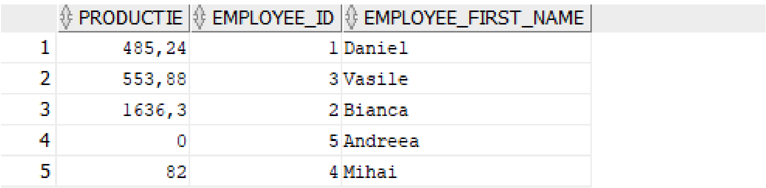


	ICE_CREAM_DESCRIPTION
1	Breyers
2	Carte dOr
3	Klondike

14.

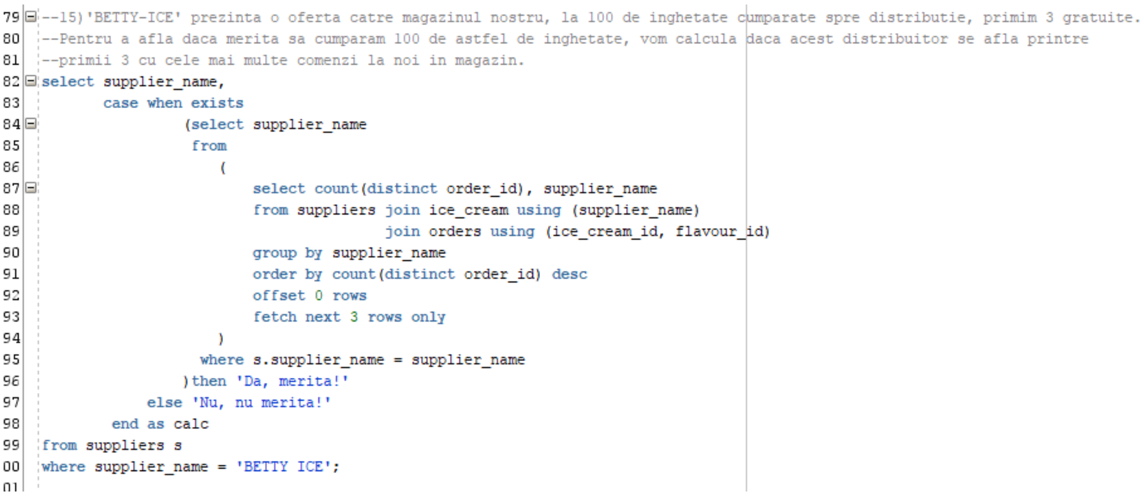


```sql
--14)Angajatul anului 2021 este considerat cel care a adus cele mai multe vanzari magazinului.
--Afisati vanzarile tuturor angajatilor in anul 2021.
select sum(coalesce(PRICE * ORDER_QTY + TOPPING_PRICE * ORDER_QTY, 0)) productie, employee_id, employee_first_name
from employees left join customers using(employee_id)
                left join orders o using (CUSTOMER_ID)
                left join ice_cream i using (ICE_CREAM_ID, FLAVOUR_ID)
                left join ice_cream_descriptions using (ICE_CREAM_ID)
                left join order_toppings t using (ORDER_ID)
                left join toppings using (TOPPING_ID)
where extract(year from PURCHASE_DATE) = 2021 or extract(year from PURCHASE_DATE) is null
group by employee_id, employee_first_name;
```

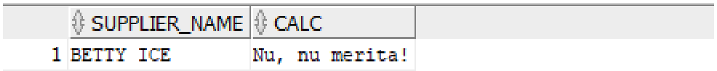


	PRODUCTIE	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME
1	485,24	1	Daniel
2	553,88	3	Vasile
3	1636,3	2	Bianca
4	0	5	Andreea
5	82	4	Mihai

15.

