**PROIECT**

**Baze de date**

**RESTAURANT**

**Cuprins**

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare
2. Prezentarea constrângerilor
3. Descrierea entităților
4. Descrierea relațiilor
5. Descrierea atributelor
6. Diagrama entitate-relație
7. Diagrama conceptuală
8. Enumerarea schemelor relaționale
9. Normalizările FN1, FN2, FN3
10. Crearea tabelelor SQL și inserarea datelor
11. Cereri SQL
12. Operații de actualizare și suprimare a datelor
13. Secvența utilizată în inserarea înregistrărilor în tabele
14. Vizualizare compusă
15. Index pentru a optimiza o cerere de tip căutare cu 2 criterii
16. Cerere ce utilizează operația de outer-join și cereri care utilizează operația division
17. Optimizarea unei cereri
18. Normalizarile BCNF, FN4, FN5

**1. Descrierea modelului real, a utilității acestuia și a**

**regulilor de funcționare**

Acest proiect cuprinde proiectarea unei baze de date pentru un restaurant, în cadrul căreia sunt reținute informații despre personalul, comenzile, meniul, clienți, mașinile pentru livrare și furnizorii acestei afaceri.

Restaurantul are și o platforma online (website și aplicație) prin cadrul căreia clienții pot da comandă.

Personalul din cadrul restaurantului este format din șef (sau șefi daca sunt mai multe personae care dețin împreuna restaurantul), manageri, bucatar-șef, bucatar, ajutor-bucatar, pizzer, ajutor-pizzer, barman, ajutor-barman, ospătar, șef de sală, livrator.

Furnizorii reprezintă diferite firme locale sau lanțuri de magazine prin care restaurantul se reaprovizionează.

Clienții pot fi de doua tipuri: în locație sau online. Plata comenzii se poate face numerar sau card, în funcție de preferința clientului, indiferent de tipul acestuia.

Angajații pot ocupa mai multe posturi, spre exemplu un ospătar poate fi și șef de sală.

**2. Constrângeri**

* Fiecare comandă trebuie să aibă o singură metodă de plată și un singur client
* În cadrul unei comenzi pot participa mai mulți bucătari (spre exemplu pizza este preparata de pizzer, iar desertul de bucatarul șef)
* Ingredientele alimentelor sunt furnizate de către o singură entitate comercială
* Dacă comanda este în locație, un singur ospătar se ocupă de aceasta, iar dacă comanda este pentru livrare, un singur livrator o va duce.
* Fiecare livrator are o singură mașină
* Fiecare mașină are ITP-ul făcut pe un an

**3. Descrierea entităților**

Entitățile necesare acestei afaceri sunt: comanda, mod\_plata, meniu, ingredient, furnizor, angajat, masina\_livrare, recepție, card, numerar.

Toate entitățile care vor fi prezentate sunt independente, mai puțin CARD și NUMERAR care sunt subentități ale superentității MOD\_PLATA.

|  |  |  |
| --- | --- | --- |
| **Entitate** | **Cheie Primară** | **Observații** |
| COMANDA | id\_comanda | Informații despre comandă |
| ANGAJAT | id\_angajat | Informații despre angajat |
| CLIENT | id\_client | Informații specifice clientului |
| MENIU | id\_produs | Informații despre fiecare preparat |
| FURNIZOR | id\_furnizor | Informații despre furnizor (societate comercială) |
| MOD\_PLATA | id\_plata | Modalitatea prin care se efectueaza plata |
| MASINA\_LIVRARE | nr\_inmatriculare | Informații specifice unei mașini |
| INGREDIENT | id\_ingredient | Informații despre ingredient |
| RECEPTIE | id\_receptie | Informații specifice recepției de marfă |
| CARD | id\_plata | Informații despre cardul folosit pentru plată |
| NUMERAR | id\_plata | Informații despre plata în numerar |

**4. Descrierea relațiilor**

|  |  |  |
| --- | --- | --- |
| **Relație** | **Cardinalitate** | **Observații** |
| are | * livrator->masina\_livrare   many-to-one   * comanda->mod\_plata   one-to-one   * comanda->produs   many-to-many   * ingredient->furnizor   many-to-one | * Un livrator are o singură mașină, însă o mașina poate avea niciunul, unul sau mai mulți șoferi; * Fiecare comandă are o singură modalitate de plată, însă mai multe comenzi pot avea aceeași modalitate de plată. Cardinalitatea minimă este 0; * O comandă are măcar un produs și un produs poate apărea in mai multe comenzi; * Un ingredient are un singur furnizor, însă un furnizor poate aproviziona cu mai multe ingrediente |
| este | * livrator->angajat   one-to-one   * card->mod\_plata   one-to-one   * numerar->mod\_plata   one-to-one | * Un livrator este angajat, însă un angajat poate avea alte funcții, deci cardinalitatea minimă este 0 și maximă 1; * Modalitatea card este o modalitate de plată, însă nu orice modalitate de plată este de tip card. Cardinalitatea minimă este 0; * Analog cu relația card->mod\_plata; |
| se ocupa de | * angajat->comanda   many-to-many | * Mai mulți angajați se pot ocupa de aceeași comandă (spre exemplu și barmanul și bucatarul). Cardinalitatea minima este 1; |
| face | * client->comanda   one-to-many | * Pentru a fi client, acesta are nevoie de minim o comandă dată. Cardinalitatea minimă este 1. Fiecare comandă are un singur client; |
| contine | * produs->ingredient   many-to-many | * Un produs conține mai multe ingrediente și un ingredient poate fi conținut în mai multe produse; |
| trimite | * furnizor->receptie   one-to-many | * Un furnizor poate trimite mai multe comenzi de aprovizionare către restaurant, însă o recepție are un singur furnizor de la care provine; |

**5. Descrierea atributelor**

ENTITATE: COMANDA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valori posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_comanda | number | 8 |  | NOT NULL |
| valoare | float | (8, 2) |  | NOT NULL |
| preparator | varchar | 256 | Numele angajatilor din bucatarie/bar | NOT NULL |
| continut | varchar | 512 | Nume din produsele din entitatea meniu | NOT NULL |
| persoana\_legatura | varchar | 30 | Numele unui angajat de tip ospatar/livrator | Persoana de legatura este ori un livrator ori un ospatar; NOT NULL |
| adresa | varchar | 512 |  | NULL |
| discount | float | 2 | [0,1] | NULL |

ENTITATE: ANGAJAT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valori posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_angajat | number | 8 |  | NOT NULL |
| nume | varchar | 20 |  | NOT NULL |
| prenume | varchar | 20 |  | NOT NULL |
| cnp | varchar | 13 |  | NOT NULL |
| nr\_telefon | varchar | 15 |  | NOT NULL |
| data\_angajare | date |  |  | NOT NULL |
| functie | varchar | 15 |  | NOT NULL |
| salariu | float | 8 | 0 | NOT NULL |

SUBENTITATE: LIVRATOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valori posibile și valori default** | **Observații, obligatoriu/opțional** |
| serie\_sasiu | varchar | 17 |  | Reprezinta masina alocata unui livrator; NOT NULL |

ENTITATE: CLIENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_client | number | 8 |  | NOT NULL |
| nume | varchar | 20 |  | NOT NULL |
| prenume | varchar | 20 |  | NOT NULL |
| nr\_telefon | varchar | 15 |  | NOT NULL |

ENTITATE: RECEPTIE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_receptie | number | 8 |  | NOT NULL |
| valoare | float | 10 |  | NOT NULL |
| data\_receptie | date |  |  | NOT NULL |
| produse | varchar | 2048 |  | NOT NULL |

ENTITATE: PRODUS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_produs | number | 8 |  | NOT NULL |
| nume | varchar | 30 |  | NOT NULL |
| gramaj | float | 5 |  | NOT NULL |
| pret | float | 8 |  | NOT NULL |
| ingrediente | varchar | 512 |  | NOT NULL |

ENTITATE: FURNIZOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_furnizor | number | 8 |  | NOT NULL |
| nume | varchar | 30 |  | NOT NULL |
| nr\_telefon | varchar | 15 |  | NOT NULL |

ENTITATE: MASINA\_LIVRARE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| serie\_sasiu | varchar | 17 |  | NOT NULL |
| nr\_inmatriculare | varchar | 8 |  | NOT NULL |
| data\_itp | date |  |  | NOT NULL, reprezintă data la care a fost efectuată ultima inspecție |

ENTITATE: MOD\_PLATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_plata | number | 8 |  | NOT NULL |
| valoare | float | (8, 2) |  | NOT NULL |

SUBENTITATE: CARD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| nr\_card | varchar | 16 |  | NOT NULL |
| nume | varchar | 20 |  | NOT NULL |
| prenume | varchar | 20 |  | NOT NULL |
| data\_expirare | date |  |  | NOT NULL |
| cvv | varchar | 3 |  | NOT NULL |

SUBENTITATE: NUMERAR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| observatii | varchar | 216 |  | NULL |

ENTITATE: INGREDIENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_ingredient | number | 8 |  | NOT NULL |
| nume | varchar | 20 |  | NOT NULL |
| stoc | float | 8 |  | NOT NULL |

