

Nedelcu Ionuț-Daniel

Grupa 142

Baza de date

nidle

## CUPRINS

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare. ....	4
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului. ....	6
3. Descrierea entităților, incluzând precizarea cheii primare. ....	10
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora. ....	13
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor. ....	16
6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5. ....	24
7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ. ....	25
8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7. .....	26
9. Realizarea normalizării până la forma normală 3 (FN1-FN3) .....	27
Forma normală 1 .....	27
Forma normală 2 .....	28
Forma normală 3 .....	29
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11). ....	31
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea. (minimum 5 înregistrări în fiecare tabel neasociativ; minimul 10 înregistrări în tabelele asociative). ....	33
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente: .....	68
Cererea 1 .....	68
Cererea 2 .....	71
Cererea 3 .....	73
Cererea 4 .....	74
Cererea 5 .....	77
13. Implementarea a 3 operații de actualizare și suprimare a datelor utilizând subcereri. ....	79
Operația 1 .....	79
Operația 2 .....	80
Operația 3 .....	81
14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă. ....	83
Creare vizualizare .....	83
Operație LMD permisă .....	85

Operație LMD nepermisă .....	86
15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n. ....	87
OUTER-JOIN pe minimum 4 tabele .....	87
Operația DIVISION .....	89
Analiza TOP-N .....	90

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

- Tema:

Tema proiectului este o bază de date realizată pentru un brand online de haine și accesorii personalizate, lucrate manual, „Nidle”. Aceasta are ca scop monitorizarea comenzilor și produselor realizate.

- Descrierea modelului real:

Această bază de date este folosită pentru a urmări și actualiza istoricul de comenzi al clienților, colecțiile, activitatea angajaților, dar și interacțiunile cu furnizorii, stocând și gestionând informațiile despre toate aceste aspecte.

Entitatea Client păstrează informații despre un client, precum numele, prenumele, emailul, dar și un identificator unic atribuit de sistem. Entitatea Adresa stochează adresa unui client, fiind conectată de entitatea Client. Entitatea Comanda conține informațiile despre o comandă, cum ar fi totalul de plată și tipul de transport pentru livrare; conectată de entitatea Client, de entitatea Feedback, care conține informații despre feedbackul primit pentru comandă (nota, descriere), și de entitatea Produs. Entitatea Produs stochează informații despre un produs, de exemplu numele produsului, prețul, descrierea, dacă este pe stoc sau nu; asociată la rândul ei cu entitățile Colecție, Sarcina, și Tipar.

Entitatea Colecție păstrează informațiile despre o colecție a brandului: numele colecției, tematica, anotimpul, și anul. Entitatea Sarcina se referă la atribuirea unui produs către un designer și un croitor, păstrând ca informație descrierea (sau cerințele) sarcinii, conectată de entitățile Croitor și Designer. Entitatea Croitor reține informațiile despre un croitor (sau persoană care confecționează produsul) – nume, prenume, email, salariu, data angajării. Entitatea Designer reține informații despre un designer – nume, prenume, email, salariu, comisionul per produs, stilul de design; aceasta fiind conectată, precum entitatea Produs, la entitatea Tipar.

Entitatea Tipar reprezintă tiparul unei bucăți dintr-un anumit produs, păstrând ca date doar dimensiunile tiparului, fiind asociată apoi cu entitatea Material. Entitatea Material reține informațiile următoare despre un material: numele, tipul materialului, compoziția acestuia; și este conectată cu entitatea Contract. Entitatea Contract se referă la datele unui contract cu un furnizor, stocând doar data acestuia, și fiind legată în final de entitatea Furnizor. Entitatea Furnizor reține informații despre un furnizor: numele firmei furnizoare și codul fiscal al acesteia.

- Utilitatea:

Această bază de date va fi folosită pentru gestionarea tuturor comenzilor, oferind brandului Nidle libertate de gestiune și acces la toate datele și informațiile despre furnizori, clienți, designeri, feedback.

Informațiile vor fi folosite pentru a urmări popularitatea colecțiilor de haine și accesorii, costurile, câștigurile, astfel că atât clienții, cât și persoana care gestionează baza de date, pot vizualiza o multitudine de informații legate de brand, care se marketează ca fiind transparent.

