Baza de date a unui lant de pizzerii

Realizat de: Suditu Darius-Cosmin

Cuprins

1. [Descrierea modelului](#ex1)
2. [Constrangeri (restrictii, reguli)](#ex2)
3. [Entitati](#b3)
4. [Relatii](#b4)
5. [Atribute](#b5)
6. [Diagrama Entitate-Relatie](#b6)
7. [Diagrama Conceptuala](#b7)
8. [Schema Relationala](#b8)
9. [Normalizarea](#b9)
   1. NON FN1 – FN1
   2. NON FN2 – FN2
   3. NON FN3 – FN3
10. [Crearea unei secvente](#b10)
11. [Crearea tabelelor si inserarea datelor](#b11)
12. [Cereri SQL](#b12)
    1. Cerinta1 + Rezolvare1
    2. Cerinta2 + Rezolvare2
    3. Cerinta3 + Rezolvare3
    4. Cerinta4 + Rezolvare4
    5. Cerinta5 + Rezolvare5
13. [Actualizare si Suprimare](#b13)

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

Pentru acest proiect, am ales realizarea bazei de date a unui lant de pizzerii, ce are mai multe locatii in diferite orase precum Bucuresti, Constanta, Cluj. Clientii pot face rezervare intr-una din locatii si pot comanda diverse preparate din meniu. Acest lucru se poate realiza cu ajutorul angajatilor, care sunt de doua tipuri, fie parte din personal, fie bucatari, ce au rolul de a servi comenzile, respectiv de a le prepara. Bucatarii gatesc diverse preparate din meniu, gatite cu ajutorul unor ingrediente. De asemenea, clientii pot oferi review-uri in functie de cat de satisfacuti au fost de cele comandate si primi diferite promotii avantajoase. Utilitatea principala a acestei baze de date este gestionarea optima a informatiilor referitoare la clienti, comenzi, rezervari, angajati si preparate in cadrul lantului de pizzerii. Ea permite inregistrarea si stocarea datelor relevante pentru functionarea eficienta a afacerii.

1. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Restrictii si reguli:

* Fiecare angajat lucreaza intr-o singura locatie, iar fiecare locatie are cel putin un bucatar si cel putin un angajat ce face parte din personal.
* Angajatii lucreaza fie ca bucatari, fie ca personal, avand diferite specializari. Acestia au un ID unic de angajat dar si un ID pentru departamentul din care fac parte.
* Comenzile sunt unice si pot fi identificate printr-un ID.
* Pizzeriile se afla in diverse locatii din diferite orase, identifiabile prin ID.
* Salariul angajatilor este direct proportional cu experienta lor.
* Clientii pot efectua rezervari unice ce pot fi identificate printr-un ID.
* Clientii pot lasa recenzii unice ce pot fi identificate printr-un ID.
* Pizzelor le sunt aplicate promotii in functie de ofertele curente.

Constrangeri:

* Cheie primara – PK
* Cheie externa – FK

1. Descrierea entităților, incluzând precizarea cheii primare.

***CLIENT***(ID\_CLIENT#) – entitate ce va contine detalii despre un client al pizzeriei.

***LOCATIE*** (ID\_LOCATIE#) – entitate ce va contine detalii despre locatiile in care se afla pizzeriile.

***PIZZERIE***(ID\_PIZZERIE#) – entitate ce va contine informatii despre o pizzerie din lantul de pizzerii.

***ANGAJAT*** (ID\_ANGAJAT#) – entitate ce contine detalii despre toti angajatii firmei.

***PERSONAL***(ID\_PERSONAL#) – entitate ce va contine detalii despre angajatii din pizzerie ce interactioneaza cu clientii.

***BUCATAR****(*ID\_BUCATAR#*)*– entitate care contine detalii despre angajatii din pizzerii care prepara mancarea.

***PIZZA*** (ID\_PIZZA#) – entitate ce va contine detalii despre pizzele disponibile.

***INGREDIENT***(ID\_INGREDIENT#) – entitate ce va contine detalii despre ingredientele folosite in gatirea pizzei.

***PREPARARE*** (ID\_PIZZA#ID\_INGREDIENT#) – entitate ce va contine detalii despre modul de gatire al fiecarei pizza.

***COMANDA***(ID\_COMANDA#) – entitate ce va contine detalii despre o comanda facuta de un client.

***REZERVARE***(ID\_REZERVARE#) – entitate ce va contine detalii despre o rezervare facuta de un client.

***RECENZIE***(ID\_RECENZIE#) – entitate ce va contine detalii despre o recenzie facuta de un client.

***PROMOTIE***(ID\_PROMOTIE#) – entitate ce va contine detalii despre o promotie a pizzeriei.

1. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

***CLIENT-COMANDA:*** un client poate plasa mai multe comenzi, dar o comanda poate fi plasata de un singur client.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(1)

***CLIENT-RECENZIE:*** un client poate plasa mai multe recenzii, dar o recenzie poate fi plasata de un singur client.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

***CLIENT-REZERVARE:*** un client poate face mai multe rezervari, dar o rezervare poate fi facuta de un singur client.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

***PIZZA-PROMOTIE:*** o pizza poate beneficia de mai multe promotii, dar o promotie corespunde unei singure pizza.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

***ANGAJAT-BUCATAR,PERSONAL***: angajatul este fie bucatar, fie parte din personalul pizzeriei.

* Tipul relatiei: ISA
* Cardinalitate: 1-1(0)

***PIZZA-INGREDIENT***: o pizza este confectionata din mai multe ingrediente, iar mai multe ingrediente sunt folosite pt confectionarea unei pizza.

* Tipul relatiei: many-many
* Cardinalitate: M(1)-M(1)

***BUCATAR-PIZZERIE***: intr-o pizzerie pot lucra mai multi bucatari, dar un bucatar poate lucra intr-o singura pizzerie.

* Tipul relatiei: many-one
* Cardinalitate: M(1)-1

***PIZZERIE-LOCATIE***: o pizzerie corespunde unei locatii.

* Tipul relatiei: one-one
* Cardinalitate: 1-1

***PIZZERIE-PERSONAL***: intr-o pizzerie pot lucra mai multi angajati, dar un angajat poate lucra intr-o singura pizzerie.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(1)

***PIZZERIE-COMANDA***: o pizzerie poate primi mai multe comenzi, dar o comanda se poate da dintr-o singura pizzerie.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

***PIZZERIE-REZERVARE***: o pizzerie poate primi mai multe rezervari, dar o rezervare se poate face pentru o singura pizzerie.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

***PIZZERIE-RECENZIE***: o pizzerie poate primi mai multe recenzii, dar o recenzie se poate face pentru o singura pizzerie.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

***COMANDA-PIZZA***: o comanda poate contine un singur tip de pizza, dar o pizza poate fi in mai multe comenzi.

* Tipul relatiei: one-many
* Cardinalitate: 1-M(0)

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

***CLIENT***

ID\_CLIENT# int not null

NUME varchar2(20) not null

EMAIL varchar2(30)

ID\_CLIENT# => cheie primara de tip int, diferita de null, care identifica unic un client.

NUME => atribut de tip sir de caractere, diferit de null, de lungime maxima 0, ce va contine numele intreg al clientului.

EMAIL => atribut de tip sir de caractere, de lungime maxima 30, ce va contine adresa de email a clientului.

***LOCATIE***

ID\_LOCATIE# int not null

ORAS varchar2(20) not null

STRADA varchar2(30)

COD\_POSTAL varchar2(10) not null

ID\_LOCATIE # => cheie primara de tip int, diferita de null, care identifica unic o locatie

ORAS => atribut de tip sir de caractere diferit de null ce va contine numele orasului

STRADA => atribut de tip sir de caractere ce va contine numele strazii

COD\_POSTAL => atribut de tip sir de caractere ce va contine codul postal (10 caractere).

***PIZZERIE***

ID\_PIZZERIE# int not null

NUME varchar2(20) not null

NUMAR\_TELEFON varchar2(10)

CAPACITATE int not null

ID\_LOCATIE FK not null

ID\_PIZZERIE # => cheie primara de tip int, diferita de null, care identifica unic o pizzerie

NUME => atribut de tip sir de caractere, diferit de null, de lungime maxima 20, ce va contine numele pizzeriei.

NUMAR\_TELEFON => atribut de tip sir de caractere, de lungime maxima 10, ce va contine nr de telefon al pizzeriei.

CAPACITATE => atribut de tip int, diferit de null, ce va contine capacitatea maxima a pizzeriei.

FK ID\_LOCATIE => cheie externa ce va identifica id-ul de locatie al locatiei in care este amplasata pizzeria.

***ANGAJAT***

ID\_ANGAJAT# int not null

NUME varchar2(50) not null

SALARIU int not null

DATA\_ANGAJARE date not null

NUMAR\_TELEFON varchar2(10)

ID\_ANGAJAT # => cheie primara de tip int, diferita de null, care identifica unic un angajat

NUME => atribut de tip sir de caractere, diferit de null, de lungime maxima 50, ce va contine numele angajatului.

DATA\_ANGAJARE => atribut de tip sir data care va contine data angajarii fiecarui angajat.

NUMAR\_TELEFON => atribut de tip sir de caractere, de lungime maxima 10, ce va contine nr de telefon al angajatului.

