Proiect - Baze de Date

- Gestiunea unui lanț de restaurante -

Popescu Paullo Robertto Karloss

Grupa 131

2021 Mai

Cuprins:

1. Prezentarea bazei de date.
   1. Tehnologii folosite pentru realizarea proiectului.
   2. Descrierea temei alese.
   3. Prezentarea constrângerilor impuse asupra modelului.
   4. Descrierea entităților.
   5. Descrierea relațiilor.
   6. Descrierea atributelor.
2. Diagrama Entitate-Relație (ER).
3. Diagrama Conceptuală.
4. Schemele relaționale corespunzătoare diagramei conceptuale.
5. Normalizarea până la forma normal 3 (FN1-FN3).
6. Implementarea bazei de date în Oracle.
   1. Crearea tabelelor și a constrângerilor.
   2. Inserarea datelor coerente în tabele (minium 5 înregistrări în fiecare tabel neasociativ, minimum 10 înregistrări în tabelele asociative) + crearea unei secvențe.
   3. Diagrama generată în sql după crearea tabelelor și inserarea datelor.
7. Crearea a 5 cereri complexe în SQL.
8. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.
9. O cerere care utilizează operația *outer-join* pe minium 4 tabele și două cereri ce utilizează operația *division*.
10. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.
11. a) Realizarea normalizării BCNF, FN4, FN5.
12. Aplicarea denormalizării.

**1. Prezentarea bazei de date**

* 1. ***Tehnologii folosite pentru realizarea proiectului.***

Pentru proiectul din cadrul cursului de Baze de Date am folosit versiunea **o19c** a *Oracle Database*.

Ce aplicații am folosit?

* Oracle SQL Developer
  1. ***Descrierea temei alese. Pentru ce ar fi folosită***

În cadrul acestui proiect, am ales ca temă ***Gestiunea unui lanț de restaurante***. Scopul ei este de a face mai usoară ținerea în evidență a mai multor restaurante (comenzi, angajați, facturi etc).

* 1. ***Prezentarea constrângerilor impuse asupra modelului.***

Un angajat trebuie să lucreze la un singur restaurant.

Un restaurant poate avea mai mulți angajați.

Angajații pot fi chelneri, casieri, bucătari sau manageri.

Într-o locație se poate găsii un singur restaurant.

Produsele sunt făcute cu ingrediente care sunt fabricate de un producător.

Într-o factură se poate găsii o singură comandă.

O comandă conține unul sau mai multe produse.

Un produs poate să aparțină mai multor restaurante.

Produsele pot fi preparate de un bucătar, dar există și produse care nu trebuiesc preparate (de exemplu vin, șampanie etc).

La o factură se pot preciza detalii, cum ar fi: dăcă a fost achitată cash sau cu ajutorul unui card de credit

* 1. ***Descrierea entităților.***

Cum am precizat mai sus, ideea proiectului este *Gestionarea unui lanț de restaurante*, de la care clienții pot comanda diferite produse. Acesta cuprinde următoarele entități:

* Entitatea **Restaurant**, care va conține numele restaurantului. Cheia primară fiind *id-ul restaurantului*.
* Entitatea **Locație,** care va conține numele țării unde se află restaurantul, orașul, respectiv codul postal și strada. Cheia primară fiind *id-ul locației*.
* Entitatea **Produs**, care va conține numele fiecărui produs, cantitatea și o scurtă descriere a sa. Cheia primară fiind *id-ul produsului*.
* Entitatea **Ingredient**, care va conține numele ingredientului. Cheia primară fiind *id-ul ingredientului*.
* Entitatea **Producător**, care va conține numele și numărul de telefon al unui producător de ingrediente. Cheia primară fiind *id-ul producătorului*.
* Entitatea **Comandă**, care va conține prețul și data pentru fiecare comandă plasată. Cheia primară fiind *id-ul comenzii*.
* Entitatea **Angajat**, care va conține numele, prenumele și data angajării. Cheia primară fiind id-ul angajatului.
* Subentitatea **Bucătar**, care va conține numărul de stele al fiecărui bucătar. Cheia primară fiind id-ul angajatului.
* Subentitatea **Casier**, care va conține numărul de ani de studii al fiecărui casier. Cheie primară fiind id-ul angajatului.
* Subentitatea **Chelner**, care va conține numărul numărul de ani de experiență al fiecărui chelner. Cheie primară fiind id-ul angajatului.
* Subentitatea **Manager**, care nu are atribute, are doar cheia primară id-ul angajatului.
* Entitatea **Client**, care va conține numele, prenumele și numărul personal de telefon, întrucât acestea sunt necesare atunci când se plasează o comandă. Cheia primară fiind *id-ul clientului*.
* Entitatea **Factură**, care va conține valoarea pe care clientul este nevoit să o plătească pentru comanda sa, dar și câteva detalii cum ar fi dacă a fost achitată cash sau cu ajutorul unui card de credit. Cheia primară fiind *id-ul facturii*.
  1. ***Descrierea relațiilor.***

Un anumit produs poate aparține mai multor restaurante.

Într-o locație poate fi găsit un singur restaurant.

Un produs se poate regăsii în mai multe comenzi.

Într-o comandă se pot găsii mai multe produse, dar cel puțin unul.

Un produs poate fi preparat de mai mulți bucătari.

Un ingredient trebuie să fie produs de un producător.

Produsele trebuie să conțină măcar un ingredient.

Un bucătar poate să prepare mai multe produse, dar trebuie să prepare minim un produs.

Un client poate plasa una sau mai multe comenzi (devine client după ce plasează minim o comandă).

O factură poate conține o singură comandă.

La o factură este atașat un singur casier, dar un casier poate fi atașat la mai multe facturi, nu doar una.

Un angajat trebuie să fie fie chelner, casier, bucătar sau manager.

* 1. ***Descrierea atributelor.***
* Entitatea **restaurant** va avea ca atribute id-ul și numele restaurantului curent (de exemplu “Yamas”, “Ivans” etc).

Tipul de date pentru id-ul restaurantului va fi *number(10)*, id-locație *number(10)*, iar pentru nume va fi *varchar2(32)*.

Cheia externă va fi id\_locație (provenită din tabelul locație), cu tipul de date *number(10).*

PK-ul va fi id-ul restaurantului (cheia primară).

Constrângere pentru numele restaurantului: Unique.

* Entitatea **locație** va avea ca atribute: id-ul locației, țara, orașul, codul poștal și strada pentru restaurantul respectiv.

Tipul de date pentru id-ul locației va fi *number(10),* țara *varchar2(20)*, orașul *varchar2(20)*, codul poștal *varchar2(15)* și strada *varchar2(50)*.

PK-ul va fi id-ul locației (cheia primară).

Constrângere pentru codul poștal: Unique.

* Entitatea **produs** va avea ca atribute: id-ul produsului, numele, cantitatea și o scurtă descriere pentru produsul comercializat.

Tipul de date pentru id-ul produsului va fi *number(10)*, nume *varchar2(25)*, gramaj *number(10)* și descriere *varchar2(100)*.

PK-ul va fi id-ul produsului (cheia primară).

Constrângere pentru gramajul produsului: check gramaj > 0.

* Tabelul asociativ **meniu** va avea ca atribute: id-ul produsului, id-ul restaurantului și prețul. Avem o cheie primară compusă formată din id-ul produsului (FK din tabelul produs) și id-ul restaurantului (FK din tabelulul restaurant).

Tipul de date pentru id-ul produsului va fi *number(10)*, id-ul restaurantului *number(10)* și prețul *number(10)*.

Constrângere pentru prețul meniului: check preț > 0.

* Tabelul asociativ **cantitate**\_**produs** va avea ca atribute: id-ul produsului, id-ul ingredientului și cantitatea.

Avem o cheie primară compusă formată din id-ul produsului (FK din tabelul produs) și id-ul ingredient (FK din tabelul ingredient).

Tipul de date pentru id-ul produsului va fi *number(10)*, id-ul ingredientului *number(12)* și cantitatea *number(10)*.

Constrângere pentru cantitate: check cantitate >= 0.

* Entitatea **ingredient** va avea ca atribute: id-ul și numele ingredientului.

Ingredient va avea ca cheie externă id-producător (provenită din tabelul producător).

Tipul de date pentru id-ul ingredientului va fi *number(12)*, id-ul producătorului *number(10)*, iar pentru numele ingredientului *varchar2(32)*.

PK-ul va fi id-ul ingredientului (cheia primară).

* Entitatea **producător** va avea ca atribute: id-ul, numele și numărul de telefon al producătorului respectiv.

Tipul de date pentru id va fi *number(10)*, numele *varchar2(32)* și numărul de telefon *varchar2(15)*.

PK-ul va fi id-ul producătorului (cheia primară).

Constrângere pentru numărul de telefon: Unique.

* Entitatea **comandă** va avea ca atribute: id-ul, prețul și data unei comenzi plasate de client, și va avea ca cheie externă id-ul clientului (provenită din tabelul client), dar și id-ul facturii (provenită din tabelul factură).

Tipul de date pentru id-ul comenzii va fi *number(20)*, id-ul clientului *number(10)*, id-ul facturii *number(15)*, data *date*,iar prețul *number(10)*.

PK-ul va fi id-ul comenzii (cheia primară).

Constrângere pentru prețul comenzii: NOT\_NULL, dar și check preț > 0.

Atributul data va avea setat ca default ziua curentă (sysdate).

* Tabelul asociativ **conținut**\_**comandă** va avea ca atribute: id-ul produsului, id-ul comenzii și numărul de produse.

Avem o cheie primară compusă formată din id-ul produsului (FK din tabelul produs) și id-ul comenzii (FK din comandă).

Tipul de date pentru id-ul produsului va fi *number(10)*, id-ul comenzii *number(20)* și numărul de produse din comandă *number(10)*.

Constrângere pentru numărul de produse: check număr\_produse > 0.

* Entitatea **angajat** va avea ca atribute: id-ul, numele, prenumele și data angajării(am ales această abordare pentru că niciun subtip nu avea atribute separate de angajat).

Angajat va avea ca cheie externă id-restaurant (provenită din tabelul restaurant).

Tipul de date pentru id-ul angajatului va fi *number(10)*, id-ul restaurantului *number(10)*, nume *varchar2(32)*, prenume *varchar2(32)*, iar data angajării *date* .

Atributul data\_angajare va avea setat ca default ziua curentă (sysdate).

* Subentitatea **bucătar** va avea ca atribute id-ul angajatului și numărul de stele.

Tipul de date pentru id-ul angajatului va fi *number(10)*, iar pentru numărul de stele *number(7)*.

PK va fi id\_angajat (care provine din tabelul angajat).

* Subentitatea **casier** va avea ca atribute id-ul angajatului și numărul de ani de studii.

Tipul de date pentru id-ul angajatului va fi *number(10)*, iar pentru numărul de ani de studii *number(5)*.

PK va fi id\_angajat (care provine din tabelul angajat).

* Subentitatea **chelner** va avea ca atribute id-ul angajatului și numărul de ani de experiență.

Tipul de date pentru id-ul angajatului va fi *number(10)*, iar pentru numărul de ani de experiență *number(5)*.

PK va fi id\_angajat (care provine din tabelul angajat).

* Subentitatea **manager** va avea ca atribut id-ul angajatulul.

Tipul de date pentru id-ul angajatului va fi *number(10)*.

PK va fi id\_angajat (care provine din tabelul angajat).

* Tabelul asociativ **preparare** va avea ca atribute: id-ul produsului, id-ul comenzii, id-ul bucătarului și durata.

Avem o cheie primară compusă formată din id-ul produsului (FK din tabelul produs), id-ul comenzii (FK din comandă) și id-ul bucătarului (FK din angajat), acestea reies din relația de tip 3 dintre produs, comandă și bucătar.

Tipul de date pentru id-ul produsului va fi *number(10)*, id-ul comenzii *number(20)*, id-ul bucătarului *number(10)* și durata *number(10)*.

* Entitatea **client** va avea ca atribute: id-ul, numele, prenumele și numărul de telefon al unui client.

Tipul de date pentru id-ul clientului va fi *number(10)*, numele *varchar2(32)*, prenumele *varchar2(32)* și numărul de telefon *varchar2(15)*.

Constrângere pentru numele și prenumele: NOT\_NULL.

PK-ul va fi id-ul clientului (cheia primară).

* Entitatea **factură** va avea ca atribute: id-ul, valoarea și câteva mici detalii legate de factură.

Constrângere pentru valoare : NOT\_NULL. Va avea cheia externă id-ul id-ul casierului.

Tipul de date pentru id-ul facturii va fi *number(15)*, valoarea *number(10)*, detalii *varchar2(50)* și id-ul casierului *number(10)*.

Constrângere pentru valoarea facturii: check valoare > 0.

PK-ul va fi id-ul facturii (cheia primară).

