Gestionarea datelor unei francize de restaurante

Proiect realizat de:

*Fugulin Victor*

*-Grupa 132-*

Cuprins

[**1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.** 2](#_Toc168086446)

[**2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.** 4](#_Toc168086447)

[**3. Descrierea entităților, incluzând precizarea cheii primare.** 4](#_Toc168086448)

[**4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.** 5](#_Toc168086449)

[**5.** **Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.** 6](#_Toc168086450)

[**6.** **Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.** 12](#_Toc168086451)

[**7.** **Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.** 13](#_Toc168086452)

[**8.** **Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.** 14](#_Toc168086453)

[**9.** **Realizarea normalizării până la forma normală 3 (FN1-FN3).** 14](#_Toc168086454)

[**10.** **Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).** 16](#_Toc168086455)

[create sequence secventa\_100 16](#_Toc168086456)

[**11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).** 16](#_Toc168086457)

[**12. Formularea în limbaj natural și implementarea a 5 cereri SQL complexe.** 36](#_Toc168086458)

[**13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.** 40](#_Toc168086459)

[**14.Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă** 41](#_Toc168086460)

# **1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Descrierea Contextului unei Baze de Date pentru o Franciză de Restaurante

O franciză de restaurante este o organizație complexă care operează sub un brand unic, dar cu multiple locații răspândite geografic. Fiecare locație funcționează autonom în ceea ce privește operațiunile zilnice, dar toate locațiile respectă aceleași standarde de calitate și servicii stabilite de francizor. Aceste restaurante pot varia ca mărime și specific, de la fast-food-uri la restaurante fine dining, dar toate împărtășesc un model de afaceri similar.

Modelul proiectat de mine gestionează atât problemele interne ale fiecărui restaurant al francizei, cum ar fi angajații, ierarhia și distribuirea acestora, meniul, locația și rezervarea meselor, cât și problemele externe care țin de transportul produselor, gestionarea comenzilor și a livrărilor.

Utilitatea Bazei de Date

Baza de date pentru o franciză de restaurante are rolul de a centraliza și organiza informațiile esențiale pentru operarea eficientă a fiecărei locații și pentru managementul general al francizei. Scopul principal este de a asigura accesul rapid și precis la date, facilitând deciziile informate la nivel de management, precum și operațiunile zilnice ale fiecărei locații.

Centralizarea datelor într-o bază de date unică permite o gestionare mai eficientă a francizei. Aceasta facilitează accesul la informații, reduce redundanța și riscul de erori și asigură consistența datelor în toate locațiile. De asemenea, permite generarea de rapoarte complexe și analize detaliate, esențiale pentru luarea deciziilor strategice și operaționale.

Reguli de functionare

1. Fiecare restaurant are un proprietar care poate schimba modul de funcționare a restaurantului atâta timp cât rămâne în standardele impuse de francizor. Un proprietar poate să dețină mai multe restaurante pe care le poate modifica diferit pe fiecare dintre ele, acestea diferențiindu-se prin ID-ul lor.
2. Alegând un restaurant, utilizatorul poate să afle toate detaliile acestuia și să vadă modificările aduse de către proprietar, cum ar fi locația restaurantului, meniul, personalul angajat și profitul.
3. Dacă un client vrea să facă o rezervare pentru o anumită zi, la o anumită oră, la un anumit restaurant al francizei, se va verifica dacă este disponibilă această rezervare consultând tabelul destinat rezervărilor fiecărui restaurant. Dacă nu există altă rezervare în aceeași zi, la aceeași oră, la același restaurant, atunci cererea clientului va fi îndeplinită și rezervarea va fi introdusă în tabel.
4. Pentru livrări, clientul vizualizează meniul restaurantului și plasează o comandă care este preluată de un șofer, angajat al francizei, și transportată la adresa dată de client.

5. Modelul are două tipuri de angajați, cei care lucrează la restaurant și cei care se ocupă cu transportul. Al doilea tip nu este legat de niciun restaurant. Atunci când într-un restaurant este plasată o comandă, cel mai apropiat șofer disponibil este chemat să ridice comanda și să o livreze clientului.

# **2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.**

1. Un proprietar poate să dețină mai multe restaurante, dar un restaurant nu poate să aibă mai mulți proprietari. De asemenea, un restaurant nu trebuie să aibă neapărat un proprietar; în acest caz, restaurantul va fi condus de un manager general angajat de franciză sau restaurantul ramane gol până când va fi cumpărat.

2. Fiecare restaurant **trebuie** să aibă un singur meniu. Acesta poate fi modificat în cazul în care doresc să fie introduse produse speciale (produse de sezon, de exemplu), dar nu pot să existe mai multe versiuni ale meniului simultan in cadrul aceluiasi restaurant.

3. Un angajat nu poate să aibă mai multe joburi. Nu există angajați care lucrează într-un restaurant și care lucrează în transport sau un angajat care sa fie casier si bucatar in acelasi timp.

4. Nu pot să existe mai multe restaurante în aceeași locație; acest lucru este făcut pentru a face serviciile francizei cât mai disponibile pentru cât mai mulți oameni și pentru a mări popularitatea acesteia

5. Un angajat al unui restaurant nu poate sa lucreze la mai multe restaurante in acelasi timp si **trebuie** sa lucreze intr-un restaurant. Restrictia nu se aplica si angajatilor care lucreaza in transport, acestia lucreaza pentru franciza, nu pentru un restaurant anume.

6. Atunci cand un client vrea sa comande, acesta este redirectionat direct catre meniul restaurantului cel mai apropiat de adresa lui curenta, astfel acesta nu poate sa vizualizeze mai multe meniuri.

# **3. Descrierea entităților, incluzând precizarea cheii primare.**

1. **RESTAURANT** reprezintă fiecare unitate de afaceri a francizei de restaurante. Fiecare restaurant are un ID unic, nume, o locație specifică și poate fi deținut de un proprietar sau condus de un manager general angajat de franciză. Restaurantul are asociat un meniu specific, are un tip de restaurant care este oferit de franciza si un telefon fix pentru a primi comenzi de la clienti. **Cheia primara este cod\_restaurant#.**

2. **PROPRIETAR** reprezintă persoana sau entitatea care deține unul sau mai multe restaurante în cadrul francizei. Proprietarul are un nume, adresa de e-mail și număr de telefon asociat și este responsabil pentru gestionarea și conducerea restaurantelor deținute. **Cheia primara este cod\_proprietar#.**

3. **REZERVARE** înregistrează detaliile fiecărei rezervări făcute de către clienți pentru un anumit restaurant din cadrul francizei. O rezervare este definită de data la care a fost făcută, restaurantul pentru care este făcută rezervarea și detaliile clientului care a făcut rezervarea. **Cheia primara este cod\_rezervare#.**

4. **LOCATIE** definește fiecare loc specific în care se află un restaurant al francizei. O locație este caracterizată prin strada si numarul strazii în care se află și poate fi asociata cu un restaurant. **Cheia primara este cod\_locatie#.**

5. **MENIU** definește lista de produse și prețurile acestora oferite de fiecare restaurant din franciză. Meniul este definit de codul acestuia, de o scurta descriere, de pretul minim si maxim al produselor continute in meniu. **Cheia primara este cod\_meniu#.**

6. **CLIENT** reprezintă persoanele care plasează comenzi sau fac rezervări în restaurantele francizei. Fiecare client are un profil care include nume, prenume, număr de telefon și adresă de e-mail. **Cheia primara este cod\_client#.**

7. **COMANDA** înregistrează detaliile fiecărei comenzi plasate de către clienți pentru livrare. O comandă este definită de codul acesteia, de pret si de numarul de produse comandate. **Cheia primara este cod\_comanda#.**

8. **ADRESA\_LIVRARE** conține informații despre adresele la care sunt livrate comenzile plasate de clienți pentru serviciul de livrare. Fiecare adresă de livrare este identificată printr-un ID unic și include strada si numarul strazii. **Cheia primara este cod\_adresa\_livrare#.**

9. **ANGAJAT\_LIVRARE** reprezintă persoanele angajate de franciză pentru a efectua livrările la domiciliu pentru comenzile plasate de clienț, cat si transportul de marfa de la depozit la restaurant. Fiecare angajat de livrare are un profil care include nume, prenume, număr de telefon și adresă de e-mail si vehicul. **Cheia primara este cod\_angajat\_livrare#.**

10. **ANGAJAT\_RESTAURANT** reprezintă personalul angajat de proprietar sau de franciza pentru a lucra în restaurantele din cadrul francizei. Fiecare angajat are un salariu incadrat in salariul minim si maxim oferit de jobul sau și trebuie sa fie asociat cu un anumit restaurant. **Cheia primara este cod\_angajat\_restaurant#.**

11. **JOB** definește rolurile angajaților în cadrul restaurantelor francizei. **Cheia primara este cod\_job#.**

12. **ISTORIC\_JOB** înregistrează istoricul angajaților din cadrul francizei și schimbările de pe care le-a avut un anumit job. **Cheia primara este cod\_istoric\_job#.**

# **4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

1. RESTAURANT\_are\_MENIU = “are” este relatia care leaga entitatile RESTAURANT si MENIU. Fiecare restaurant poate avea un singur meniu asociat si trebuie sa aiba un meniu. Fiecare meniu poate sa aiba un singur restaurant asociat lui si trebuie sa aiba un restaurant asociat. Cardinalitatea minima este 1:1, iar cardinalitatea maxima este tot 1:1.