***PERSONAL***

ID\_PERSONAL# int not null

SPECIALIZARE varchar2(20) not null

ID\_ANGAJAT FK not null

ID\_PIZZERIE FK not null

ID\_Personal# => cheie primara de tip int, diferita de null, care identifica unic un angajat din pizzerie.

SPECIALIZARE => atribut de tip sir de caractere ce reprezinta specializarea angajatului

FK ID\_ANGAJAT => cheie externa ce va identifica id-ul de angajat al angajatului din pizzerie.

FK ID\_PIZZERIE => cheie externa ce va identifica id-ul de pizzerie al pizzeriei in care lucreaza angajatul.

***BUCATAR***

ID\_BUCATAR# int not null

SPECIALIZARE varchar2(20) not null

ID\_ANGAJAT FK not null

ID\_PIZZERIE FK not null

ID\_BUCATAR# => cheie primara de tip int, diferita de null, care identifica unic un bucatar din pizzerie.

SPECIALIZARE => atribut de tip sir de caractere ce reprezinta specializarea bucatarului.

FK ID\_ANGAJAT => cheie externa ce va identifica id-ul de angajat al angajatului din pizzerie.

FK ID\_PIZZERIE => cheie externa ce va identifica id-ul de pizzerie al pizzeriei in care lucreaza bucatarul.

***PIZZA***

ID\_PIZZA# int not null

NUME varchar2(20) not null

TIP varchar2(20) not null

PRET int not null

ID\_PIZZA# => cheie primara de tip int, diferita de null, care identifica unic o pizza.

NUME => atribut de tip sir de caractere ce reprezinta numele pizzei.

TIP => atribut de tip sir de caractere ce reprezinta tipul pizzei.

PRET => atribut de tip int ce reprezinta pretul pizzei.

***INGREDIENT***

ID\_INGREDIENT# int not null

NUME varchar2(50) not null

CANTITATE int

ID\_INGREDIENT# => cheie primara de tip int, diferita de null, care identifica unic un ingredient.

NUME => atribut de tip sir de caractere ce va contine numele unui ingredient.

CANTITATE => atribut de tip int ce va spune cantitatea de ingredient folosita.

***PREPARARE***

PK FK ID\_PIZZA

PK FK ID\_INGREDIENT

***COMANDA***

ID\_COMANDA# int not null

CANTITATE int not null

PRET int not null

DATA\_COMANDA date

ID\_PIZZA FK not null

ID\_PIZZERIE FK not null

ID\_CLIENT FK not null

ID\_COMANDA# => cheie primara de tip int, diferita de null, care identifica unic o comanda data.

CANTITATE => atribut de tip int ce va identifica numarul de pizze comandate.

PRET => atribut de tip int ce va identifica pretul total al unei comenzi.

DATA\_COMANDA => atribut de tip date ce va identifica data in care a fost efectuata o comanda.

FK ID\_PIZZA => cheie externa ce va identifica id-ul pizzei comandate.

FK ID\_PIZZERIE => cheie externa ce va identifica id-ul pizzeriei din care este efectuata comanda.

FK ID\_CLIENT => cheie externa ce va identifica id-ul clientului ce efectueaza comanda.

***RECENZIE***

ID\_RECENZIE# int not null

DATA\_RECENZIE date

RATING int not null

COMENTARIU varchar2(50)

ID\_PIZZERIE FK not null

ID\_CLIENT FK not null

ID\_RECENZIE# => cheie primara de tip int, diferita de null, care identifica unic o recenzie data.

DATA\_RECENZIE => atribut de tip date ce va reprezenta data in care a fost trimisa o recenzie.

RATING => atribut de tip int ce va reprezenta rating-ul oferit in recenzie.

COMENTARIU => atribut de tip sir de caractere ce va reprezenta comentariul oferit in recenzie.

FK ID\_PIZZERIE => cheie externa ce va identifica ID-ul pizzeriei pentru care este data recenzia.

FK ID\_CLIENT => cheie externa ce va identifica ID-ul clientului ce efectueaza recenzia.

***REZERVARE***

ID\_REZERVARE# int not null

DATA\_REZERVARE date

ORA varchar2(8)

NUMAR\_PERSOANE int not null

ID\_PIZZERIE FK not null

ID\_CLIENT FK not null

ID\_REZERVARE# => cheie primara de tip int, diferita de null, care identifica unic o rezervare facuta.

DATA\_REZERVARE => atribut de tip date ce va reprezenta data in care a fost facuta o rezervare.

NUMAR\_PERSOANE => atribut de tip int ce va reprezenta numarul de persoane pt care este facuta rezervarea.

ORA => atribut de tip sir de caractere ce va reprezenta ora la care este facuta rezervarea.

FK ID\_PIZZERIE => cheie externa ce va identifica ID-ul pizzeriei pentru care este facuta rezervarea.

FK ID\_CLIENT => cheie externa ce va identifica ID-ul clientului ce efectueaza rezervarea.

***PROMOTIE***

ID\_PROMOTIE# int not null

DATA\_START date

DATA\_STOP date

VALOARE\_REDUCERE int not null

ID\_PIZZA FK not null

ID\_PROMOTIE# => cheie primara de tip int, diferita de null, care identifica unic o promotie existenta.

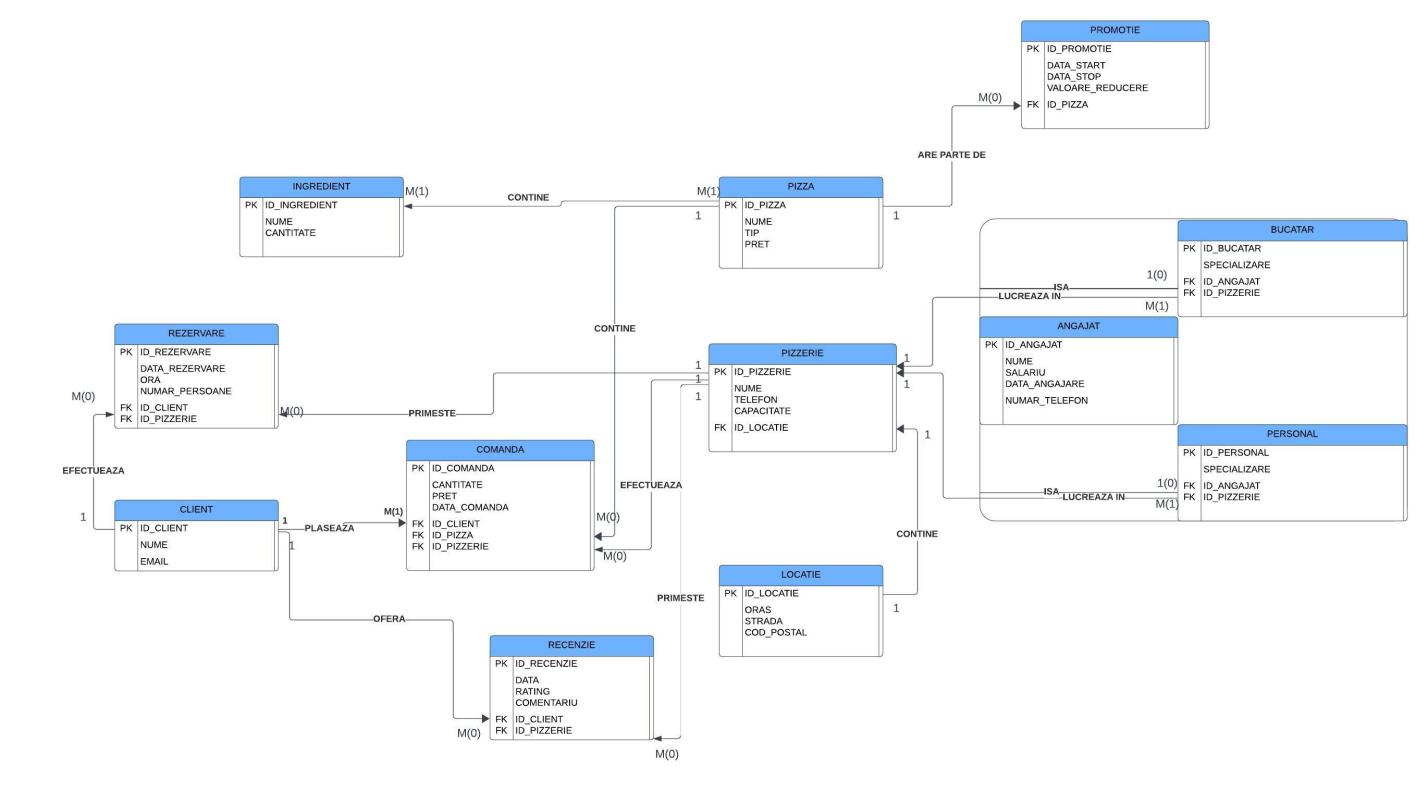
DATA\_START => atribut de tip date ce va reprezenta data la care incepe promotia.

DATA\_STOP => atribut de tip date ce va reprezenta data la care se termina promotia.