**2. Diagrama Entitate-Relație (ER):**

1

furnizează

PRODUCĂTOR

INGREDIENT

se\_află

LOCAȚIE

1

M(0)

M(1)

are

1

M(0)

conține

COMANDĂ

are

PRODUS

RESTAURANT

M(1)

1

M(1)

M(0)

1

M(1)

M(1)

M(1)

M(0)

atașat\_la

FACTURĂ

CLIENT

lucrează

ISA

ISA

ISA

ISA

1

1

1

1

1(0)

1(0)

1(0)

1(0)

CASIER

MANAGER

ANGAJAT

prepară

are

cere

M(1)

M(0)

BUCĂTAR

1

1

1

M(0)

CHELNER

PRODUCĂTOR

#ID-PRODUCĂTOR

Nume

Nr\_telefon

**3. Diagrama Conceptuală:**

INGREDIENT

#ID-INGREDIENT

Nume



LOCAȚIE

#ID-LOCAȚIE

Țara

Oraș

Cod\_Poștal

Strada



CANTITATE\_PRODUS

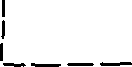
#ID-PRODUS

#ID-INGREGDIENT

Cantitate



se\_află



RESTAURANT

#ID-RESTAURANT

Nume



COMANDĂ

#ID-COMANDĂ

Preț

Data



CONȚINUT\_COMANDĂ

#ID-PRODUS

#ID-COMANDĂ

Număr\_Produse

PRODUS

#ID-PRODUS

Nume

Gramaj

Descriere

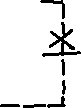


MENIU

#ID-RESTAURANT

#ID-PRODUS

Preț



PREPARARE

#ID-PRODUS

#ID-COMANDĂ

#ID-BUCĂTAR

Durată



ANGAJAT

#ID-ANGAJAT

Nume

Prenume

Data\_Angajare



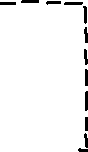
CLIENT

#ID-CLIENT

Nume

Prenume

Nr\_telefon



BUCĂTAR

nr\_stele



CASIER

ani\_studiu

FACTURĂ

#ID-FACTURĂ

Valoare

Detalii



CHELNER

ani\_experiență



MANAGER



**4. Schema relațională:**

* RESTAURANT (#ID-RESTAURANT, id-locație, Nume)
* LOCAȚIE (#ID-LOCAȚIE, Țara, Oraș, Cod\_Poștal, Strada)
* MENIU (#ID-RESTAURANT, #ID-PRODUS, Preț)
* PRODUS (#ID-PRODUS, Nume, Gramaj, Descriere)
* CANTITATE\_PRODUS (#ID-PRODUS, #ID-INGREDIENT, Cantitate)
* INGREDIENT (#ID-INGREDIENT, id-producător, Nume)
* PRODUCĂTOR (#ID-PRODUCĂTOR, Nume, Nr\_telefon)
* CONȚINUT\_COMANDĂ (#ID-PRODUS, #ID-COMANDĂ, Număr\_Produse)
* COMANDĂ (#ID-COMANDĂ, Preț, Data, id-factură, id-client)
* CLIENT (#ID-CLIENT, Nume, Prenume, Nr\_telefon)
* ANGAJAT (#ID-ANGAJAT, id-restaurant, Nume, Prenume, data\_angajare)
* BUCĂTAR (#ID-ANGAJAT, nr\_stele)
* CASIER (#ID-ANGAJAT, ani\_studiu)
* CHELNER (#ID-ANGAJAT, ani\_experiență)
* MANAGER (#ID-ANGAJAT)
* FACTURĂ (#ID-FACTURĂ, Valoare, Detalii, id-casier)
* PREPARARE (#ID-PRODUS, #ID-COMANDĂ, #ID-BUCĂTAR, Durată)

**5. Normalizarea până la forma normal 3 (FN1 – FN3):**

Schema noastră ar fi scoasă din FN1 dacă entitatea produs ar conține atributul ingrediente, întrucât un produs are mai multe ingrediente (de exemplu piper,boia, sare etc).

**Schema Non-FN1:**

**PRODUS** (#ID-PRODUS, Nume, Gramaj, Descriere, ***Ingrediente***)

Schema noastră ar fi scoasă din FN2 dacă în loc să existe două entități ingredient și producător ar exista doar entitatea Ingredient (#id-ingredient, #id-producător, nume\_ingredient, nume\_producător, nr\_telefon\_producător), pentru ca nr de telefon nu depinde de id-ul ingredientului.

**Schema Non-FN2:**

**INGREDIENT** (#ID-INGREDIENT, #ID-PRODUCĂTOR, Nume\_ingredient, Nume\_Producător, ***Nr\_telefon***)

Schema noastră ar fi scoasă din FN3 dacă entitatea comandă ar conține ca atribut numărul de telefon al clientului (dependența tranzitivă între oraș-țara).

**Schema Non-FN3:**

**COMANDĂ** (#ID-COMANDĂ, Preț, Data, id\_factură, id\_client, ***nr\_telefon\_client***)

**6. Implementarea bazei de date în Oracle:**

***6.1 Crearea tabelelor și a constrângerilor.***

--- CREAREA TABELELOR ---

-- LOCATIE --

CREATE TABLE locatie (

id\_locatie NUMBER(10) PRIMARY KEY,

tara VARCHAR2(20),

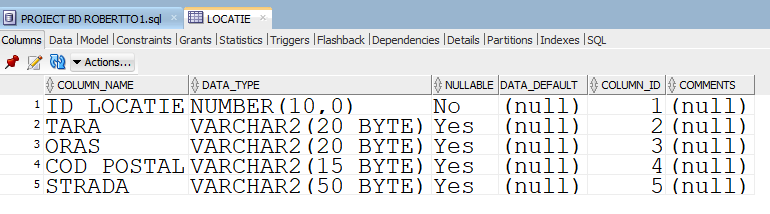
oras VARCHAR2(20),

cod\_postal VARCHAR2(15) UNIQUE,

strada VARCHAR2(50)

);

***Print-Screen:***



-- RESTAURANT --

CREATE TABLE restaurant (

id\_restaurant NUMBER(10) PRIMARY KEY,

id\_locatie NUMBER(10),

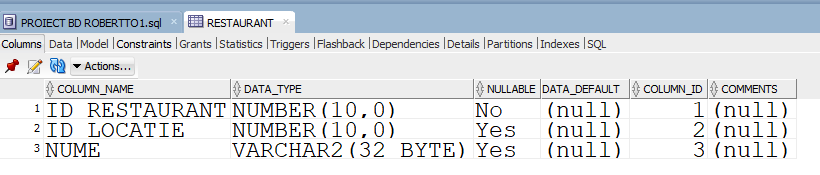
nume VARCHAR2(32) UNIQUE,

CONSTRAINT fk\_restaurant\_locatie FOREIGN KEY ( id\_locatie )

REFERENCES locatie ( id\_locatie )

);

***Print-Screen:***



-- PRODUS --

CREATE TABLE produs (

id\_produs NUMBER(10) PRIMARY KEY,

nume VARCHAR2(25),

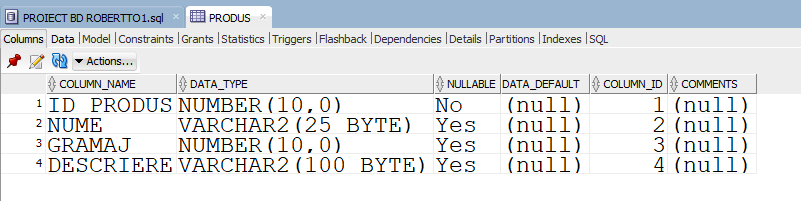
gramaj NUMBER(10),

descriere VARCHAR2(100),

CONSTRAINT chk\_gramaj CHECK ( gramaj > 0 )

);

***Print-Screen:***



-- MENIU --

CREATE TABLE meniu (

id\_produs NUMBER(10),

id\_restaurant NUMBER(10),

pret NUMBER(10),

CONSTRAINT meniu\_pk PRIMARY KEY ( id\_produs,

id\_restaurant ),

CONSTRAINT meniu\_produs\_fk FOREIGN KEY ( id\_produs )

REFERENCES produs ( id\_produs ),

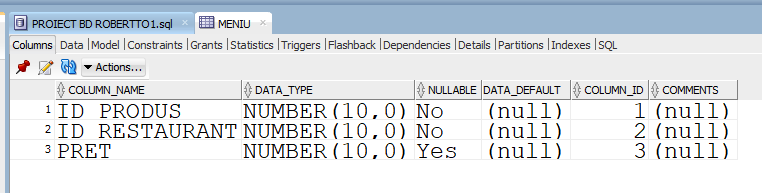
CONSTRAINT meniu\_restaurant\_fk FOREIGN KEY ( id\_restaurant )

REFERENCES restaurant ( id\_restaurant ),

CONSTRAINT chk\_pret CHECK ( pret > 0 )

);

***Print-Screen:***



-- PRODUCATOR --

CREATE TABLE producator (

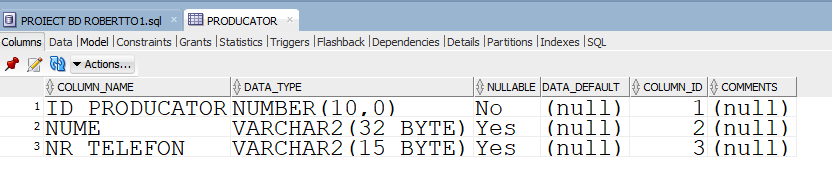
id\_producator NUMBER(10) PRIMARY KEY,

nume VARCHAR2(32),

nr\_telefon VARCHAR2(15) UNIQUE

);

***Print-Screen:***



-- INGREDIENT --

CREATE TABLE ingredient (

id\_ingredient NUMBER(12) PRIMARY KEY,

id\_producator NUMBER(10),

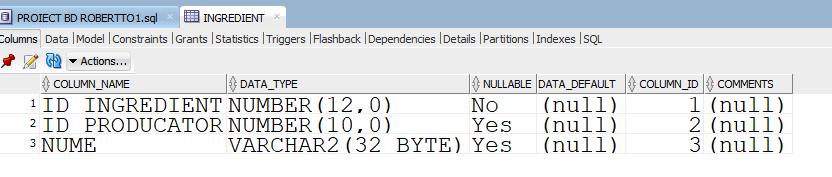
nume VARCHAR2(32),

CONSTRAINT fk\_ingredient\_producator FOREIGN KEY ( id\_producator )

REFERENCES producator ( id\_producator )

);

***Print-Screen:***



-- CANTITATE\_PRODUS --

CREATE TABLE cantitate\_produs (

id\_produs NUMBER(10),

id\_ingredient NUMBER(12),

cantitate NUMBER(10),

CONSTRAINT cantitate\_produs\_pk PRIMARY KEY ( id\_produs,

id\_ingredient ),

CONSTRAINT cantitate\_produs\_fk FOREIGN KEY ( id\_produs )

REFERENCES produs ( id\_produs ),

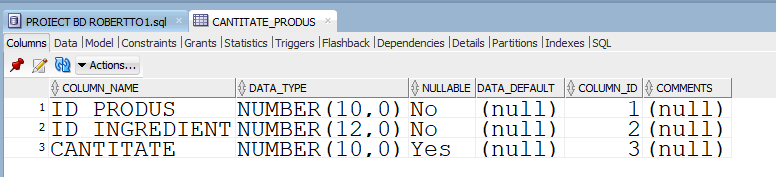
CONSTRAINT cantitate\_ingredient\_fk FOREIGN KEY ( id\_ingredient )

REFERENCES ingredient ( id\_ingredient ),

CONSTRAINT chk\_cantitate\_produs CHECK ( cantitate > 0 )

);

***Print-Screen:***



-- ANGAJAT --

CREATE TABLE angajat (

id\_angajat NUMBER(10) PRIMARY KEY,

id\_restaurant NUMBER(10),

nume VARCHAR2(32),

prenume VARCHAR2(32),

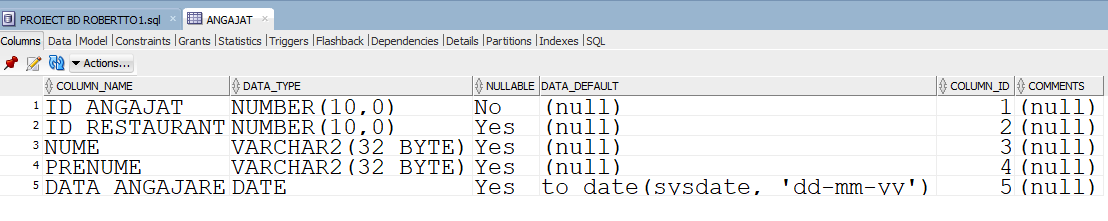
data\_angajare DATE DEFAULT to\_date(sysdate, 'dd-mm-yy'),

CONSTRAINT angajat\_restaurant\_fk FOREIGN KEY ( id\_restaurant )

REFERENCES restaurant ( id\_restaurant )

);

***Print-Screen:***



-- CHELNER --

CREATE TABLE chelner (

id\_angajat NUMBER(10),

ani\_experienta NUMBER(5),

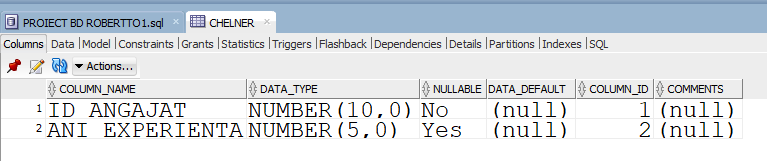
CONSTRAINT chelner\_pk PRIMARY KEY ( id\_angajat ),

CONSTRAINT angajat\_fk FOREIGN KEY ( id\_angajat )

REFERENCES angajat ( id\_angajat )

);