2. CLIENT\_vizualizeaza\_MENIU = “vizualizeaza” este relatia care leaga entitatile CLIENT si MENIU. Un client poate sa vizualizeze un singur meniu (restrictia nr. 7) si nu trebuie sa vizualizeze niciun meniu. Un meniu poate sa fie vizualizat de mai multi clienti si nu trebuie sa fie vizualizat de niciunul. Cardinalitatea minima este 0:0, iar cardinalitatea maxima este M:1.

3. CLIENT\_plaseaza\_COMANDA = “plaseaza” este relatia care leaga entitatile CLIENT si COMANDA. Un client poate sa plaseze mai multe comenzi si nu trebuie sa plaseze nicio comanda. O comanda poate fi plasata de un singur client si trebuie sa fie plasata de un client. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:M.

4. ANGAJAT\_RESTAURANT\_lucreaza\_RESTAURANT = “lucreaza” este relatia care leaga entitatile RESTAURANT si ANGAJAT\_RESTAURANT. Un angajat poate sa lucreze la un singur restaurant si trebuie sa lucreze la un restaurant. Un restaurant poate sa aiba mai mult angajati si nu trebuie sa aiba niciun angajat. Cardinalitatea minima este 0:1 si cardinalitatea maxima este M:1.

5. ANGAJAT\_RESTAURANT\_ocupa\_JOB = “ocupa” este relatia care leaga entitatile ANGAJAT\_RESTAURANT si JOB. Un angajat poate sa ocupe un singur job si trebuie sa ocupe un job. Un job poate as fie ocupat de mai multi angajati si un job nu trebuie sa fie ocupat de niciun angajat. Cardinalitatea minima este 0:1 si cardinalitatea maxima este M:1.

6. ANGAJAT\_LIVRARI\_ocupa\_JOB = “ocupa” este relatia care leaga entitatile ANGAJAT\_LIVRARI si JOB. Un angajat de la livrari poate sa ocupe un singur job si trebuie sa ocupe un job. Un job poate sa fie ocupat de mai multi angajati de la livrari si nu trebuie sa fie ocupat de niciun angajat. Cardinalitatea minima este 0:1 si cardinalitatea maxima este M:1.

7. PROPRIETAR\_detine\_RESTAURANT = “detine” este relatia care leaga entitatile PROPRIETAR si RESTAURANT. Un proprietar poate sa detina mai multe restaurante si trebuie sa detina unul. Un restaurant poate sa fie detinut de un singur proprietar si nu trebuie sa fie detinut de niciunul. Cardinalitatea minima este 0:1 si cardinaliteatea maxima este 1:M.

8. RESTAURANT\_se\_alfa\_in\_LOCATIE = “se afla” este relatia care leaga entitatile RESTAURANT si LOCATIE. Un restaurant se poate afla intr-o singura locatie si trebuie sa se afle intr-o locatie. Intr-o locatie se poate afla un singur restaurant si nu trebuie sa se afle niciunul. Cardinaliteatea minima este 0:1 si cardinalitetea maxima este 1:1.

9. JOB\_asociat\_JOB\_ISTORIC = “asociat” este relatia care leaga entitatile JOB si JOB\_ISTORIC. Un job poate fi asociat cu mai multe inregistrati din job\_istoric si nu trebuie sa fie asociat cu niciuna. O inregistrare din job\_istoric poate sa fie asociata cu un un singur job si trebuie sa fie asociata cu un job. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:M.

10. ANGAJAT\_LIVRARI\_livreaza\_COMANDA\_la\_ADRESA\_LIVRARE = “livreaza\_la” este relatia de tip trei care leaga entitatile ANGAJAT\_LIVRARI, COMANDA SI ADRESA\_LIVRARE. Un angajat livreaza o comanda sau mai multe comenzi la o adresa.

11. CLIENT\_face\_REZERVARE = “face” este relatia care leaga entitatile CLIENT SI REZERVARE. Un client poate sa faca o singura rezervare si nu trebuie sa faca niciuna. O rezervare poate sa fie facuta de un singur client si trebuie sa fie facuta de cel putin unul. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:1.

12. RESTAURANT\_programata\_REZERVARE = “programata” este relatia care leaga entitatile RESTAURANT SI REZERVARE. Un restaurant poate sa aiba mai multe rezervari programate si nu trebuie sa aiba niciuna programata. O rezervare poate sa fie programata la un singur restaurant si trebuie sa fie progamata la un restaurant. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:M.

# **Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.**

1. Atributele entitatii RESTAURANT:
   1. **Cod\_restaurant**: variabila de tip intreg, de lungime 3, care reprezinta codul restaurantului, are valoare unica si obligatorie si este cheia primara a entitatii.(Valori posibile: 100, 101, 102...)
   2. **Nr\_telefon**: variabila de tip intreg, de lungime maxim 12, care reprezinta numarul de telefon al telefonului fix aflat in acel restaurant, are valoare unica si obligatorie.(Valori posibile: 40787645002, 3801012376…)
   3. **Vanzari\_saptamanale**: variabila de tip intreg, de lungime maxim 6, care reprezinta venitul saptamanal al restaurantului in euro. Valoare implicita 0. (Valori posibile: 20.000, 40.000…)
   4. **nume\_restaurant**: este o variabila de tip caracter, de lungime maxim 50, care reprezinta numele restaurantului, de obicei numele francizei + ceva reprezentativ locului; are valoare unica si obligatorie.

(Valori posibile: Franciza Topoloveni, Franciza Bradu…)

* 1. **Cod\_proprietar:** variabila de tip intreg, de lungime 2, care reprezinta codul proprietarului care detine restaurantul.
  2. **Tip\_restaurant:** variabila de tip caracter, de lungime maxima 50, care reprezinta tipul restaurantului, de obicei, dimensiunea acestuia. Are valoare obligatorie. (Valori posibile: Fast-Food mare, Fine-dinning mic…)