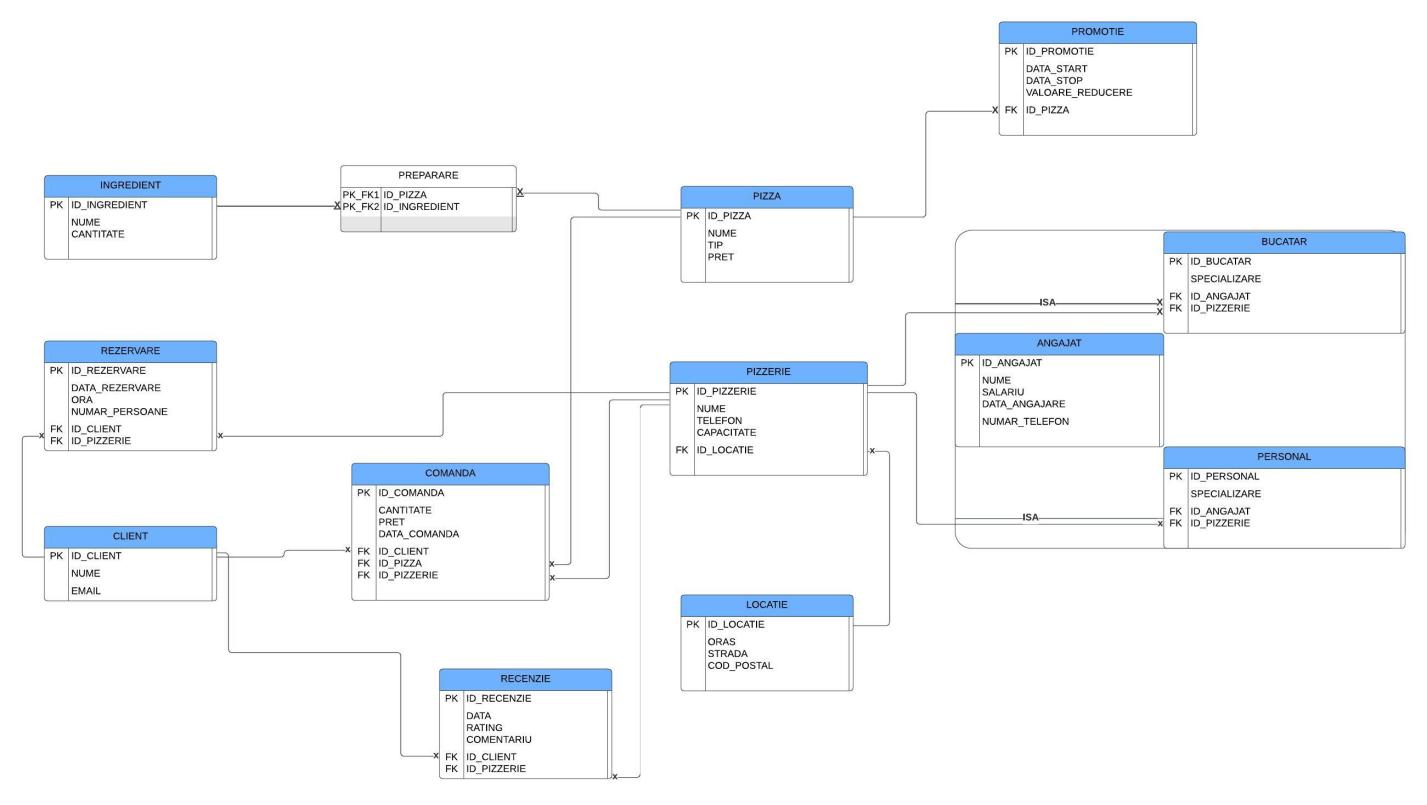
VALOARE\_REDUCERE => atribut de tip int ce va reprezenta valoarea reducerii.

FK ID\_PIZZA => cheie externa ce va identifica ID-ul pizzei pt care se aplica promotia.

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



1. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



1. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

***CLIENT***

(ID\_CLIENT#, NUME, EMAIL)

***LOCATIE***

(ID\_LOCATIE#, ORAS, STRADA, COD\_POSTAL)

***PIZZERIE***

(ID\_PIZZERIE#, NUME, NUMAR\_TELEFON, ID\_LOCATIE FK)

***ANGAJAT***

(ID\_ANGAJAT#, NUME, SALARIU, DATA\_ANGAJARE, NUMAR\_TELEFON)

***PERSONAL***

(ID\_PERSONAL#, SPECIALIZARE, ID\_ANGAJAT FK)

***BUCATAR***

(ID\_BUCATAR#, SPECIALIZARE, ID\_ANGAJAT FK)

***PIZZA***

(ID\_PIZZA#, NUME, TIP, PRET)

***INGREDIENT***

(ID\_INGREDIENT#, NUME, CANTITATE)

***PREPARARE***

(ID\_PIZZA PK FK, ID\_INGREDIENT PK FK)

***COMANDA***

(ID\_COMANDA#, CANTITATE, PRET, DATA\_COMANDA, ID\_PIZZA FK, ID\_PIZZERIE FK, ID\_CLIENT FK)

***REZERVARE***

(ID\_REZERVARE#, DATA\_REZERVARE, ORA, NUMAR\_PERSOANE, ID\_PIZZERIE FK, ID\_CLIENT FK)

***RECENZIE***

(ID\_RECENZIE#, DATA\_RECENZIE, RATING, COMENTARIU, ID\_PIZZERIE FK, ID\_CLIENT FK)

***PROMOTIE***

(ID\_PROMOTIE#, DATA\_START, DATA\_STOP, VALOARE\_REDUCERE, ID\_PIZZA FK)

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

**NON FN1:**

|  |  |
| --- | --- |
| Pizzerie | |
| ID\_Pizzerie - PK | Recenzie |
| P1 | R1,R2,R3 |
| P2 | R4,R5 |

Prima forma normala elimina grupurile repetitive, asigurand ca fiecare camp contine o singura valoare si ca fiecare inregistrare este identificata unic printr-o cheie primara.

Realizam un nou tabel pentru recenzie, nu le adaugam in pizzerie.

**FN1:**

|  |
| --- |
| Pizzerie |
| ID\_Pizzerie - PK |
| P1 |
| P2 |

|  |  |  |
| --- | --- | --- |
| Recenzie | | |
| ID\_Recenzie - PK | Rating | ID\_Pizzerie# |
| R1 | 4 | P1 |
| R2 | 5 | P1 |
| R3 | 4 | P1 |
| R4 | 4 | P2 |
| R5 | 3 | P2 |

**NON FN2:**

|  |  |  |
| --- | --- | --- |
| Construire | | |
| ID\_Pizza - PK | ID\_Ingredient- PK | Nume\_Ingredient |
| PIZ1 | 1 | Mozzarela |
| PIZ2 | 2 | Pepperoni |

Avand o dependenta partiala, realizam un nou tabel pentru Ingredient deoarece numele depinde doar de acesta.

**FN2:**

|  |  |
| --- | --- |
| Construire | |
| ID\_Pizza - PK | ID\_Ingredient - PK |
| PIZ1 | 1 |
| PIZ2 | 2 |

|  |  |
| --- | --- |
| Ingredient | |
| ID\_Ingredient - PK | Nume\_Ingredient |
| 1 | Mozzarela |
| 2 | Pepperoni |

**NON FN3:**

|  |  |  |
| --- | --- | --- |
| Comanda | | |
| ID\_Comanda - PK | Cantitate | Nume client |
| 1 | 3 | Ana Popescu |
| 2 | 4 | Mihai Ionescu |

Realizam un nou tabel pentru client deoarece numele clientului nu depinde de cheia primara.

**FN3:**

|  |  |  |
| --- | --- | --- |
| Comanda | | |
| ID\_Comanda - PK | Cantitate | ID\_Client - FK |
| 1 | 3 | C1 |
| 2 | 4 | C2 |

|  |  |
| --- | --- |
| Client | |
| ID\_Client - PK | Nume |
| C1 | Ana Popescu |
| C2 | Mihai Ionescu |

10.Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in tabele.

CREATE SEQUENCE seq\_id\_ingredient

START WITH 1

INCREMENT BY 1

NOCYCLE;



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).