RELATIE: ANGAJAT\_se\_ocupa\_de\_COMANDA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_comanda | number | 8 |  | NOT NULL |
| id\_angajat | number | 8 |  | NOT NULL |

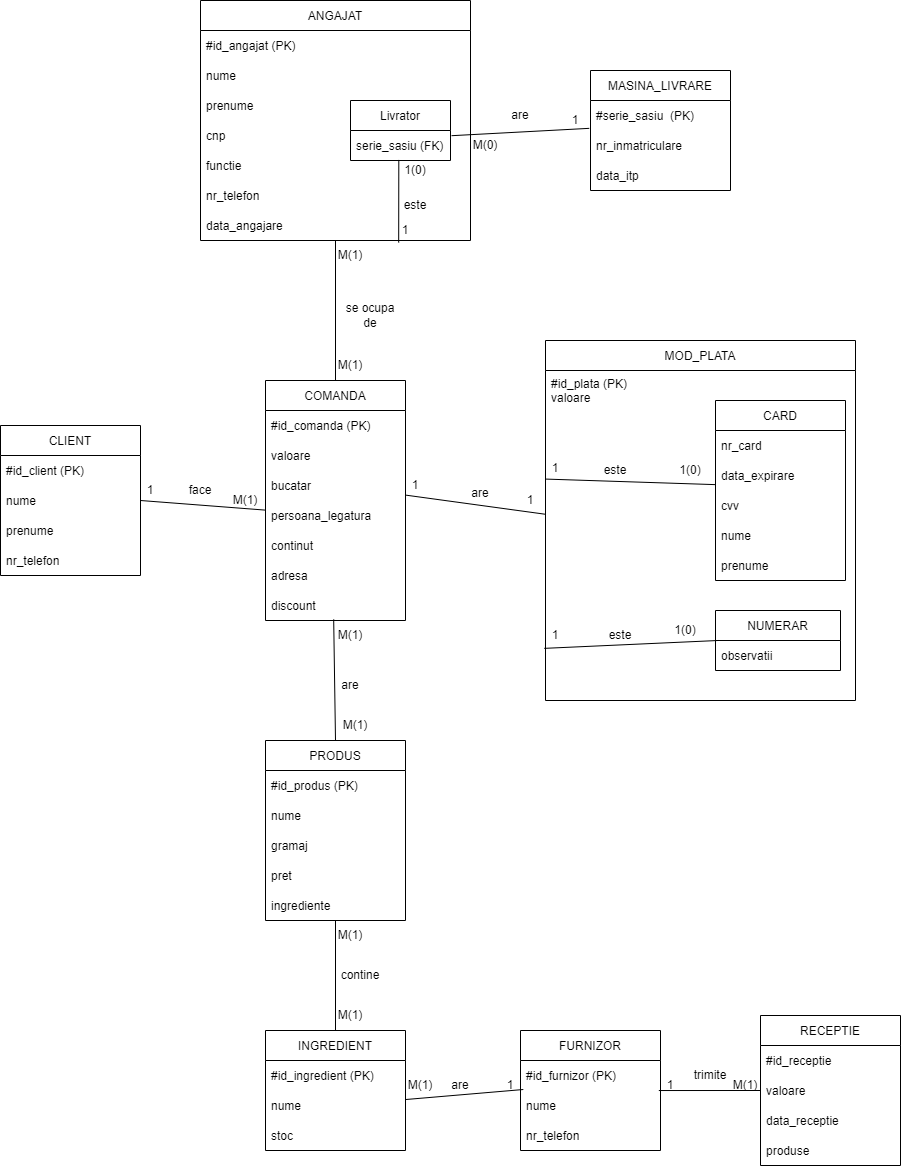
RELATIE: COMANDA\_are\_PRODUS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_produs | number | 8 |  | NOT NULL |
| id\_comanda | number | 8 |  | NOT NULL |
| cantitate | number | 4 | 1 | NOT NULL |

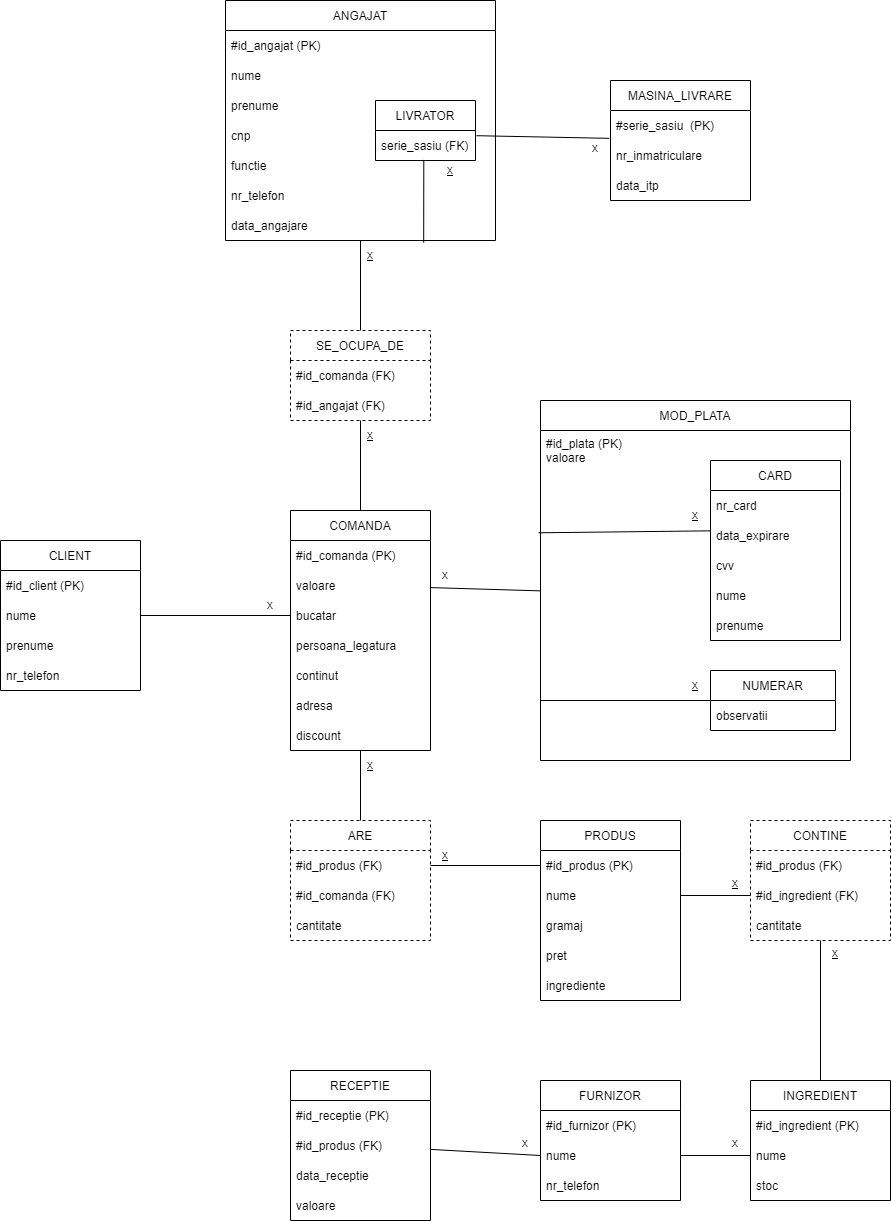
RELATIE: PRODUS\_contine\_INGREDIENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume coloana** | **Tip** | **Dimensiune/precizie** | **Valoari posibile și valori default** | **Observații, obligatoriu/opțional** |
| id\_produs | number | 8 |  | NOT NULL |
| id\_ingredient | number | 8 |  | NOT NULL |
| cantitate | number | 4 | 1 | NOT NULL |

**6. Diagrama entitate-relație**



**7. Diagrama conceptuală + 8. Enumerarea schemelor relaționale**

****

**9. FN1, FN2, FN3**

**FN1:**

Prima normalizare presupune ca fiecare relație/entitate să aibă atribute cu o valoare indivizibilă.

Astfel înainte de normalizare, tabelul ARE ar fi fost astfel:

|  |  |  |
| --- | --- | --- |
| #id\_produs | #id\_comanda | cantitate |
| 100, 123, 115, 120 | 25 | 1 |
| 30, 45, 98, 115, 115 | 26 | 1 |

Iar după normalizare arată astfel:

|  |  |  |
| --- | --- | --- |
| #id\_produs | #id\_comanda | cantitate |
| 100 | 25 | 1 |
| 123 | 25 | 1 |
| 115 | 25 | 1 |
| 120 | 25 | 1 |
| 30 | 26 | 1 |
| 45 | 26 | 1 |
| 98 | 26 | 1 |
| 115 | 26 | 2 |

**FN2:**

Pentru ca o relație să fie în NF2 trebuie să respecte NF1 și în plus, fiecare atribut care nu este cheie să fie dependent de întreaga cheie primară.

Entitatea PRODUS conține atributul ingrediente, care presupune o înlănțuire de denumiri de ingrediente, prezente și în entitatea INGREDIENT.