1. Atributele entitatii PROPRIETAR:
   1. **Cod\_proprietar**: variabila de tip intreg, de lungime 2, care reprezinta codul proprietarului, are valoare unica si obligatorie, fiind cheia primara a entitatii. (Valori posibile: 10, 11, 12…).
   2. **Nume**: variabila de tip caracter, de lungime maxim 20, care reprezinta numele proprietarului. Are valoare obligatorie. (Valori posibile: Fugulin, Pojoga, Popescu…).
   3. **Prenume**: variabila de tip intreg, de lungime maxim 20, care reprezinta prenumele proprietarului. Are valoare obligatorie. (Valori posibile: Victor, Mihai-Sebastian, Alex…).
   4. **Nr\_telefon**: variabila de tip intreg, de lungime maxim 12, care reprezinta numarul de telefon al proprietarului, are valoare unica si obligatorie.(Valori posibile: 40787645002, 3801012376…)
   5. **Email**: variabila de tip caracter, de lungime maxim 50, care reprezinta adresa de email al proprietarului, are valoare unica si obligatorie. (Valori posibile: [victor.fugulin@yahoo.com](mailto:victor.fugulin@yahoo.com)).
   6. **Profit**: variabila de tip numar, de lungime maxim 7, care reprezinta profitul pe care l-a acumulat proprietarul care detine restaurante ale francizei. Are valoare implicita de 0.
2. Atributele entitatii LOCATIE:
   1. **Cod\_locatie:** variabila de tip intreg, de lungime 4, care reprezinta codul locatiei. Are valoare obligatorie si unica, fiind cheia primara a entitatii (Valori posibile: 1000, 1001).
   2. **Strada:** variabila de tip caracter, de lungime maxim 30, care reprezinta numele strazii unei anumite locatii. Are valoare obligatorie. (Valori posibile: Calea Bucuresti, Mihai Eminescu…)
   3. **Nr\_strada:** variabila de tip intreg, de lungime maxim 5, care reprezinta numarul strazii unei anumite locatii. Are valoare obligatorie. (Valori posibile: 125, 1…).
   4. **Cod\_restaurant**: variabila de tip intreg, de lungime 3, care reprezinta codul restaurantului, are valoare unica, deoarece doua restaurante nu se pot afla in aceeasi locatie. (Valori posibile: 100, 101, 102...).
3. Atributele entitatii ANGAJAT\_RES:
   1. **Cod\_angajat:** variabila de tip intreg, de lungime 3, care reprezinta codul angajatului. Are valoare unica si obligatorie, fiind cheia primara a entitatii (Valori posibile: 100, 101, 102).
   2. **Nume**: variabila de tip caracter, de lungime maxim 20, care reprezinta numele angajatului. Are valoare obligatorie. (Valori posibile: Fugulin, Pojoga, Popescu…).
   3. **Prenume**: variabila de tip caracter, de lungime maxim 20, care reprezinta prenumele angajatului. Are valoare obligatorie. (Valori posibile: Fugulin, Pojoga, Popescu…).
   4. **Cod\_job:** variabila de tip intreg, de lungime maxim 3, care reprezinta codul jobului pe care il ocupa angajatul. Are valoare obligatorie (Valori posibile: 100, 140, 123…).
   5. **Salariu:** variabila de tip intreg, de lungime maxim 5, care reprezinta salariul lunar in euro al angajatului. Are valoare obligatorie. (Valori posibile: 2000, 3000, 5000…).
   6. **Data\_inceput:** variabila de tip data calenderistica, care reprezinta data in care a fost angajat persoana. Are valoare obligatorie (Valori posibile: 13-AUG-1988).
   7. **Email**: variabila de tip caracter, de lungime maxim 50, care reprezinta adresa de email al angajatului, are valoare unica si obligatorie. (Valori posibile: [victor.fugulin@yahoo.com](mailto:victor.fugulin@yahoo.com)).
   8. **Nr\_telefon**: variabila de tip intreg, de lungime maxim 12, care reprezinta numarul de telefon al angajatului, are valoare unica si obligatorie (Valori posibile: 40787645002, 3801012376…)
   9. **Cod\_restaurant**: variabila de tip intreg, de lungime 3, care reprezinta codul restaurantului, are valoare obligatorie, deoarece un angajat trebuie sa lucreze intr-un restaurant. (Valori posibile: 100, 101, 102...).
   10. **Cod\_manager:** variabila de tip intreg, de lungime 3, care reprezinta codul managerului direct al angajatului. (Valori posibile: 100, 101, 102).
4. Atributele entitatii JOB:
   1. **Cod\_job:** variabila de tip intreg, de lungime 3, care reprezinta codul jobului pe care il ocupa angajatul. Are valoare unica si obligatorie, fiind cheia primara a entitatii (Valori posibile: 100, 140, 123…).
   2. **Denumire\_job:** variabila de tip caracter, de lungime maxim 30, care reprezinta numele jobului, are valoare unica si obligatorie (Valori posibile: Bucatar, Casier, Manager\_general, Sofer\_livrari\_marfa, Sofer\_livrari\_comanda, Bucatar\_sef, Manager\_livrari).
   3. **Salariu\_minim:** variabila de tip intreg, de lungime maxima 5, care reprezinta salariul minim pe care il poate primi un angajat care ocupa acest job. Are valoare obligatorie (Valori posibile: 2000, 3000…).
   4. **Salariu\_maxim:** variabila de tip intreg, de lungime maxima 5, care reprezinta salariul maxim pe care il poate primi un angajat care ocupa acest job. Are valoare obligatorie (Valori posibile: 10000, 20000...).
5. Atributele entitatii ISTORIC\_JOB:
   1. **Cod\_istoric:** variabila de tip intreg, de lungime 3, care reprezinta codul istoricului. Are valoare unica si obligatorie, fiind cheia primara a entitatii (Valori posibile: 100, 132…).
   2. **Data\_inceput:** variabila de tip data calenderistica, care reprezinta data de inceput al jobului din trecut, are valoare obligatorie (Valori posibile: 12-MAI-1975…).
   3. **Data\_final:** variabila de tip data calenderistica, care reprezinta data in care s-a sfarsit contractul acelui job, are valoare obligatorie (Valori posibile: 25-IAN-2003).
   4. **Nume:** variabila de tip intreg, care reprezinta numele angajatului care a lucrat la acest job in trecut, are valoare obligatorie (Valori posibile: Popescu…).
   5. **Prenume:** variabila de tip intreg, care reprezinta prenumele angajatului care a lucrat la acest job in trecut, are valoare obligatorie (Valori posibile: Alex…).
   6. **Cod\_job:** variabila de tip intreg, de lungime 3, care reprezinta codul jobului pentru care a fost intocmit istoricul. Are valoare obligatorie (Valori posibile: 100, 140, 123…).
   7. **Nume\_restaurant**: variabila de tip caracter, de lungime maxima 30, care reprezinta numele restaurantului la care a lucrat angajatul in trecut. Are valoare oblogatorie. (Valori posibile: Franciza Bradu…)
6. Atributele entitatii ANGAJAT\_LIVRARI:
   1. **Cod\_ang\_liv:** variabila de tip intreg, de lungime 3, care reprezinta codul angajatului care se ocupa cu livrarile. Are valoare obligatorie si unica, fiind cheia primara a entitatii(Valori posibile: 200, 201, 202…).
   2. **Nume**: variabila de tip caracter, de lungime maxim 20, care reprezinta numele angajatului care se ocupa cu livrarile. Are valoare obligatorie. (Valori posibile: Fugulin, Pojoga, Popescu…).
   3. **Prenume**: variabila de tip caracter, de lungime maxim 20, care reprezinta prenumele angajatului care se ocupa cu livrarile. Are valoare obligatorie. (Valori posibile: Victor, Alex Andrei…).
   4. **Salariu:** variabila de tip intreg, de lungime maxim 5, care reprezinta salariul lunar in euro al angajatului care se ocupa cu livrarile. Are valoare obligatorie. (Valori posibile: 2000, 3000, 5000…).
   5. **Vehicul:** variabila de tip caracter, de lungime maxim 20, care reprezinta vehiculul pe care il conduce angajatul pentru a livra marfa sau comenzi. (Valoare posibila: MASINA, BICICLETA, MOTOCICLETA…).
   6. **Nr\_telefon**: variabila de tip intreg, de lungime maxim 12, care reprezinta numarul de telefon al soferului, are valoare obligatorie (Valori posibile: 40787645002, 3801012376…)
   7. **Email**: variabila de tip caracter, de lungime maxim 50, care reprezinta adresa de email al soferului, are valoare obligatorie. (Valori posibile: [victor.fugulin@yahoo.com](mailto:victor.fugulin@yahoo.com)).
   8. **Cod\_job:** variabila de tip intreg, de lungime maxim 3, care reprezinta codul jobului pe care il ocupa angajatul care se ocupa cu livrarile. Are valoare obligatorie (Valori posibile: 100, 140, 123…).
   9. **Cod\_manager:** variabila de tip intreg, de lungime 3, care reprezinta codul managerului direct al angajatului care se ocupa cu livrarile, are valoare unica, deoarece un angajat poate sa aiba un singur manager direct (Valori posibile: 100, 101, 102).
7. Atributele entitatii MENIU:
   1. **Cod\_meniu:** variabila de tip intreg, de lungime 3, care reprezinta codul meniului. Are valoare obliatorie si unica, fiind cheia primara a entitatii (Valori posibile: 100, 120, 321).
   2. **Descriere\_meniu:** variabila de tip caracter, de lungime maxim 50, care reprezinta o scurta descriere a meniului (Valori posibile: Editie Speciala de Caraciun).
   3. **numar\_pagini:** variabila de tip intreg, de lungime maxim 2, care reprezinta numarul de pagini pe care il are meniul. Are valoare obligatorie (Valori posibile: 10, 32, 7…).
   4. **Pret\_produs\_minim:** variabila de tip intreg, de lungime maxim 2, care reprezinta pretul minim, in euro, al produselor continute in meniu. Are valoare obliatorie (Valori posibile: 10, 15, 8…)
   5. **Pret\_produs\_minim:** variabila de tip intreg, de lungime maxim 3, care reprezinta pretul maxim, in euro, al produselor continute in meniu. Are valoare obliatorie (Valori posibile: 70, 100, 85…)
   6. **Cod\_restaurant:** variabila de tip intreg, de lungime 3, care reprezinta codul restaurantului caruia ii este asociat meniul, are valoare obligatorie si unica (Valori posibile: 100, 101, 102…).
8. Atributele entiatii CLIENT:
   1. **Cod\_client:** variabila de tip intreg, de lungime 5, care reprezinta codul clientului care poate sa vizualizeze un meniu sau care poate face o rezervare. Are valoare obligatorie si unica, fiind cheia primara a entitatii (Valori posibile: 10000, 10101, 20312…).
   2. **Nume**: variabila de tip caracter, de lungime maxim 20, care reprezinta numele clientului. Are valoare obligatorie. (Valori posibile: Fugulin, Pojoga, Popescu…).
   3. **Preume**: variabila de tip caracter, de lungime maxim 20, care reprezinta prenumele clientului. Are valoare obligatorie. (Valori posibile: Victor, Alex, Sebastian…).
   4. **Telefon\_client**: variabila de tip intreg de lungime maxim 12, care reprezinta numarul de telefon al clientului. Are valoare obligatorie (Valori posibile: 40787645044).
   5. **Cod\_meniu:** variabila de tip intreg, de lungime 3, care reprezinta codul meniului pe care il vizualizeaza clientul. (Valori posibile: 100, 120, 321).
9. Atributele entitatii REZERVARE:
   1. **Cod\_rezervare:** variabila de tip intreg, de lungime 4, care reprezinta codul rezervarii. Are valoare obligatorie si unica, fiind cheia primara a entitatii (Valori posibile: 1000, 1212, 3210…).
   2. **Cod\_restaurant:** variabila de tip intreg, de lungime 3, care reprezinta codul restaurantului la care este programata rezervarea, are valoare obligatorie si unica (Valori posibile: 100, 101, 102…).
   3. **Zi\_rezervare:** este variabila de tip data calendaristica, care reprezinta data la care a fost facuta rezervarea. Are valoare obligatorie (Valori posibile: 28-IAN-2025).
   4. **Nume\_client:** variabila de tip caracter, de lungime maxim 30, care reprezinta numele clientului care a facut rezevarea. Are valoare obligatorie (Valori posibile: Alex, Victor, Denisa-Ioana…).
   5. **Nr\_telefon:** variabila de tip intreg, de lungime maxim 12, care reprezinta numarul de telefon al clientului care a facut rezervarea. Are valoare obligatorie (Valori posibile: 40787645044).
   6. **Cod\_client:** variabila de tip intreg, de lungime 5, care reprezinta codul clientului care, a facut rezervarea. Are valoare obligatorie (Valori posibile, 10101, 10000, 30312…).
10. Atributele entitatii COMANDA:
    1. **Cod\_comanda:** variabila de tip intreg, de lungime 5, care reprezinta codul comenzii. Are valoare obligatorie si unica, fiind cheia primara a entitatii. (Valori posibile: 10101, 22222…).
    2. **Pret**: variabila de tip intreg, de lungime maxim 4, care reprezinta pretul pe care il are de platit clientul pentru comanda. Are valoare obligatorie (Valori posibile: 100, 75, 50…).
    3. **Nr\_produse**: variabila de tip intreg, de lungime maxim 2, care reprezina numarul de produse comandate de client. Are valoare obligatorie (Valori posibile: 5, 10, 20…).
    4. **Cod\_client:** variabila de tip intreg, de lungime 5, care reprezinta codul clientului care, a plasat comanda. Are valoare obligatorie (Valori posibile, 10101, 10000, 30312…).
11. Atributele entitatii ADRESA\_LIVRARE:
    1. **Cod\_adresa:** variabila de tip intreg, de lungime 5, care reprezinta codul adresei unde trebuie sa fie livrata comanda. Are valoare obligatorie si unica, fiind cheia primara a entitatii (Valori posibile: 10000, 10202…).
    2. **Strada:** variabila de tip caracter, de lungime maxim 20, care reprezinta numele strazii adresei unde trebuie sa fie livrata comanda. Are valoare obligatorie. (Valori posibile: Calea Bucuresti, Mihai Eminescu…)
    3. **Nr\_strada:** variabila de tip intreg, de lungime maxim 5, care reprezinta numarul strazii adresei unde trebuie sa fie livrata comanda. Are valoare obligatorie. (Valori posibile: 125, 1…).