--CREARE--

CREATE TABLE Pizza

(

ID\_Pizza VARCHAR2(10) NOT NULL,

Nume VARCHAR2(20) NOT NULL,

Tip VARCHAR2(20) NOT NULL,

Pret int not null,

PRIMARY KEY (ID\_Pizza)

);

create TABLE Angajat(

ID\_Angajat int not null,

Nume VARCHAR2(50) not null,

Salariu int not null,

Data\_Angajare date not null,

Numar\_Telefon VARCHAR2(10),

PRIMARY KEY (ID\_Angajat)

);

CREATE TABLE Locatie

(

ID\_Locatie VARCHAR2(10) NOT NULL,

Oras VARCHAR2(20) NOT NULL,

Strada VARCHAR2(30),

Cod\_postal VARCHAR2(10) NOT NULL,

PRIMARY KEY (ID\_Locatie)

);

CREATE TABLE Pizzerie

(

ID\_PIZZERIE VARCHAR2(20) NOT NULL,

Nume VARCHAR2(10) NOT NULL,

numar\_telefon VARCHAR2(10),

Capacitate int not null,

ID\_Locatie VARCHAR2(10) NOT NULL,

PRIMARY KEY (ID\_Pizzerie),

FOREIGN KEY (ID\_Locatie) REFERENCES Locatie(ID\_Locatie)

);

CREATE TABLE Bucatar

(

ID\_Bucatar INT NOT NULL,

Specializare VARCHAR2(20) NOT NULL,

ID\_Pizzerie VARCHAR2(10) NOT NULL,

ID\_Angajat not null,

PRIMARY KEY (ID\_Bucatar),

FOREIGN KEY (ID\_Angajat) REFERENCES Angajat(ID\_Angajat),

FOREIGN KEY (ID\_Pizzerie) REFERENCES Pizzerie(ID\_Pizzerie)

);

CREATE TABLE Client

(

ID\_Client INT NOT NULL,

Nume VARCHAR2(20) NOT NULL,

Email VARCHAR2(30),

PRIMARY KEY (ID\_Client)

);

CREATE TABLE Personal

(

ID\_Personal INT NOT NULL,

Specializare VARCHAR2(20) NOT NULL,

ID\_ANGAJAT not null,

ID\_PIZZERIE VARCHAR2(10) NOT NULL,

PRIMARY KEY (ID\_Personal),

FOREIGN KEY (ID\_Angajat) REFERENCES Angajat(ID\_Angajat),

FOREIGN KEY (ID\_PIZZERIE) REFERENCES Pizzerie(ID\_Pizzerie)

);

create TABLE Rezervare(

ID\_Rezervare int not null primary key,

Data\_Rezervare date,

Ora VARCHAR2(8),

Numar\_Persoane int not null,

ID\_Client int not null,

ID\_Pizzerie VARCHAR2(10) not null,

foreign key(ID\_Client) references Client(ID\_Client),

foreign key(ID\_Pizzerie) references Pizzerie(ID\_Pizzerie)

);

create TABLE Recenzie(

ID\_Recenzie int not null primary key,

Data\_Recenzie date,

Rating int not null,

Comentariu VARCHAR2(50) NOT NULL,

ID\_Client int not null,

ID\_Pizzerie VARCHAR2(10) not null,

foreign key(ID\_Client) references Client(ID\_Client),

foreign key(ID\_Pizzerie) references Pizzerie(ID\_Pizzerie)

);

create TABLE Promotie(

ID\_Promotie int not null primary key,

Data\_Start date,

Data\_Stop date,

Valoare\_Reducere int not null,

ID\_Pizza VARCHAR2(10) not null,

foreign key(ID\_Pizza) references Pizza(ID\_Pizza)

);

create TABLE Ingredient(

ID\_Ingredient int not null primary key,

Nume VARCHAR2(50) not null,

Cantitate int NOT NULL

);

CREATE TABLE Preparare (

ID\_Pizza VARCHAR2(10) NOT NULL,

ID\_Ingredient INT NOT NULL,

PRIMARY KEY (ID\_Pizza, ID\_Ingredient),

FOREIGN KEY (ID\_Pizza) REFERENCES Pizza(ID\_Pizza),

FOREIGN KEY (ID\_Ingredient) REFERENCES Ingredient(ID\_Ingredient)

);

CREATE TABLE Comanda

(

ID\_Comanda INT NOT NULL,

Cantitate INT NOT NULL,

Pret INT NOT NULL,

Data\_Comanda date,

ID\_Pizza VARCHAR2(10) NOT NULL,

ID\_Pizzerie VARCHAR2(10) NOT NULL,

ID\_Client INT NOT NULL,

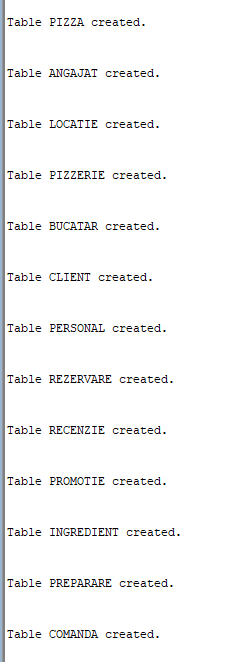
PRIMARY KEY (ID\_Comanda),

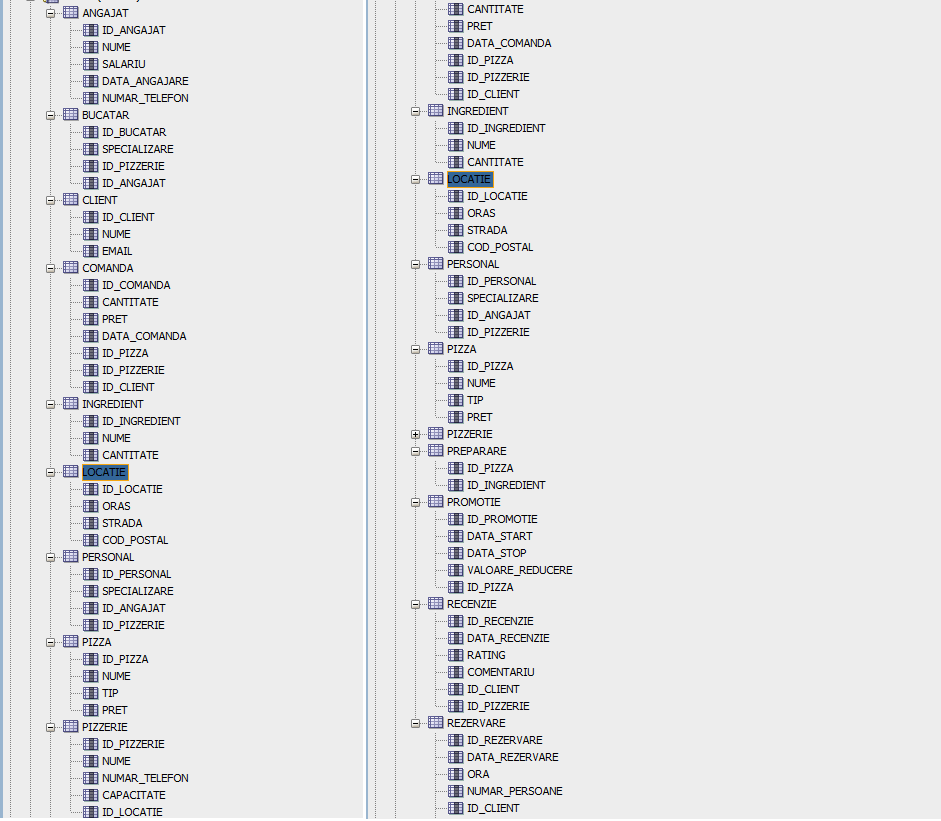
FOREIGN KEY (ID\_Pizza) REFERENCES Pizza(ID\_Pizza),

FOREIGN KEY (ID\_Pizzerie) REFERENCES Pizzerie(ID\_Pizzerie),

FOREIGN KEY (ID\_Client) REFERENCES Client(ID\_Client)

);





--INSERARE--

insert into Client (ID\_Client, nume, email) VALUES (1, 'Ana Popescu', 'ana.popescu@example.com');

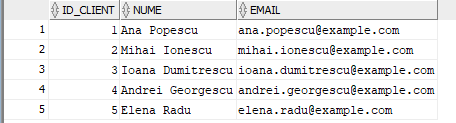
insert into Client (ID\_Client, nume, email) VALUES(2, 'Mihai Ionescu', 'mihai.ionescu@example.com');

insert into Client (ID\_Client, nume, email) VALUES(3, 'Ioana Dumitrescu', 'ioana.dumitrescu@example.com');

insert into Client (ID\_Client, nume, email) VALUES(4, 'Andrei Georgescu', 'andrei.georgescu@example.com');

insert into Client (ID\_Client, nume, email) VALUES(5, 'Elena Radu', 'elena.radu@example.com');

select \* from client;



insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO001', 'Bucharest', 'Nordului', '010101');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO002', 'Cluj-Napoca', 'Gheorghe Titeica', '020202');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO003', 'Timisoara', 'Brailita', '030303');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO004', 'Iasi', 'Mugur Mugurel', '040404');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO005', 'Constanta', 'Poenaru', '050505');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO006', 'Brasov', 'Sforii', '060606');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO007','Iasi', 'Radului', '101010');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO008', 'Bucuresti', 'Floreasca', '202020');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO009', 'Bucuresti', 'Arcului', '303030');

insert into Locatie (ID\_Locatie, Oras, Strada, Cod\_postal) VALUES('RO010', 'Constanta', 'Plajei', '001122');

select \* from locatie;



insert into Pizzerie (ID\_Pizzerie, Nume, numar\_telefon,Capacitate, ID\_Locatie) VALUES ('P1', 'Pizzerie1', '1234567890',88, 'RO001');