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

- Regulile și restricțiile bazei de date:

### 1) Tabelul ADRESE

- *id\_adresa* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *tara* trebuie să fie un șir de caractere cu lungime maximă de 40 de caractere
- *judet* trebuie să fie un șir de caractere cu lungime maximă de 40 de caractere
- *oras* trebuie să fie un șir de caractere cu lungime maximă de 40 de caractere
- *strada* trebuie să fie un șir de caractere cu lungime maximă de 40 de caractere
- *numar* trebuie să fie un număr întreg pozitiv (număr natural)
- *scara* trebuie să fie un număr întreg pozitiv (număr natural) sau null
- *apartament* trebuie să fie un număr întreg pozitiv (număr natural) sau null
- *id\_client* este o cheie externă către tabelul CLIENTI, trebuie să referențieze un *id\_client* existent în acel tabel

### 2) Tabelul CLIENTI

- *id\_client* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nume* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *prenume* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *email* trebuie să fie un șir de caractere cu lungime maximă de 40 de caractere
- *nr\_tel* trebuie să fie un șir de caractere cu lungime maximă de 15 caractere

### 3) Tabelul COMENZI

- *id\_comanda* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *pret* trebuie să fie un număr real cu două zecimale
- *livrare* trebuie să fie una dintre valorile „curier”, „posta” sau „personal”
- *stare* trebuie să fie una dintre valorile „preluata”, „în lucru”, „finalizata” sau „livrata”
- *id\_client* este o cheie externă către tabelul CLIENTI, trebuie să referențieze un *id\_client* existent în acel tabel

### 4) Tabelul FEEDBACKURI

- *id\_feedback* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nota* trebuie să fie un număr întreg pozitiv (număr natural) mai mic sau egal cu 10
- *descriere* trebuie să fie un șir de caractere cu lungime maximă de 500 de caractere

- *id\_comanda* este o cheie externă către tabelul COMENZI, trebuie să referențieze un *id\_comanda* existent în acel tabel, face parte de asemenea din cheia primară a tabelului

#### 5) Tabelul PRODUSE

- *id\_produs* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nume* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *pret* trebuie să fie un număr real cu două zecimale
- *descriere* trebuie să fie un șir de caractere cu lungime maximă de 100 de caractere
- *id\_comanda* este o cheie externă către tabelul COMENZI, trebuie să referențieze un *id\_comanda* existent în acel tabel
- *id\_colectie* este o cheie externă către tabelul COLECTII, trebuie să referențieze un *id\_colectie* existent în acel tabel

#### 6) Tabelul COLECTII

- *id\_colectie* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nume* trebuie să fie un șir de caractere cu lungime maximă de 60 de caractere
- *tematica* trebuie să fie un șir de caractere cu lungime maximă de 60 de caractere
- *an* trebuie să fie un număr întreg pozitiv (număr natural)
- *anotimp* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere

#### 7) Tabelul SARCINI

- *id\_sarcina* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *descriere* trebuie să fie un șir de caractere cu lungime maximă de 100 de caractere
- *id\_croitor* este o cheie externă către tabelul CROITORI, trebuie să referențieze un *id\_croitor* existent în acel tabel
- *id\_designer* este o cheie externă către tabelul DESIGNERI, trebuie să referențieze un *id\_designer* existent în acel tabel
- *id\_produs* este o cheie externă către tabelul PRODUSE, trebuie să referențieze un *id\_produs* existent în acel tabel

#### 8) Tabelul ANGAJATI

- *id\_angajat* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nume* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *prenume* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *tip* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *email* trebuie să fie un șir de caractere cu lungime maximă de 40 de caractere
- *nr\_tel* trebuie să fie un șir de caractere cu lungime maximă de 15 caractere

- *salariu* trebuie să fie un număr întreg pozitiv
- *comision* trebuie să fie un număr real pozitiv cu două zecimale
- *data\_angajare* trebuie să fie o dată calendaristică

#### 9) Tabelul CROITORI

- *id\_croitor* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *id\_angajat* este o cheie externă către tabelul ANGAJATI, trebuie să referențieze un *id\_angajat* existent în acel tabel, face parte de asemenea din cheia primară a tabelului

#### 10) Tabelul DESIGNERI

- *id\_designer* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *id\_angajat* este o cheie externă către tabelul ANGAJATI, trebuie să referențieze un *id\_angajat* existent în acel tabel, face parte de asemenea din cheia primară a tabelului

#### 11) Tabelul TIPARE

- *id\_tipar* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *lungime* trebuie să fie un număr întreg pozitiv
- *lățime* trebuie să fie un număr întreg pozitiv
- *id\_designer* este o cheie externă către tabelul DESIGNERI, trebuie să referențieze un *id\_designer* existent în acel tabel
- *id\_produs* este o cheie externă către tabelul PRODUSE, trebuie să referențieze un *id\_produs* existent în acel tabel
- *id\_material* este o cheie externă către tabelul MATERIALE, trebuie să referențieze un *id\_material* existent în acel tabel

#### 12) Tabelul MATERIALE

- *id\_material* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nume* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *tip* trebuie să fie un șir de caractere cu lungime maximă de 30 de caractere
- *compozitie* trebuie să fie un șir de caractere cu lungime maximă de 100 de caractere