# **Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.**

O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat

Diagrama realizata folosing site-ul: draw.io

Nota autor: diagramaER nu are toate atributele. Modelul a suferit niste modificari dupa crearea diagramei si din pacate nu am mai gasit folderul cu aceasta diagrama pentru a o modifica. Toate atributele se pot gasi totusi la punctul 8 al acestui proiect. De asemenea, au fost eliminate anumite atribute din modelul initial care inca se regasesc in diagrama.

# **Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.**

O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat

Diagrama realizata folosing site-ul: draw.io

# **Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.**

PROPRIETAR(cod\_proprietar#, nume, prenume, profit, nr\_telefon, email).

RESTAURANT(cod\_restaurant#, nume\_restaurant, tip\_restaurant, vanzari\_saptamanale, nr\_telefon, cod\_proprietar).

LOCATIE(cod\_locatie#, strada, nr\_strada, cod\_restaurant).

ANGAJAT\_RES(cod\_angajat#, nume, prenume, salariu, data\_inceput, email, nr\_telefon, cod\_restaurant, cod\_job, cod\_manager).

ANGAJAT\_LIVRARI(cod\_ang\_liv#, nume, prenume, salariu, vehicul, nr\_telefon, email, cod\_job, cod\_manager).

JOB(cod\_job#, denumire\_job, salariu\_minim, salariu\_maxim).

ISTORIC\_JOB(cod\_istoric#, data\_inceput, data\_final, nume, prenume, cod\_job, cod\_restaurant) .

MENIU(cod\_meniu#, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant).

CLIENT(cod\_client#, nume, prenume, telefon\_client, cod\_meniu).

REZERVARE(cod\_rezervare#, cod\_restaurant, zi\_rezervare, nr\_telefon, cod\_client).

COMANDA(cod\_comanda#, pret, nr\_produse, cod\_client).

ADRESA\_LIVRARE(cod\_adresa#, strada, nr\_strada).

LIVRARE(cod\_comanda#, cod\_adresa#, cod\_ang\_liv#, durata\_estimativa).

# **Realizarea normalizării până la forma normală 3 (FN1-FN3).**

**FN1**

Modelul este normalizat in forma normala 1, deoarece fiecarui atribut al modelului ii corespunde o valoare atomica.

Un exemplu de **non-FN1** ar fi daca entitatea ANGAJAT\_LIVRARI ar avea il loc de atributul “vehicul”, atributul “vehicule” si atunci ar aparea valori multiple. De exemplu, angajatul cu codul 100 ar putea sa aiba trei vehicule cu care livreaza, atunci atributul “vehicule” ar fi compus din 3 valori (vehicule: DACIA, MERCEDES, BMW) si atunci modelul nu ar mai fi in FN1. Ca sa il transformam in FN1, atributul “vehicule” devine “vehicul”, iar in loc de o linie in entitatea ANGAJAT\_LIVRARI, o sa folosim 3 linii pentru angajatul cu codul 100, o linie pentru fiecare vehicul cu care livreaza.

**FN2**

Modelul este normalizat in forma normala 2, deoarece este in forma normala 1 si modelul meu nu are chei primare compuse.

Un exemplu de non-FN2 ar fi daca in entitatea LOCATIE cheia primara ar fi fost compusa din: strada, cod\_postal. In acest caz, atributul “oras” nu depinde de toata cheia primara, ci doar partial, adica doar de cod\_postal, deoarece strazi cu aceelasi nume exista in orase diferite. Pentru a transforma in FN2, eliminam atributul “oras” din entitatea LOCATIE si aplicand Casey-Delobel, faceam o noua entitate a carei cheie primara ar fi fost “cod\_postal” si in aceasta noua entitate puteam pune atributul “oras”.

**FN3**

Modelul este normalizat in forma normala 3, deoarece este in forma normala 2 si toate atributele, care nu fac parte din cheia primara, depind direct de aceasta. Adica, nu exista dependente tranzitive de la cheia primara la un alt atribut.

Un exemplu de non\_FN3 ar fi daca in entitatea RESTAURANT aveam atributul “nr\_locuri” care reprezenta numarul de clienti care ar putea sa ocupe un loc simultan la restaurant. Acest atribut depinde direct de atributul “tip\_restaruant”, care reprezinta tipul si implicit, dimensiunea, respectivului restaurant, care la randul sau depinde de cheia primara. Astfel atributul “nr\_locuri” depinde tranzitiv de cheia primara, prin intermediul atributului “tip\_restaurant”, ceea ce face ca modelul sa nu fie in FN3. Pentru a transforma in FN-3, eliminam atributul “nr\_locuri” din entitatea RESTAURANT si aplicand Casey-Delobel am mai face o noua entitate a carei cheie primara o sa fie atributul “tip\_restaurant”, iar in aceasta entitate putem pune atributul “nr\_locuri”.