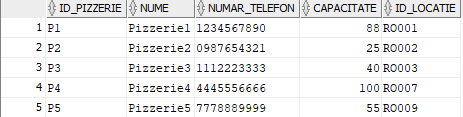
insert into Pizzerie (ID\_Pizzerie, Nume, numar\_telefon,Capacitate, ID\_Locatie) VALUES('P2', 'Pizzerie2', '0987654321',25, 'RO002');

insert into Pizzerie (ID\_Pizzerie, Nume, numar\_telefon,Capacitate, ID\_Locatie) VALUES('P3', 'Pizzerie3', '1112223333',40, 'RO003');

insert into Pizzerie (ID\_Pizzerie, Nume, numar\_telefon,Capacitate, ID\_Locatie) VALUES('P4', 'Pizzerie4', '4445556666',100, 'RO007');

insert into Pizzerie (ID\_Pizzerie, Nume, numar\_telefon,Capacitate, ID\_Locatie) VALUES('P5', 'Pizzerie5', '7778889999',55, 'RO009');

select \* from Pizzerie;



insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(1, 'Ion Popescu', 3000, TO\_DATE('2023-01-15', 'YYYY-MM-DD'), '0712345678');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(2, 'Maria Ionescu', 3500, TO\_DATE('2022-11-20', 'YYYY-MM-DD'), '0723456789');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(3, 'Ana Dumitrescu', 3200, TO\_DATE('2023-02-10', 'YYYY-MM-DD'), '0734567890');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(4, 'Andrei Georgescu', 4000, TO\_DATE('2023-03-05', 'YYYY-MM-DD'), '0745678901');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(5, 'Elena Radu', 3800, TO\_DATE('2022-09-25', 'YYYY-MM-DD'), '0756789012');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(6, 'Mihai Stan', 3400, TO\_DATE('2022-12-10', 'YYYY-MM-DD'), '0767890123');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(7, 'Cristina Stoica', 3700, TO\_DATE('2023-04-15', 'YYYY-MM-DD'), '0778901234');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(8, 'Alexandra Marin', 3600, TO\_DATE('2023-05-20', 'YYYY-MM-DD'), '0789012345');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(9, 'Florin Munteanu', 3900, TO\_DATE('2023-07-01', 'YYYY-MM-DD'), '0790123456');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(10, 'Diana Andreescu', 3300, TO\_DATE('2023-08-10', 'YYYY-MM-DD'), '0801234567');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(11, 'Gabriel Popa', 3100, TO\_DATE('2023-09-15', 'YYYY-MM-DD'), '0812345678');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(12, 'Andreea Dumitrascu', 3400, TO\_DATE('2023-10-20', 'YYYY-MM-DD'), '0823456789');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(13, 'Catalin Stanciu', 3200, TO\_DATE('2023-11-10', 'YYYY-MM-DD'), '0834567890');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(14, 'Larisa Radulescu', 3800, TO\_DATE('2023-12-05', 'YYYY-MM-DD'), '0845678901');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(15, 'Marius Iorgulescu', 3700, TO\_DATE('2024-01-15', 'YYYY-MM-DD'), '0856789012');

insert into Angajat (ID\_Angajat, nume, salariu, data\_angajare, numar\_telefon) VALUES(16, 'Dan Beleaua', 4300, TO\_DATE('2024-01-19', 'YYYY-MM-DD'), '0732456852');

select \* from angajat;



insert into Personal (ID\_Personal, Specializare, ID\_Angajat, ID\_Pizzerie) VALUES(1, 'Ospatar', 1, 'P1');

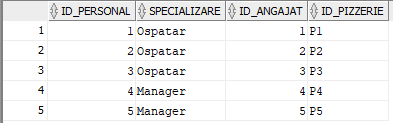
insert into Personal (ID\_Personal, Specializare, ID\_Angajat, ID\_Pizzerie) VALUES(2, 'Ospatar', 2, 'P2');

insert into Personal (ID\_Personal, Specializare, ID\_Angajat, ID\_Pizzerie) VALUES(3, 'Ospatar', 3, 'P3');

insert into Personal (ID\_Personal, Specializare, ID\_Angajat, ID\_Pizzerie) VALUES(4, 'Manager', 4, 'P4');

insert into Personal (ID\_Personal, Specializare, ID\_Angajat, ID\_Pizzerie) VALUES(5, 'Manager', 5, 'P5');

select \* from Personal;



insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(6, 'Diavola', 'P1', 6);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(7, 'Quattro Formaggi', 'P2', 7);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(8, 'Quattro Formaggi', 'P3', 8);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(9, 'Prosciutto Funghi', 'P4', 9);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(10, 'Prosciutto Funghi', 'P5', 10);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(11, 'Diavola', 'P1', 11);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(12, 'Prosciutto Funghi', 'P2', 12);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(13, 'Prosciutto Funghi', 'P3', 13);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(14, 'Diavola', 'P4', 14);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(15, 'Quattro Formaggi', 'P5', 15);

insert into Bucatar (ID\_Bucatar, Specializare, ID\_Pizzerie, ID\_Angajat) VALUES(16, 'Diavola', 'P1', 16);

select \* from Bucatar;



insert into Pizza (ID\_Pizza, Nume, Tip, Pret) VALUES ('PIZ001', 'Margherita', 'Vegetarian', 30);

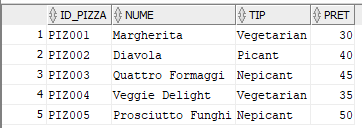
insert into Pizza (ID\_Pizza, Nume, Tip, Pret) VALUES ('PIZ002', 'Diavola', 'Picant', 40);

insert into Pizza (ID\_Pizza, Nume, Tip, Pret) VALUES ('PIZ003', 'Quattro Formaggi', 'Nepicant', 45);

insert into Pizza (ID\_Pizza, Nume, Tip, Pret) VALUES ('PIZ004', 'Veggie Delight', 'Vegetarian', 35);

insert into Pizza (ID\_Pizza, Nume, Tip, Pret) VALUES ('PIZ005', 'Prosciutto Funghi', 'Nepicant', 50);

SELECT \* FROM Pizza;



insert into Ingredient (ID\_Ingredient, Nume, Cantitate) VALUES (seq\_id\_ingredient.NEXTVAL, 'Sos de rosii', 100);

insert into Ingredient (ID\_Ingredient, Nume, Cantitate) VALUES (seq\_id\_ingredient.NEXTVAL, 'Mozzarela', 200);

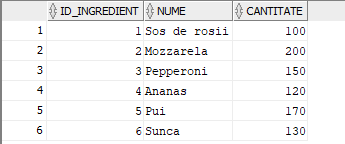
insert into Ingredient (ID\_Ingredient, Nume, Cantitate) VALUES (seq\_id\_ingredient.NEXTVAL, 'Pepperoni', 150);

insert into Ingredient (ID\_Ingredient, Nume, Cantitate) VALUES (seq\_id\_ingredient.NEXTVAL, 'Ananas', 120);

insert into Ingredient (ID\_Ingredient, Nume, Cantitate) VALUES (seq\_id\_ingredient.NEXTVAL, 'Pui', 170);

insert into Ingredient (ID\_Ingredient, Nume, Cantitate) VALUES (seq\_id\_ingredient.NEXTVAL, 'Sunca', 130);

SELECT \* FROM Ingredient;



insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ001', 1);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ001', 2);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ002', 1);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ002', 2);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ002', 3);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ003', 1);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ003', 2);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ003', 4);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ004', 1);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ004', 2);

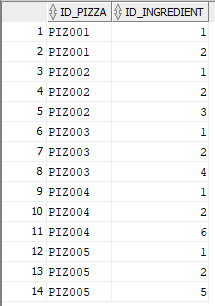
insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ004', 6);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ005', 1);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ005', 2);

insert into Preparare (ID\_Pizza, ID\_Ingredient) VALUES ('PIZ005', 5);

SELECT \* FROM Preparare;



insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (1, 2, 60, TO\_DATE('2024-05-01', 'YYYY-MM-DD'), 'PIZ001', 'P1', 1);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (2, 1, 40, TO\_DATE('2024-05-02', 'YYYY-MM-DD'), 'PIZ002', 'P2', 2);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (3, 3, 135, TO\_DATE('2024-05-03', 'YYYY-MM-DD'), 'PIZ003', 'P3', 3);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (4, 2, 70, TO\_DATE('2024-05-04', 'YYYY-MM-DD'), 'PIZ004', 'P4', 4);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (5, 1, 50, TO\_DATE('2024-05-05', 'YYYY-MM-DD'), 'PIZ005', 'P5', 5);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (6, 2, 60, TO\_DATE('2024-05-01', 'YYYY-MM-DD'), 'PIZ001', 'P1', 5);

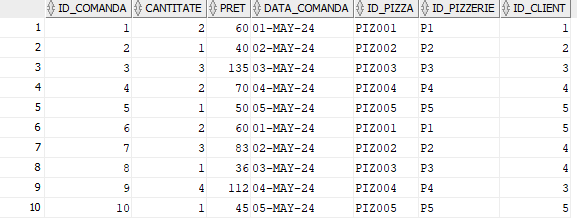
insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (7, 3, 83, TO\_DATE('2024-05-02', 'YYYY-MM-DD'), 'PIZ002', 'P2', 4);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (8, 1, 36, TO\_DATE('2024-05-03', 'YYYY-MM-DD'), 'PIZ003', 'P3', 4);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (9, 4, 112, TO\_DATE('2024-05-04', 'YYYY-MM-DD'), 'PIZ004', 'P4', 3);

insert into Comanda (ID\_Comanda, Cantitate, Pret, Data\_Comanda, ID\_Pizza, ID\_Pizzerie, ID\_Client) VALUES (10, 1, 45, TO\_DATE('2024-05-05', 'YYYY-MM-DD'), 'PIZ005', 'P5', 5);