FN1->FN2:

COMANDA(#id\_comanda, valoare, bucatar, persoana\_legatura, continut, adresa, discount) -> COMANDA(#id\_comanda, valoare, bucatar, #id\_persoana\_legatura, adresa, discount);

PRODUS(#id\_produs, nume, gramaj, pret, ingrediente) -> PRODUS(#id\_produs, nume, gramaj, pret);

RECEPTIE(#id\_receptie, data\_receptie, valoare, ingrediente) -> RECEPTIE(#id\_receptie, #id\_ingredient, data\_receptie, valoare, cantitate);

**FN3:**

O relație este în FN3 dacă se regăsește in FN2 și dacă fiecare atribut care nu este cheie depinde direct de cheia primară.

Putem observa că în relația COMANDA și în MOD\_PLATA avem atributul valoare reținut redundant, mai exact acesta se poate calcula din prețul fiecărui produs, cantitatea acestuia în o comandă și discount-ul aplicat comenzii.

FN2 -> FN3:

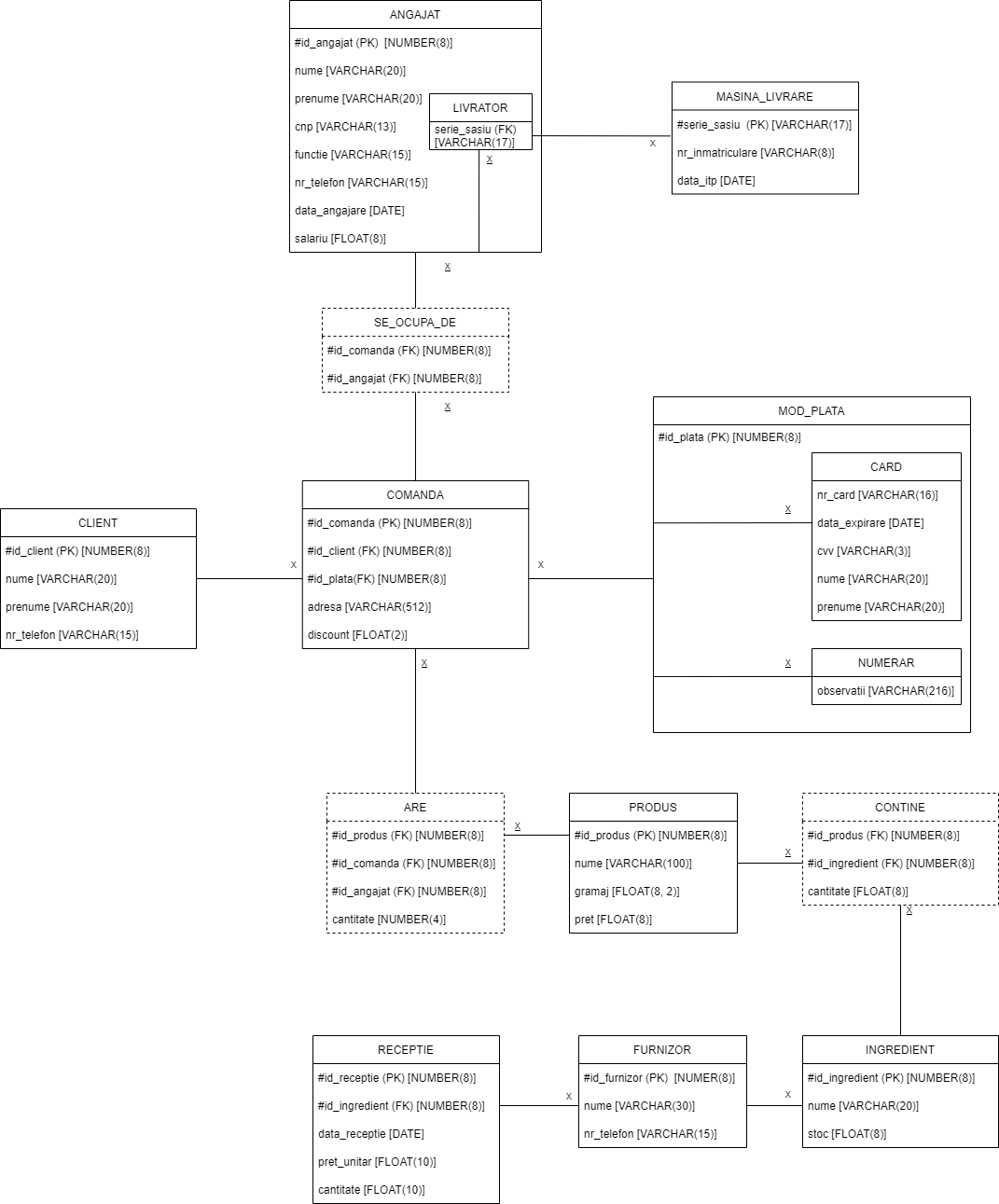
MOD\_PLATA(#id\_plata, valoare) -> MOD\_PLATA(#id\_plata)

COMANDA(#id\_comanda, #id\_client, #id\_persoana\_legatura, valoare, adresa, discount) -> COMANDA(#id\_comanda, #id\_client, adresa, discount);

ARE(#id\_produs, #id\_comanda, cantitate) -> ARE(#id\_produs, #id\_comanda, #id\_angajat, cantitate)

RECEPTIE(#id\_receptie, #id\_produs, data\_receptie, valoare) -> RECEPTIE(#id\_receptie, #id\_produs, data\_receptie, pret\_unitar, cantitate)

Astfel schema conceptuala dupa cele trei normalizari este:



**10. Crearea tabelelor SQL și inserarea datelor + 13. Secvența utilizată în inserarea înregistrărilor în tabele**

**ANGAJATI:**

Creare tabel:

CREATE TABLE angajati

(id\_angajat number(8),

nume varchar(20),

prenume varchar(20),

cnp varchar(13),

functie varchar(15),

nr\_telefon varchar(15),

data\_angajare date,

salariu float(8),

CONSTRAINT angajat\_pk PRIMARY KEY (id\_angajat));

Creare secvență:

CREATE SEQUENCE SEQ\_ANGAJAT

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 1000

NOCYCLE;

Introducere date:

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Mihai', 'Radu', '2323232323323', 'bucatar-sef', '0747347365', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 4500);

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Roman', 'Anastasia', '2323232323323', 'bucatar', '0747308635', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 3500);

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Bran', 'Andrei', '2323232323323', 'manager', '0747347365', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 4400);

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Paduraru', 'Adi', '2323232323323', 'barman', '0747349365', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 3000);

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Ana', 'Andreea', '2323232323323', 'ospatar', '0747893365', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 2500);

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Popescu', 'Andrei', '2323232323323', 'pizzer', '0747374365', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 4400);

INSERT INTO angajati values(SEQ\_ANGAJAT.NEXTVAL, 'Axinte', 'Iulian', '2323232323323', 'livrator', '0744597365', TO\_DATE('29/12/2015', 'DD/MM/YYYY'), 2200);



**MASINI\_LIVRARE:**

Creare tabel:

CREATE TABLE masini\_livrare

(serie\_sasiu varchar(17),

nr\_inmatriculare varchar(8),

data\_itp date,

CONSTRAINT masina\_livrare\_pk PRIMARY KEY (serie\_sasiu));

Introducere date:

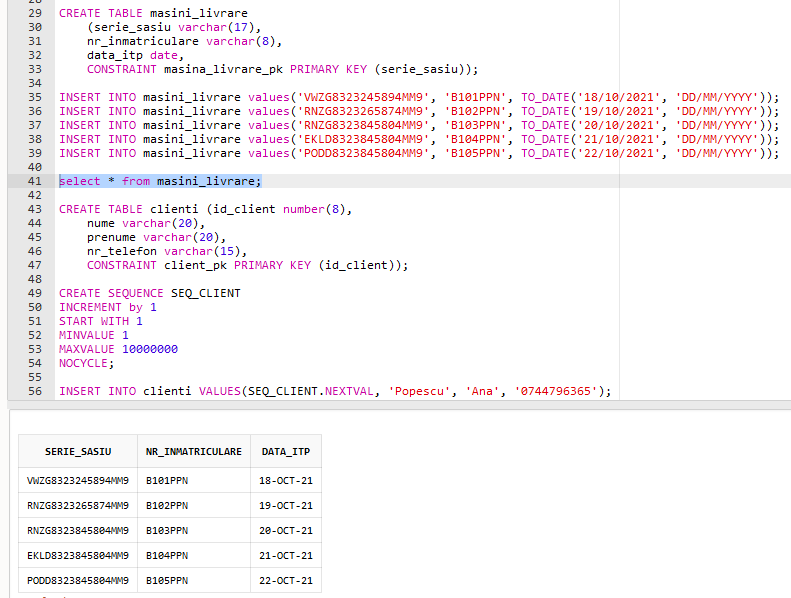
INSERT INTO masini\_livrare values('VWZG8323245894MM9', 'B101PPN', TO\_DATE('18/10/2021', 'DD/MM/YYYY'));

INSERT INTO masini\_livrare values('RNZG8323265874MM9', 'B102PPN', TO\_DATE('19/10/2021', 'DD/MM/YYYY'));