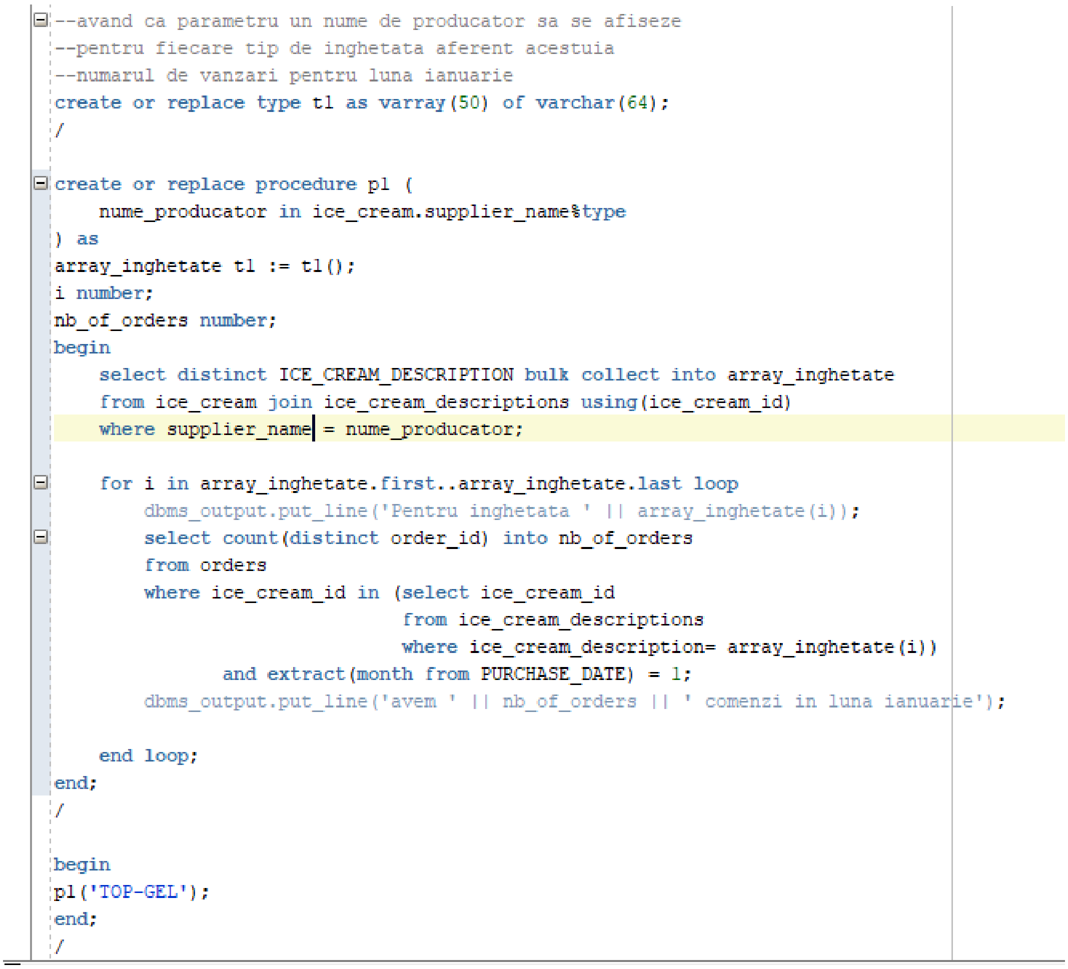


```
--15)'BETTY-ICE' prezinta o oferta catre magazinul nostru, la 100 de inghetate cumparate spre distributie, primim 3 gratuite.
--Pentru a afla daca merita sa cumparam 100 de astfel de inghetate, vom calcula daca acest distribuitor se afla printre
--primii 3 cu cele mai multe comenzi la noi in magazin.
select supplier_name,
       case when exists
                (select supplier_name
                 from
                    (
                        select count(distinct order_id), supplier_name
                        from suppliers join ice_cream using (supplier_name)
                                       join orders using (ice_cream_id, flavour_id)
                        group by supplier_name
                        order by count(distinct order_id) desc
                        offset 0 rows
                        fetch next 3 rows only
                    )
                  where s.supplier_name = supplier_name
                )then 'Da, merita!'
            else 'Nu, nu merita!'
        end as calc
from suppliers s
where supplier_name = 'BETTY ICE';
```

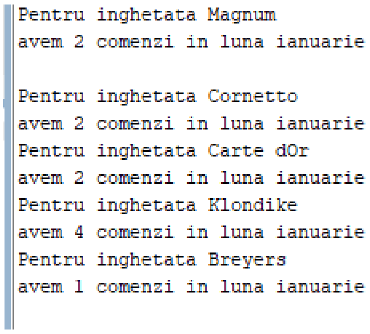


	SUPPLIER_NAME	CALC
1	BETTY ICE	Nu, nu merita!