***Print-Screen:***



-- CASIER --

CREATE TABLE casier (

id\_angajat NUMBER(10),

ani\_studiu NUMBER(5),

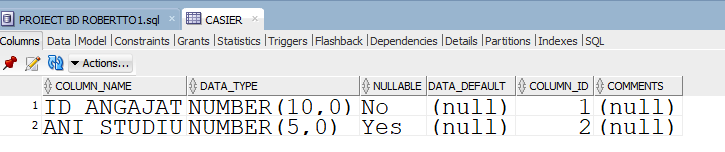
CONSTRAINT casier\_pk PRIMARY KEY ( id\_angajat ),

CONSTRAINT angajat\_casier\_fk FOREIGN KEY ( id\_angajat )

REFERENCES angajat ( id\_angajat )

);

***Print-Screen:***



-- BUCATAR --

CREATE TABLE bucatar (

id\_angajat NUMBER(10),

nr\_stele NUMBER(7),

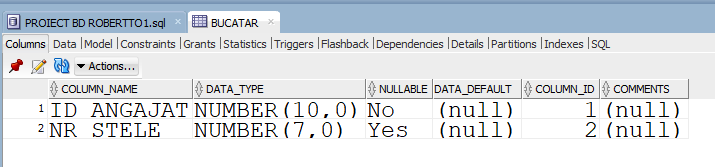
CONSTRAINT bucatar\_pk PRIMARY KEY ( id\_angajat ),

CONSTRAINT angajat\_bucatar\_fk FOREIGN KEY ( id\_angajat )

REFERENCES angajat ( id\_angajat )

)

***Print-Screen:***



-- MANAGER --

CREATE TABLE manager (

id\_angajat NUMBER(10),

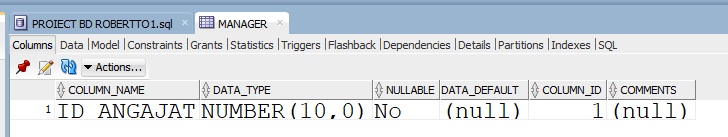
CONSTRAINT manager\_pk PRIMARY KEY ( id\_angajat ),

CONSTRAINT angajat\_manager\_fk FOREIGN KEY ( id\_angajat )

REFERENCES angajat ( id\_angajat )

);

***Print-Screen:***



-- FACTURA --

CREATE TABLE factura (

id\_factura NUMBER(15) PRIMARY KEY,

id\_angajat NUMBER(10),

valoare NUMBER(10),

detalii VARCHAR2(50),

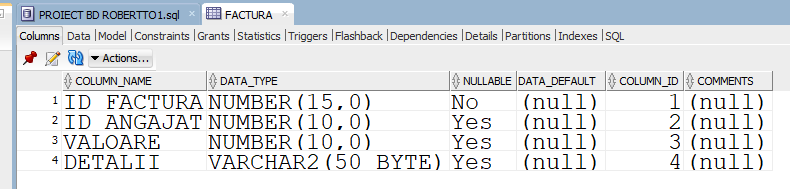
CONSTRAINT fk\_factura\_casier FOREIGN KEY ( id\_angajat )

REFERENCES casier ( id\_angajat ),

CONSTRAINT chk\_factura\_valoare CHECK ( valoare > 0 )

);

***Print-Screen:***



-- CLIENT --

CREATE TABLE client (

id\_client NUMBER(10) PRIMARY KEY,

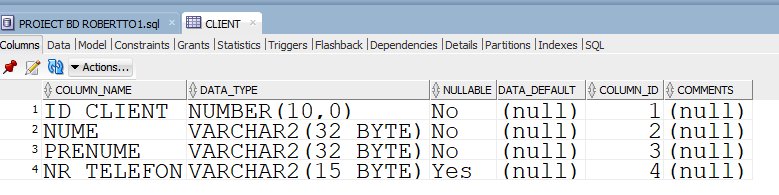
nume VARCHAR2(32) NOT NULL,

prenume VARCHAR2(32) NOT NULL,

nr\_telefon VARCHAR2(15)

);

***Print-Screen:***



-- COMANDA --

CREATE TABLE comanda (

id\_comanda NUMBER(20) PRIMARY KEY,

id\_client NUMBER(10),

id\_factura NUMBER(15),

pret NUMBER(10) NOT NULL,

data DATE DEFAULT to\_date(sysdate, 'dd-mm-yy'),

CONSTRAINT fk\_comanda\_client FOREIGN KEY ( id\_client )

REFERENCES client ( id\_client ),

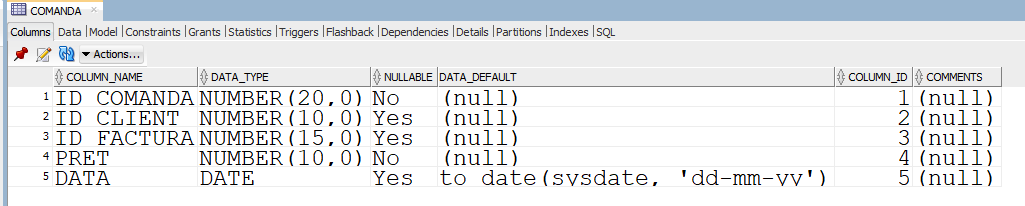
CONSTRAINT fk\_comanda\_factura FOREIGN KEY ( id\_factura )

REFERENCES factura ( id\_factura ),

CONSTRAINT chk\_comanda CHECK ( pret > 0 )

);

***Print-Screen:***



-- CONTINUT\_COMANDA --

CREATE TABLE continut\_comanda (

id\_produs NUMBER(10),

id\_comanda NUMBER(20),

numar\_produse NUMBER(10),

CONSTRAINT continut\_comanda\_pk PRIMARY KEY ( id\_produs,

id\_comanda ),

CONSTRAINT continut\_produs\_fk FOREIGN KEY ( id\_produs )

REFERENCES produs ( id\_produs ),

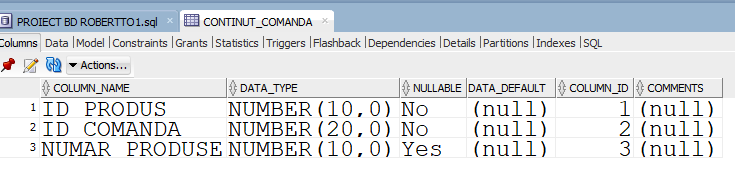
CONSTRAINT continut\_comanda\_fk FOREIGN KEY ( id\_comanda )

REFERENCES comanda ( id\_comanda ),

CONSTRAINT chk\_continut\_comanda CHECK ( numar\_produse > 0 )

);

***Print-Screen:***



-- PREPARARE --

CREATE TABLE preparare (

id\_produs NUMBER(10),

id\_comanda NUMBER(20),

id\_angajat NUMBER(10),

durata NUMBER(10),

CONSTRAINT preparare\_pk PRIMARY KEY ( id\_produs,

id\_comanda,

id\_angajat ),

CONSTRAINT preparare\_casier\_fk FOREIGN KEY ( id\_angajat )

REFERENCES bucatar ( id\_angajat ),

CONSTRAINT preparare\_produs\_fk FOREIGN KEY ( id\_produs )

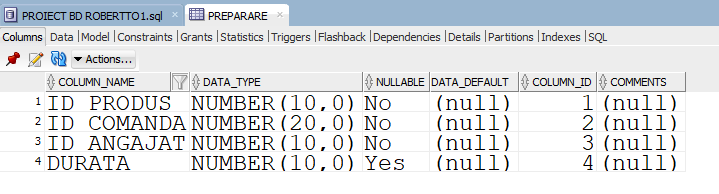
REFERENCES produs ( id\_produs ),

CONSTRAINT preparare\_comanda\_fk FOREIGN KEY ( id\_comanda )

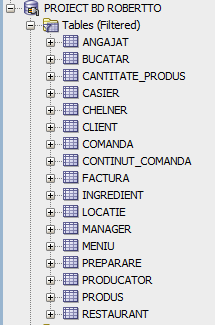
REFERENCES comanda ( id\_comanda )

);

***Print-Screen:***



***Print-Screen:***



***6.2 Inserarea datelor coerente în tabele (minimum 5 înregistrări în fiecare tabel neasociativ, minimum 10 înregistrări în tabelele asociative) + crearea unei secvențe.***

--- INSERAREA DATELOR IN TABELE ---

-- PENTRU TABELUL LOCATIE --

create sequence id\_locatie

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into locatie

values (id\_locatie.nextval, 'Romania' , 'Bucuresti', '010051', 'Batista'); --1

insert into locatie

values (id\_locatie.nextval, 'Romania', 'Cluj', '400033', 'Mihai Eminescu'); --2

insert into locatie

values (id\_locatie.nextval, 'Romania', 'Iasi', '700547', 'Rediu'); --3

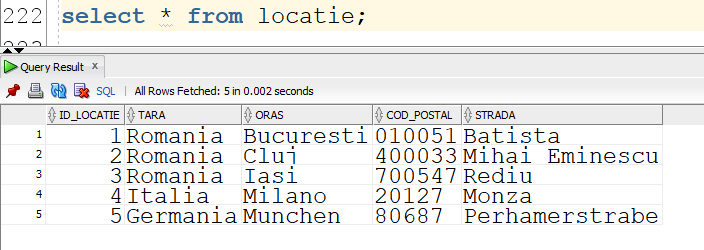
insert into locatie

values (id\_locatie.nextval, 'Italia', 'Milano', '20127', 'Monza'); --4

insert into locatie

values (id\_locatie.nextval, 'Germania', 'Munchen', '80687', 'Perhamerstrabe'); --5

***Print-Screen:***



-- PENTRU TABELUL RESTAURANT --

create sequence id\_restaurant

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into restaurant

values (id\_restaurant.nextval, 2, 'Gurmandul'); --1

insert into restaurant

values (id\_restaurant.nextval, 3, 'Yamas'); --2

insert into restaurant

values (id\_restaurant.nextval, 1, 'Ivans'); --3

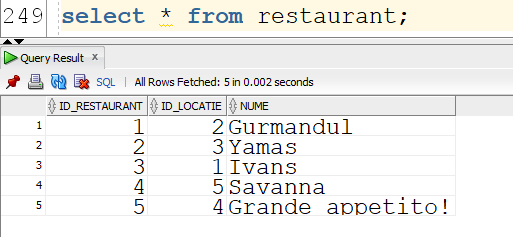
insert into restaurant

values (id\_restaurant.nextval, 5, 'Savanna'); --4

insert into restaurant

values (id\_restaurant.nextval, 4, 'Grande appetito!'); --5

***Print-Screen:***



-- PENTRU TABELUL MENIU --

-- pretul este in lei --

insert into meniu (id\_restaurant, id\_produs, pret)

values (4, 1, 35); --1

insert into meniu (id\_restaurant, id\_produs, pret)

values (4, 2, 50); --2

insert into meniu (id\_restaurant, id\_produs, pret)

values (1, 5, 20); --3

insert into meniu (id\_restaurant, id\_produs, pret)

values (1, 4, 33); --4

insert into meniu (id\_restaurant, id\_produs, pret)

values (5, 2, 60); --5

insert into meniu (id\_restaurant, id\_produs, pret)

values (2, 4, 15); --6

insert into meniu (id\_restaurant, id\_produs, pret)

values (5, 4, 13); --7

commit;

insert into meniu (id\_restaurant, id\_produs, pret)

values (3, 5, 12); --8

insert into meniu (id\_restaurant, id\_produs, pret)

values (4, 4, 30); --9

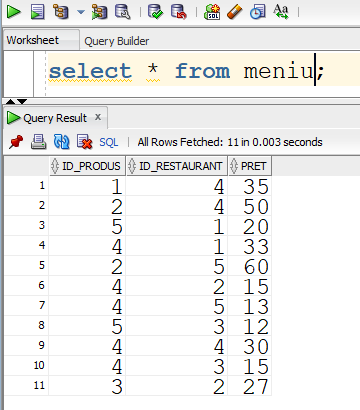
insert into meniu (id\_restaurant, id\_produs, pret)

values (3, 4, 15); --10

insert into meniu (id\_restaurant, id\_produs, pret)

values (2, 3, 27); --11

***Print-Screen:***



-- PENTRU TABELUL PRODUS --

-- aici cantitatea este in grame

create sequence id\_produs

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into produs

values (id\_produs.nextval, 'Spaghete', 200, 'picante'); --1

insert into produs

values (id\_produs.nextval, 'Pizza', 150, 'dulce aromata'); --2

insert into produs

values (id\_produs.nextval, 'Spaghete', 200, 'picante'); --3

-- aici am updatat linia pentru paste bolognese

update produs

set nume = 'Paste bolognese', descriere = 'cu sos dulce'

where id\_produs = 3;

insert into produs

values (id\_produs.nextval, 'Sarmale', 250, 'ca la mama acasa'); --4

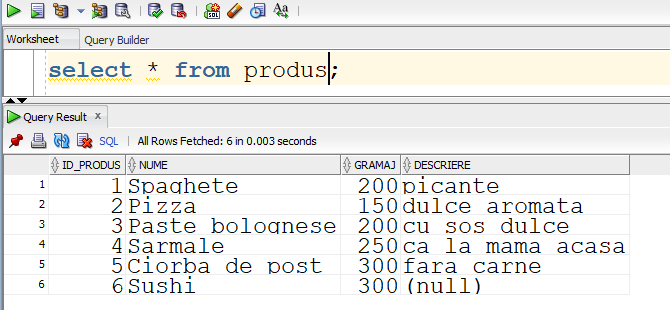
insert into produs

values (id\_produs.nextval, 'Ciorba de post', 300, 'fara carne'); --5

insert into produs(id\_produs,nume,gramaj)