#### 13) Tabelul CONTRACTE

- *id\_contract* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *data* trebuie să fie o dată calendaristică



- *id\_furnizor* este o cheie externă către tabelul FURNIZORI, trebuie să referențieze un *id\_furnizor* existent în acel tabel, face parte de asemenea din cheia primară a tabelului
- *id\_material* este o cheie externă către tabelul MATERIALE, trebuie să referențieze un *id\_material* existent în acel tabel face parte de asemenea din cheia primară a tabelului

#### 14) Tabelul FURNIZORI

- *id\_furnizor* este cheia primară și trebuie să fie unică pentru fiecare înregistrare în tabel
- *nume* trebuie să fie un șir de caractere cu lungime maximă de 50 de caractere
- *cod\_fiscal* trebuie să fie un număr întreg pozitiv (număr natural)

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

#### 1) Entitatea ADRESA

Această entitate conține date despre adresa unui client. Fiecare înregistrare este identificată cu ajutorul cheii primare `id_adresa`. Entitatea stochează informațiile necesare pentru o adresă: țara, județ, oraș, strada, număr, și dacă este cazul, scara și apartament, fiind necesare pentru a putea trimite coletele (comenzile) clienților care nu au optat pentru ridicarea personală a acestora.

#### 2) Entitatea CLIENT

Această entitate conține date despre un client. Fiecare înregistrare este identificată cu ajutorul cheii primare `id_client`. Entitatea stochează informațiile necesare pentru identificarea unui client în baza de date: nume, prenume, email, număr de telefon (`nr_tel`). Este conectată prin cheia străină (FK) `id_adresa` de entitatea ADRESA pentru a putea accesa adresa/adresele corespunzătoare clientului.

#### 3) Entitatea COMANDA

Această entitate conține date despre o comandă. Fiecare înregistrare este identificată cu ajutorul cheii primare `id_comanda`. Entitatea stochează informațiile necesare pentru identificarea unei comenzi în baza de date: prețul (calculat în funcție de tipul de livrare a comenzii), tipul de livrare a comenzii (livrare: curier, posta, personal), starea comenzii (stare: preluata, în lucru, finalizata, livrata). Este conectată prin cheia străină (FK) `id_client` de entitatea CLIENT pentru a putea vedea cărui client îi aparține comanda.

#### 4) Entitatea FEEDBACK

Această entitate conține date despre un feedback (corespunzător sau nu). Fiecare înregistrare este identificată cu ajutorul cheii primare `id_feedback`. Entitatea stochează informațiile necesare pentru procesarea feedbackului de la clienți: nota oferită (nota) și textul/descrierea feedbackului (descriere). Este conectată prin cheia străină (FK) `id_comanda` de entitatea COMANDA (face parte din cheia primară a entității) pentru a putea vedea comanda căreia îi aparține, și implicit clientul.

#### 5) Entitatea PRODUS

Această entitate conține date despre un produs al brandului. Fiecare înregistrare este identificată cu ajutorul cheii primare `id_produs`. Entitatea stochează informațiile importante despre produs: nume, preț, descriere. Este conectată prin cheia străină (FK) `id_comanda` de entitatea COMANDA pentru a putea accesa comanda corespunzătoare, implicit clientul

acesteia, și prin cheia străină (FK) id\_colectie, de entitatea COLECTIE, deci colecția în care se încadrează, dacă produsul face parte dintr-o colecție.

#### 6) Entitatea COLECTIE

Această entitate conține date despre o colecție de haine și accesorii a brandului. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_colectie. Entitatea stochează informațiile importante despre colecție: nume, tema colecției (tematica), anul căruia îi corespunde (an), anotimpul/sezonul (anotimp). Informațiile acestei entități sunt importante pentru a putea vizualiza direcția artistică a brandului, dar și cele mai populare colecții.

#### 7) Entitatea SARCINA

Această entitate conține date despre o sarcină corespunzătoare unui produs. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_sarcina. Entitatea stochează cerințele/descrierea assignmentului (descriere). Este conectată prin: cheia străină (FK) id\_croitor de entitatea CROITOR pentru a putea accesa informațiile despre croitorul (sau după caz responsabilul, etc.) pentru sarcină; cheia străină (FK) id\_designer de entitatea DESIGNER pentru a putea accesa informațiile despre designerul care colaborează pentru sarcină; prin cheia străină (FK) id\_produs de entitatea PRODUS, pentru accesarea produsului pentru care este nevoie de această sarcină. Oricare din aceste chei străine poate lipsi, întrucât o sarcină nu necesită obligatoriu să fie conectată de toate celelalte trei entități.