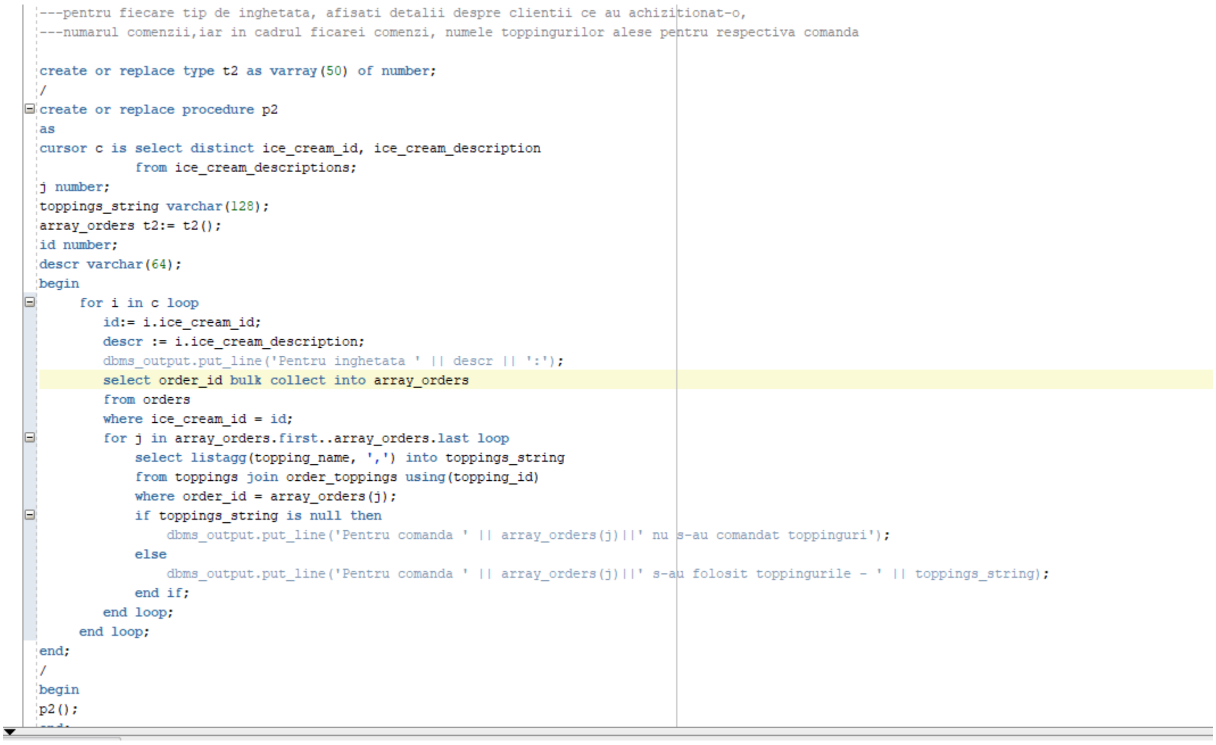
g+h) Proceduri, functii si triggeri



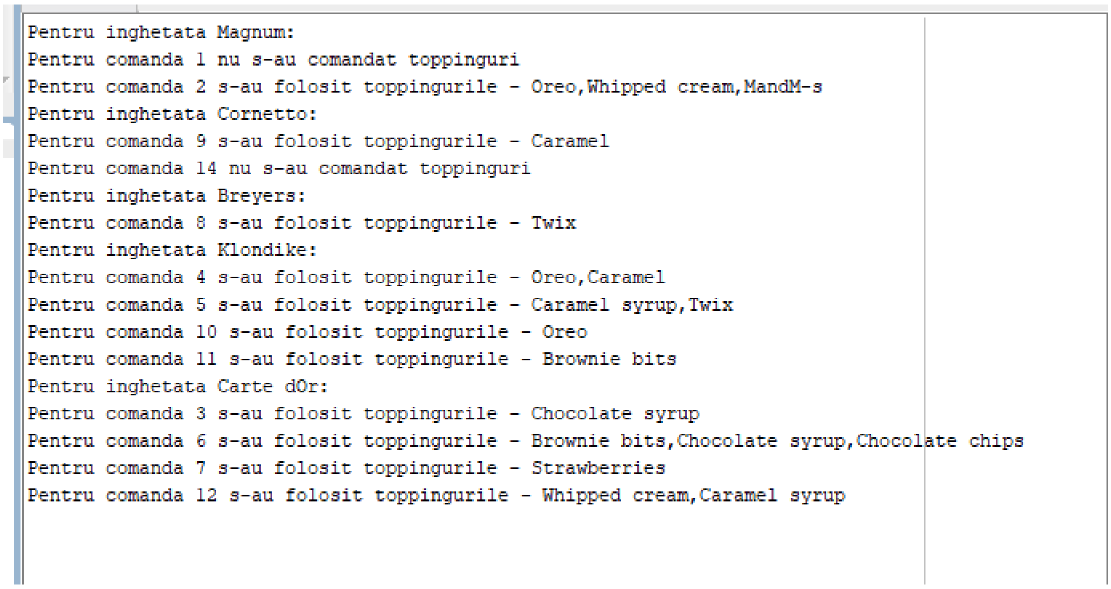
```
--avand ca parametru un nume de producator sa se afiseze
--pentru fiecare tip de inghetata aferent acestuia
--numarul de vanzari pentru luna ianuarie
create or replace type t1 as varray(50) of varchar(64);
/

create or replace procedure p1 (
    nume_producator in ice_cream.supplier_name%type
) as
array_inghetate t1 := t1();
i number;
nb_of_orders number;
begin
    select distinct ICE_CREAM_DESCRIPTION bulk collect into array_inghetate
    from ice_cream join ice_cream_descriptions using(ice_cream_id)
    where supplier_name = nume_producator;

    for i in array_inghetate.first..array_inghetate.last loop
        dbms_output.put_line('Pentru inghetata ' || array_inghetate(i));
        select count(distinct order_id) into nb_of_orders
        from orders
        where ice_cream_id in (select ice_cream_id
                               from ice_cream_descriptions
                               where ice_cream_description= array_inghetate(i))
               and extract(month from PURCHASE_DATE) = 1;
        dbms_output.put_line('avem ' || nb_of_orders || ' comenzi in luna ianuarie');

    end loop;
end;
/

begin
p1('TOP-GEL');
end;
/
```