values (id\_produs.nextval, 'Sushi', 300); --6

***Print-Screen:***



-- PENTRU TABELUL CANTITATE\_PRODUS --

-- cantitatea este masurata in grame --

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (1, 1, 85); --1

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (1, 4, 50); --2

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (2, 4, 60); --3

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (2, 3, 150); --4

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (2, 1, 100); --5

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (3, 4, 200); --6

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (3, 5, 55); --7

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (4, 4, 100); --8

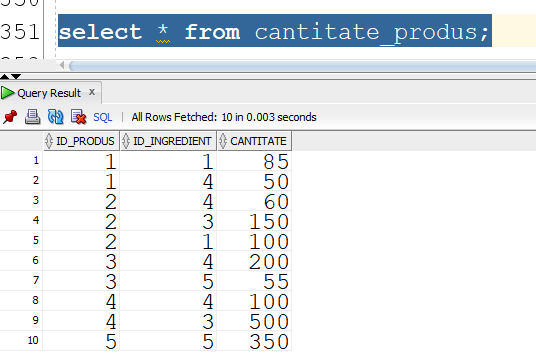
insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (4, 3, 500); --9

insert into cantitate\_produs (id\_produs, id\_ingredient, cantitate)

values (5, 5, 350); --10

***Print-Screen:***



-- PENTRU TABELUL INGREDIENT --

create sequence id\_ingredient

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into ingredient

values (id\_ingredient.nextval, 1, 'rosii'); --1

insert into ingredient

values (id\_ingredient.nextval, 3, 'varza murata'); --2

insert into ingredient

values (id\_ingredient.nextval, 4, 'ulei floarea soarelui'); --3

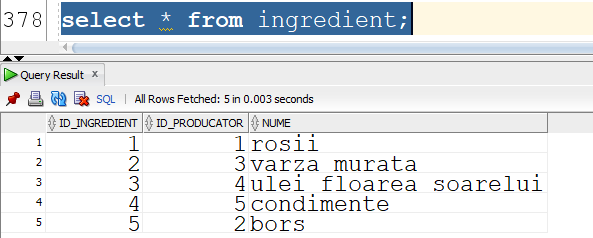
insert into ingredient

values (id\_ingredient.nextval, 5, 'condimente'); --4

insert into ingredient

values (id\_ingredient.nextval, 2, 'bors'); --5

***Print-Screen:***



-- PENTRU TABELUL PRODUCATOR --

create sequence id\_producator

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into producator

values (id\_producator.nextval, 'Livada cu de toate', '0770573182'); --1

insert into producator

values (id\_producator.nextval, 'Olivers', '0754754318'); --2

insert into producator

values (id\_producator.nextval, 'FreshOnly', '021999123'); --3

insert into producator

values (id\_producator.nextval, 'ION MOS', '021129123'); --4

insert into producator

values (id\_producator.nextval, 'FUCHS', '021592555'); --5

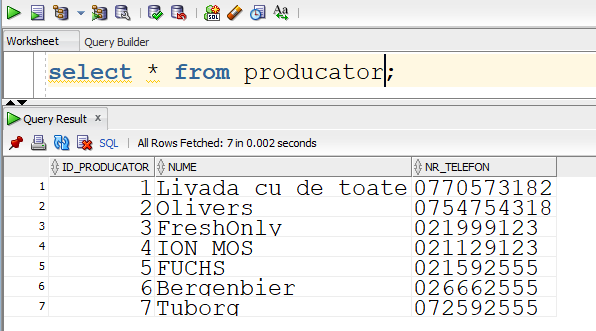
insert into producator

values (id\_producator.nextval, 'Bergenbier', '026662555'); --6

insert into producator

values (id\_producator.nextval, 'Tuborg', '072592555'); --7

***Print-Screen:***



-- PENTRU TABELUL CONTINUT\_COMANDA --

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (1, 4, 2); --1

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (1, 2, 1); --2

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (2, 2, 1); --3

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (2, 1, 1); --4

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (3, 5, 1); --5

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (3, 4, 1); --6

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (4, 5, 1); --7

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (5, 4, 1); --8

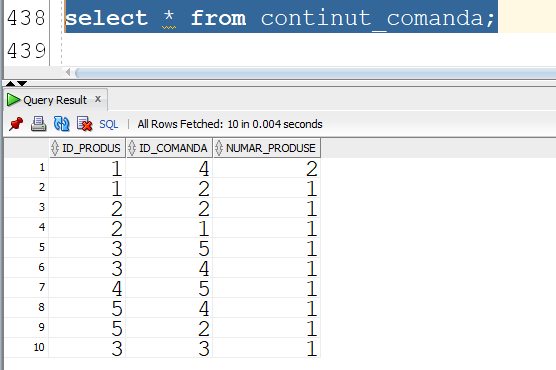
insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (5, 2, 1); --9

insert into continut\_comanda (id\_produs, id\_comanda, numar\_produse)

values (3, 3, 1); --10

***Print-Screen:***



-- PENTRU TABELUL COMANDA --

create sequence id\_comanda

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into comanda (id\_comanda, id\_client, id\_factura, pret)

values (id\_comanda.nextval, 1, 4, 50); --1

insert into comanda (id\_comanda, id\_client, id\_factura, pret)

values (id\_comanda.nextval, 3, 5, 130); --2

insert into comanda (id\_comanda, id\_client, id\_factura, pret)

values (id\_comanda.nextval, 2, 3, 20); --3

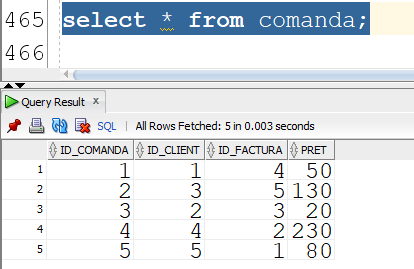
insert into comanda (id\_comanda, id\_client, id\_factura, pret)

values (id\_comanda.nextval, 4, 2, 230); --4

insert into comanda (id\_comanda, id\_client, id\_factura, pret)

values (id\_comanda.nextval, 5, 1, 80); --5

***Print-Screen:***



-- PENTRU TABELUL CLIENT --

create sequence id\_client

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into client

values (id\_client.nextval, 'Ionescu', 'Marian', '0721666212'); --1

insert into client (id\_client, nume, prenume, nr\_telefon)

values (id\_client.nextval, 'Dumitrescu', 'Mircel', '0721611211'); --2

insert into client (id\_client, nume, prenume, nr\_telefon)

values (id\_client.nextval, 'Gheorghe', 'Sebastian', '0744573419'); --3

insert into client (id\_client, nume, prenume, nr\_telefon)

values (id\_client.nextval, 'Salam','Florin', '0210116666'); --4

insert into client (id\_client, nume, prenume, nr\_telefon)

values (id\_client.nextval, 'Biju', 'Costel', '0211999913'); --5

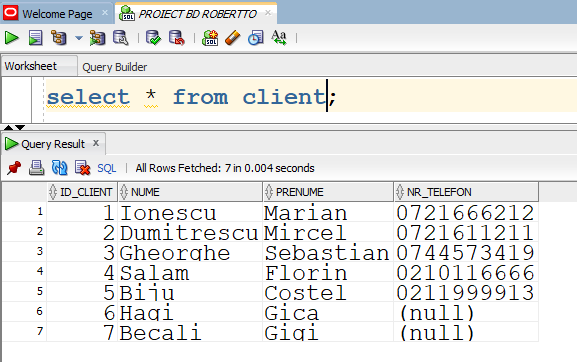
insert into client (id\_client, nume, prenume)

values (id\_client.nextval, 'Hagi', 'Gica'); --6

insert into client (id\_client, nume, prenume)

values (id\_client.nextval, 'Becali', 'Gigi'); --7

***Print-Screen:***



-- PENTRU TABELUL FACTURA --

-- fiecare factura are o valoare care include mai mult taxe etc --

create sequence id\_factura

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into factura (id\_factura, id\_angajat, valoare, detalii)

values (id\_factura.nextval, 16, 100, 'CASH'); --1

insert into factura (id\_factura, id\_angajat, valoare, detalii)

values (id\_factura.nextval, 12, 250, 'CARD'); --2

insert into factura (id\_factura, id\_angajat, valoare, detalii)

values (id\_factura.nextval, 8, 50, 'CASH'); --3

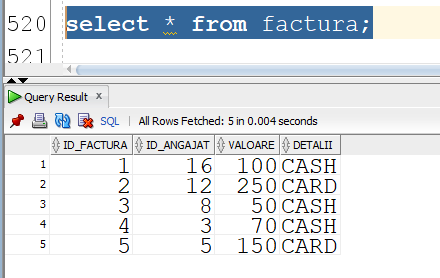
insert into factura (id\_factura, id\_angajat, valoare, detalii)

values (id\_factura.nextval, 3, 70, 'CASH'); --4

insert into factura (id\_factura, id\_angajat, valoare, detalii)

values (id\_factura.nextval, 5, 150, 'CARD'); --5

***Print-Screen:***



-- PENTRU TABELUL CASIER --

insert into casier(id\_angajat, ani\_studiu)

values (3, 10); --1

insert into casier(id\_angajat, ani\_studiu)

values (16, 2); --2

insert into casier(id\_angajat, ani\_studiu)

values (5, 0); --3

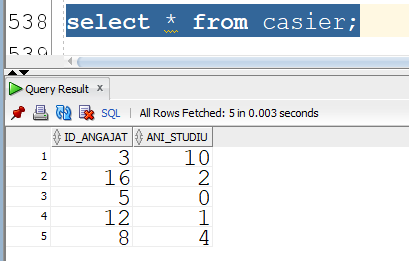
insert into casier(id\_angajat, ani\_studiu)

values (12, 1); --4

insert into casier(id\_angajat, ani\_studiu)

values (8, 4); --5

***Print-Screen:***



-- PENTRU TABELUL ANGAJAT --

create sequence id\_angajat

start with 1

increment by 1

minvalue 0

maxvalue 9999

nocycle;

insert into angajat (id\_angajat, id\_restaurant, nume, prenume)

values (id\_angajat.nextval, 1, 'Popescu', 'Robertto'); --1

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 2, 'Escobar', 'Ricardo', to\_date('15-02-21', 'dd-mm-yy')); --2

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 4, 'Marinescu', 'Teodora', to\_date('10-01-20','dd-mm-yy')); --3

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 3, 'Marinescu', 'Petre', to\_date('15-01-21','dd-mm-yy')); --4

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 2, 'Manole', 'Alexandru', to\_date('20-06-19','dd-mm-yy')); --5

insert into angajat (id\_angajat, id\_restaurant, nume, prenume)

values (id\_angajat.nextval, 1, 'Voicu', 'Andrei'); --6

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 3, 'Radoi', 'Raisa', to\_date('15-05-20','dd-mm-yy')); --7

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 4, 'Stan', 'Mihnea', to\_date('01-01-20','dd-mm-yy')); --8

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 5, 'Filip', 'Mihnea', to\_date('21-09-20','dd-mm-yy')); --9

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 1, 'Peste', 'Florin', to\_date('25-07-20','dd-mm-yy')); --10

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 2, 'Raducanu', 'Sorin', to\_date('01-04-21', 'dd-mm-yy')); --11

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 3, 'Bida', 'Marian', to\_date('25-02-21', 'dd-mm-yy')); --12

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 4, 'Fredo', 'Magiore', to\_date('01-05-21', 'dd-mm-yy')); --13

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 5, 'Cercel', 'Florin', to\_date('02-09-20','dd-mm-yy')); --14

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 4, 'Stanescu', 'Gigel', to\_date('02-07-20', 'dd-mm-yy')); --15

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 2, 'Dociu', 'Mihai', to\_date('17-08-20', 'dd-mm-yy')); --16

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 1, 'Dumitrescu', 'Florin', to\_date('04-03-21', 'dd-mm-yy')); --17

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 3, 'Sociu', 'Razvan', to\_date('29-12-20', 'dd-mm-yy')); --18

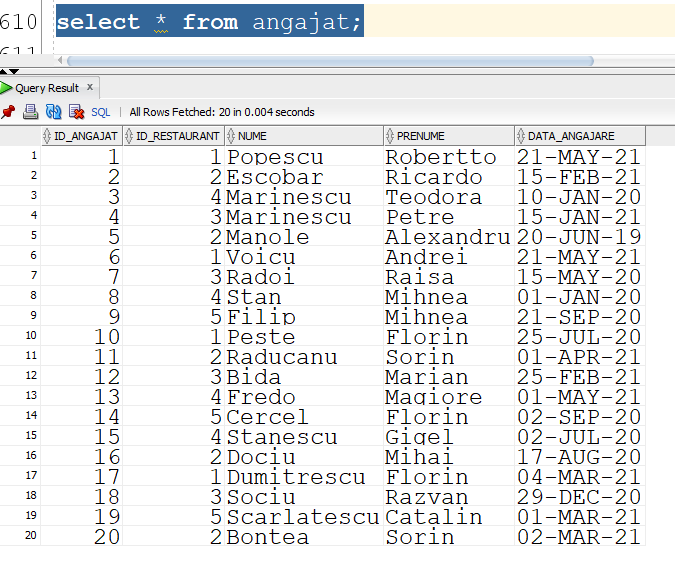
insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 5, 'Scarlatescu', 'Catalin', to\_date('01-03-21', 'dd-mm-yy')); --19

insert into angajat (id\_angajat, id\_restaurant, nume, prenume, data\_angajare)

values (id\_angajat.nextval, 2, 'Bontea', 'Sorin', to\_date('02-03-21', 'dd-mm-yy')); --20