#### 8) Entitatea ANGAJAT

Această entitate conține date despre un angajat. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_angajat. Entitatea stochează informațiile despre acesta: nume, prenume, email, salariu, comision, data angajare (data\_angajare). Informațiile acestei entități sunt folosite pentru a putea gestiona angajații brandului. Entitatea ANGAJAT este o supraentitate pentru entitățile CROITOR și DESIGNER.

#### 9) Entitatea CROITOR

Această entitate este o subentitate pentru entitatea ANGAJAT, și reține astfel dacă un angajat este (și) croitor. Este conectată prin cheia străină (FK) de entitatea ANGAJAT, care face parte din cheia primară a entității.

#### 10) Entitatea DESIGNER

Această entitate este o subentitate pentru entitatea ANGAJAT, și reține astfel dacă un angajat este (și) designer. Este conectată prin cheia străină (FK) de entitatea ANGAJAT, care face parte din cheia primară a entității.

#### 11) Entitatea TIPAR

Această entitate conține date despre un tipar. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_tipar. Entitatea stochează dimensiunile unui tipar pentru un anumit produs: lungime, latime. Este conectată prin: cheia străină (FK) id\_designer de entitatea DESIGNER pentru a putea accesa informațiile despre designerul corespunzător, cheia străină (FK) id\_produs de entitatea PRODUS pentru a putea accesa informațiile despre produsul căruia îi corespunde; prin cheia străină (FK) id\_material de entitatea MATERIAL, pentru accesarea informațiilor materialului necesar pentru tipar.

#### 12) Entitatea MATERIAL

Această entitate conține date despre un material folosit pentru produse. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_material. Entitatea stochează informațiile de bază ale unui material: nume, tip, compoziție.

#### 13) Entitatea CONTRACT

Această entitate conține date despre un contract cu un furnizor. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_contract. Entitatea stochează data la care a fost semnat contractul. Este conectată prin cheia străină (FK) id\_furnizor de entitatea FURNIZOR pentru a putea accesa informațiile despre furnizorul corespunzător contractului, și prin cheia străină (FK) id\_material de entitatea MATERIAL pentru a putea accesa informațiile despre materialul corespunzător contractului. Aceste două chei străine fac parte din cheia primară.

#### 14) Entitatea FURNIZOR

Această entitate conține date despre un furnizor. Fiecare înregistrare este identificată cu ajutorul cheii primare id\_furnizor. Entitatea stochează numele entității legale (care poate fi persoană fizică, firmă, etc.) furnizoare și codul fiscal al acesteia (CNP, CUI, etc. după caz).

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

1) Entitatea ADRESA are o relație de tipul „many-to-one” cu entitatea CLIENT, deoarece un client poate avea mai multe adrese, dar o adresă corespunde unui singur client. În cazul în care doi clienți locuiesc la aceeași adresă, aceasta este reținută individual pentru fiecare din ei, cu id diferit (explicație: dacă unul dintre ei modifică adresa, aceasta trebuie să se modifice doar pentru acel client). Există posibilitatea ca o adresă să nu corespundă nimănui, și doar să fie reținută în baza de date. Astfel, cardinalitatea ADDRESS-CLIENT este  $M(0):1(0)$ .

2) Entitatea CLIENT are o relație cu ADRESA, de tip „one-to-many”, detaliată anterior, dar și o relație cu entitatea COMANDA.

Relația CLIENT-COMANDA („one-to-many”): un client poate avea mai multe comenzi, dar o comandă nu poate corespunde mai multor clienți. Un client poate să nu aibă nicio comandă (doar i-au fost reținute datele), iar o comandă anulată nu va mai avea niciun client căreia îi este atribuită. Astfel, cardinalitatea CLIENT-ORDER este  $1(0):M(0)$ .

3) Entitatea COMANDA are o relație cu CLIENT, de tip „many-to-one”, detaliată anterior, dar și relații cu FEEDBACK și PRODUS.

Relația COMANDA-FEEDBACK („one-to-one”): o comandă poate avea cel mult un feedback, iar un feedback corespunde obligatoriu unei comenzi. Astfel, cardinalitatea COMANDA-FEEDBACK este  $1(0):1(0)$ .

Relația COMANDA-PRODUS („one-to-many”): o comandă poate avea mai multe produse, sau poate să nu aibă niciunul (subînțeleasă ca donație/sponsorizare), iar un produs poate corespunde unei singure comenzi, sau niciuneia (având în vedere că fiecare produs este unicat, acesta nu poate corespunde mai multor comenzi). Astfel, cardinalitatea COMANDA-PRODUS este  $1(0):M(0)$ .

4) Entitatea FEEDBACK are o relație cu entitatea COMANDA, de tip „one-to-one”, detaliată anterior.

5) Entitatea PRODUS are o relație cu entitatea COMANDA, de tip „many-to-one”, detaliată anterior, dar și relații cu COLECTIE, SARCINA, și TIPAR.