INSERT INTO masini\_livrare values('RNZG8323845804MM9', 'B103PPN', TO\_DATE('20/10/2021', 'DD/MM/YYYY'));

INSERT INTO masini\_livrare values('EKLD8323845804MM9', 'B104PPN', TO\_DATE('21/10/2021', 'DD/MM/YYYY'));

INSERT INTO masini\_livrare values('PODD8323845804MM9', 'B105PPN', TO\_DATE('22/10/2021', 'DD/MM/YYYY'));



**CLIENTI:**

Creare tabel:

CREATE TABLE clienti

(id\_client number(8),

nume varchar(20),

prenume varchar(20),

nr\_telefon varchar(15),

CONSTRAINT client\_pk PRIMARY KEY (id\_client));

Creare secvență:

CREATE SEQUENCE SEQ\_CLIENT

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 10000000

NOCYCLE;

Introducere date:

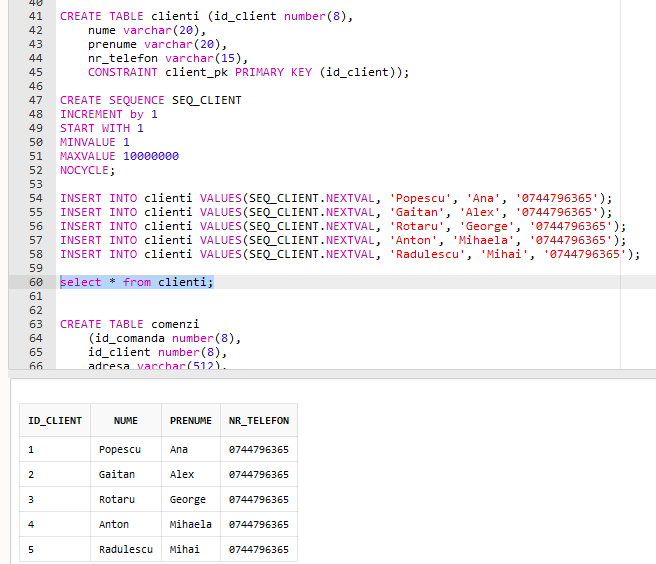
INSERT INTO clienti VALUES(SEQ\_CLIENT.NEXTVAL, 'Popescu', 'Ana', '0744796365');

INSERT INTO clienti VALUES(SEQ\_CLIENT.NEXTVAL, 'Gaitan', 'Alex', '0744796365');

INSERT INTO clienti VALUES(SEQ\_CLIENT.NEXTVAL, 'Rotaru', 'George', '0744796365');

INSERT INTO clienti VALUES(SEQ\_CLIENT.NEXTVAL, 'Anton', 'Mihaela', '0744796365');

INSERT INTO clienti VALUES(SEQ\_CLIENT.NEXTVAL, 'Radulescu', 'Mihai', '0744796365');



**MOD\_PLATA:**

Creare tabel:

CREATE TABLE mod\_plata

(id\_plata number (8),

CONSTRAINT mod\_plata\_pk PRIMARY KEY (id\_plata));

Creare secvență:

CREATE SEQUENCE SEQ\_MOD\_PLATA

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 10000000

NOCYCLE;

Introducere date:

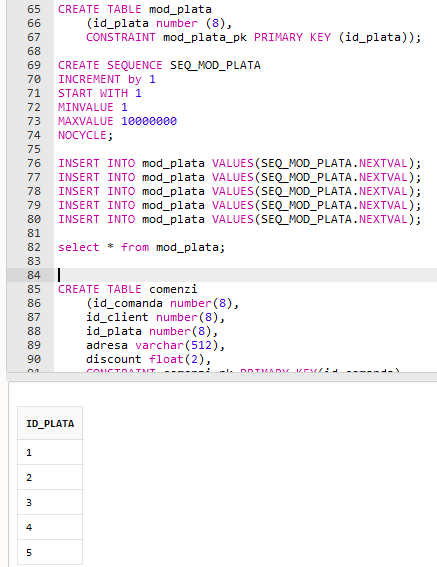
INSERT INTO mod\_plata VALUES(SEQ\_MOD\_PLATA.NEXTVAL);

INSERT INTO mod\_plata VALUES(SEQ\_MOD\_PLATA.NEXTVAL);

INSERT INTO mod\_plata VALUES(SEQ\_MOD\_PLATA.NEXTVAL);

INSERT INTO mod\_plata VALUES(SEQ\_MOD\_PLATA.NEXTVAL);

INSERT INTO mod\_plata VALUES(SEQ\_MOD\_PLATA.NEXTVAL);



**COMENZI:**

Creare tabel:

CREATE TABLE comenzi

(id\_comanda number(8),

id\_client number(8),

id\_plata number(8),

adresa varchar(512),

discount float(2),

CONSTRAINT comenzi\_pk PRIMARY KEY(id\_comanda),

CONSTRAINT comenzi\_id\_client\_fk FOREIGN KEY(id\_client) REFERENCES clienti(id\_client),

CONSTRAINT comenzi\_id\_plata\_fk FOREIGN KEY(id\_plata) REFERENCES mod\_plata(id\_plata));

Creare secvență:

CREATE SEQUENCE SEQ\_COMENZI

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 10000000

NOCYCLE;

Introducere date:

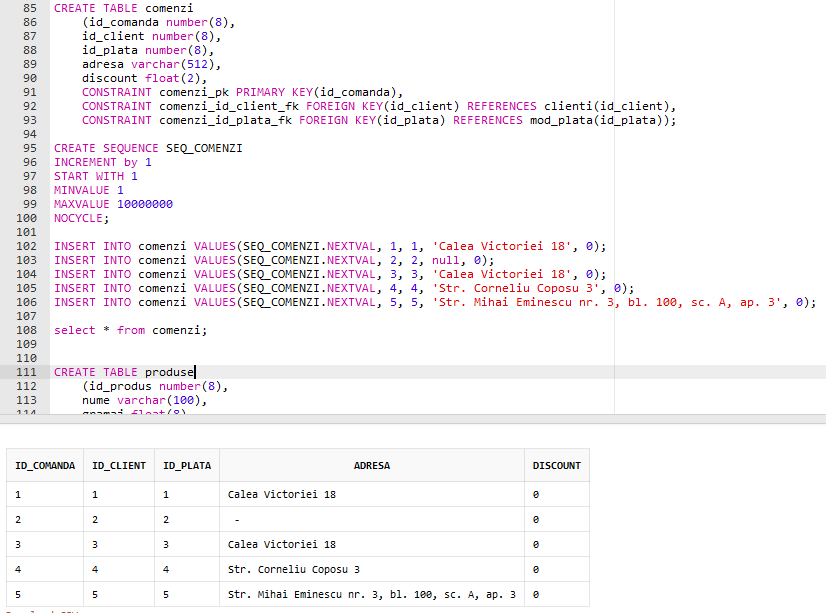
INSERT INTO comenzi VALUES(SEQ\_COMENZI.NEXTVAL, 1, 1, 'Calea Victoriei 18', 0);

INSERT INTO comenzi VALUES(SEQ\_COMENZI.NEXTVAL, 2, 2, null, 0);

INSERT INTO comenzi VALUES(SEQ\_COMENZI.NEXTVAL, 3, 3, 'Calea Victoriei 18', 0);

INSERT INTO comenzi VALUES(SEQ\_COMENZI.NEXTVAL, 4, 4, 'Str. Corneliu Coposu 3', 0);

INSERT INTO comenzi VALUES(SEQ\_COMENZI.NEXTVAL, 5, 5, 'Str. Mihai Eminescu nr. 3, bl. 100, sc. A, ap. 3', 0);



**PRODUSE:**

Creare tabel:

CREATE TABLE produse

(id\_produs number(8),

nume varchar(100),

gramaj float(8),

pret float(8),

CONSTRAINT produse\_pk PRIMARY KEY (id\_produs));

Creare secvență:

CREATE SEQUENCE SEQ\_PRODUSE

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 1000

NOCYCLE;

Introducere date:

INSERT INTO produse VALUES(SEQ\_PRODUSE.NEXTVAL, 'Salata Greceasca', 350, 24.5);

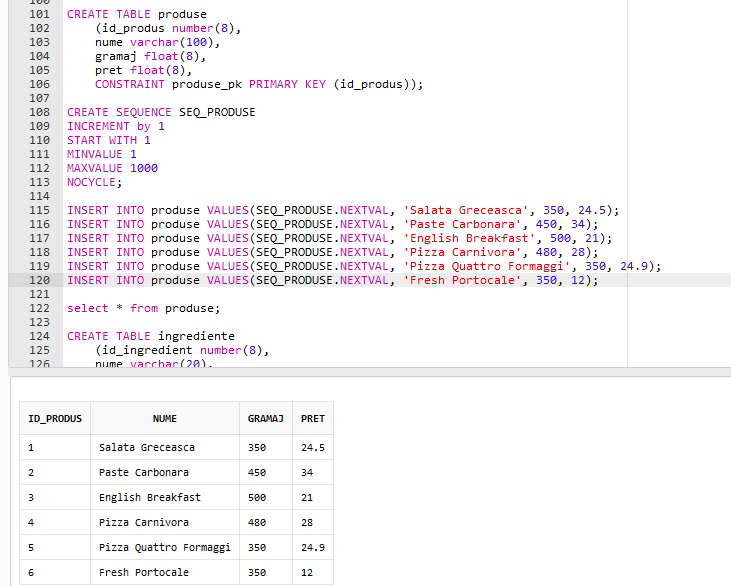
INSERT INTO produse VALUES(SEQ\_PRODUSE.NEXTVAL, 'Paste Carbonara', 450, 34);