SELECT \* FROM Comanda;



insert into Rezervare (ID\_Rezervare, Data\_Rezervare, Ora,Numar\_Persoane, ID\_Client, ID\_Pizzerie) VALUES (1, TO\_DATE('2024-06-01', 'YYYY-MM-DD'), 18,4, 1, 'P1');

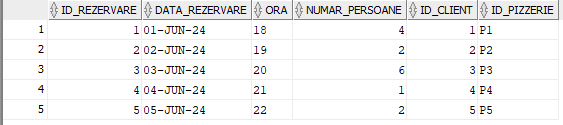
insert into Rezervare (ID\_Rezervare, Data\_Rezervare, Ora,Numar\_Persoane, ID\_Client, ID\_Pizzerie) VALUES (2, TO\_DATE('2024-06-02', 'YYYY-MM-DD'), 19,2, 2, 'P2');

insert into Rezervare (ID\_Rezervare, Data\_Rezervare, Ora,Numar\_Persoane, ID\_Client, ID\_Pizzerie) VALUES (3, TO\_DATE('2024-06-03', 'YYYY-MM-DD'), 20,6, 3, 'P3');

insert into Rezervare (ID\_Rezervare, Data\_Rezervare, Ora,Numar\_Persoane, ID\_Client, ID\_Pizzerie) VALUES (4, TO\_DATE('2024-06-04', 'YYYY-MM-DD'), 21,1, 4, 'P4');

insert into Rezervare (ID\_Rezervare, Data\_Rezervare, Ora,Numar\_Persoane, ID\_Client, ID\_Pizzerie) VALUES (5, TO\_DATE('2024-06-05', 'YYYY-MM-DD'), 22,2, 5, 'P5');

SELECT \* FROM Rezervare;



insert into Promotie (ID\_Promotie, Data\_Start, Data\_Stop, Valoare\_Reducere, ID\_Pizza) VALUES (1, TO\_DATE('2024-06-01', 'YYYY-MM-DD'), TO\_DATE('2024-06-15', 'YYYY-MM-DD'), 20, 'PIZ001');

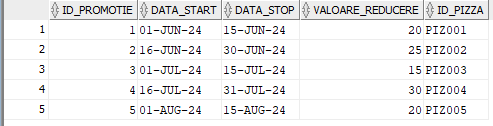
insert into Promotie (ID\_Promotie, Data\_Start, Data\_Stop, Valoare\_Reducere, ID\_Pizza) VALUES (2, TO\_DATE('2024-06-16', 'YYYY-MM-DD'), TO\_DATE('2024-06-30', 'YYYY-MM-DD'), 25, 'PIZ002');

insert into Promotie (ID\_Promotie, Data\_Start, Data\_Stop, Valoare\_Reducere, ID\_Pizza) VALUES (3, TO\_DATE('2024-07-01', 'YYYY-MM-DD'), TO\_DATE('2024-07-15', 'YYYY-MM-DD'), 15, 'PIZ003');

insert into Promotie (ID\_Promotie, Data\_Start, Data\_Stop, Valoare\_Reducere, ID\_Pizza) VALUES (4, TO\_DATE('2024-07-16', 'YYYY-MM-DD'), TO\_DATE('2024-07-31', 'YYYY-MM-DD'), 30, 'PIZ004');

insert into Promotie (ID\_Promotie, Data\_Start, Data\_Stop, Valoare\_Reducere, ID\_Pizza) VALUES (5, TO\_DATE('2024-08-01', 'YYYY-MM-DD'), TO\_DATE('2024-08-15', 'YYYY-MM-DD'), 20, 'PIZ005');

SELECT \* FROM Promotie;



INSERT INTO Recenzie (ID\_Recenzie, Data\_Recenzie, Rating, Comentariu, ID\_Client, ID\_Pizzerie) VALUES (1, TO\_DATE('2024-05-10', 'YYYY-MM-DD'), 5, 'Pizza excelenta!', 1, 'P1');

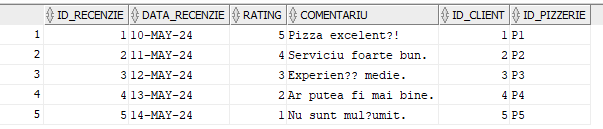
INSERT INTO Recenzie (ID\_Recenzie, Data\_Recenzie, Rating, Comentariu, ID\_Client, ID\_Pizzerie) VALUES (2, TO\_DATE('2024-05-11', 'YYYY-MM-DD'), 4, 'Serviciu foarte bun.', 2, 'P2');

INSERT INTO Recenzie (ID\_Recenzie, Data\_Recenzie, Rating, Comentariu, ID\_Client, ID\_Pizzerie) VALUES (3, TO\_DATE('2024-05-12', 'YYYY-MM-DD'), 3, 'Experien?? medie.', 3, 'P3');

INSERT INTO Recenzie (ID\_Recenzie, Data\_Recenzie, Rating, Comentariu, ID\_Client, ID\_Pizzerie) VALUES (4, TO\_DATE('2024-05-13', 'YYYY-MM-DD'), 2, 'Ar putea fi mai bine.', 4, 'P4');

INSERT INTO Recenzie (ID\_Recenzie, Data\_Recenzie, Rating, Comentariu, ID\_Client, ID\_Pizzerie) VALUES (5, TO\_DATE('2024-05-14', 'YYYY-MM-DD'), 1, 'Nu sunt mul?umit.', 5, 'P5');

SELECT \* FROM Recenzie;



12.Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

* + 1. • subcereri sincronizate în care intervin cel puțin 3 tabele
    2. • subcereri nesincronizate în clauza FROM
    3. • grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)
    4. • ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)
    5. • utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
    6. • utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

**12.1 Exercitiul 1**

**Cerere cu grupari de date, functii grup, filtrare la nivel de grupuri cu subcereri nesincronizate (in clauza having) in care intervin cel putin 3 tabele.**

Sa se afiseze numele clientilor si numarul de pizze diferite pe care le-au comandat. Selecteaza doar clientii care au cumparat un numar de pizze diferite mai mare decat media numarului de pizze diferite cumparate de toti clientii.Rezultatele vor fi ordonate descrescator dupa nr de pizze diferite.

SELECT

C.Nume AS Nume\_Client,

COUNT(DISTINCT P.ID\_Pizza) AS Numar\_Pizze\_Diferite

FROM

Client C

JOIN

Comanda CM ON C.ID\_Client = CM.ID\_Client

JOIN

Pizza P ON CM.ID\_Pizza = P.ID\_Pizza

JOIN

Pizzerie PI ON CM.ID\_Pizzerie = PI.ID\_Pizzerie

GROUP BY

C.Nume

HAVING

COUNT(DISTINCT P.ID\_Pizza) > (

SELECT AVG(Numar\_Pizze)