# **Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).**

# create sequence secventa\_100

start with 100

increment by 1

maxvalue 999;

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

# **11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).**

**1.Proprietar**

create table PROPRIETAR(

cod\_proprietar number(2) constraint pk\_cod\_proprietar primary key,

nume varchar2(20) not null,

prenume varchar2(20) not null,

nr\_telefon number(12) not null constraint telefon\_unic unique,

email varchar2(50) not null constraint email\_unic unique,

profit number(7) default 0

);

insert into PROPRIETAR(cod\_proprietar, nume, prenume, nr\_telefon, email, profit) values (10, 'Victor', 'Fugulin', 787645044, 'victor.fugulin@gmail.com', 35000);

insert into PROPRIETAR(cod\_proprietar, nume, prenume, nr\_telefon, email, profit) values (11, 'Jones', 'Mary', 5551234567, 'mary.jones@gmail.com', 27500);

insert into PROPRIETAR(cod\_proprietar, nume, prenume, nr\_telefon, email, profit) values (12, 'Patel', 'Ahmed', 9998887777, 'ahmed.patel@gmail.com', 18500);

insert into PROPRIETAR(cod\_proprietar, nume, prenume, nr\_telefon, email, profit) values (13, 'Petrova', 'Elena', 7775551234, 'elena.petrova@gmail.com', 16200);

insert into PROPRIETAR(cod\_proprietar, nume, prenume, nr\_telefon, email, profit) values (14, 'Martinez', 'Juan', 1234567890, 'juan.martinez@gmail.com', 25800);

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

**2. Restaurant**

create table RESTAURANT(

cod\_restaurant number(3) constraint pk\_cod\_restaurant primary key,

nume\_restaurant varchar2(50) not null constraint nume\_unic unique,

vanzari\_saptamanale number(6) default 0,

nr\_telefon number(12) not null constraint telefon\_fix\_unic unique,

tip\_restaurant varchar2(50) not null,

cod\_proprietar number(2),

constraint fk\_proprietar foreign key(cod\_proprietar) references proprietar(cod\_proprietar) on delete cascade

);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Timisoara', 9800, 0255557777, 'Mic', 14);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Brasov', 10200, 0266665555, 'Mediu', 14);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Iasi', 9400, 0277776666, 'Mic', 13);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Constanta', 8900, 0299994444, 'Mediu', 12);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Bucuresti', 12700, 0287883331, 'Mare', 11);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Topoloveni', 6800, 0787645044, 'Mic', 10);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Baneasa', 9230, 078764321, 'Mic', 14);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant, cod\_proprietar) values (secventa\_100.nextval, 'Franciza Afi', 3800, 073264321, 'Mediu', 10);

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant) values (secventa\_100.nextval, 'Franciza Vivo Mall', 5200, 0787324321, 'Mare');

insert into RESTAURANT(cod\_restaurant, nume\_restaurant, vanzari\_saptamanale, nr\_telefon, tip\_restaurant) values (secventa\_100.nextval, 'Franciza Plaza', 7000, 0725134321, 'Mediu');

**O imagine care conține text, captură de ecran, software, Pictogramă computer

Descriere generată automat**

**3. Locatie**

create table LOCATIE(

cod\_locatie number(4) constraint pk\_cod\_locatie primary key,

strada varchar2(30) not null,

nr\_strada number(5)not null,

cod\_restaurant number(3) constraint fk\_restaurant\_unic unique,

constraint fk\_cod\_restaurant foreign key(cod\_restaurant) references restaurant(cod\_restaurant) on delete cascade

);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1000, 'Strada Victoriei', 10, 100);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1001, 'Bulevardul Unirii', 225, 101);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1002, 'Strada Mihai Viteazu', 15, 102);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1003, 'Calea Dorobantilor', 530, 103);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1004, 'Bulevardul Eroilor', 120, 104);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1005, 'Strada Avram Iancu', 18, 105);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1006, 'Bulevardul Independentei', 40, 106);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1007, 'Strada Ion Creanga', 12, 107);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1008, 'Bulevardul Traian', 22, 108);

insert into LOCATIE(cod\_locatie, strada, nr\_strada, cod\_restaurant) values (1009, 'Strada Horea', 17, 109);

insert into LOCATIE(cod\_locatie, strada, nr\_strada) values (1021, 'Calea Bucuresti', 126);

insert into LOCATIE(cod\_locatie, strada, nr\_strada) values (1022, 'Ion Luca Caragiale', 1);

insert into LOCATIE(cod\_locatie, strada, nr\_strada) values (1023, 'Ion Mincu', 40);

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat

**4. Job**

create table JOB(

cod\_job number(3) constraint pk\_cod\_job primary key,

denumire\_job varchar(30) not null constraint nume\_job\_unic unique,

salariu\_minim number(5) not null,

salariu\_maxim number(5) not null);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (100, 'Bucatar sef', 6000, 12000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (101, 'Sous Chef', 5000, 10000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (102, 'Manager Restaurant', 5500, 11000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (103, 'Chelner/Chelnerita', 3000, 6000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (104, 'Barman/Barmanita', 3200, 6400);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (105, 'Bucatar adjunct', 3500, 7000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (106, 'Spalator de Vase', 2500, 5000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (107, 'Gazda', 2800, 5600);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (108, 'Manager General Restaurant', 8000, 15000);

insert into JOB(cod\_job, denumire\_job, salariu\_minim, salariu\_maxim) values (109, 'Manager de Marketing', 7500, 14000);

insert into job (cod\_job, denumire\_job, salariu\_minim, salariu\_maxim)values (110, 'Manager Transport', 3500, 5000);

insert into job (cod\_job, denumire\_job, salariu\_minim, salariu\_maxim)values (111, 'Sofer Livrare', 2000, 3000);

insert into job (cod\_job, denumire\_job, salariu\_minim, salariu\_maxim)values (112, 'Gestionar Depozit', 2500, 4000);

insert into job (cod\_job, denumire\_job, salariu\_minim, salariu\_maxim)values (113, 'Sofer Marfa', 2500, 3500);

insert into job (cod\_job, denumire\_job, salariu\_minim, salariu\_maxim)values (114, 'Coordonator Transport Marfa', 3000, 4500);

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat

**5.Angajat\_res**

create table Angajat\_Res(

cod\_angajat number(3) constraint pk\_cod\_angajat primary key,

nume varchar2(20) not null,

prenume varchar2(20) not null,

salariu number(5) not null,

email varchar2(50) not null constraint email\_ang\_unic unique,

nr\_telefon number(12) not null constraint nrtel\_ang\_unic unique,

cod\_manager number(3),

data\_inceput date not null,

constraint fk\_cod\_manager foreign key(cod\_manager) references angajat\_res(cod\_angajat),

cod\_job number(3),

constraint fk\_cod\_job foreign key(cod\_job) references job(cod\_job) on delete cascade,

cod\_restaurant number(3) not null,

constraint fk2\_cod\_restaurant foreign key(cod\_restaurant) references restaurant(cod\_restaurant) on delete cascade