INSERT INTO produse VALUES(SEQ\_PRODUSE.NEXTVAL, 'English Breakfast', 500, 21);

INSERT INTO produse VALUES(SEQ\_PRODUSE.NEXTVAL, 'Pizza Carnivora', 480, 28);

INSERT INTO produse VALUES(SEQ\_PRODUSE.NEXTVAL, 'Pizza Quattro Formaggi', 350, 24.9);

INSERT INTO produse VALUES(SEQ\_PRODUSE.NEXTVAL, 'Fresh Portocale', 350, 12);



**INGREDIENTE:**

Creare tabel:

CREATE TABLE ingrediente

(id\_ingredient number(8),

nume varchar(20),

stoc float(8),

CONSTRAINT ingrediente\_pk PRIMARY KEY (id\_ingredient));

Creare secvență:

CREATE SEQUENCE SEQ\_INGREDIENTE

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 1000

NOCYCLE;

Introducere date:

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'ou', 50);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'lapte', 6);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'faina', 50);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'pepperoni', 5);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'cheddar', 10);

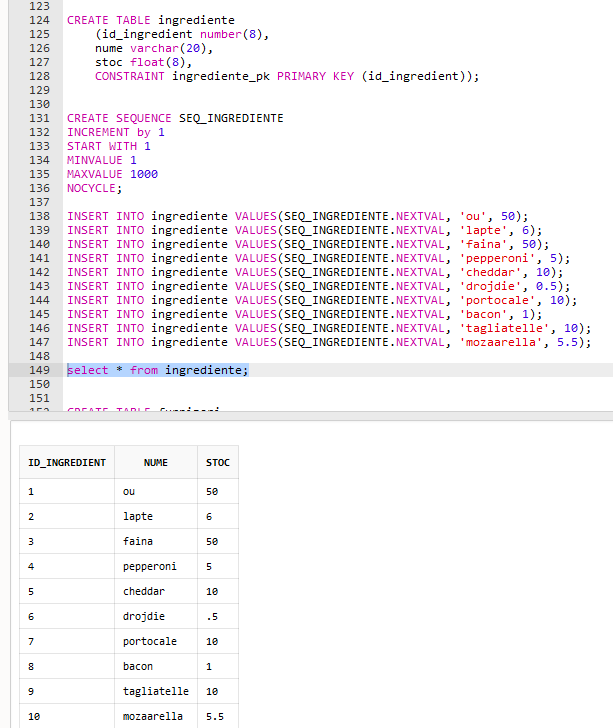
INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'drojdie', 0.5);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'portocale', 10);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'bacon', 1);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'tagliatelle', 10);

INSERT INTO ingrediente VALUES(SEQ\_INGREDIENTE.NEXTVAL, 'mozzarella', 5.5);



**FURNIZORI:**

Creare tabel:

CREATE TABLE furnizori

(id\_furnizor number(8),

nume varchar(30),

nr\_telefon varchar(15),

CONSTRAINT furnizori\_pk PRIMARY KEY (id\_furnizor));

Creare secvență:

CREATE SEQUENCE SEQ\_FURNIZOR

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 1000

NOCYCLE;

Introducere date:

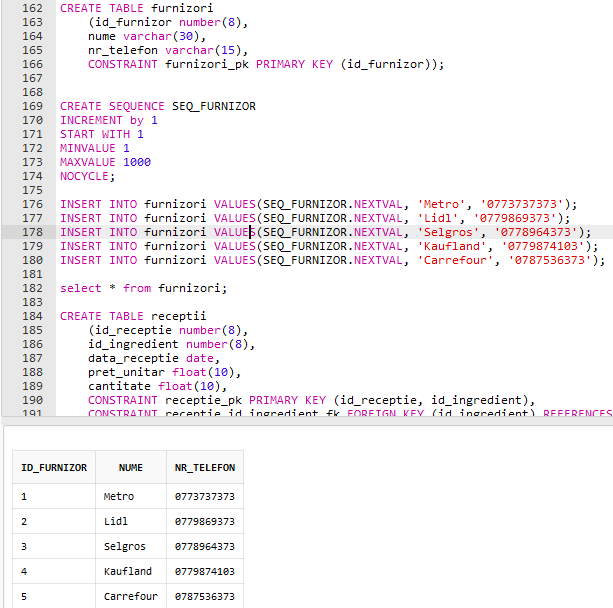
INSERT INTO furnizori VALUES(SEQ\_FURNIZOR.NEXTVAL, 'Metro', '0773737373');

INSERT INTO furnizori VALUES(SEQ\_FURNIZOR.NEXTVAL, 'Lidl', '0779869373');

INSERT INTO furnizori VALUES(SEQ\_FURNIZOR.NEXTVAL, 'Selgros', '0778964373');

INSERT INTO furnizori VALUES(SEQ\_FURNIZOR.NEXTVAL, 'Kaufland', '0779874103');

INSERT INTO furnizori VALUES(SEQ\_FURNIZOR.NEXTVAL, 'Carrefour', '0787536373');



**RECEPTII:**

Creare tabel:

CREATE TABLE receptii

(id\_receptie number(8),

id\_ingredient number(8),

id\_furnizor number(8),

data\_receptie date,

pret\_unitar float(10),

cantitate float(10),

CONSTRAINT receptie\_pk PRIMARY KEY (id\_receptie, id\_ingredient, id\_furnizor),

CONSTRAINT receptie\_id\_ingredient\_fk FOREIGN KEY (id\_ingredient) REFERENCES ingrediente(id\_ingredient),

CONSTRAINT receptie\_id\_furnizor\_fk FOREIGN KEY(id\_furnizor) REFERENCES furnizori(id\_furnizor));

Creare secvență:

CREATE SEQUENCE SEQ\_RECEPTIE

INCREMENT by 1

START WITH 1

MINVALUE 1

MAXVALUE 1000

NOCYCLE;

Introducere date:

INSERT INTO receptii VALUES(SEQ\_RECEPTIE.NEXTVAL, 1, 1, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 0.5, 50);

INSERT INTO receptii VALUES(1, 2, 1, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 4.33, 6);

INSERT INTO receptii VALUES(1, 3, 1, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 3.5, 50);

INSERT INTO receptii VALUES(1, 4, 1, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 23.5, 5);

INSERT INTO receptii VALUES(1, 5, 1, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 15.59, 10);

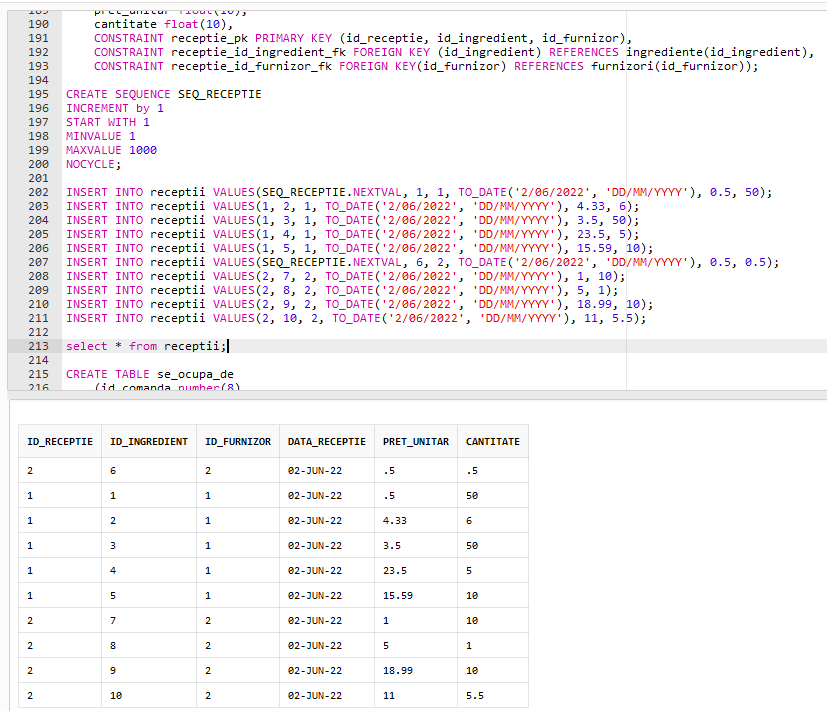
INSERT INTO receptii VALUES(SEQ\_RECEPTIE.NEXTVAL, 6, 2, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 0.5, 0.5);

INSERT INTO receptii VALUES(2, 7, 2, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 1, 10);

INSERT INTO receptii VALUES(2, 8, 2, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 5, 1);

INSERT INTO receptii VALUES(2, 9, 2, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 18.99, 10);