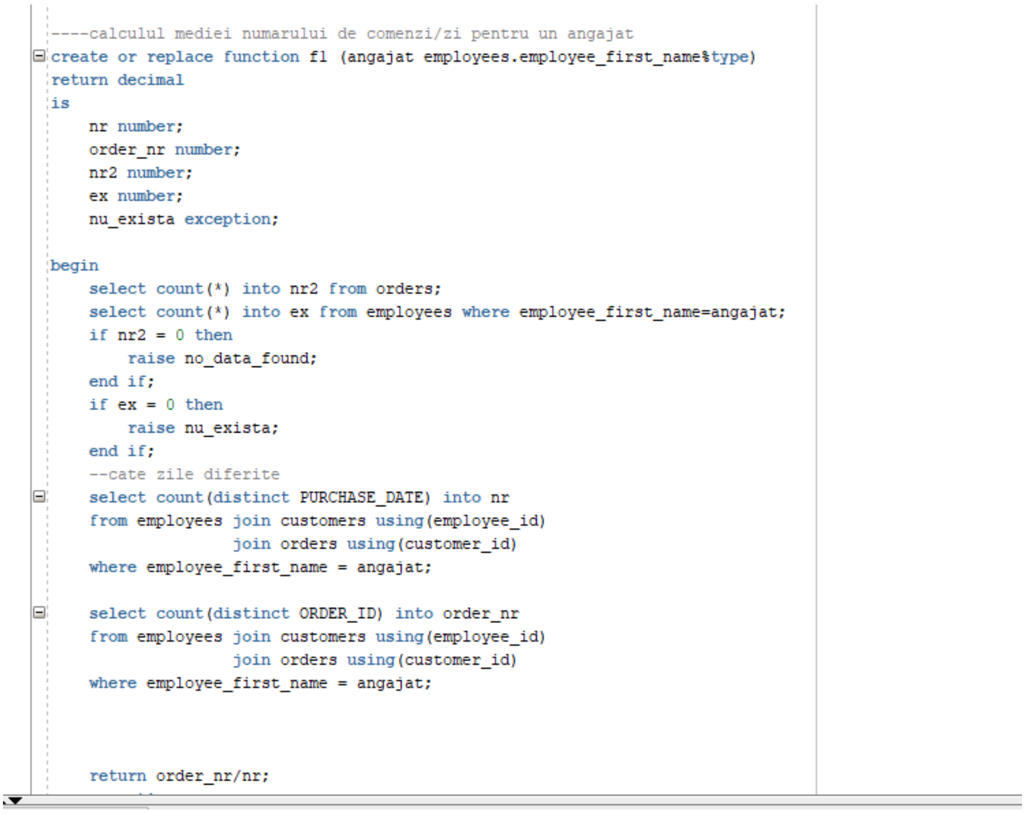
```
Pentru inghetata Magnum
avem 2 comenzi in luna ianuarie

Pentru inghetata Cornetto
avem 2 comenzi in luna ianuarie
Pentru inghetata Carte dOr
avem 2 comenzi in luna ianuarie
Pentru inghetata Klondike
avem 4 comenzi in luna ianuarie
Pentru inghetata Breyers
avem 1 comenzi in luna ianuarie
```



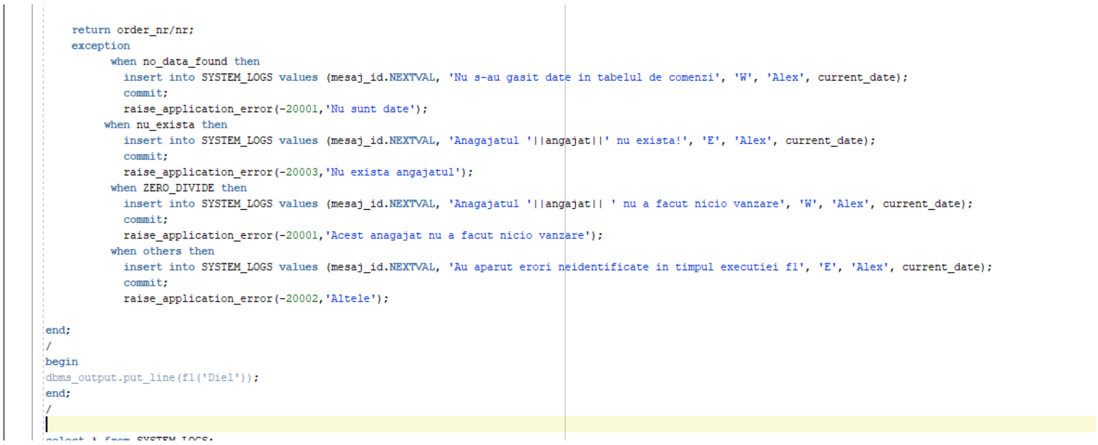
```
---pentru fiecare tip de inghetata, afisati detalii despre clientii ce au achizitionat-o,
---numarul comenzii,iar in cadrul ficarei comenzi, numele toppingurilor alese pentru respectiva comanda

create or replace type t2 as varray(50) of number;
/
create or replace procedure p2
as
cursor c is select distinct ice_cream_id, ice_cream_description
            from ice_cream_descriptions;
j number;
toppings_string varchar(128);
array_orders t2:= t2();
id number;
descr varchar(64);
begin
     for i in c loop
        id:= i.ice_cream_id;
        descr := i.ice_cream_description;
        dbms_output.put_line('Pentru inghetata ' || descr || ':');
        select order_id bulk collect into array_orders
        from orders
        where ice_cream_id = id;
        for j in array_orders.first..array_orders.last loop
            select listagg(topping_name, ',') into toppings_string
            from toppings join order_toppings using(topping_id)
            where order_id = array_orders(j);
            if toppings_string is null then
                dbms_output.put_line('Pentru comanda ' || array_orders(j)||' nu s-au comandat toppinguri');
            else
                dbms_output.put_line('Pentru comanda ' || array_orders(j)||' s-au folosit toppingurile - ' || toppings_string);
            end if;
        end loop;
     end loop;
end;
/
begin
p2();
```