Relația PRODUS-COLECTIE („many-to-one”): un produs poate aparține unei singure colecții sau niciuneia, fiind de sine stătător, iar o colecție poate avea mai multe produse. Astfel, cardinalitatea PRODUS-COLECTIE este  $M(0):1(0)$ .

Relația PRODUS-SARCINA („one-to-many”): un produs poate necesita mai multe sarcini, sau niciuna, iar o sarcină poate corespunde unui produs, sau poate să nu corespundă niciunuia. Astfel, cardinalitatea PRODUS-SARCINA este 1(0):M(0).

Relația PRODUS-TIPAR („one-to-many”): un produs poate avea nevoie de tipare, sau nu, iar un tipar poate să corespundă unui produs, sau niciunuia, dar tot să existe în baza de date. Astfel, cardinalitatea PRODUS-TIPAR este 1(0):M(0).

6) Entitatea COLECTIE are o relație cu PRODUS, de tip „one-to-many”, detaliată anterior.

7) Entitatea SARCINA are o relație cu PRODUS, de tip „many-to-one”, detaliată anterior, dar și relații cu CROITOR și DESIGNER. De menționat: entitatea SARCINA apare ca tabel asociativ între PRODUS, CROITOR și DESIGNER.

Relația SARCINA-CROITOR („many-to-one”): o sarcină îi este atribuită unui singur creator, sau niciunuia, dacă nu necesită unul. Un creator poate primi mai multe sarcini, sau poate să nu aibă niciuna. Astfel, cardinalitatea SARCINA-CROITOR este M(0):1(0).

Relația SARCINA-DESIGNER („many-to-one”): o sarcină îi este atribuită unui singur designer, sau niciunuia, dacă nu necesită unul. Un designer poate primi mai multe sarcini, sau poate să nu aibă niciuna. Astfel, cardinalitatea SARCINA-DESIGNER este M(0):1(0).

8) Entitatea ANGAJAT are relații de tip „one-to-one” (IsA) cu CROITOR și cu DESIGNER, întrucât CROITOR și DESIGNER sunt subtipuri de ANGAJAT, deci subentități.

9) Entitatea CROITOR are o relație cu SARCINA de tip „one-to-many”, detaliată anterior.

10) Entitatea DESIGNER are o relație cu SARCINA de tip „one-to-many”, detaliată anterior, dar și cu TIPAR.

Relația DESIGNER-TIPAR („one-to-many”): un designer poate face mai multe tipare, sau poate să nu facă niciunul, iar un tipar poate corespunde unui singur designer, sau niciunuia. Astfel, cardinalitatea DESIGNER-TIPAR este 1(0):M(0).

11) Entitatea TIPAR are relații de tip „many-to-one” cu PRODUS și DESIGNER, detaliate anterior, dar și cu TIPAR.

Relația TIPAR-MATERIAL („many-to-one”): un tipar poate avea nevoie de un singur material, sau de niciunul (caz în care este interpretat ca fiind schiță), iar un material poate fi necesar mai multor tipare, sau niciunuia, în cazul în care nu a fost folosit încă. Astfel, cardinalitatea TIPAR-MATERIAL este  $M(0):1(0)$ .

12) Entitatea MATERIAL are o relație de tip „one-to-many” cu TIPAR, detaliată anterior, dar și cu CONTRACT.

Relația MATERIAL-CONTRACT („one-to-many”): un material poate fi achiziționat prin intermediul mai multor contracte (implicit, de la mai mulți furnizori), iar printr-un contract se poate achiziționa un material, și numai unul. Astfel, cardinalitatea MATERIAL-CONTRACT este  $1(1):M(0)$ .

13) Entitatea CONTRACT are o relație de tip „many-to-one” cu MATERIAL, detaliată anterior, dar și cu FURNIZOR. De menționat: entitatea CONTRACT apare ca tabel asociativ între MATERIAL și FURNIZOR.

Relația CONTRACT-FURNIZOR („many-to-one”): un contract poate fi încheiat cu un singur furnizor, iar cu un furnizor pot fi semnate mai multe contracte, sau niciunul, dacă încă nu au fost făcute achiziții de la acesta. Astfel, cardinalitatea CONTRACT-FURNIZOR este  $M(0):1(1)$ .

14) Entitatea FURNIZOR are o relație de tip „one-to-many” cu CONTRACT, detaliată anterior.