FROM (

SELECT

CM2.ID\_Client,

COUNT(DISTINCT P2.ID\_Pizza) AS Numar\_Pizze

FROM

Comanda CM2

JOIN

Pizza P2 ON CM2.ID\_Pizza = P2.ID\_Pizza

JOIN

Pizzerie PI2 ON CM2.ID\_Pizzerie = PI2.ID\_Pizzerie

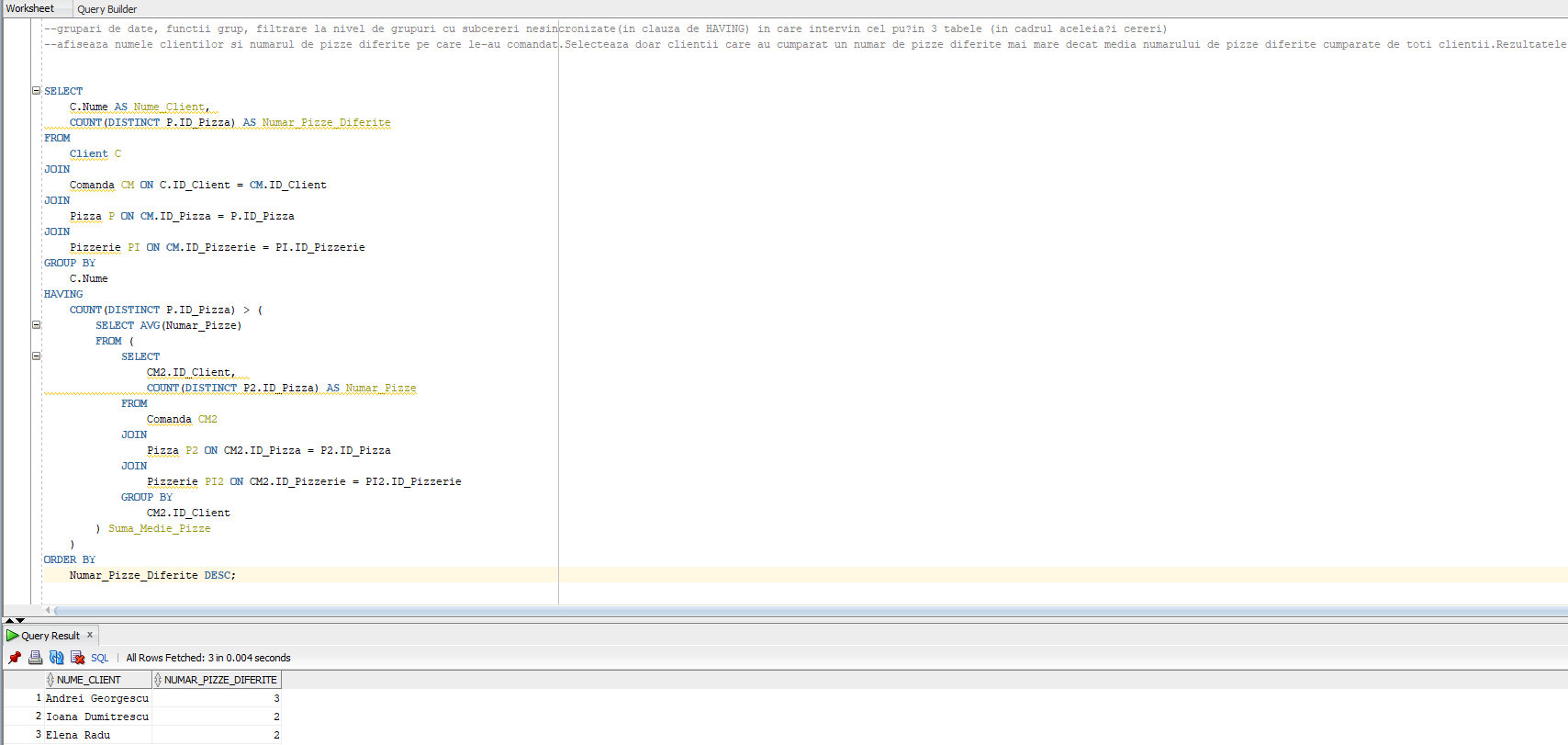
GROUP BY

CM2.ID\_Client

) Suma\_Medie\_Pizze

)

ORDER BY

Numar\_Pizze\_Diferite DESC;

**12.2 Exercitiul 2**

**Subcereri nesincronizate in clauza from, expresia CASE.**

Sa se afiseze numele si email-ul pt fiecare client. Clientii care au comenzi mai mari de 500 de lei devin "Clienti premium", iar ceilalti "Clienti uzuali".

SELECT

C.Nume,

C.Email,

CASE

WHEN NVL(Suma\_Comenzi, 0) > 200 THEN 'Client premium'

ELSE 'Client uzual'

END AS Tip\_Client

FROM

Client C

LEFT JOIN

(

SELECT

ID\_Client,

SUM(Pret) AS Suma\_Comenzi

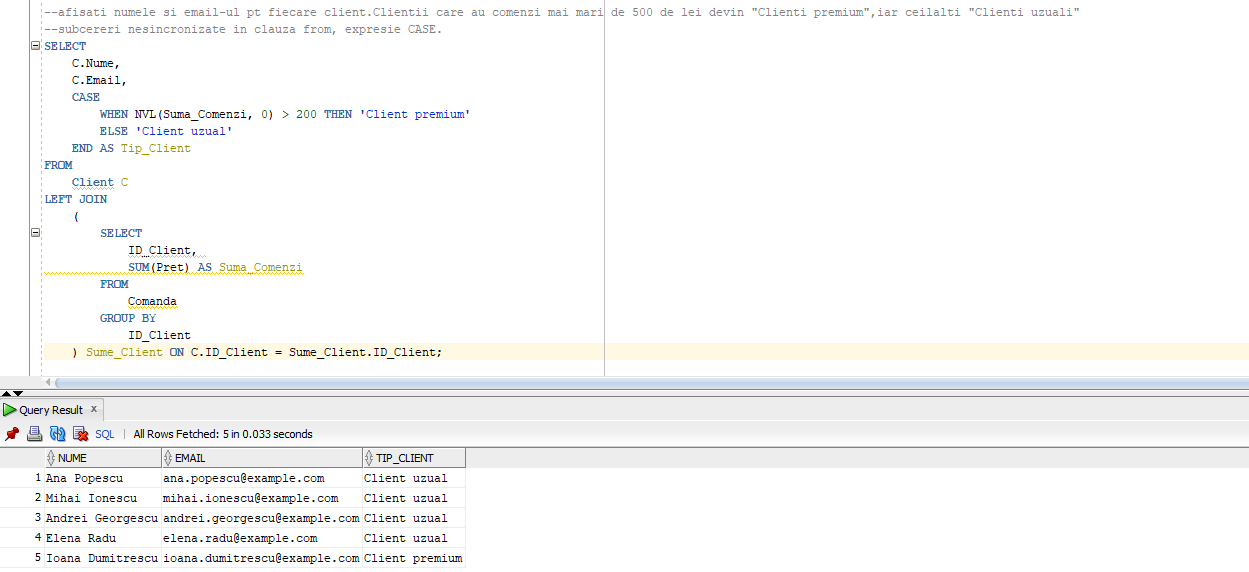
FROM

Comanda

GROUP BY

ID\_Client

) Sume\_Client ON C.ID\_Client = Sume\_Client.ID\_Client;



**12.3 Exercitiul 3**

**NVL, DECODE, ORDER BY**

Sa se afiseze numele , id-ul si numele pizzeriei in care lucreaza , si salariul vechi si majorat stiind ca se majoreaza astfel: 20% pt Pizzeria1,15% pentru Pizzeria2, 10% pentru Pizzeria3, 10 % pentru Pizzeria4 si 20% pentru Pizzeria5. Salariul este majorat angajatilor care s-au angajat dupa 2022 si se afiseaza in ordinea magazinelor in functie de id.

SELECT

A.Nume,

A.ID\_Angajat,

PI.Nume,

A.Salariu AS Salariu\_Vechi,

NVL(

DECODE(

PI.ID\_Pizzerie,

'P1', A.Salariu \* 1.20,

'P2', A.Salariu \* 1.15,

'P3', A.Salariu \* 1.10,

'P4', A.Salariu \* 1.10,

'P5', A.Salariu \* 1.20

),

A.Salariu

) AS Salariu\_Majorat

FROM

Angajat A

INNER JOIN

Personal AM ON A.ID\_Angajat = AM.ID\_Angajat

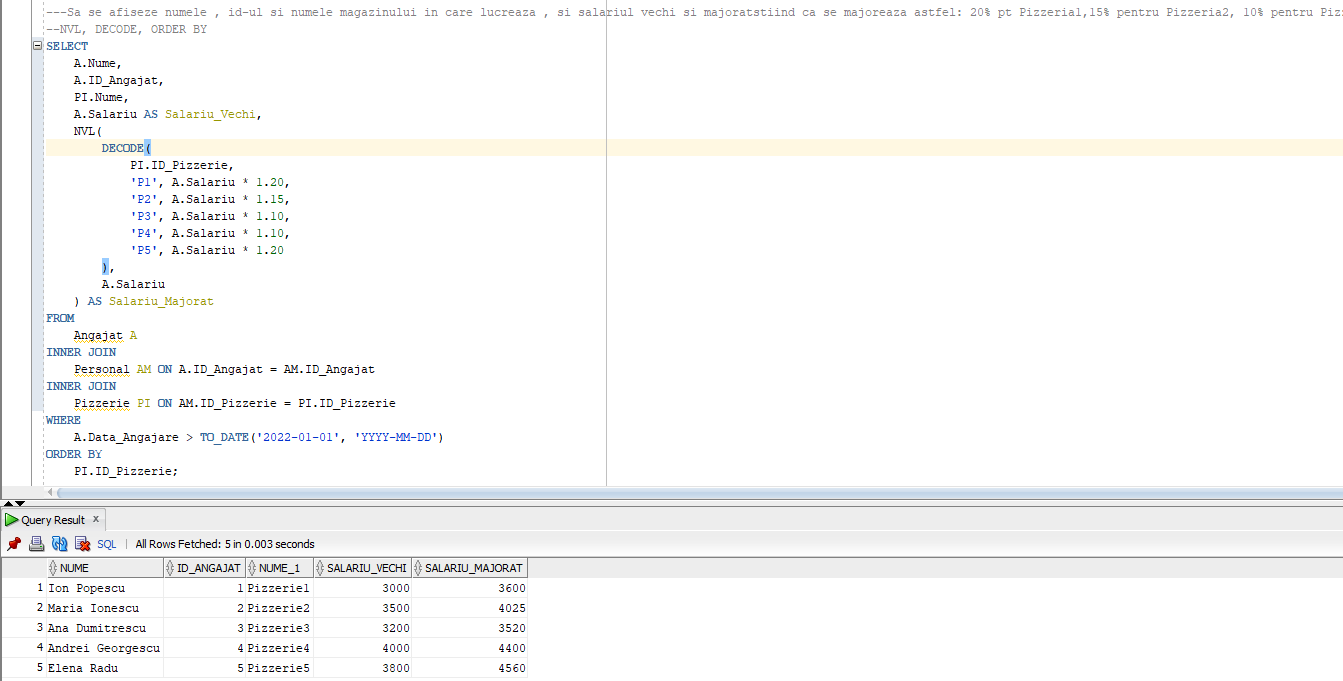
INNER JOIN

Pizzerie PI ON AM.ID\_Pizzerie = PI.ID\_Pizzerie

WHERE

A.Data\_Angajare > TO\_DATE('2022-01-01', 'YYYY-MM-DD')

ORDER BY

PI.ID\_Pizzerie; 

**12.4 Exercitiul 4**

**Subcereri sincronizate in care intervin cel putin 3 tabele, functii pe date calendaristice.**

Pentru toate comenzile plasate in anul 2024 sa se afiseze urmatoarele informatii: numele clientului, numele pizzeriei, data in care a fost efectuata comanda, numele pizzei comandate, respectiv numarul de ani si luni trecuti de la efectuarea comenzii.