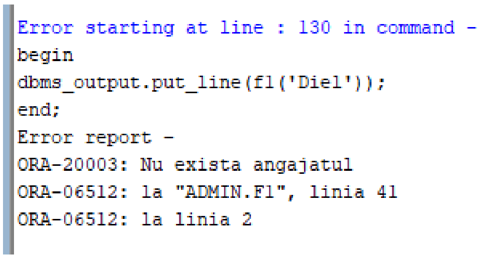
```
Pentru inghetata Magnum:
Pentru comanda 1 nu s-au comandat toppinguri
Pentru comanda 2 s-au folosit toppingurile - Oreo,Whipped cream,MandM-s
Pentru inghetata Cornetto:
Pentru comanda 9 s-au folosit toppingurile - Caramel
Pentru comanda 14 nu s-au comandat toppinguri
Pentru inghetata Breyers:
Pentru comanda 8 s-au folosit toppingurile - Twix
Pentru inghetata Klondike:
Pentru comanda 4 s-au folosit toppingurile - Oreo,Caramel
Pentru comanda 5 s-au folosit toppingurile - Caramel syrup,Twix
Pentru comanda 10 s-au folosit toppingurile - Oreo
Pentru comanda 11 s-au folosit toppingurile - Brownie bits
Pentru inghetata Carte dOr:
Pentru comanda 3 s-au folosit toppingurile - Chocolate syrup
Pentru comanda 6 s-au folosit toppingurile - Brownie bits,Chocolate syrup,Chocolate chips
Pentru comanda 7 s-au folosit toppingurile - Strawberries
Pentru comanda 12 s-au folosit toppingurile - Whipped cream,Caramel syrup
```



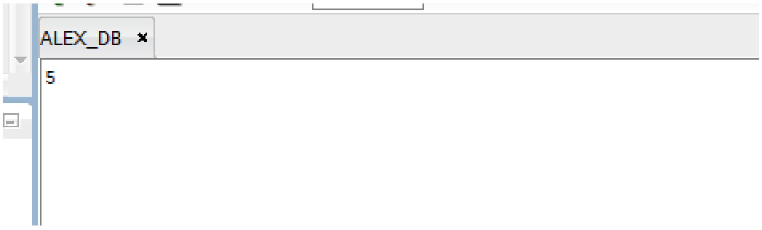
```
----calculul mediei numarului de comenzi/zi pentru un angajat
create or replace function f1 (angajat employees.employee_first_name%type)
return decimal
is
    nr number;
    order_nr number;
    nr2 number;
    ex number;
    nu_exista exception;

begin
    select count(*) into nr2 from orders;
    select count(*) into ex from employees where employee_first_name=angajat;
    if nr2 = 0 then
        raise no_data_found;
    end if;
    if ex = 0 then
        raise nu_exista;
    end if;
    --cate zile diferite
    select count(distinct PURCHASE_DATE) into nr
    from employees join customers using(employee_id)
                   join orders using(customer_id)
    where employee_first_name = angajat;

    select count(distinct ORDER_ID) into order_nr
    from employees join customers using(employee_id)
                   join orders using(customer_id)
    where employee_first_name = angajat;



    return order_nr/nr;
```