***Print-Screen:***



-- PENTRU TABELUL PREPARARE --

-- durata este masurata in minute --

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (1, 1, 17, 60); --1

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (1, 2, 9, 45); --2

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (2, 4, 19, 35); --3

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (3, 4, 19, 25); --4

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (4, 4, 19, 120); --5

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (5, 4, 17, 75); --6

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (3, 3, 20, 70); --7

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (3, 5, 9, 16); --8

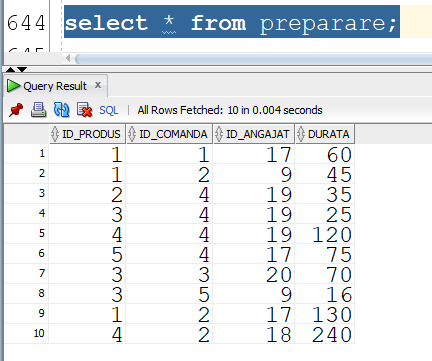
insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (1, 2, 17, 130); --9

insert into preparare (id\_produs, id\_comanda, id\_angajat, durata)

values (4, 2, 18, 240); --10

***Print-Screen:***



-- PENTRU TABELUL BUCATAR --

insert into bucatar (id\_angajat, nr\_stele)

values (9, 1); --1

insert into bucatar (id\_angajat, nr\_stele)

values (17, 3); --2

insert into bucatar (id\_angajat, nr\_stele)

values (18, 4); --3

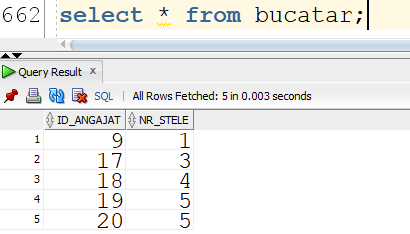
insert into bucatar (id\_angajat, nr\_stele)

values (19, 5); --4

insert into bucatar (id\_angajat, nr\_stele)

values (20, 5); --5

***Print-Screen:***



-- PENTRU TABELUL CHELNER --

insert into chelner (id\_angajat, ani\_experienta)

values (4, 2); --1

insert into chelner (id\_angajat, ani\_experienta)

values (7, 5); --2

insert into chelner (id\_angajat, ani\_experienta)

values (11, 1); --3

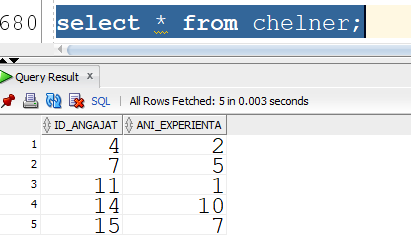
insert into chelner (id\_angajat, ani\_experienta)

values (14, 10); --4

insert into chelner (id\_angajat, ani\_experienta)

values (15, 7); --5

***Print-Screen:***



-- PENTRU TABELUL MANAGER --

insert into manager (id\_angajat)

values (1); --1

insert into manager (id\_angajat)

values (2); --2

insert into manager (id\_angajat)

values (6); --3

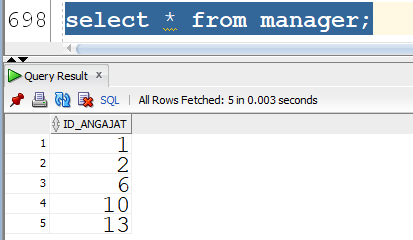
insert into manager (id\_angajat)

values (10); --4

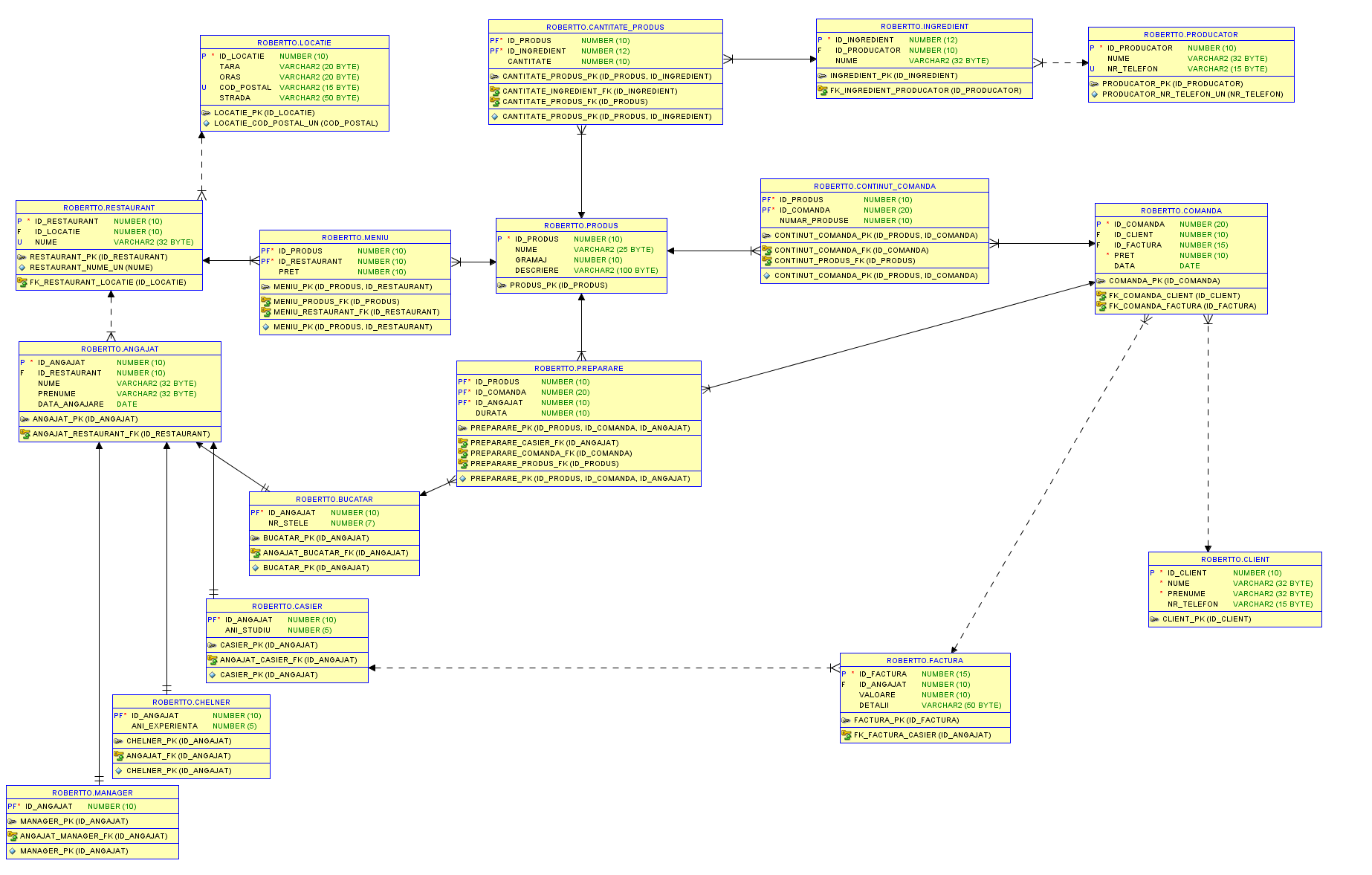
insert into manager (id\_angajat)

values (13); --5

***Print-Screen:***



***6.2 Diagrama generată în sql după crearea tabelelor și inserarea datelor.***



**7. Crearea a 5 cereri complexe în SQL:**

-- CERINTA 11 --

-- 1) Sa se afiseze numele si id-ul tuturor produselor comandate de un client vreodata.

SELECT DISTINCT

cl.id\_client,

cl.nume,

cl.prenume,

prod.id\_produs,

prod.nume

FROM

comanda c,

client cl,

continut\_comanda cmd,

produs prod

WHERE

cl.id\_client = c.id\_client

AND c.id\_comanda = cmd.id\_comanda

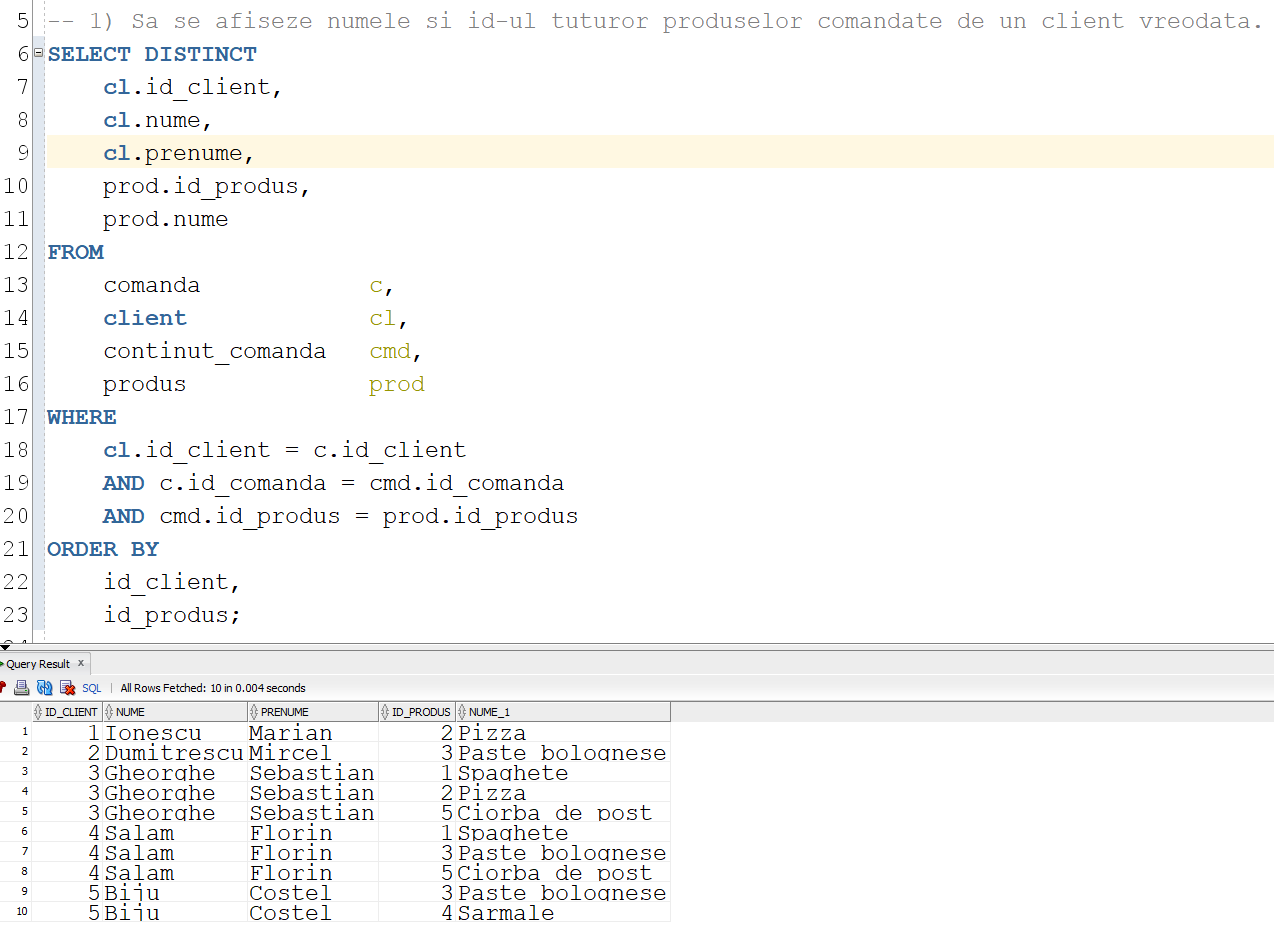
AND cmd.id\_produs = prod.id\_produs

ORDER BY

id\_client,

id\_produs;

***Print-Screen:***



-- 2) Pentru fiecare bucatar angajat inanul curent, afisati profitul mediu pe care il poate aduce daca ar vinde

-- din fiecare mancare pe care stie sa o prepare exact o bucata.