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

### 1) Tabelul ADRESE

- *id\_adresa* (PK, INT): cheia primară a tabelului ADRESE
- *tara* (VARCHAR2(40)): reprezintă țara adresei, reținută ca șir de cel mult 40 de caractere
- *judet* (VARCHAR2(40)): reprezintă județul adresei, reținut ca șir de cel mult 40 de caractere
- *oras* (VARCHAR2(40)): reprezintă orașul adresei, reținut ca șir de cel mult 40 de caractere
- *strada* (VARCHAR2(40)): reprezintă strada adresei, reținută ca șir de cel mult 40 de caractere
- *numar* (INT): numărul locuinței pe strada, reținut ca număr întreg pozitiv (număr natural)
- *scara* (INT): numărul scării în bloc (corespunzător coloanei *numar*), reținut ca număr întreg pozitiv (număr natural) sau null, dacă nu există
- *apartament* (INT): numărul apartamentului pe scară (corespunzător coloanei *scara*), reținut ca număr întreg pozitiv (număr natural) sau null, dacă nu există
- *id\_client* (FK, INT): cheie străină către tabelul CLIENTI, opțională (poate fi null), nu face parte din cheia primară

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID_ADRESA	NUMBER	No	(null)	1 (null)	
2	TARA	VARCHAR2(40 BYTE)	Yes	(null)	2 (null)	
3	JUDET	VARCHAR2(40 BYTE)	Yes	(null)	3 (null)	
4	ORAS	VARCHAR2(40 BYTE)	Yes	(null)	4 (null)	
5	STRADA	VARCHAR2(40 BYTE)	Yes	(null)	5 (null)	
6	NUMAR	NUMBER	Yes	(null)	6 (null)	
7	SCARA	NUMBER	Yes	(null)	7 (null)	
8	APARTAMENT	NUMBER	Yes	(null)	8 (null)	
9	ID_CLIENT	NUMBER	Yes	(null)	9 (null)	

	❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE	❖ STATUS	❖ DEFERRABLE	❖ VALIDATED
1	FK_ADRESE_CLIENTI	Foreign_Key	(null)	DANIEL	CLIENTI	SYS_C008318	NO ACTION	ENABLED	NOT DEFERRABLE	VALIDATED
2	SYS_C008319	Check	NUMAR > 0	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
3	SYS_C008320	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED



## 2) Tabelul CLIENTI

- *id\_client* (PK, INT): cheia primară a tabelului CLIENTI
- *nume* (VARCHAR2(30)): reprezintă numele clientului, reținut ca șir de cel mult 30 de caractere
- *prenume* (VARCHAR2(30)): reprezintă prenumele clientului, reținut ca șir de cel mult 30 de caractere
- *email* (VARCHAR2(40)): reprezintă emailul clientului, reținut ca șir de cel mult 40 de caractere
- *nr\_tel* (VARCHAR2(15)): reprezintă numărul de telefon al clientului, reținut ca șir de cel mult 15 caractere

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_CLIENT		NUMBER		No	(null)		1		(null)
2	NUME		VARCHAR2 (30 BYTE)		Yes	(null)		2		(null)
3	PRENUME		VARCHAR2 (30 BYTE)		Yes	(null)		3		(null)
4	EMAIL		VARCHAR2 (40 BYTE)		Yes	(null)		4		(null)
5	NR_TEL		VARCHAR2 (15 BYTE)		Yes	(null)		5		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE	❖	VALIDATED
1	SYS_C008318		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE		VALIDATED

## 3) Tabelul COMENZI

- *id\_comanda* (PK, INT): cheia primară a tabelului COMENZI
- *pret* (DECIMAL(10,2)): prețul unei comenzi, reținut ca număr real cu două zecimale
- *livrare* (ENUM('curier', 'posta', 'personal')): tipul de livrare a comenzii, reținut ca tip enumerat cu valorile 'curier', 'posta' sau 'personal'
- *stare* (ENUM('preluata', 'in lucru', 'finalizata', 'livrata')): starea comenzii, reținută ca tip enumerat cu valorile 'preluata', 'in lucru', 'finalizata' sau 'livrata'
- *id\_client* (FK, INT): cheie străină către tabelul CLIENTI, obligatorie (nu poate fi null), nu face parte din cheia primară

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_COMANDA		NUMBER		No	(null)		1		(null)
2	PRET		NUMBER (10,2)		Yes	(null)		2		(null)
3	LIVRARE		VARCHAR2 (10 BYTE)		Yes	(null)		3		(null)
4	STARE		VARCHAR2 (10 BYTE)		Yes	(null)		4		(null)
5	ID_CLIENT		NUMBER		No	(null)		5		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME
1	FK_COMENZI_CLIENTI		Foreign_Key	(null)		DANIEL		CLIENTI		SYS_C00
2	SYS_C008322		Check	"ID_CLIENT" IS NOT NULL		(null)		(null)		(null)
3	SYS_C008323		Check	UPPER(LIVRARE) IN ('CURIER', 'POSTA', 'PERSONAL')		(null)		(null)		(null)
4	SYS_C008324		Check	UPPER(STARE) IN ('PRELUATA', 'IN LUCRU', 'FINALIZATA', 'LIVRATA')		(null)		(null)		(null)
5	SYS_C008325		Primary_Key	(null)		(null)		(null)		(null)