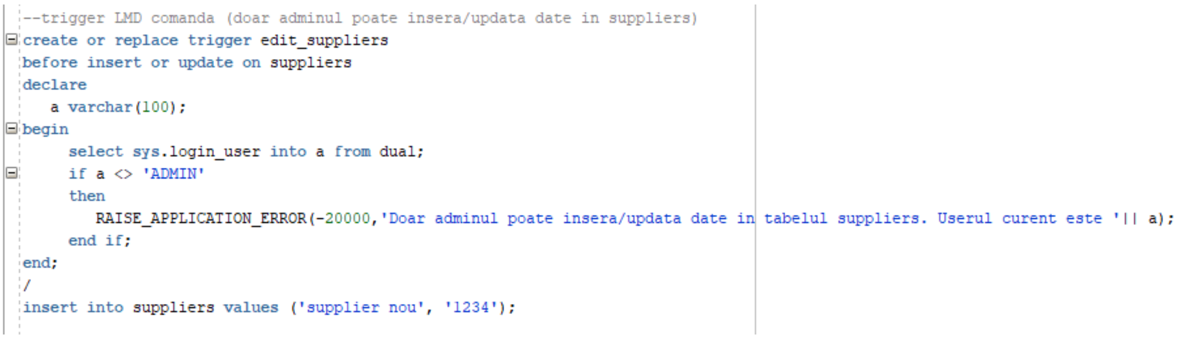
```
    return order_nr/nr;
    exception
          when no_data_found then
            insert into SYSTEM_LOGS values (mesaj_id.NEXTVAL, 'Nu s-au gasit date in tabelul de comenzi', 'W', 'Alex', current_date);
            commit;
            raise_application_error(-20001,'Nu sunt date');
         when nu_exista then
            insert into SYSTEM_LOGS values (mesaj_id.NEXTVAL, 'Anagajatul '||angajat||' nu exista!', 'E', 'Alex', current_date);
            commit;
            raise_application_error(-20003,'Nu exista angajatul');
          when ZERO_DIVIDE then
            insert into SYSTEM_LOGS values (mesaj_id.NEXTVAL, 'Anagajatul '||angajat|| ' nu a facut nicio vanzare', 'W', 'Alex', current_date);
            commit;
            raise_application_error(-20001,'Acest anagajat nu a facut nicio vanzare');
          when others then
            insert into SYSTEM_LOGS values (mesaj_id.NEXTVAL, 'Au aparut erori neidentificate in timpul executiei f1', 'E', 'Alex', current_date);
            commit;
            raise_application_error(-20002,'Altele');

end;
/
begin
dbms_output.put_line(f1('Diel'));
end;
/

select * from SYSTEM_LOGS;
```



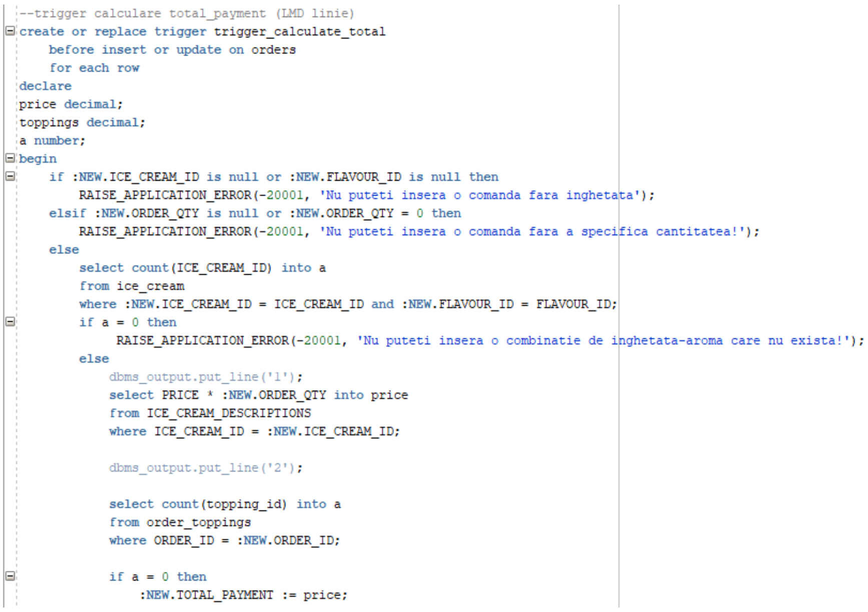
```
Error starting at line : 130 in command -
begin
dbms_output.put_line(f1('Diel'));
end;
Error report -
ORA-20003: Nu exista angajatul
ORA-06512: la "ADMIN.F1", linia 41
ORA-06512: la linia 2
```