);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(100, 'Popescu', 'Ion', 7000, 'popescu.ion@gmail.com', 720123456, null, TO\_DATE('2012-01-15', 'YYYY-MM-DD'), 102, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(101, 'Ionescu', 'Maria', 4000, 'ionescu.maria@gmail.com', 720234567, 100, TO\_DATE('2004-08-13', 'YYYY-MM-DD'), 103, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(102, 'Pop', 'Andrei', 3200, 'pop.andrei@gmail.com', 720345678, 100, TO\_DATE('2003-04-10', 'YYYY-MM-DD'), 104, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(103, 'Constantinescu', 'Ana', 8000, 'const.ana@gmail.com', 720456789, 100, TO\_DATE('2000-02-20', 'YYYY-MM-DD'), 100, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(104, 'Dumitrescu', 'Alex', 2800, 'dumi.alex@gmail.com', 720567890, 100, TO\_DATE('2007-12-12', 'YYYY-MM-DD'), 106, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(105, 'Stan', 'Elena', 6000, 'stan.elena@gmail.com', 720678901, 103, TO\_DATE('2006-07-21', 'YYYY-MM-DD'), 101, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(106, 'Munteanu', 'Cristian', 5500, 'munteanu.cris@gmail.com', 720789012, 103, TO\_DATE('2020-11-08', 'YYYY-MM-DD'), 105, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(107, 'Georgescu', 'Laura', 7000, 'georg.laura@gmail.com', 720890123, 100, TO\_DATE('2013-11-13', 'YYYY-MM-DD'), 107, 100);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(108, 'Vasilescu', 'Mihai', 9200, 'vasi.mihai@gmail.com', 720901234, null, TO\_DATE('2009-09-17', 'YYYY-MM-DD'), 102, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(109, 'Iordache', 'Ana', 7800, 'ior.ana@gmail.com', 73463456, 108, TO\_DATE('2005-10-10', 'YYYY-MM-DD'), 100, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(110, 'Popa', 'Catalin', 3400, 'popa.catalin@gmail.com', 7203124567, 108, TO\_DATE('2017-03-25', 'YYYY-MM-DD'), 103, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(111, 'Dragomir', 'Diana', 4800, 'drago.diana@gmail.com', 720234678, 108, TO\_DATE('2000-02-12', 'YYYY-MM-DD'), 104, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(112, 'Stanciu', 'Razvan', 6500, 'stan.razvan@gmail.com', 720534789, 109, TO\_DATE('2007-12-12', 'YYYY-MM-DD'), 101, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(113, 'Nistor', 'Adriana', 5400, 'nistor.adriana@gmail.com', 762347890, 109, TO\_DATE('2013-02-14', 'YYYY-MM-DD'), 105, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(114, 'Popescu', 'Andreea', 7900, 'popescu.andreea@gmail.com', 722368901, null, TO\_DATE('2014-03-20', 'YYYY-MM-DD'), 102, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(115, 'Ionescu', 'Marian', 7200, 'ion.marian@gmail.com', 7207234512, 114, TO\_DATE('2015-04-05', 'YYYY-MM-DD'), 100, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(116, 'Constantin', 'Diana', 5300, 'const.diana@gmail.com', 726234123, 115, TO\_DATE('2016-05-10', 'YYYY-MM-DD'), 101, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(117, 'Stoica', 'Radu', 3700, 'stoica.radu@gmail.com', 720623234, 114, TO\_DATE('2017-06-15', 'YYYY-MM-DD'), 103, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(118, 'Moldovan', 'Cristina', 3200, 'mold.cristina@gmail.com', 72014156, 114, TO\_DATE('2017-07-20', 'YYYY-MM-DD'), 103, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(119, 'Dumitru', 'Gabriel', 4800, 'dumi.gabriel@gmail.com', 7272367, 114, TO\_DATE('2013-08-25', 'YYYY-MM-DD'), 104, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(120, 'Iordache', 'Elena', 7600, 'ior.elena@gmail.com', 729923478, 114, TO\_DATE('2014-09-30', 'YYYY-MM-DD'), 107, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(121, 'Georgescu', 'Mihai', 7200, 'georg.mihai@gmail.com', 720412439, 115, TO\_DATE('2015-10-05', 'YYYY-MM-DD'), 105, 104);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(122, 'Popa', 'Ana', 8000, 'popa.ana@gmail.com', 72532290, null, TO\_DATE('2016-11-10', 'YYYY-MM-DD'), 102, 105);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(123, 'Vasilescu', 'Cristian', 7000, 'vasi.cris@gmail.com', 72023151, null, TO\_DATE('2017-12-15', 'YYYY-MM-DD'), 102, 106);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(124, 'Voiculescu', 'Niculae', 9000, 'voicu.nicu@gmail.com', 72353151, null, TO\_DATE('2017-12-15', 'YYYY-MM-DD'), 102, 101);

insert into angajat\_res (cod\_angajat, nume, prenume, salariu, email, nr\_telefon, cod\_manager, data\_inceput, cod\_job, cod\_restaurant) values(125, 'Badescu', 'Stefan', 7000, 'bad.stef@gmail.com', 72323551, null, TO\_DATE('2017-12-15', 'YYYY-MM-DD'), 102, 102);

O imagine care conține text, captură de ecran, software, Site web

Descriere generată automat

O imagine care conține text, captură de ecran, Dreptunghi

Descriere generată automat

**6. Istoric\_job**

create table istoric\_job(

cod\_istoric number(3) constraint pk\_cod\_istoric primary key,

data\_inceput date not null,

data\_final date not null,

nume varchar2(20) not null,

prenume varchar2(20) not null,

nume\_restaurant varchar(50) not null,

cod\_job number(3) not null,

constraint fk2\_cod\_job foreign key(cod\_job) references job(cod\_job) on delete cascade

);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (100, to\_date('1999-01-01', 'YYYY-MM-DD'), to\_date('2004-12-31', 'YYYY-MM-DD'), 'Popescu', 'Ion', 'Franciza Topoloveni', 102);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (102, to\_date('2000-02-15', 'YYYY-MM-DD'), to\_date('2002-06-30', 'YYYY-MM-DD'), 'Ionescu', 'Maria', 'Franciza Timisoara', 103);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (103, to\_date('1999-01-01', 'YYYY-MM-DD'), to\_date('2004-12-31', 'YYYY-MM-DD'), 'Stoian', 'Ionut', 'Franciza Timisoara', 100);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (104, to\_date('2000-02-15', 'YYYY-MM-DD'), to\_date('2005-06-30', 'YYYY-MM-DD'), 'Balan', 'Maria', 'Franciza Timisoara', 101);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (105, to\_date('2001-03-20', 'YYYY-MM-DD'), to\_date('2006-09-30', 'YYYY-MM-DD'), 'Radulescu', 'Andrei', 'Franciza Topoloveni', 102);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (106, to\_date('2002-04-10', 'YYYY-MM-DD'), to\_date('2007-12-31', 'YYYY-MM-DD'), 'Ilie', 'Ana', 'Franciza Topoloveni', 103);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (107, to\_date('2003-05-05', 'YYYY-MM-DD'), to\_date('2008-10-31', 'YYYY-MM-DD'), 'Dobre', 'Alex', 'Franciza Bucuresti', 104);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (108, to\_date('2004-06-01', 'YYYY-MM-DD'), to\_date('2009-12-15', 'YYYY-MM-DD'), 'Cretu', 'Elena', 'Franciza Bucuresti', 105);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (109, to\_date('2005-07-15', 'YYYY-MM-DD'), to\_date('2010-02-28', 'YYYY-MM-DD'), 'Marin', 'Cristian', 'Franciza Bucuresti', 106);

insert into istoric\_job (cod\_istoric, data\_inceput, data\_final, nume, prenume, nume\_restaurant, cod\_job)values (110, to\_date('2006-08-10', 'YYYY-MM-DD'), to\_date('2011-04-30', 'YYYY-MM-DD'), 'Alexandrescu', 'Laura', 'Franciza Bucuresti', 107);

O imagine care conține text, captură de ecran, software, Dreptunghi

Descriere generată automat

**7.Angajat\_livrari**

create table angajat\_livrari(

cod\_ang\_liv number(3) constraint pk\_cod\_ang\_liv primary key,

nume varchar2(20) not null,

prenume varchar2(20) not null,

salariu number(5) not null,

vehicul varchar(20),

email varchar2(50) not null,

nr\_telefon number(12) not null,

cod\_manager number(3),

constraint fk\_cod\_manager\_liv foreign key(cod\_manager) references angajat\_livrari(cod\_ang\_liv) on delete cascade,

cod\_job number(3) not null,

constraint fk\_cod\_job\_liv foreign key(cod\_job) references job(cod\_job) on delete cascade);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, email, nr\_telefon, cod\_manager, cod\_job)values (200, 'Popescu', 'Andrei', 5000, 'popescu.andrei@gmail.com', 720123456, null, 110);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (201, 'Ionescu', 'Maria', 3000, 'Dacia', 'ionescu.maria@gmail.com', 720234567, 200, 111);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (202, 'Ionescu', 'Maria', 3000, 'BMW', 'ionescu.maria@gmail.com', 720234567, 200, 111);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (203, 'Constantinescu', 'Radu', 3500, 'Bicicleta', 'const.radu@gmail.com', 720345678, 200, 111);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, email, nr\_telefon, cod\_manager, cod\_job)values (204, 'Dumitrescu', 'Cristina', 3300, 'dumitrescu.cristina@gmail.com', 720456789, null, 112);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, email, nr\_telefon, cod\_manager, cod\_job)values (205, 'Georgescu', 'Mihai', 3100, 'georg.mihai@gmail.com', 720567890, 200, 114);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (206, 'Stanciu', 'Ana', 3400, 'BMW', 'stanciu.ana@gmail.com', 720678901, 205, 113);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (207, 'Iordache', 'Andrei', 3200, 'Motocicleta', 'iordache.andrei@gmail.com', 720789012, 205, 113);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (208, 'Vasile', 'Ioana', 3000, 'Mercedes', 'vasile.ioana@gmail.com', 720890123, 200, 111);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (209, 'Vasile', 'Ioana', 3000, 'Dacia', 'vasile.ioana@gmail.com', 720890123, 200, 111);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (210, 'Vasile', 'Ioana', 3000, 'Motocicleta', 'vasile.ioana@gmail.com', 720890123, 200, 111);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, vehicul, email, nr\_telefon, cod\_manager, cod\_job)values (211, 'Mihai', 'Cristian', 3500, 'Motocicleta', 'mihai.cristian@gmail.com', 720901234, 205, 113);

insert into angajat\_livrari (cod\_ang\_liv, nume, prenume, salariu, email, nr\_telefon, cod\_manager, cod\_job)values (212, 'Radu', 'Elena', 3300, 'radu.elena@gmail.com', 720123456, null, 112);