#### 4) Tabelul FEEDBACKURI

- *id\_feedback* (PK, INT): cheia primară a tabelului FEEDBACKURI
- *nota* (INT): nota oferită în feedback, reținută ca număr întreg pozitiv (număr natural) mai mic sau egal cu 10
- *descriere* (VARCHAR2(500)): reprezintă descrierea feedbackului, reținută ca șir de cel mult 500 de caractere
- *id\_comanda* (FK, INT): cheie străină către tabelul COMENZI, obligatorie (nu poate fi null), nu face parte din cheia primară

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_FEEDBACK	NUMBER	No	(null)	1 (null)	
2 NOTA	NUMBER	Yes	(null)	2 (null)	
3 DESCRIERE	VARCHAR2(500 BYTE)	Yes	(null)	3 (null)	
4 ID_COMANDA	NUMBER	No	(null)	4 (null)	

❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE	❖ STATUS
1 FK_FEEDBACKURI_COMENZI	Foreign_Key	(null)	DANIEL	COMENZI	SYS_C008325	NO ACTION	ENABLED
2 SYS_C008327	Check	"ID_COMANDA" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
3 SYS_C008328	Check	NOTA > 0 AND NOTA <= 10	(null)	(null)	(null)	(null)	ENABLED
4 SYS_C008329	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED

#### 5) Tabelul PRODUSE

- *id\_produs* (PK, INT): cheia primară a tabelului PRODUSE
- *nume* (VARCHAR2(30)): reprezintă numele produsului, reținut ca șir de cel mult 30 de caractere
- *pret* (DECIMAL(10,2)): prețul produsului, reținut ca număr real cu două zecimale
- *descriere* (VARCHAR2(100)): reprezintă descrierea produsului, reținută ca șir de cel mult 100 de caractere
- *id\_comanda* (FK, INT): cheie străină către tabelul COMENZI, opțională (poate fi null), nu face parte din cheia primară
- *id\_colectie* (FK, INT): cheie străină către tabelul COLECTII, opțională (poate fi null), nu face parte din cheia primară

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_PRODUS	NUMBER	No	(null)	1 (null)	
2 NUME	VARCHAR2(30 BYTE)	Yes	(null)	2 (null)	
3 PRET	NUMBER(10,2)	Yes	(null)	3 (null)	
4 DESCRIERE	VARCHAR2(100 BYTE)	Yes	(null)	4 (null)	
5 ID_COMANDA	NUMBER	Yes	(null)	5 (null)	
6 ID_COLECTIE	NUMBER	Yes	(null)	6 (null)	

❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE	❖ STATUS	❖ DEFERRABLE
1 FK_PRODUSE_COLECTII	Foreign_Key	(null)	DANIEL	COLECTII	SYS_C008332	NO ACTION	ENABLED	NOT DEFERRABLE
2 FK_PRODUSE_COMENZI	Foreign_Key	(null)	DANIEL	COMENZI	SYS_C008325	NO ACTION	ENABLED	NOT DEFERRABLE
3 SYS_C008341	Check	PRET > 0	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
4 SYS_C008342	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

## 6) Tabelul COLECTII

- *id\_colectie* (PK, INT): cheia primară a tabelului COLECTII
- *nume* (VARCHAR2(60)): reprezintă numele colecției, reținut ca șir de cel mult 60 de caractere
- *tematica* (VARCHAR2(60)): reprezintă tema unei colecții, reținută ca șir de cel mult 60 de caractere
- *an* (INT): anul colecției, reținut ca număr întreg pozitiv (număr natural)
- *anotimp* (VARCHAR2(30)): reprezintă anotimpul sau sezonul colecției, reținut ca șir de cel mult 30 de caractere

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_COLECTIE		NUMBER		No	(null)		1		(null)
2	NUME		VARCHAR2(60 BYTE)		Yes	(null)		2		(null)
3	TEMATICA		VARCHAR2(60 BYTE)		Yes	(null)		3		(null)
4	AN		NUMBER		Yes	(null)		4		(null)
5	ANOTIMP		VARCHAR2(30 BYTE)		Yes	(null)		5		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE	❖	VALIDATED
1	SYS_C008331		Check	an > 1999		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE		VALIDATED
2	SYS_C008332		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE		VALIDATED