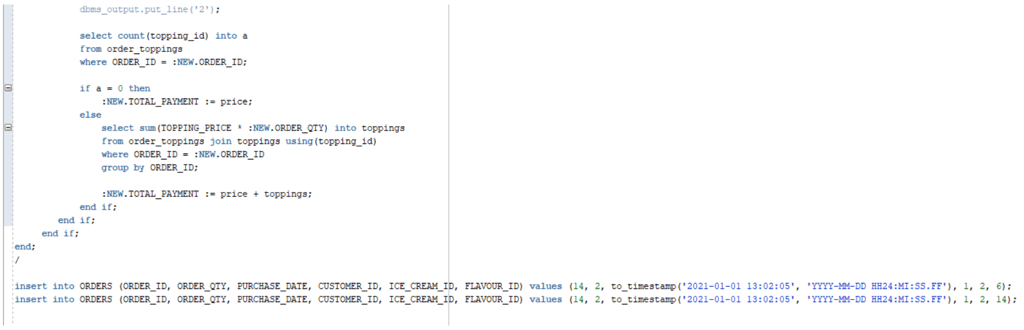
```
5
```



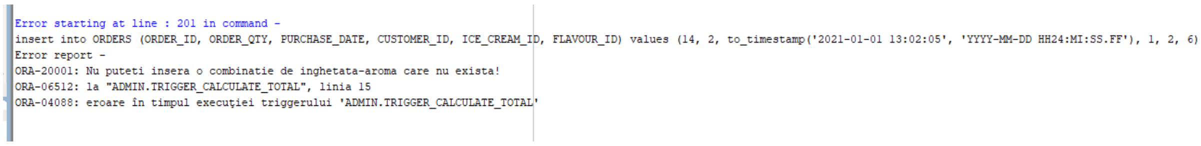
```
--trigger LMD comanda (doar adminul poate insera/updata date in suppliers)
create or replace trigger edit_suppliers
before insert or update on suppliers
declare
    a varchar(100);
begin
      select sys.login_user into a from dual;
      if a <> 'ADMIN'
      then
         RAISE_APPLICATION_ERROR(-20000,'Doar adminul poate insera/updata date in tabelul suppliers. Userul curent este '|| a);
      end if;
end;
/
insert into suppliers values ('supplier nou', '1234');
```



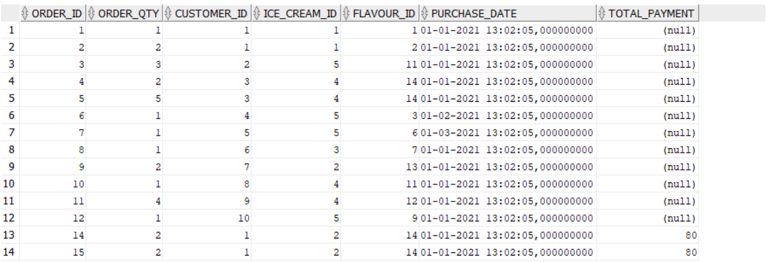
```
--trigger calculare total_payment (LMD linie)
create or replace trigger trigger_calculate_total
    before insert or update on orders
    for each row
declare
price decimal;
toppings decimal;
a number;
begin
    if :NEW.ICE_CREAM_ID is null or :NEW.FLAVOUR_ID is null then
        RAISE_APPLICATION_ERROR(-20001, 'Nu puteti insera o comanda fara inghetata');
    elsif :NEW.ORDER_QTY is null or :NEW.ORDER_QTY = 0 then
        RAISE_APPLICATION_ERROR(-20001, 'Nu puteti insera o comanda fara a specifica cantitatea!');
    else
        select count(ICE_CREAM_ID) into a
        from ice_cream
        where :NEW.ICE_CREAM_ID = ICE_CREAM_ID and :NEW.FLAVOUR_ID = FLAVOUR_ID;
        if a = 0 then
             RAISE_APPLICATION_ERROR(-20001, 'Nu puteti insera o combinatie de inghetata-aroma care nu exista!');
        else
            dbms_output.put_line('1');
            select PRICE * :NEW.ORDER_QTY into price
            from ICE_CREAM_DESCRIPTIONS
            where ICE_CREAM_ID = :NEW.ICE_CREAM_ID;

            dbms_output.put_line('2');

            select count(topping_id) into a
            from order_toppings
            where ORDER_ID = :NEW.ORDER_ID;

            if a = 0 then
                :NEW.TOTAL_PAYMENT := price;
```



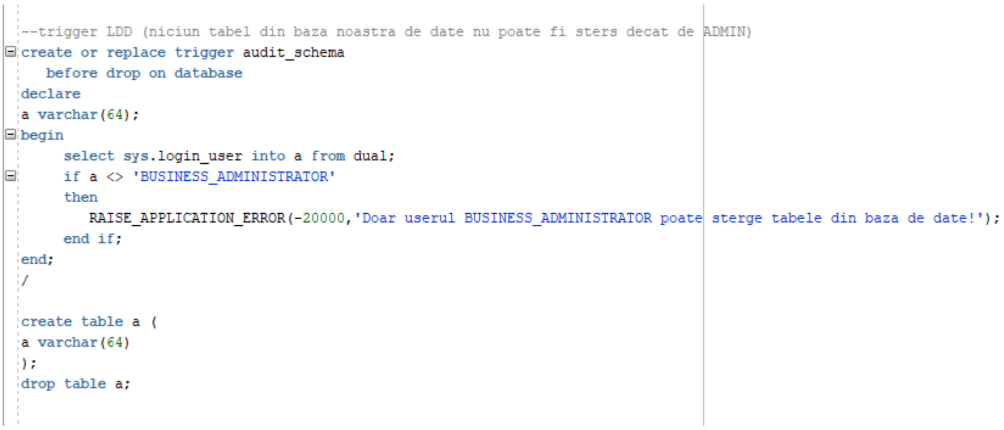
```
            dbms_output.put_line('2');

            select count(topping_id) into a
            from order_toppings
            where ORDER_ID = :NEW.ORDER_ID;

            if a = 0 then
                :NEW.TOTAL_PAYMENT := price;
            else
                select sum(TOPPING_PRICE * :NEW.ORDER_QTY) into toppings
                from order_toppings join toppings using(topping_id)
                where ORDER_ID = :NEW.ORDER_ID
                group by ORDER_ID;

                :NEW.TOTAL_PAYMENT := price + toppings;
            end if;
        end if;
    end if;
end;
/

insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (14, 2, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 1, 2, 6);
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (14, 2, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 1, 2, 14);
```