INSERT INTO receptii VALUES(2, 10, 2, TO\_DATE('2/06/2022', 'DD/MM/YYYY'), 11, 5.5);



**SE\_OCUPA\_DE:**

Creare tabel:

CREATE TABLE se\_ocupa\_de

(id\_comanda number(8),

id\_angajat number(8),

CONSTRAINT se\_ocupa\_de\_id\_comanda\_fk FOREIGN KEY (id\_comanda) REFERENCES comenzi(id\_comanda),

CONSTRAINT se\_ocupa\_de\_id\_angajat\_fk FOREIGN KEY (id\_angajat) REFERENCES angajati(id\_angajat));

Introducere date:

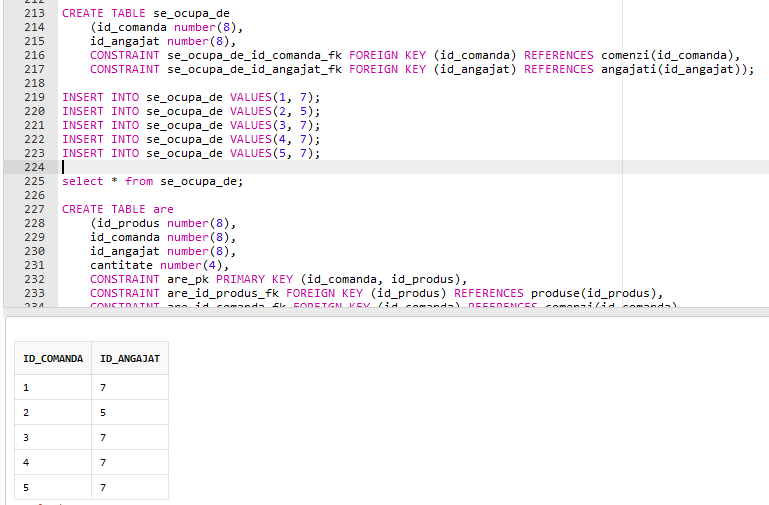
INSERT INTO se\_ocupa\_de VALUES(1, 7);

INSERT INTO se\_ocupa\_de VALUES(2, 5);

INSERT INTO se\_ocupa\_de VALUES(3, 7);

INSERT INTO se\_ocupa\_de VALUES(4, 7);

INSERT INTO se\_ocupa\_de VALUES(5, 7);



**ARE:**

Creare tabel:

CREATE TABLE are

(id\_produs number(8),

id\_comanda number(8),

id\_angajat number(8),

cantitate number(4),

CONSTRAINT are\_pk PRIMARY KEY (id\_comanda, id\_produs),

CONSTRAINT are\_id\_produs\_fk FOREIGN KEY (id\_produs) REFERENCES produse(id\_produs),

CONSTRAINT are\_id\_comanda\_fk FOREIGN KEY (id\_comanda) REFERENCES comenzi(id\_comanda),

CONSTRAINT are\_id\_angajat\_fk FOREIGN KEY (id\_angajat) REFERENCES angajati(id\_angajat));

Introducere date:

INSERT INTO are VALUES(1, 1, 1, 1);

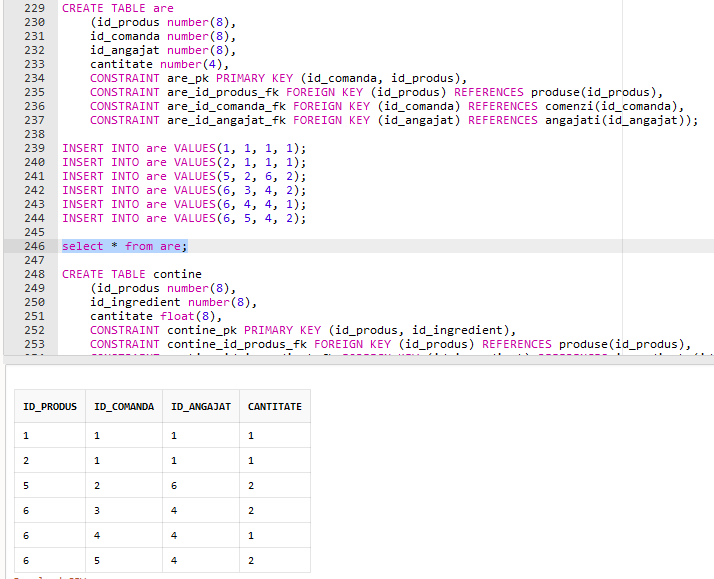
INSERT INTO are VALUES(2, 1, 1, 1);

INSERT INTO are VALUES(5, 2, 6, 2);

INSERT INTO are VALUES(6, 3, 4, 2);

INSERT INTO are VALUES(6, 4, 4, 1);

INSERT INTO are VALUES(6, 5, 4, 2);



**CONTINE:**

Creare tabel:

CREATE TABLE contine

(id\_produs number(8),

id\_ingredient number(8),

cantitate float(8),

CONSTRAINT contine\_pk PRIMARY KEY (id\_produs, id\_ingredient),

CONSTRAINT contine\_id\_produs\_fk FOREIGN KEY (id\_produs) REFERENCES produse(id\_produs),

CONSTRAINT contine\_id\_ingredient\_fk FOREIGN KEY (id\_ingredient) REFERENCES ingrediente(id\_ingredient));

Introducere date:

INSERT INTO contine VALUES(1, 8, 0.3);

INSERT INTO contine VALUES(1, 10, 0.1);

INSERT INTO contine VALUES(2, 9, 0.4);

INSERT INTO contine VALUES(2, 10, 0.1);

INSERT INTO contine VALUES(3, 1, 2);

INSERT INTO contine VALUES(3, 8, 0.1);

INSERT INTO contine VALUES(4, 6, 0.025);

INSERT INTO contine VALUES(4, 3, 0.2);

INSERT INTO contine VALUES(4, 8, 0.1);

INSERT INTO contine VALUES(4, 4, 0.1);

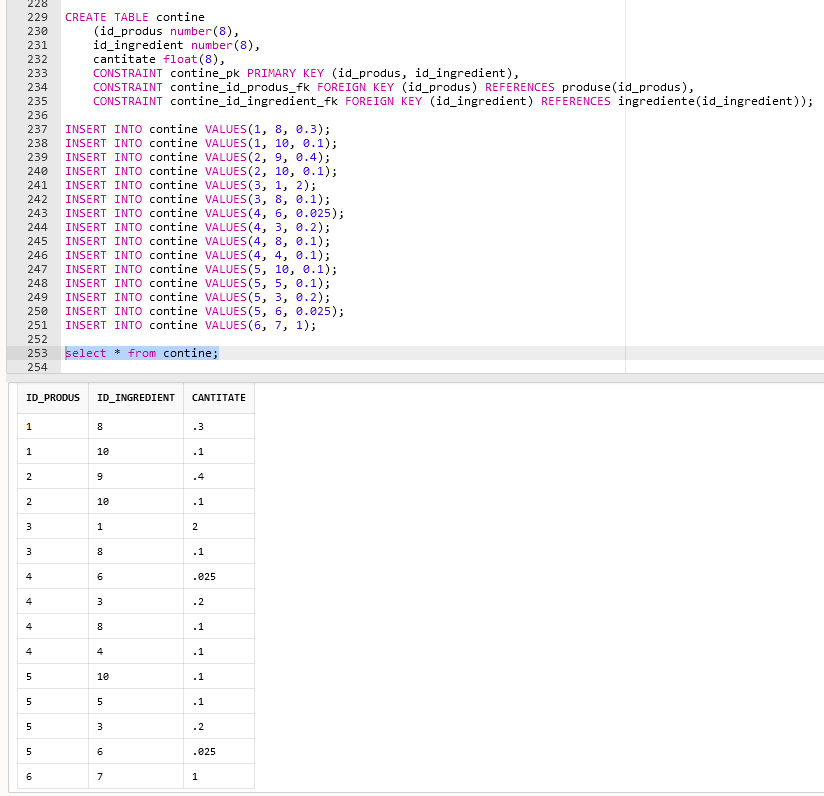
INSERT INTO contine VALUES(5, 10, 0.1);

INSERT INTO contine VALUES(5, 5, 0.1);

INSERT INTO contine VALUES(5, 3, 0.2);

INSERT INTO contine VALUES(5, 6, 0.025);

INSERT INTO contine VALUES(6, 7, 1);



**11. Cereri SQL**

*1. Să se listeze toate ingredientele, numarul comenzii, clientul și plata comenzilor care au fost livrate de livratorul Axinte Iulian, rezultatele vor fi ordonate în funcție de id-ul comenzii.*

SELECT Comenzi.id\_comanda, Ingrediente.nume, Clienti.nume, Clienti.prenume, Mod\_plata.id\_plata FROM comenzi Comenzi

JOIN clienti Clienti ON (Comenzi.id\_client=Clienti.id\_client)

JOIN mod\_plata Mod\_plata ON (Comenzi.id\_plata=Mod\_plata.id\_plata)

JOIN are are ON (Comenzi.id\_comanda=Are.id\_comanda)

JOIN produse Produse ON (Produse.id\_produs=Are.id\_produs)

JOIN contine Contine ON (Produse.id\_produs=Contine.id\_produs)