**O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat**

**8. Meniu**

create table meniu(

cod\_meniu number(3) constraint pk\_cod\_meniu primary key,

descriere\_meniu varchar2(50),

numar\_pagini number(2) not null,

pret\_produs\_minim number(2) not null,

pret\_produs\_maxim number(2) not null,

cod\_restaurant not null constraint cod2\_restaurant\_unic unique,

constraint fk3\_cod\_restaurant foreign key(cod\_restaurant) references restaurant(cod\_restaurant) on delete cascade);

insert into meniu (cod\_meniu, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (300, 'Meniu Extra', 4, 15, 40, 100);

insert into meniu (cod\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (301, 6, 20, 45, 101);

insert into meniu (cod\_meniu, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (302, 'Meniu editite speciala', 8, 25, 50, 102);

insert into meniu (cod\_meniu, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (303, 'Meniu fructe de mare', 5, 18, 35, 103);

insert into meniu (cod\_meniu, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (304, 'Meniu Racoritor de Vara', 3, 10, 25, 104);

insert into meniu (cod\_meniu, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (305, 'Meniu Extra', 2, 8, 20, 105);

insert into meniu (cod\_meniu, descriere\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (306, 'Meniu oferte limitate', 4, 12, 30, 106);

insert into meniu (cod\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (307, 3, 15, 40, 107);

insert into meniu (cod\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant)values (308, 7, 30, 55, 108);

insert into meniu (cod\_meniu, numar\_pagini, pret\_produs\_minim, pret\_produs\_maxim, cod\_restaurant) values (309, 6, 25, 50, 109);

**O imagine care conține text, captură de ecran, software, număr

Descriere generată automat**

**9.Client**

create table client(

cod\_client number(5) constraint pk\_cod\_client primary key,

nume varchar2(20) not null,

prenume varchar2(20) not null,

telefon\_client number(12) not null,

cod\_meniu number(3),

constraint fk\_cod\_meniu foreign key(cod\_meniu) references meniu(cod\_meniu) on delete cascade);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10000, 'Popescu', 'Ana', 720123456, 300);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10001, 'Ionescu', 'Mihai', 720234567, 304);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10002, 'Constantinescu', 'Maria', 720345678, 305);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10003, 'Dumitrescu', 'Ion', 720456789, 300);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10004, 'Georgescu', 'Elena', 720567890, 304);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10005, 'Stanciu', 'Radu', 720678901, 305);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10006, 'Iordache', 'Andreea', 720789012, 300);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10007, 'Vasile', 'Gabriel', 720890123, 304);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10008, 'Mihai', 'Cristina', 720901234, 305);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10009, 'Radu', 'Andrei', 720123456, 300);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10010, 'Popescu', 'Ioana', 720234567, 304);

insert into client (cod\_client, nume, prenume, telefon\_client, cod\_meniu) values (10011, 'Ionescu', 'Catalin', 720345678, 305);

insert into client (cod\_client, nume, prenume, telefon\_client) values (10012, 'Constantinescu', 'Ana', 720456789);

insert into client (cod\_client, nume, prenume, telefon\_client) values (10013, 'Dumitrescu', 'Elena', 720567890);

insert into client (cod\_client, nume, prenume, telefon\_client) values (10014, 'Georgescu', 'Ionut', 720678901);

**O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat**

**10.Rezervare**

create table rezervare(

cod\_rezervare number(4) constraint pk\_cod\_rezervare primary key,

data\_rezervare date not null,

nume\_client varchar2(20) not null,

nr\_telefon number(12) not null,

cod\_restaurant number(3) not null,

constraint fk4\_cod\_restaurant foreign key(cod\_restaurant) references restaurant(cod\_restaurant) on delete cascade,

cod\_client number(5) not null,

constraint fk\_cod\_client foreign key(cod\_client) references client(cod\_client) on delete cascade);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1000, TO\_DATE('2024-06-15', 'YYYY-MM-DD'), 'Popescu Ana', 720123456, 104, 10000);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1001, TO\_DATE('2024-06-16', 'YYYY-MM-DD'), 'Ionescu Mihai', 720234567, 105, 10001);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1002, TO\_DATE('2024-06-17', 'YYYY-MM-DD'), 'Constantinescu Maria', 720345678, 100, 10002);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1003, TO\_DATE('2024-06-18', 'YYYY-MM-DD'), 'Dumitrescu Ion', 720456789, 100, 10003);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1004, TO\_DATE('2024-06-19', 'YYYY-MM-DD'), 'Georgescu Elena', 720567890, 104, 10004);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1005, TO\_DATE('2024-06-20', 'YYYY-MM-DD'), 'Stanciu Radu', 720678901, 105, 10005);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1006, TO\_DATE('2024-06-21', 'YYYY-MM-DD'), 'Iordache Andreea', 720789012, 100, 10006);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1007, TO\_DATE('2024-06-22', 'YYYY-MM-DD'), 'Vasile Gabriel', 720890123, 104, 10007);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1008, TO\_DATE('2024-06-23', 'YYYY-MM-DD'), 'Mihai Cristina', 720901234, 100, 10008);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1009, TO\_DATE('2024-06-24', 'YYYY-MM-DD'), 'Radu Andrei', 720123456, 105, 10009);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1010, TO\_DATE('2024-06-25', 'YYYY-MM-DD'), 'Popescu Ana', 720123456, 104, 10000);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1011, TO\_DATE('2024-06-26', 'YYYY-MM-DD'), 'Popescu Ana', 720123456, 104, 10000);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1012, TO\_DATE('2024-06-27', 'YYYY-MM-DD'), 'Ionescu Mihai', 720234567, 105, 10001);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1013, TO\_DATE('2024-06-28', 'YYYY-MM-DD'), 'Ionescu Mihai', 720234567, 100, 10001);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1014, TO\_DATE('2024-06-29', 'YYYY-MM-DD'), 'Constantinescu Maria', 720345678, 104, 10002);

insert into rezervare (cod\_rezervare, data\_rezervare, nume\_client, nr\_telefon, cod\_restaurant, cod\_client)values (1015, TO\_DATE('2024-06-30', 'YYYY-MM-DD'), 'Constantinescu Maria', 720345678, 100, 10002);

**O imagine care conține text, captură de ecran, software

Descriere generată automat**

**11. Comanda**

create table comanda(

cod\_comanda number(5) constraint pk\_cod\_comanda primary key,

pret number(4) not null,

nr\_produse number(2) not null,

cod\_client number(5) not null,

constraint fk2\_cod\_client foreign key(cod\_client) references client(cod\_client) on delete cascade);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10000, 150, 9, 10000);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10001, 120, 5, 10001);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10002, 200, 14, 10002);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10003, 90, 6, 10006);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10004, 210, 12, 10007);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10005, 70, 3, 10008);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10006, 130, 7, 10009);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10007, 70, 7, 10009);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10008, 30, 2, 10009);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10009, 100, 8, 10000);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10010, 20, 1, 10000);

insert into comanda (cod\_comanda, pret, nr\_produse, cod\_client) values (10012, 75, 4, 10001);

**O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat**

**12. Adresa\_livrare**

create table adresa\_livrare(

cod\_adresa number(5) constraint pk\_cod\_adresa primary key,

strada varchar2(20) not null,

nr\_strada number(5) not null);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10000, 'Strada Mihai Viteazu', 10);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10001, 'Blvd. Republicii', 25);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10002, 'Strada Libertă?ii', 15);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10003, 'Aleea Crângului', 8);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10004, 'Blvd. 1 Decembrie', 45);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10005, 'Strada Trandafirilor', 30);

insert into adresa\_livrare (cod\_adresa, strada, nr\_strada) values (10006, 'Aleea Fagilor', 12);

**O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat**

**13.Livrare**

create table Livrare(

durata\_estimativa number(3) not null,

cod\_comanda number(5),

cod\_adresa number (5),

cod\_ang\_liv number(3),

constraint pk\_livrare primary key(cod\_comanda, cod\_adresa, cod\_ang\_liv));