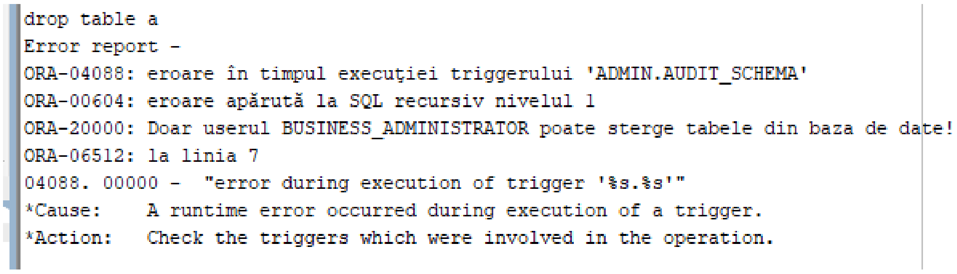
```
Error starting at line : 201 in command -
insert into ORDERS (ORDER_ID, ORDER_QTY, PURCHASE_DATE, CUSTOMER_ID, ICE_CREAM_ID, FLAVOUR_ID) values (14, 2, to_timestamp('2021-01-01 13:02:05', 'YYYY-MM-DD HH24:MI:SS.FF'), 1, 2, 6)
Error report -
ORA-20001: Nu puteti insera o combinatie de inghetata-aroma care nu exista!
ORA-06512: la "ADMIN.TRIGGER_CALCULATE_TOTAL", linia 15
ORA-04088: eroare în timpul execuţiei triggerului 'ADMIN.TRIGGER_CALCULATE_TOTAL'
```



	ORDER_ID	ORDER_QTY	CUSTOMER_ID	ICE_CREAM_ID	FLAVOUR_ID	PURCHASE_DATE	TOTAL_PAYMENT
1	1	1	1	1	1	01-01-2021 13:02:05,000000000	(null)
2	2	2	1	1	2	01-01-2021 13:02:05,000000000	(null)
3	3	3	2	5	11	01-01-2021 13:02:05,000000000	(null)
4	4	2	3	4	14	01-01-2021 13:02:05,000000000	(null)
5	5	5	3	4	14	01-01-2021 13:02:05,000000000	(null)
6	6	1	4	5	3	01-02-2021 13:02:05,000000000	(null)
7	7	1	5	5	6	01-03-2021 13:02:05,000000000	(null)
8	8	1	6	3	7	01-01-2021 13:02:05,000000000	(null)
9	9	2	7	2	13	01-01-2021 13:02:05,000000000	(null)
10	10	1	8	4	11	01-01-2021 13:02:05,000000000	(null)
11	11	4	9	4	12	01-01-2021 13:02:05,000000000	(null)
12	12	1	10	5	9	01-01-2021 13:02:05,000000000	(null)
13	14	2	1	2	14	01-01-2021 13:02:05,000000000	80
14	15	2	1	2	14	01-01-2021 13:02:05,000000000	80



```
--trigger LDD (niciun tabel din baza noastra de date nu poate fi sters decat de ADMIN)
create or replace trigger audit_schema
   before drop on database
declare
a varchar(64);
begin
     select sys.login_user into a from dual;
     if a <> 'BUSINESS_ADMINISTRATOR'
     then
        RAISE_APPLICATION_ERROR(-20000,'Doar userul BUSINESS_ADMINISTRATOR poate sterge tabele din baza de date!');
     end if;
end;
/

create table a (
a varchar(64)
);
drop table a;
```



```
drop table a
Error report -
ORA-04088: eroare în timpul execuției triggerului 'ADMIN.AUDIT_SCHEMA'
ORA-00604: eroare apărută la SQL recursiv nivelul 1
ORA-20000: Doar userul BUSINESS_ADMINISTRATOR poate sterge tabele din baza de date!
ORA-06512: la linia 7
04088. 00000 -  "error during execution of trigger '%s.%s'"
*Cause:    A runtime error occurred during execution of a trigger.
*Action:   Check the triggers which were involved in the operation.
```

Va multumesc!