SELECT

c.Nume AS NUME\_CLIENT,

p.Nume AS NUME\_PIZZERIE,

TO\_CHAR(o.Data\_Comanda, 'DD MONTH YY') AS DATA\_COMANDA,

pizza.Nume AS NUME\_PIZZA,

TRUNC(MONTHS\_BETWEEN(SYSDATE, o.Data\_Comanda) / 12) AS ANI\_TRECUTI,

MOD(TRUNC(MONTHS\_BETWEEN(SYSDATE, o.Data\_Comanda)), 12) AS LUNI\_TRECUTI

FROM

Comanda o

JOIN Client c ON o.ID\_Client = c.ID\_Client

JOIN Pizzerie p ON o.ID\_Pizzerie = p.ID\_Pizzerie

JOIN Pizza pizza ON o.ID\_Pizza = pizza.ID\_Pizza

WHERE

o.ID\_Client IN (

SELECT

o2.ID\_Client

FROM

Comanda o2

JOIN Pizza pizza2 ON o2.ID\_Pizza = pizza2.ID\_Pizza

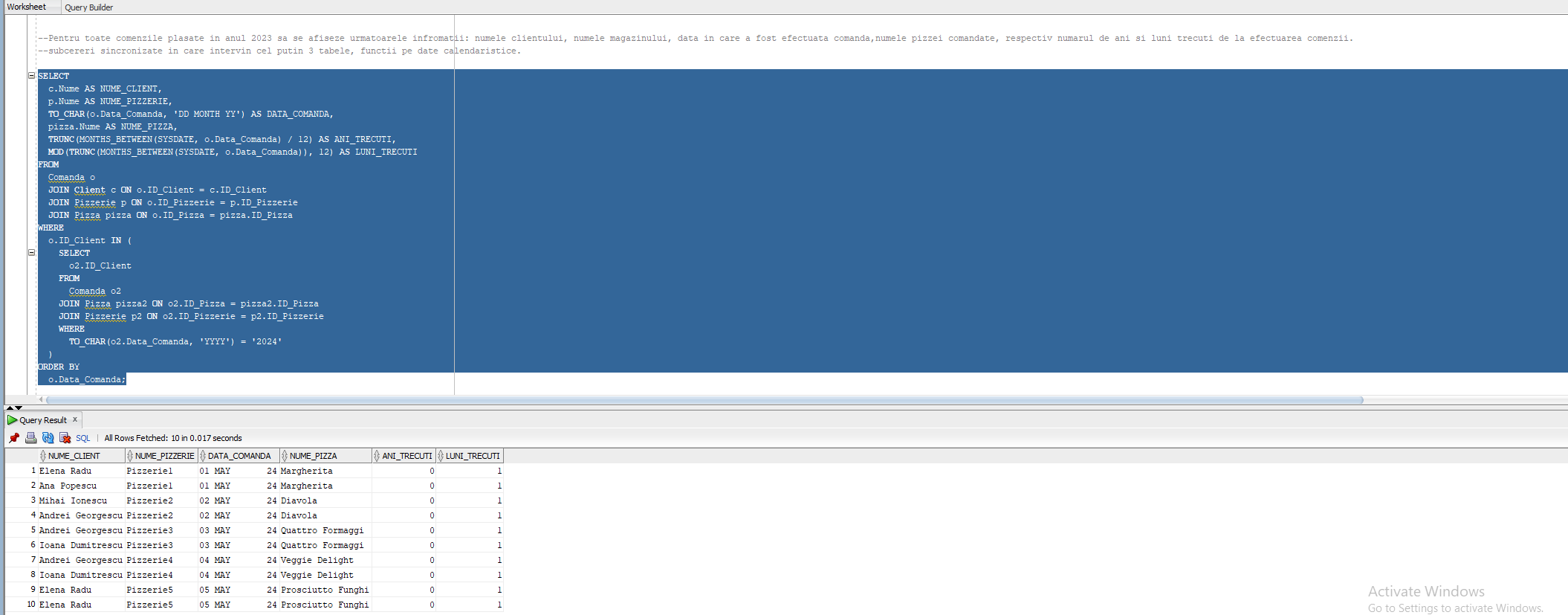
JOIN Pizzerie p2 ON o2.ID\_Pizzerie = p2.ID\_Pizzerie

WHERE

TO\_CHAR(o2.Data\_Comanda, 'YYYY') = '2024'

)

ORDER BY

o.Data\_Comanda;

**12.5 Exercitiul 5**

**Clauza WITH, functii pe siruri de caractere**

Sa se afiseze numele angajatilor si numele pizzeriilor care au cel putin o comanda plasata de un client cu un email care se termina cu ".com", si care au produs cel putin o pizza de tip vegetarian.

WITH Comenzi\_Email AS (

SELECT DISTINCT Co.ID\_Pizzerie

FROM Comanda Co

JOIN Client Cl ON Co.ID\_Client = Cl.ID\_Client

WHERE Cl.Email LIKE '%.com'

),

Pizza\_Calda AS (

SELECT DISTINCT P.ID\_Pizza

FROM Pizza P

WHERE LOWER(P.Tip) = 'vegetarian'

)

SELECT DISTINCT A.Nume, Pz.Nume AS Nume\_Pizzerie

FROM Angajat A

JOIN Personal AP ON A.ID\_Angajat = AP.ID\_Angajat

JOIN Pizzerie Pz ON AP.ID\_Pizzerie = Pz.ID\_Pizzerie

JOIN Comenzi\_Email CE ON Pz.ID\_Pizzerie = CE.ID\_Pizzerie

JOIN Comanda Co ON Pz.ID\_Pizzerie = Co.ID\_Pizzerie

JOIN Pizza\_Calda PC ON Co.ID\_Pizza = PC.ID\_Pizza;



13.Implmentarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

1.Aplicati o reducere de 20% la toate comenzile care sunt plasate de un client care a plasat 2 sau mai multe comenzi.

update Comanda

set Pret=Pret-Pret/10

where ID\_Client in

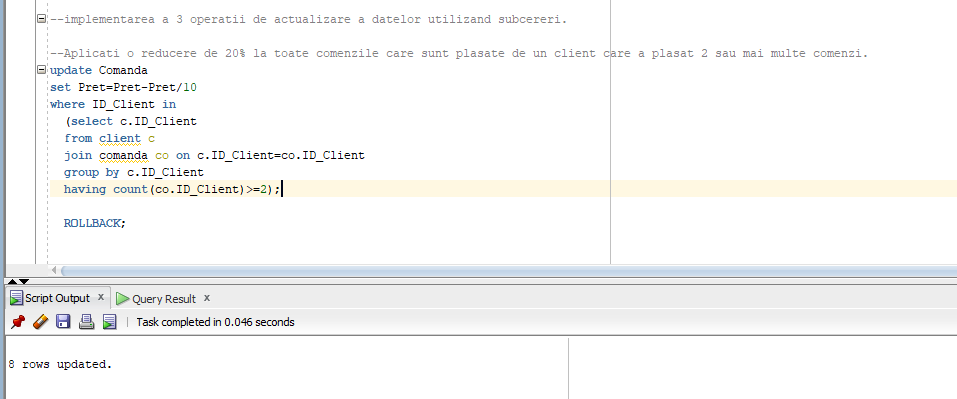
(select c.ID\_Client

from client c

join comanda co on c.ID\_Client=co.ID\_Client

group by c.ID\_Client

having count(co.ID\_Client)>=2);



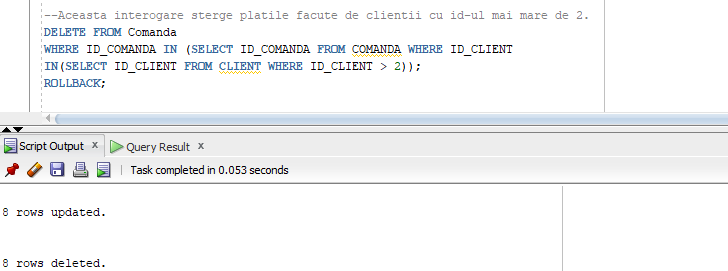
2. Sa se stearga platile facute de clientii cu id-ul mai mare de 2.

DELETE FROM Comanda

WHERE ID\_COMANDA IN (SELECT ID\_COMANDA FROM COMANDA WHERE ID\_CLIENT

IN(SELECT ID\_CLIENT FROM CLIENT WHERE ID\_CLIENT > 2));

ROLLBACK;



3.Sa se mareasca salariul angajatilor de la Pizzeria2 2 cu 20% .

UPDATE Angajat

SET Salariu = Salariu \* 1.2

WHERE ID\_Angajat IN (

SELECT AM.ID\_Angajat

FROM Personal AM

INNER JOIN Pizzerie PI ON AM.ID\_Pizzerie = PI.ID\_Pizzerie

WHERE PI.Nume = 'Pizzerie2'

);