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (30, 10000, 10000, 201);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (45, 10001, 10001, 202);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (35, 10002, 10002, 202);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (25, 10003, 10003, 203);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (40, 10004, 10004, 208);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (50, 10005, 10005, 209);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (30, 10006, 10006, 210);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (35, 10007, 10006, 210);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (40, 10008, 10006, 210);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (30, 10009, 10000, 201);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (20, 10010, 10000, 201);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (35, 10011, 10001, 202);

**Corectare creare si inserarea tabel Livrare:**

create table Livrare(

durata\_estimativa number(3) not null,

cod\_comanda number(5),

constraint fk5\_cod\_comanda foreign key(cod\_comanda) references comanda(cod\_comanda),

cod\_adresa number (5),

constraint fk5\_cod\_adresa foreign key(cod\_adresa) references adresa\_livrare(cod\_adresa),

cod\_ang\_liv number(3),

constraint fk5\_cod\_ang\_liv foreign key(cod\_ang\_liv) references angajat\_livrari(cod\_ang\_liv),

constraint pk5\_livrare primary key(cod\_comanda, cod\_adresa, cod\_ang\_liv));

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (30, 10000, 10000, 201);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (45, 10001, 10001, 202);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (35, 10002, 10002, 202);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (25, 10003, 10003, 203);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (40, 10004, 10004, 208);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (50, 10005, 10005, 209);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (30, 10006, 10006, 210);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (35, 10007, 10006, 210);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (40, 10008, 10006, 210);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (30, 10009, 10000, 201);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (20, 10010, 10000, 201);

insert into livrare (durata\_estimativa, cod\_comanda, cod\_adresa, cod\_ang\_liv) values (35, 10011, 10001, 202);

**O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat**

# **12. Formularea în limbaj natural și implementarea a 5 cereri SQL complexe.**

1. Numele, prenumele, codul de angajat si salariul, angajatiilor care lucreaza la un restaurant, care au salariul mai mare decat salariul mediu al restaurantului in care lucreaza si restaurantul este detinut de proprietarul cu numele Victor sau Martinez.

select nume, prenume, cod\_angajat, salariu

from angajat\_res a1

where a1.salariu > ( select avg(salariu)

from angajat\_res a2 join restaurant r on a2.cod\_restaurant = r.cod\_restaurant join proprietar p on p.cod\_proprietar = r.cod\_proprietar

where lower(p.nume) like 'victor' OR lower(p.nume) like 'martinez'

group by a2.cod\_restaurant

having a1.cod\_restaurant = a2.cod\_restaurant);

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

Am folosit punctul a) in rezolvarea enuntului.

2. Sa se selecteze numele, prenumele, codul restaurantului si salariul angajatiilor care lucreaza intr-un restaurant si au salariul minim din acel restaurant.

select a.nume, a.cod\_restaurant, a.salariu

from angajat\_res a, (select min(salariu) sal, cod\_restaurant

from angajat\_res

group by cod\_restaurant) minim

where minim.sal = a.salariu and minim.cod\_restaurant = a.cod\_restaurant;

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

Am folosit punctul b) in rezolvarea enuntului.

3. Sa se selecteze restaurantul cu salariul total maxim, al carui proprietar are profitul mai mare decat 20000 si restaurantul are descriere de meniu. Prin salariu total intelegem, suma tuturor salariilor angajatiilor care lucreaza in acel restaurant.

select max(sum(salariu))

from angajat\_res a join restaurant r on a.cod\_restaurant = r.cod\_restaurant

group by r.cod\_restaurant

having r.cod\_restaurant in (select r.cod\_restaurant

from restaurant r join proprietar p on p.cod\_proprietar = r.cod\_proprietar join meniu m on m.cod\_restaurant = r.cod\_restaurant

where descriere\_meniu is not null and profit > 2000);

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

Corectare clauza having, problema 3:

3. Sa se afiseze angajatii care au cele mai multe livrari:

select count(\*), nume, prenume, a.cod\_ang\_liv

from angajat\_livrari a join livrare l on a.cod\_ang\_liv = l.cod\_ang\_liv

group by a.cod\_ang\_liv, nume, prenume

having count(\*) =( select max(count(\*))

from angajat\_livrari a join livrare l on a.cod\_ang\_liv = l.cod\_ang\_liv

group by a.cod\_ang\_liv);

O imagine care conține text, captură de ecran, software, Pictogramă computer

Descriere generată automat

Am folosit punctul c) in rezolvarea enuntului.

4. Sa se selecteze codul restaurantului si numarul de angajati\_res al fiecarui restaurant (care are angajati) care au fost angajati in anul 2014, 2015, 2016, respectiv 2017 si care au codul restaurantului mai mare decat 102. Sa se ordoneze totul crescator, dupa codul\_restaurantului

with aux as (select data\_inceput, a.cod\_restaurant

from angajat\_res a join restaurant r on a.cod\_restaurant = r.cod\_restaurant

where r.cod\_restaurant > 102)

select cod\_restaurant,

sum(nvl(decode(to\_char(data\_inceput, 'yyyy'), '2014', 1),0)) "Anul 2014",

sum(nvl(decode(to\_char(data\_inceput, 'yyyy'), '2015', 1),0)) "Anul 2015",

sum(nvl(decode(to\_char(data\_inceput, 'yyyy'), '2016', 1),0)) "Anul 2016",

sum(nvl(decode(to\_char(data\_inceput, 'yyyy'), '2017', 1),0)) "Anul 2017"

from aux

group by cod\_restaurant

order by cod\_restaurant;

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

Am folosit punctele d) si f) in rezolvarea enuntului.

5. Sa se selecteze numele, prenumele, numele restaurantului in care lucreaza, data angajarii, ultima zi a lunii din care data\_inceput face parte si data actuala pentru angajatii care lucreaza la un restaurant condus de proprietarul cu numele 'Victor' sau 'Martinez', cu o coloana aditionala in care sa scrie "DA" daca angajatul a fost angajat dupa anul 2015 si "NU" daca a fost angajat inainte.

select a.nume, a.prenume, nume\_restaurant, to\_char(data\_inceput,'yyyy-mm-dd') "data angajare", last\_day(data\_inceput), sysdate, case when to\_char(data\_inceput, 'yyyy') >= '2015' then 'Da' else 'Nu' end "Dupa 2015"

from angajat\_res a join restaurant r on a.cod\_restaurant = r.cod\_restaurant join proprietar p on p.cod\_proprietar = r.cod\_proprietar

where lower(p.nume) = 'victor' or upper(p.nume) = 'MARTINEZ';

O imagine care conține text, software, captură de ecran, număr

Descriere generată automat

Am folosit punctul e) in rezolvarea enuntului.

# **13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.**

1. Sa se modifice vanzarile saptamanale la valoarea de 10000 pentru fiecare restaurant care are nume de meniu.

update restaurant

set vanzari\_saptamanale = 10000

where cod\_restaurant in (select r.cod\_restaurant

from restaurant r join meniu m on r.cod\_restaurant = m.cod\_restaurant

where descriere\_meniu is not null);

O imagine care conține text, software, captură de ecran, Pagină web

Descriere generată automat

2. Sa se stearga livrarile care a caror comanda a fost mai scumpa de 170 de lei.

delete from livrare l

where l.cod\_comanda in (select cod\_comanda

from comanda

where pret > 170);

O imagine care conține text, software, Font, Pictogramă computer

Descriere generată automat

3. Sa se adauge inca 5000 la profitul proprietarilor care au restaurant de tip 'Mare'.

O imagine care conține text, captură de ecran, software, Pictogramă computer

Descriere generată automat

# **14.Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă**

Am creat urmatorul view:

create or replace view angajati\_si\_restaurant as ( select a.cod\_angajat, r.cod\_restaurant, a.nume, a.prenume, a.nr\_telefon, a.email, a.cod\_job, a.data\_inceput, a.salariu, r.nume\_restaurant

from angajat\_res a join restaurant r on a.cod\_restaurant = r.cod\_restaurant

where r.cod\_restaurant = 100);

O imagine care conține text, captură de ecran, linie, software

Descriere generată automat

Un exemplu de operatie permisa este selectul:

select \*

from angajati\_si\_restaurant;

O imagine care conține text, captură de ecran, număr, software

Descriere generată automat

Un exemplu de operatie nepermisa este insertul:

insert into angajati\_si\_restaurant(cod\_angajat, cod\_restaurant, nume, prenume, nr\_telefon, email, cod\_job, data\_inceput, salariu, nume\_restaurant)

values(300, 400, 'ABBA', 'CHICHICITA', '07123412', 'victor.fugulin@s.unibuc.ro',to\_date('2020-10-10', 'yyyy-mm-dd') ,103, 7000, 'Franciza La Park');

O imagine care conține text, captură de ecran, Font, Pagină web

Descriere generată automat