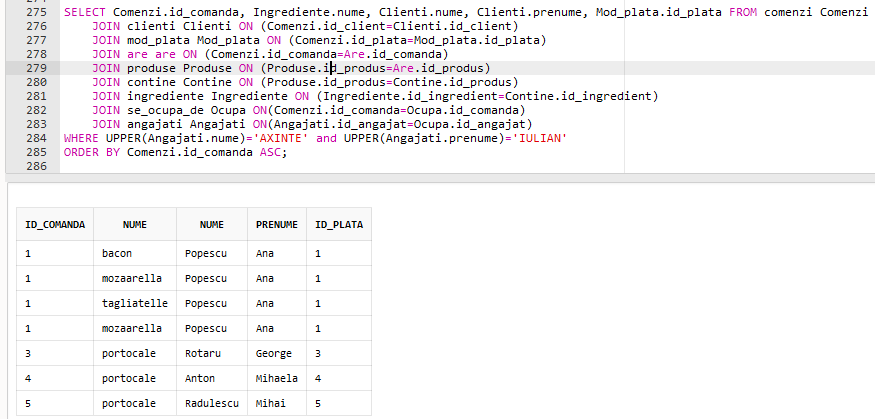
JOIN ingrediente Ingrediente ON (Ingrediente.id\_ingredient=Contine.id\_ingredient)

JOIN se\_ocupa\_de Ocupa ON(Comenzi.id\_comanda=Ocupa.id\_comanda)

JOIN angajati Angajati ON(Angajati.id\_angajat=Ocupa.id\_angajat)

WHERE UPPER(Angajati.nume)='AXINTE' and UPPER(Angajati.prenume)='IULIAN'

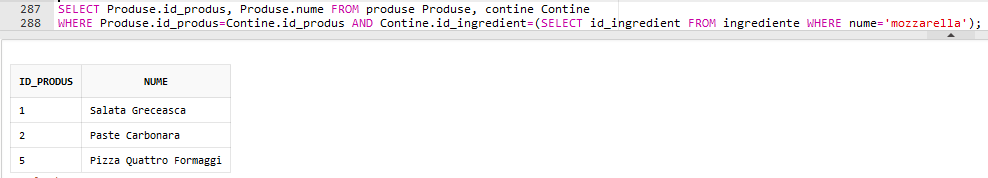
ORDER BY Comenzi.id\_comanda ASC;



*2. Să se afișeze toate produsele ce conțin mozzarella*

SELECT Produse.id\_produs, Produse.nume FROM produse Produse, contine Contine

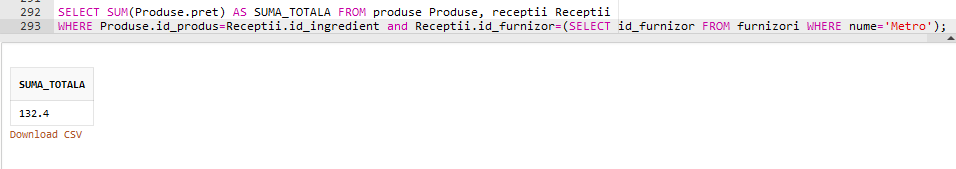
WHERE Produse.id\_produs=Contine.id\_produs AND Contine.id\_ingredient=(SELECT id\_ingredient FROM ingrediente WHERE nume='mozzarella');



*3. Să se afișeze suma totală a produselor din meniu care contin ingrediente de la furnizorul Metro*

SELECT SUM(Produse.pret) AS SUMA\_TOTALA FROM produse Produse, receptii Receptii

WHERE Produse.id\_produs=Receptii.id\_ingredient and Receptii.id\_furnizor=(SELECT id\_furnizor FROM furnizori WHERE nume='Metro');



*4. Să se afișeze totalul fiecărei comenzi, id-ul acesteia, id-ul clientului și tipul comenzii (daca este dusă de un livrator atunci tipul acesteia va fi ‘Livrare’ și în caz contrar va fi ’Restaurant’*

WITH totalComenzi AS(SELECT Comenzi.id\_comanda, Comenzi.id\_client, SUM(Produse.pret \* Are.cantitate) as total

FROM comenzi Comenzi, produse Produse, are Are

WHERE Comenzi.id\_comanda=Are.id\_comanda and Are.id\_produs=Produse.id\_produs

GROUP BY Comenzi.id\_comanda, Comenzi.id\_client),

tabel AS (SELECT DISTINCT t.id\_comanda, t.id\_client, t.total,

CASE

WHEN Ocupa.id\_comanda=t.id\_comanda and Ocupa.id\_angajat IN (SELECT id\_angajat FROM angajati WHERE UPPER(functie)='LIVRATOR') THEN 'Livrare'

ELSE 'Restaurant'

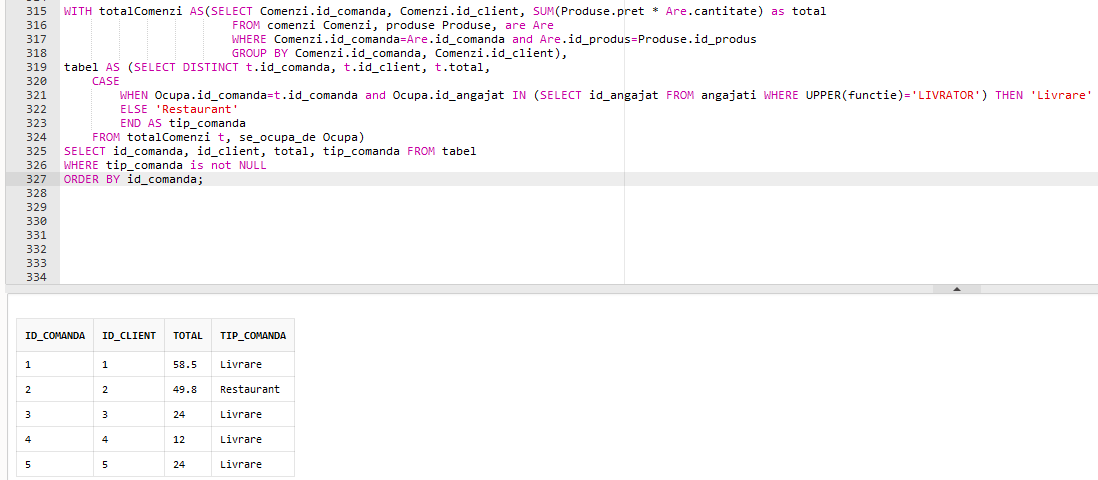
END AS tip\_comanda

FROM totalComenzi t, se\_ocupa\_de Ocupa)

SELECT id\_comanda, id\_client, total, tip\_comanda FROM tabel

WHERE tip\_comanda is not NULL

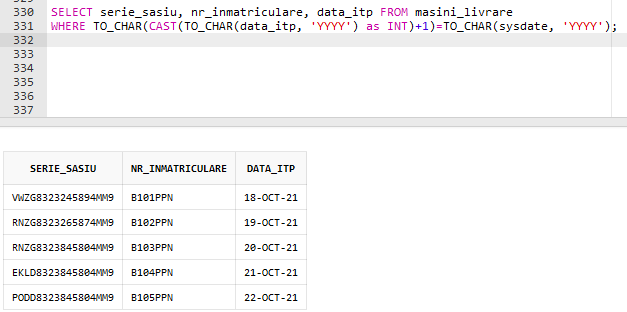
ORDER BY id\_comanda;



*5. Să se listeze toate mașinile cărora le expira ITP-ul in acest an*

SELECT serie\_sasiu, nr\_inmatriculare, data\_itp FROM masini\_livrare

WHERE TO\_CHAR(CAST(TO\_CHAR(data\_itp, 'YYYY') as INT)+1)=TO\_CHAR(sysdate, 'YYYY');

**

**12. Operații de actualizare și suprimare a datelor**

*1. Să se mărească prețul produselor ce conțin mozzarella cu 2 lei*

select \* from produse;

UPDATE produse SET pret=pret+2

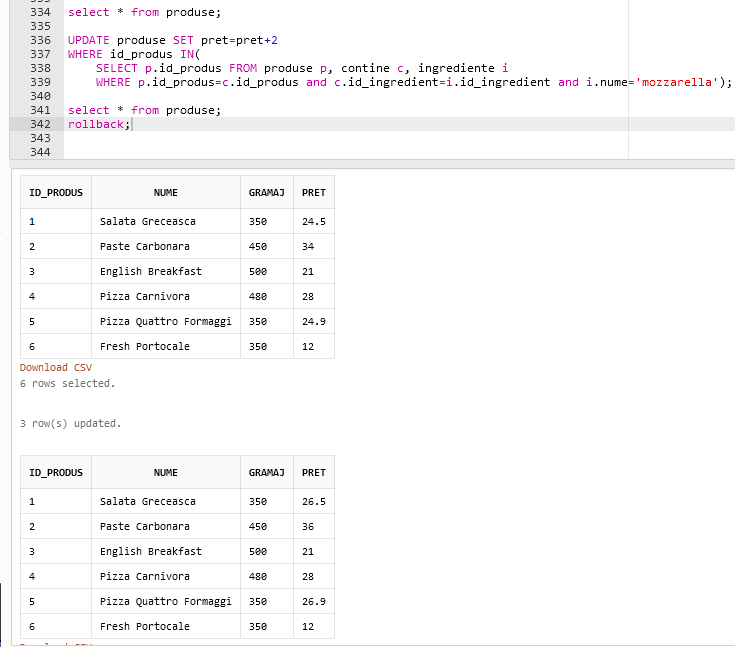
WHERE id\_produs IN(

SELECT p.id\_produs FROM produse p, contine c, ingrediente i

WHERE p.id\_produs=c.id\_produs and c.id\_ingredient=i.id\_ingredient and i.nume='mozzarella');

select \* from produse;

rollback;