SELECT

a.id\_angajat,

lower(a.nume),

lower(a.prenume),

r.id\_restaurant,

upper(r.nume),

AVG(m.pret) "castig mediu"

FROM

angajat a,

bucatar b,

preparare prep,

produs prod,

meniu m,

restaurant r

WHERE

a.id\_restaurant = r.id\_restaurant

AND a.id\_angajat = b.id\_angajat

AND b.id\_angajat = prep.id\_angajat

AND prod.id\_produs = prep.id\_produs

AND prod.id\_produs = m.id\_produs

AND m.id\_restaurant = r.id\_restaurant

AND to\_char(a.data\_angajare, 'yyyy') LIKE to\_char(sysdate, 'yyyy')

GROUP BY (

a.id\_angajat,

a.nume,

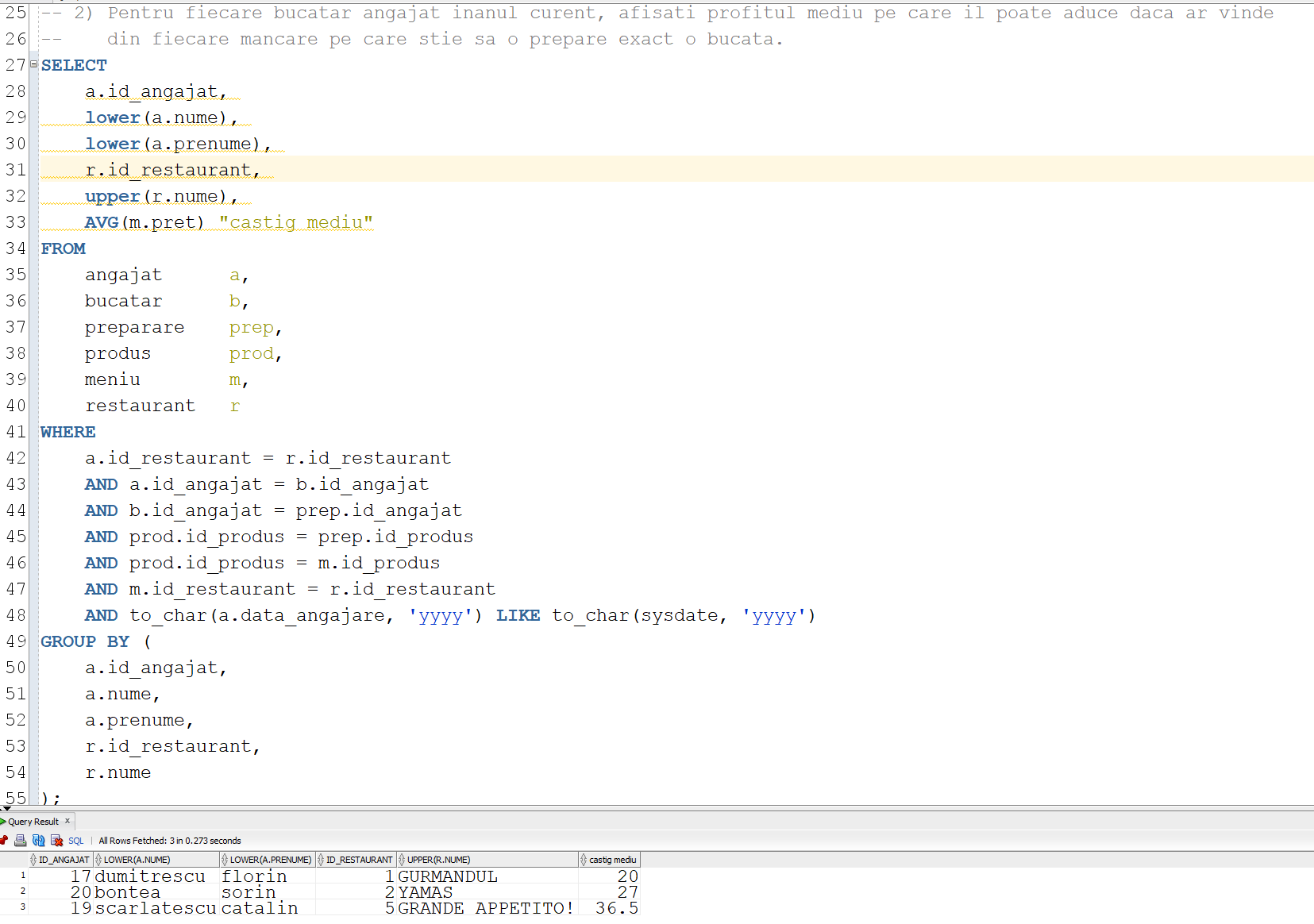
a.prenume,

r.id\_restaurant,

r.nume

);

***Print-Screen:***



-- 3) Afisati pentru fiecare produs, id-ul bucatarului care il prepara si cate stele are bucatarul

-- sau mesaj daca nu e preparat de nimeni

-- daca e preparat de mai multi, ii afisati pe toti.

-- Nesicronizata

with myProd as (

select distinct prod.ID\_PRODUS, prod.NUME, prep.ID\_ANGAJAT

from produs prod,

PREPARARE prep

where prod.ID\_PRODUS = prep.ID\_PRODUS(+))

select myProd.ID\_PRODUS,

myProd.NUME,

decode(nvl(om.ID\_ANGAJAT, -1), -1, 'nu e preparat de nimeni', om.id\_angajat) preparator,

om.NR\_STELE,

om.ID\_RESTAURANT

from myProd,

(select a.ID\_ANGAJAT, a.NUME, a.PRENUME, b.NR\_STELE, r.ID\_RESTAURANT

from ANGAJAT a,

BUCATAR b,

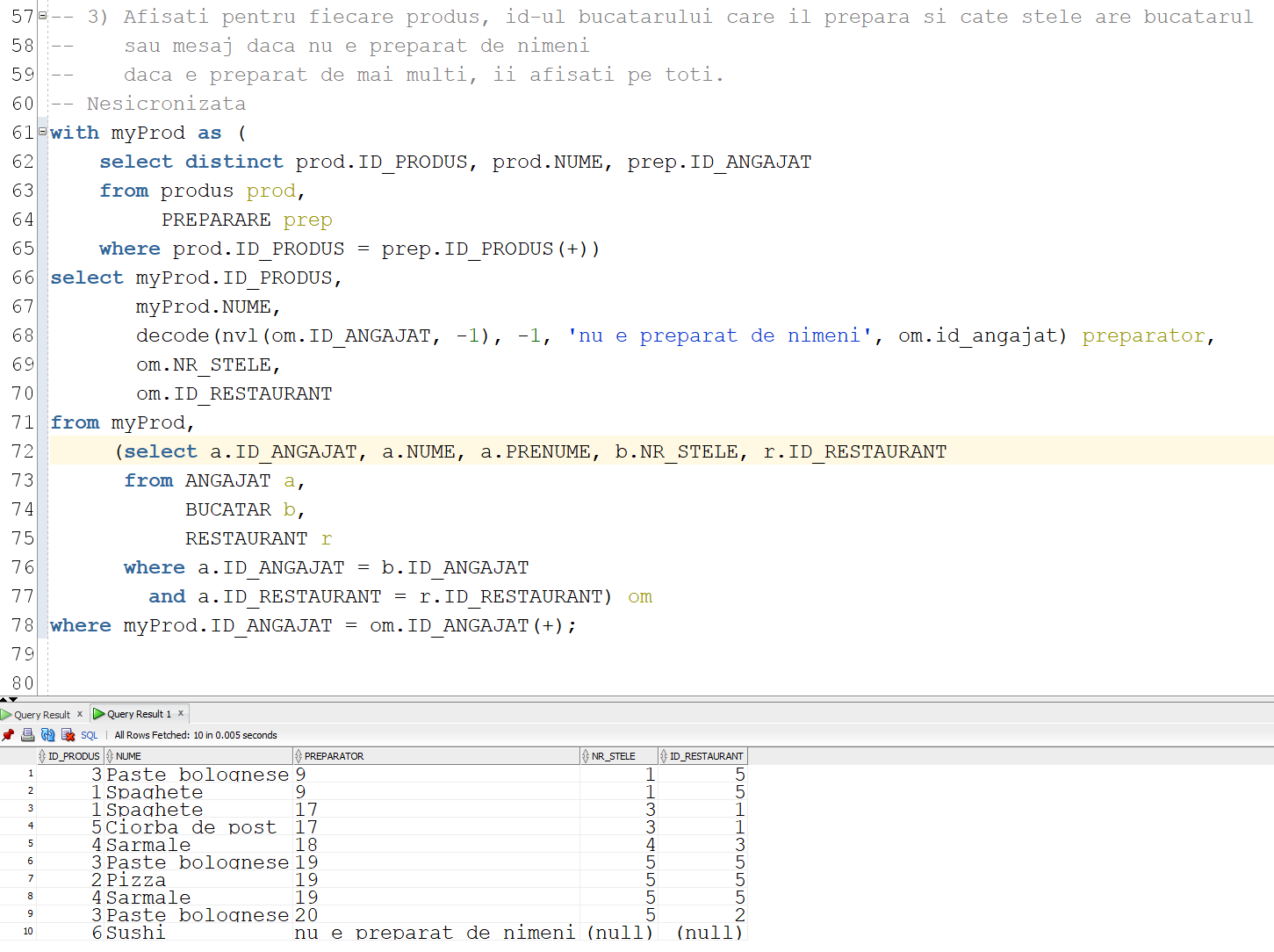
RESTAURANT r

where a.ID\_ANGAJAT = b.ID\_ANGAJAT

and a.ID\_RESTAURANT = r.ID\_RESTAURANT) om

where myProd.ID\_ANGAJAT = om.ID\_ANGAJAT(+);

***Print-Screen:***



--- 4) Afisati toti clientii cu toate comenzile lor si facturile aferente,

-- a caror suma cheltuita pana acum depaseste average-ul comenzilor cu cel putin 3 produs

-- Sincronizata

select \*

from CLIENT cl,

COMANDA cmd,

FACTURA f

where cmd.ID\_CLIENT = cl.ID\_CLIENT

and f.ID\_FACTURA = cmd.ID\_FACTURA

and (select sum(cmd2.PRET)

from COMANDA cmd2

where cmd2.ID\_CLIENT = cl.ID\_CLIENT) >

(select avg(t1.pr)

from (select cmd3.ID\_COMANDA, cmd3.PRET pr

from COMANDA cmd3,

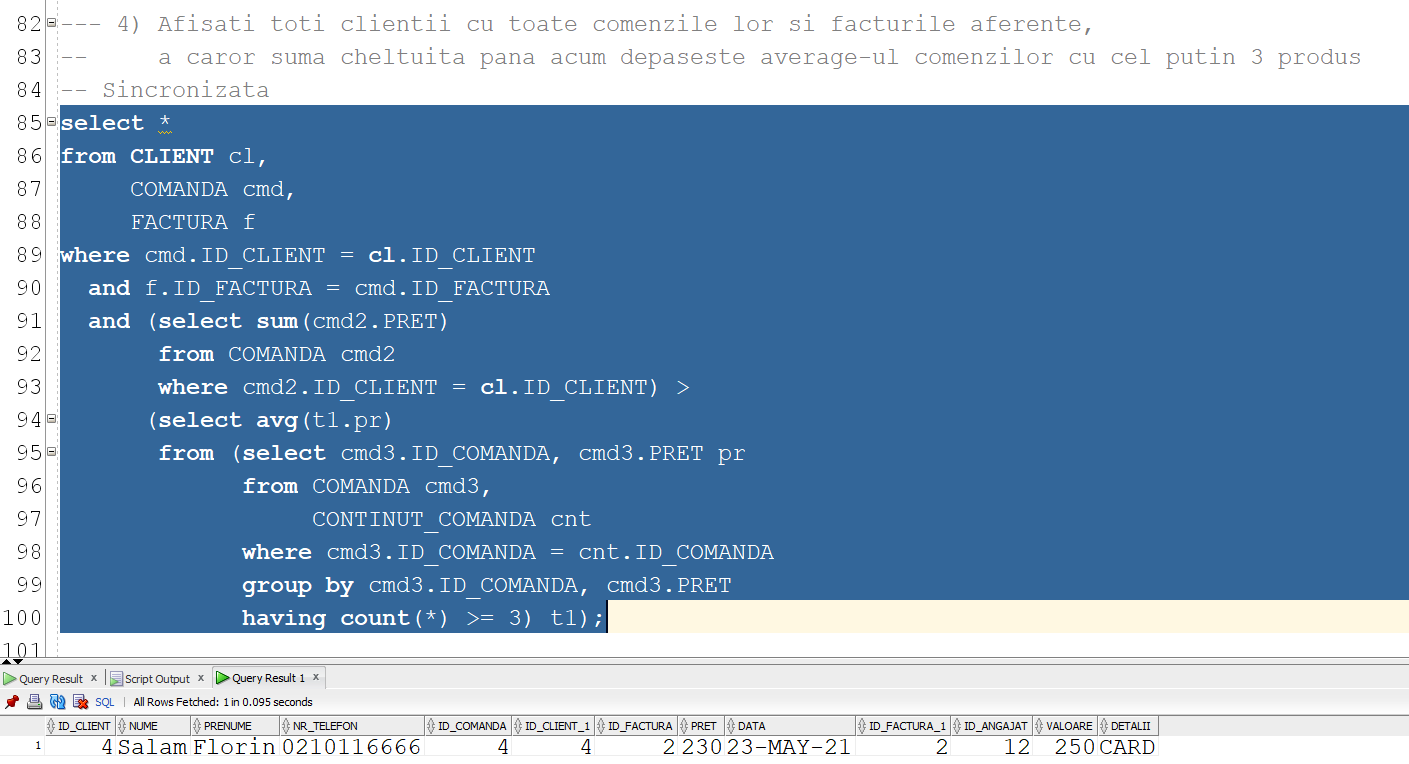
CONTINUT\_COMANDA cnt

where cmd3.ID\_COMANDA = cnt.ID\_COMANDA

group by cmd3.ID\_COMANDA, cmd3.PRET

having count(\*) >= 3) t1);

***Print-Screen:***



-- 5) Pentru fiecare pereche de angajati, afisati daca acestia se cunosc de cel putin un an.

select case

when months\_between(a1.DATA\_ANGAJARE, a2.DATA\_ANGAJARE) >= 12 then 'Angajatii ' || a1.ID\_ANGAJAT || ' si ' ||

a2.ID\_ANGAJAT ||

' se cunosc de peste un an'

else 'Angajatii ' || a1.ID\_ANGAJAT || ' si ' || a2.ID\_ANGAJAT || ' se cunosc de mai putin de un an' end

from ANGAJAT a1,

ANGAJAT a2

where a1.ID\_ANGAJAT < a2.ID\_ANGAJAT;

-- CERINTA 12 --

-- 1) suprimare

DELETE FROM producator

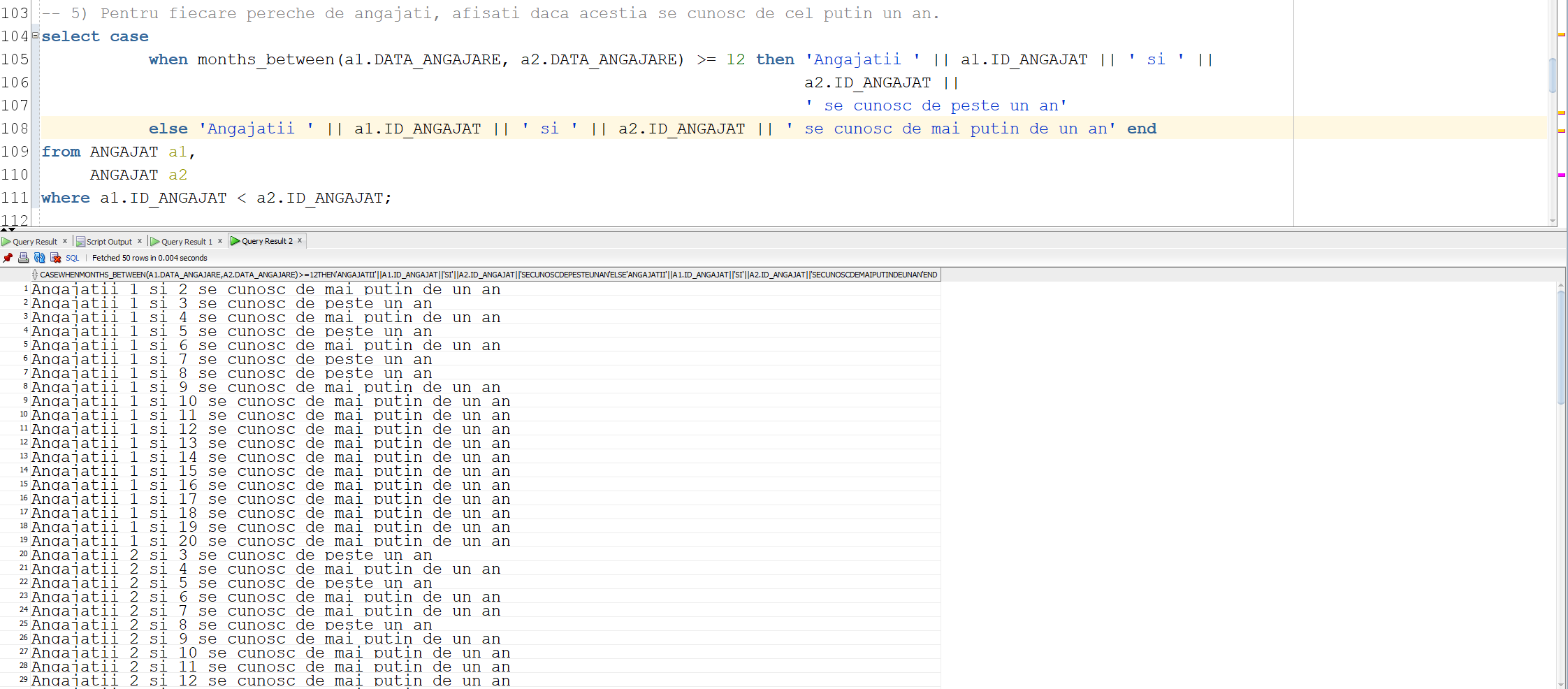
WHERE nume IN ( SELECT nume

FROM producator

where upper(nume) like '%BERGENBIER%'

);

***Print-Screen:***



**8. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri:**

-- CERINTA 12 --

-- 1) suprimare

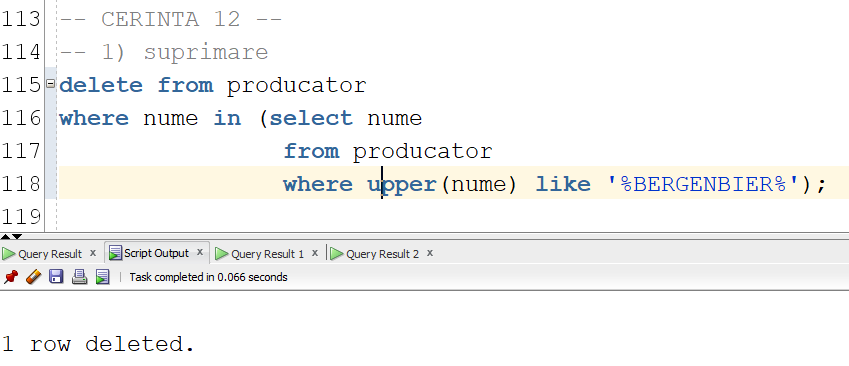
delete from producator

where nume in (select nume

from producator

where upper(nume) like '%BERGENBIER%');

***Print-Screen:***



-- 2) update

update client

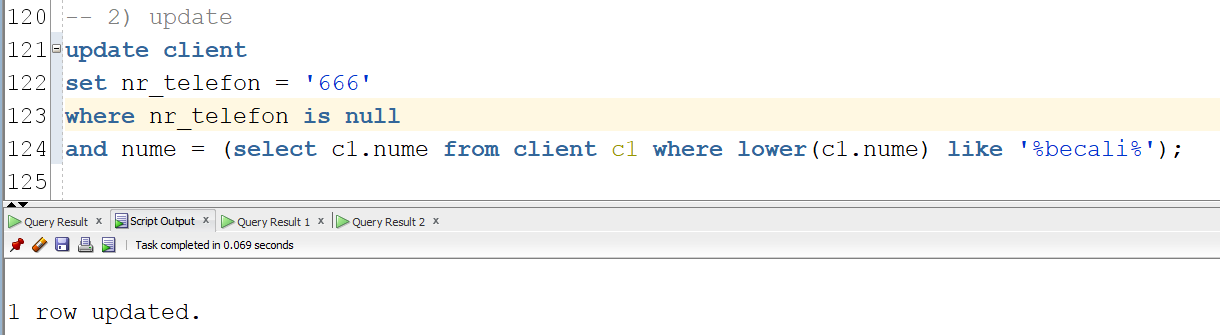
set nr\_telefon = '666'

where nr\_telefon is null

and nume = (select c1.nume from client c1 where lower(c1.nume) like '%becali%');

select \* from client;

***Print-Screen:***



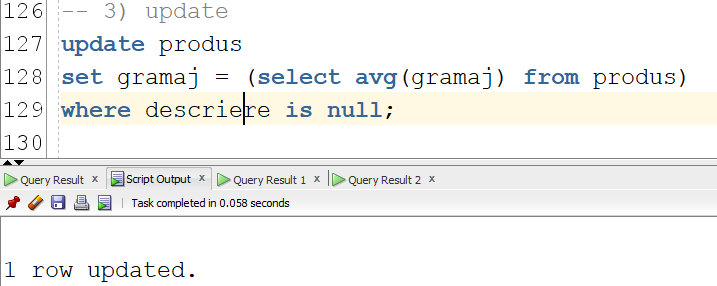
-- 3) update

update produs

set gramaj = (select avg(gramaj) from produs)

where descriere is null;

***Print-Screen:***



**9. O cerere care utilizează operația outer-join pe minium 4 tabele și două cereri ce utilizează operația division:**

-- CERINTA 16

-- Folosim outer join aici

-- Afisati pentru toate produsele din sistem, la ce restaurant se servesc, sau null daca nu e servit nicaieri.

select p.ID\_PRODUS, p.NUME, m.ID\_RESTAURANT, r.NUME, l.TARA, l.ORAS, l.STRADA

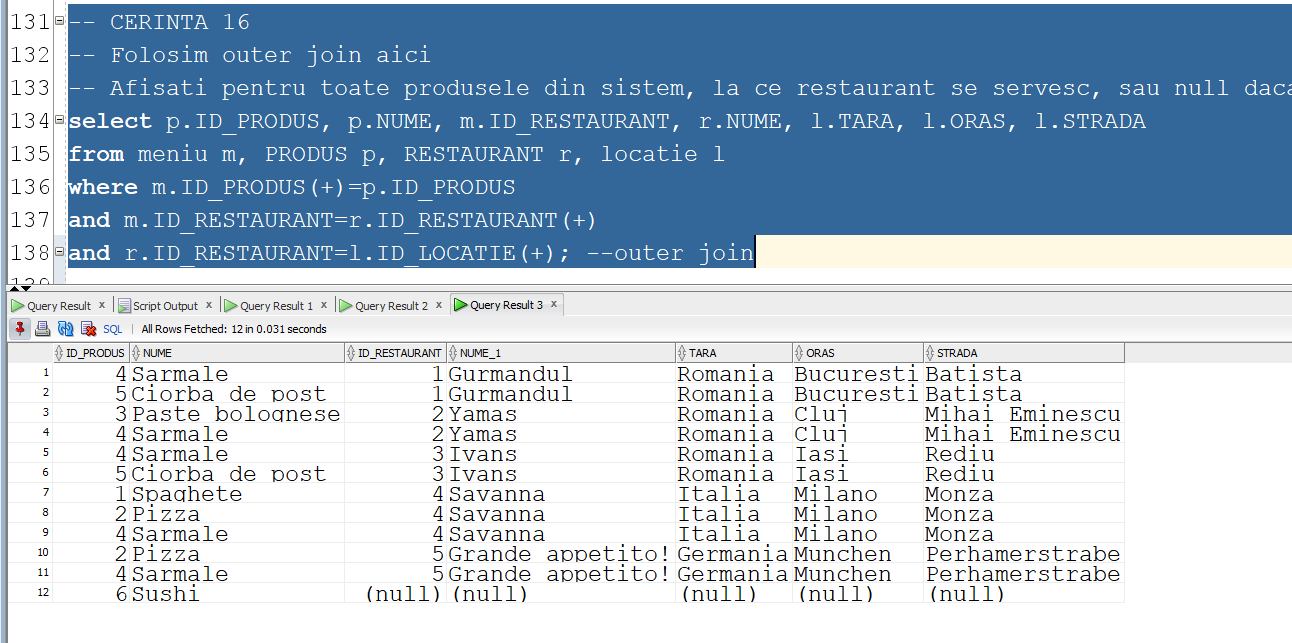
from meniu m, PRODUS p, RESTAURANT r, locatie l

where m.ID\_PRODUS(+)=p.ID\_PRODUS

and m.ID\_RESTAURANT=r.ID\_RESTAURANT(+)

and r.ID\_RESTAURANT=l.ID\_LOCATIE(+); --outer join

***Print-Screen:***



-- Afisati produsele care se gasesc in toate restaurantele.

-- Folosim division

select distinct p.ID\_PRODUS,p.NUME

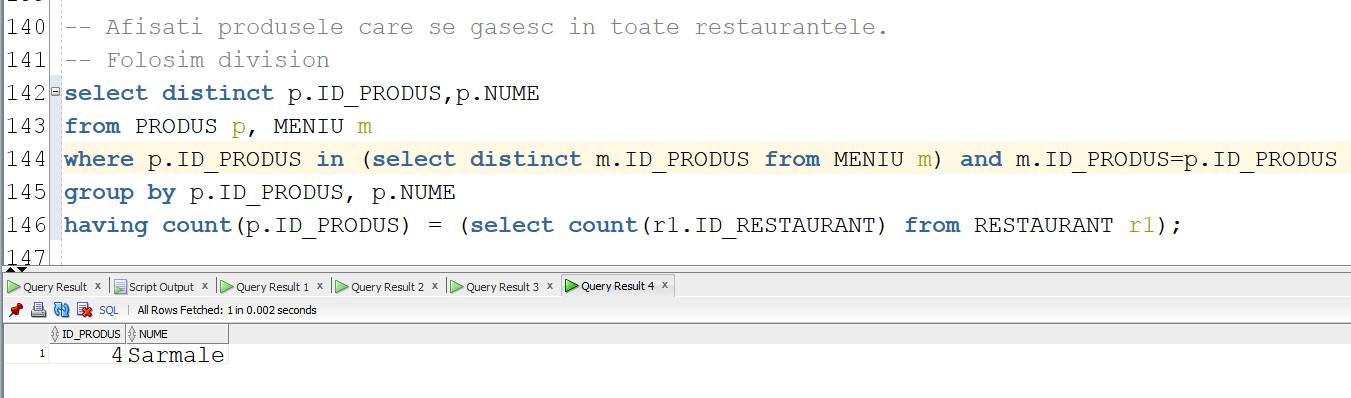
from PRODUS p, MENIU m

where p.ID\_PRODUS in (select distinct m.ID\_PRODUS from MENIU m) and m.ID\_PRODUS=p.ID\_PRODUS

group by p.ID\_PRODUS, p.NUME

having count(p.ID\_PRODUS) = (select count(r1.ID\_RESTAURANT) from RESTAURANT r1);

***Print-Screen:***



-- Afisati produsele care se gasesc in toate comenzile dintr-o anumita zi.