**

*2. Să se adauge 100 de lei la salariu șoferului care a dus comanda 1.*

select \* from angajati;

UPDATE angajati SET salariu=salariu+100

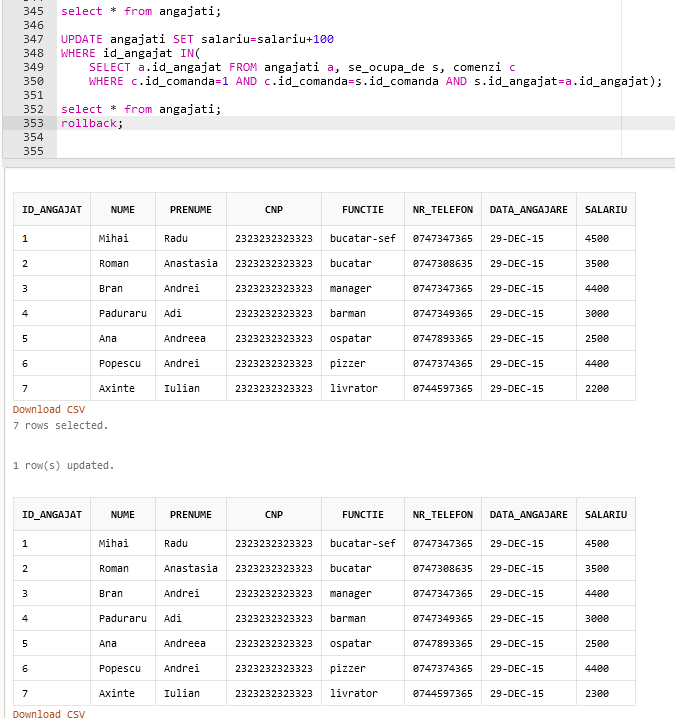
WHERE id\_angajat IN(

SELECT a.id\_angajat FROM angajati a, se\_ocupa\_de s, comenzi c

WHERE c.id\_comanda=1 AND c.id\_comanda=s.id\_comanda AND s.id\_angajat=a.id\_angajat);

select \* from angajati;

rollback;

**

*3. Să se ștearga produsul cu numele Salata Greceasca*

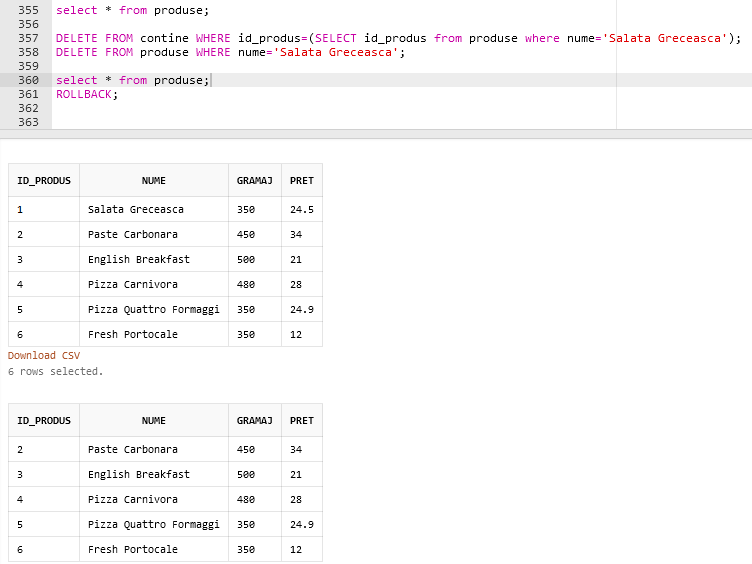
select \* from produse;

DELETE FROM contine WHERE id\_produs=(SELECT id\_produs from produse where nume='Salata Greceasca');

DELETE FROM produse WHERE nume='Salata Greceasca';

select \* from produse;

ROLLBACK;



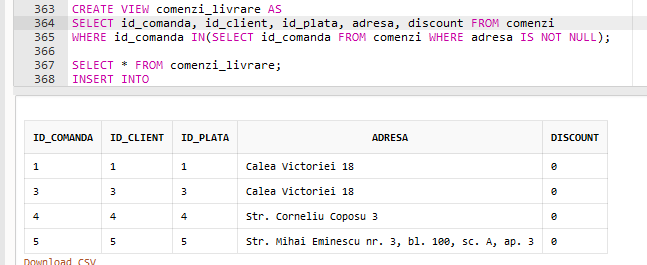
**14.** **Vizualizare compusă**

CREATE VIEW comenzi\_livrare AS

SELECT id\_comanda, id\_client, id\_plata, adresa, discount FROM comenzi

WHERE id\_comanda IN(SELECT id\_comanda FROM comenzi WHERE adresa IS NOT NULL);

SELECT \* FROM comenzi\_livrare;



Operația permisă pe un VIEW este cea de UPDATE, iar cea nepermisă este cea de DELETE, deoarece un VIEW poate avea functii GROUP în interior și dacă se dorește ștergerea uneia nu se știe exact care.

**16. Cerere ce utilizează operația de outer-join și cereri care utilizează operația division**

**Operația OUTER-JOIN:**

Să se listeze toate comenzile, clienții, plățile și angajații care s-au ocupat de acestea, pentru toate comenzile în care angajatul cu funcția de pizzer a participat la crearea produselor din comanda respectivă.

SELECT Comenzi.id\_comanda, Comenzi.adresa, Comenzi.discount, Comenzi.id\_client, Modalitati.id\_plata, Angajati.nume, Angajati.prenume, Angajati.id\_angajat FROM comenzi Comenzi

FULL OUTER JOIN clienti Clienti ON(Comenzi.id\_client=Clienti.id\_client)

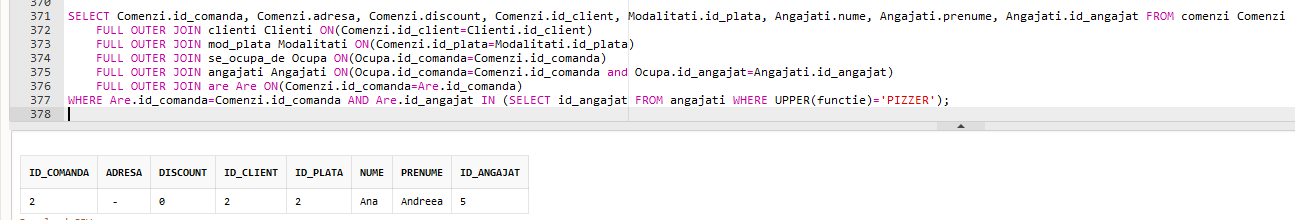
FULL OUTER JOIN mod\_plata Modalitati ON(Comenzi.id\_plata=Modalitati.id\_plata)

FULL OUTER JOIN se\_ocupa\_de Ocupa ON(Ocupa.id\_comanda=Comenzi.id\_comanda)

FULL OUTER JOIN angajati Angajati ON(Ocupa.id\_comanda=Comenzi.id\_comanda and Ocupa.id\_angajat=Angajati.id\_angajat)

FULL OUTER JOIN are Are ON(Comenzi.id\_comanda=Are.id\_comanda)

WHERE Are.id\_comanda=Comenzi.id\_comanda AND Are.id\_angajat IN (SELECT id\_angajat FROM angajati WHERE UPPER(functie)='PIZZER');



**Operația DIVISION:**

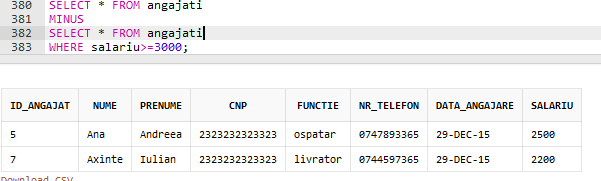
Să se listeze toți angajații care au salariul mai mic de 3000 de RON.

SELECT \* FROM angajati

MINUS

SELECT \* FROM angajati

WHERE salariu>=3000;



Să se listeze toate produsele ce nu conțin mozzarella.

SELECT p.id\_produs, p.nume, p.gramaj, p.pret FROM produse p, contine c, ingrediente i

GROUP BY p.id\_produs, p.nume, p.gramaj, p.pret

MINUS

SELECT p.id\_produs, p.nume, p.gramaj, p.pret FROM produse p, contine c, ingrediente i

WHERE p.id\_produs=c.id\_produs AND c.id\_ingredient=i.id\_ingredient and i.nume='mozzarella';