-- Folosim Division

insert into FACTURA values(1000,3,100,'nu');

insert into COMANDA values (1000,1,1000,1000,to\_date('13-07-2000','dd-mm-yyyy'));

insert into CONTINUT\_COMANDA values (4,1000,3);

with t as(select \* from CONTINUT\_COMANDA cnt0, COMANDA cmd0 where cnt0.ID\_COMANDA=cmd0.ID\_COMANDA)

select distinct p.ID\_PRODUS, p.NUME

from PRODUS p, t t1

where p.ID\_PRODUS=t1.ID\_PRODUS

and to\_char(t1.DATA,'dd-mm-yyyy')='13-07-2000'

and p.ID\_PRODUS in (select distinct t2.ID\_PRODUS from t t2

where to\_char(t2.DATA, 'dd-mm-yyyy')='13-07-2000')

group by p.ID\_PRODUS, p.NUME

having count(p.ID\_PRODUS)=(select count(\*) from t t3

where to\_char(t3.DATA, 'dd-mm-yyyy')='13-07-2000');

***Print-Screen:***



**10. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării:**

**Cerința:**

*Să se afișeze codul și numele restaurantelor care au în meniu prețuri mai mari sau egale cu 15 lei la produse care sunt picante sau cu sos dulce.*

* Mai jos avem arborele ***neoptimizat***:

REZULTAT



id\_restaurant, nume



RESTAURANT

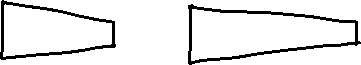


id\_produs



descriere = ‘cu sos dulce’

Preț >= 15



descriere = ‘picante’



MENIU

PRODUS

PRODUS

***ALGEBRA RELAȚIONALĂ:***

R1 = SELECT (PRODUS, descriere = ‘picante’);

R2 = SELECT (PRODUS, descriere = ‘%cu%sos%dulce%’);

R3 = UNION (R1, R2);

R4 = PROJECT (R3, id\_produs);

R5 = SELECT (MENIU, Preț >= 15);

R6 = SEMIJOIN (R5, R4);

R7 = SEMIJOIN (RESTAURANT, R6);

REZULTAT = PROJECT (R7, id\_restaurant);

***SQL:***

with

R1 as (select \* from produs where descriere like '%picante%'),

R2 as (select \* from produs where descriere like '%cu%sos%dulce%'),

R3 as (select \* from R1 union select \* from R2),

R4 as (select id\_produs from R3),

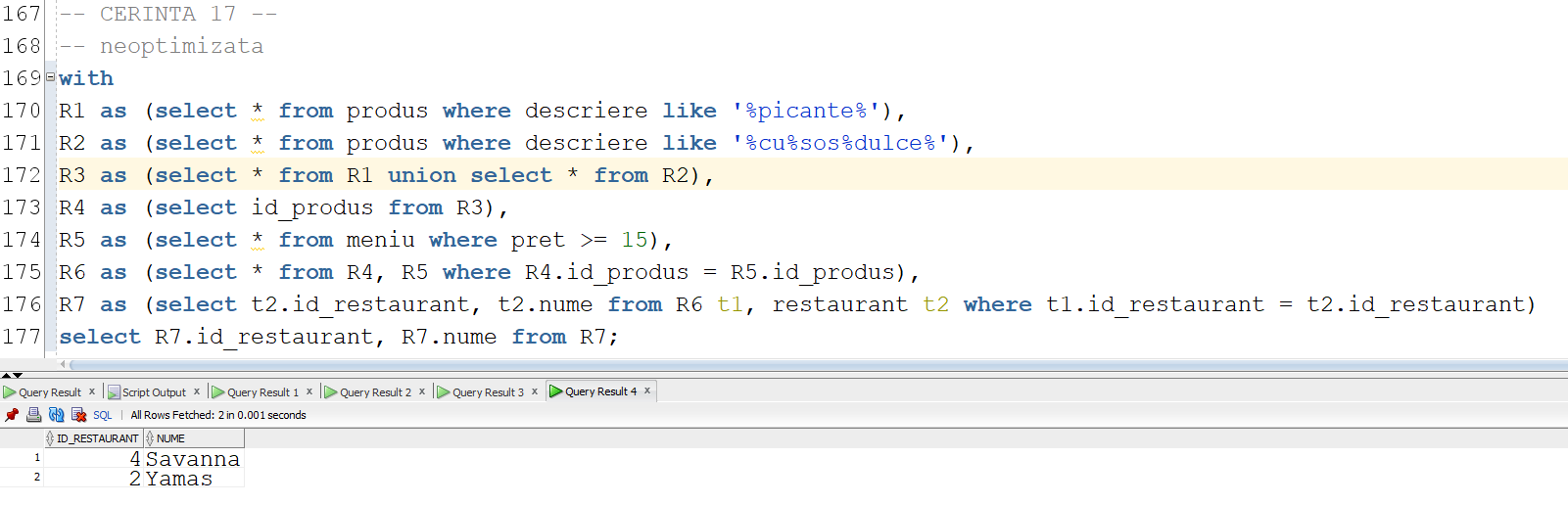
R5 as (select \* from meniu where pret >= 15),

R6 as (select \* from R4, R5 where R4.id\_produs = R5.id\_produs),

R7 as (select t2.id\_restaurant, t2.nume from R6 t1, restaurant t2 where t1.id\_restaurant = t2.id\_restaurant)

select R7.id\_restaurant, R7.nume from R7;

***Print-Screen:***



* Mai jos avem arborele ***optimizat***:

REZULTAT



id\_restaurant, nume



RESTAURANT



id\_produs

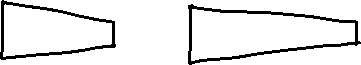
id\_produs



Preț >= 15



descriere = ‘cu sos dulce’



descriere = ‘picante’



MENIU

PRODUS

PRODUS

***ALGEBRA RELAȚIONALĂ:***

R1 = SELECT (PRODUS, descriere = ‘picante’);

R2 = SELECT (PRODUS, descriere = ‘%cu%sos%dulce%’);

R3 = PROJECT (R1, id\_produs);

R4 = PROJECT (R2, id\_produs);

R5 = UNION (R4, R3);

R6 = SELECT (MENIU, Preț >= 15);

R7 = SEMIJOIN (R5, R6);

R8 = PROJECT (RESTAURANT, id\_restaurant, nume);

REZULTAT = SEMIJOIN (R7, R8);

***SQL:***

with

R1 as (select \* from produs where descriere like '%picante%'),

R2 as (select \* from produs where descriere like '%cu%sos%dulce%'),

R3 as (select id\_produs from R1),

R4 as (select id\_produs from R2),

R5 as (select \* from R3 union select \* from R4),

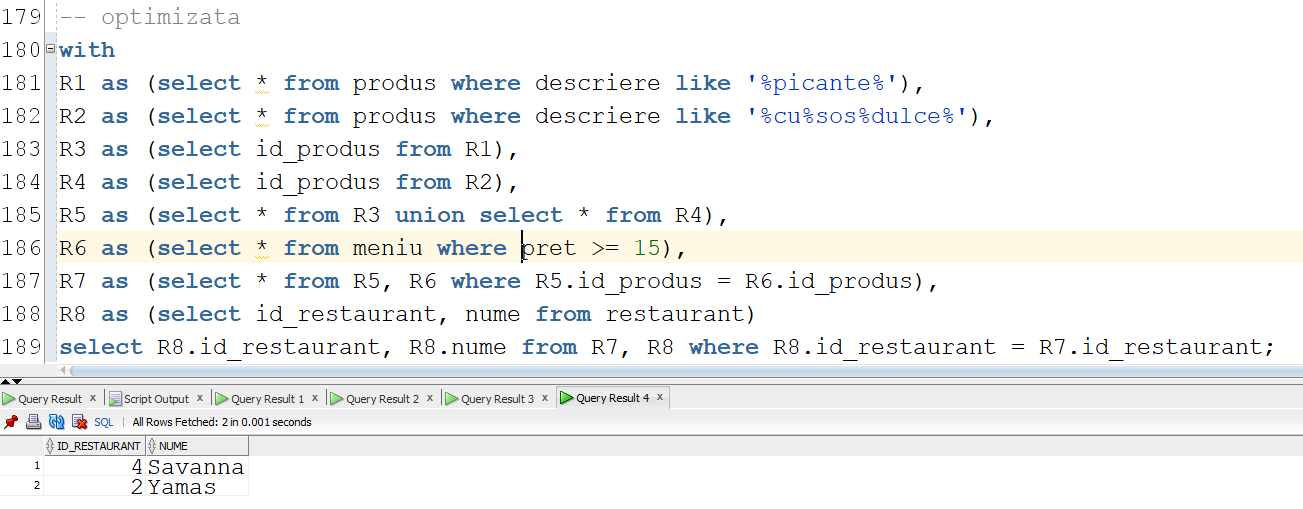
R6 as (select \* from meniu where pret >= 15),

R7 as (select \* from R5, R6 where R5.id\_produs = R6.id\_produs),

R8 as (select id\_restaurant, nume from restaurant)

select R8.id\_restaurant, R8.nume from R7, R8 where R8.id\_restaurant = R7.id\_restaurant;

***Print-Screen:***



**11. a) Realizarea normalizării BCNF, FN4, FN5:**

Schema noastră ar fi scoasă din BNCF întrucât în tabelul meniu, #id\_restaurant și #id\_produs determină prețul. Prespunem pentru modelul nostru că prețul determină #id\_produs.

Tabelul Meniu nu este în BCNF, deci aplicăm regula *Casey-Delobel* și atunci rezultă că:

* **MENIU1** (#Preț, id\_produs)
* **MENIU2** (#id\_restaurant, Preț)

Un alt caz ar putea fi atunci când #id\_produs determină preț, tragem concluzia că nu este în BCNF. Aplicăm regula *Casey-Delobel* și atunci rezultă că:

* **MENIU1** (#id\_produs, preț)
* **MENIU2** (#id\_produs, #id\_restaurant)

Observăm că aici se păstrează dependențele funcționale spre deosebire de prima afirmație.

**Schema Non-BCNF:**

**MENIU** (#ID-RESTAURANT, #ID-PRODUS, Preț)

Schema noastră ar fi scoasă din FN4 întrucât în tabelul preparare, #id\_comandă ->-> (multidetermină) #id\_produs, de unde rezultă #id\_comandă ->-> (multidetermină) #id\_bucătar. O aducem în FN4:

* **PREPARARE1** (#id\_comandă, #id\_produs, Durată)
* **PREPARARE2** (#id\_comandă, #id\_bucătar)

**Schema Non-FN4:**

**PREPARARE** (#ID-BUCĂTAR, #ID-COMANDĂ, #ID-PRODUS, Durată)

Un bucătar poate prepara mai multe produse care aparțin aceleiași comenzi sau poate prepara un produs pentru o comandă într-o perioadă de timp diferită. Presupunem că mai mulți bucătari pot prepara același produs, în aceeași perioadă de timp sau în perioade de timp diferite.

Datorită dependențelor precizate anterior, relația nu va mai fi în FN5. Ea se poate desface prin proiecție în trei relații:

* **PREPARARE1** (#id\_bucătar, #id\_comandă, #id\_produs)
* **PREPARARE2** (#id\_bucătar, Durată)
* **PREPARARE3** (#id\_comandă, #id\_produs, Durată)

Putem observa foarte clar:

PREPARARE != JOIN (PREPARARE1, PREPARARE2)

PREPARARE != JOIN (PREPARARE1, PREPARARE3)

PREPARARE != JOIN (PREPARARE2, PREPARARE3)

PREPARARE = JOIN (JOIN(PREPARARE1, PREPARARE2), PREPARARE3)

**Schema Non-FN5:**

**PREPARARE** (#ID-COMANDĂ, #ID-PRODUS, #ID-BUCĂTAR, Durată)

**b) Aplicarea denormalizării:**

Tabelul LOCAȚIE (#id\_locație, Țara, Oraș, Cod\_poștal, Strada)

Această relație nu este în FN3 întrucât cod\_poștal -> (determină) {Oraș, Țara}. Ea este mai exact în FN2.

Dacă aplicăm FN3 vom obține:

* LOCAȚIE1 (#id\_locație, Strada, Cod\_poștal)
* COD\_P (#cod\_poștal, Oraș, Țara)

Acest lucru nu este convenabil, deoarece rareori vom accesa adresa unui restaurant fără informații la Țară și Oraș. Din acest motiv am decis să nu folosesc forma FN3 pentru tabelul LOCAȚIE, reducem numărul de join-uri care nu sunt necesare.

Un alt exemplu ar putea fi acela pentru tabelul FACTURĂ, care conține #id\_factură, valoare, id\_casier, detalii. Acesta să conțină și nume\_chelner, respectiv prenume\_chelner, pentru a avea mai multe detalii despre angajatul(chelnerul) care a fost atașat la o factură. Această condiție ar reduce numărul de join-uri pentru a afla numele și prenumele chelnerului. Presupunem că nu ne interesează decât numele și prenumele acestuia. (Să se afișeze numele și prenumele chelnerului care a fost atașat la factura cu cea mai mare valoare.)