## 7) Tabelul SARCINI

- *id\_sarcina* (PK, INT): cheia primară a tabelului SARCINI
- *descriere* (VARCHAR2(100)): reprezintă descrierea sarcinii, reținută ca șir de cel mult 100 de caractere trebuie să fie un șir de caractere cu lungime maximă de 100 de caractere
- *id\_croitor* (FK, INT): cheie străină către tabelul CROITORI, opțională (poate fi null), nu face parte din cheia primară
- *id\_designer* (FK, INT): cheie străină către tabelul DESIGNERI, opțională (poate fi null), nu face parte din cheia primară
- *id\_produs* (FK, INT): cheie străină către tabelul PRODUSE, opțională (poate fi null), nu face parte din cheia primară

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_SARCINA		NUMBER		No	(null)		1		(null)
2	DESCRIERE		VARCHAR2(100 BYTE)		Yes	(null)		2		(null)
3	ID_CROITOR		NUMBER		Yes	(null)		3		(null)
4	ID_DESIGNER		NUMBER		Yes	(null)		4		(null)
5	ID_PRODUS		NUMBER		Yes	(null)		5		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE	❖	VALIDATED
1	FK_SARCINI_CROITORI		Foreign_Key	(null)		DANIEL		CROITORI		UNIQUE_ID_CROITOR		NO ACTION		ENABLED		NOT DEFERRABLE		VALIDATED
2	FK_SARCINI_DESIGNERI		Foreign_Key	(null)		DANIEL		DESIGNERI		UNIQUE_ID_DESIGNER		NO ACTION		ENABLED		NOT DEFERRABLE		VALIDATED
3	FK_SARCINI_PRODUSE		Foreign_Key	(null)		DANIEL		PRODUSE		SYS_C008342		NO ACTION		ENABLED		NOT DEFERRABLE		VALIDATED
4	SYS_C008358		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE		VALIDATED

## 8) Tabelul ANGAJATI

- *id\_angajat* (PK, INT): cheia primară a tabelului ANGAJATI
- *nume* (VARCHAR2(30)): reprezintă numele angajatului, reținut ca șir de cel mult 30 de caractere
- *prenume* (VARCHAR2(30)): reprezintă prenumele angajatului, reținut ca șir de cel mult 30 de caractere
- *tip* (VARCHAR2(30)): reprezintă tipul angajatului (fondator, colaborator, etc), reținut ca șir de cel mult 30 de caractere
- *email* (VARCHAR2(40)): reprezintă emailul angajatului, reținut ca șir de cel mult 40 de caractere
- *nr\_tel* (VARCHAR2(15)): reprezintă numărul de telefon al angajatului, reținut ca șir de cel mult 15 caractere
- *salariu* (INT): salariul angajatului, reținut ca număr întreg pozitiv (număr natural)
- *comision* (DECIMAL(10,2)): comisionul angajatului, reținut ca număr real pozitiv cu două zecimale (inclusiv 0)
- *data\_angajare* (DATE): data angajării angajatului, reținută ca dată

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_ANGAJAT	NUMBER	No	(null)	1 (null)	
2 NUME	VARCHAR2(30 BYTE)	Yes	(null)	2 (null)	
3 PRENUME	VARCHAR2(30 BYTE)	Yes	(null)	3 (null)	
4 TIP	VARCHAR2(30 BYTE)	Yes	(null)	4 (null)	
5 EMAIL	VARCHAR2(40 BYTE)	Yes	(null)	5 (null)	
6 NR_TEL	VARCHAR2(15 BYTE)	Yes	(null)	6 (null)	
7 SALARIU	NUMBER(38,0)	Yes	(null)	7 (null)	
8 COMISION	NUMBER(10,2)	Yes	(null)	8 (null)	
9 DATA_ANGAJARE	DATE	Yes	(null)	9 (null)	

❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE	❖ STATUS
1 SYS_C008345	Check	SALARIU > 0	(null)	(null)	(null)	(null)	ENABLED
2 SYS_C008346	Check	COMISION >= 0 AND COMISION <=100	(null)	(null)	(null)	(null)	ENABLED
3 SYS_C008347	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED

## 9) Tabelul CROITORI

- *id\_croitor* (PK, INT): cheia primară a tabelului CROITORI
- *id\_angajat* (FK, INT): cheie străină către tabelul ANGAJATI, obligatorie (nu poate fi null), face parte din cheia primară

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_CROITOR	NUMBER	No	(null)	1 (null)	
2 ID_ANGAJAT	NUMBER	No	(null)	2 (null)	

❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE	❖ STATUS	❖ DEFERRABLE
1 FK_CROITORI_ANGAJATI	Foreign_Key	(null)	DANIEL	ANGAJATI	SYS_C008347	NO ACTION	ENABLED	NOT DEFERRABLE
2 SYS_C008355	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
3 UNIQUE_ID_CROITOR	Unique	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

## 10) Tabelul DESIGNERI

- *id\_designer* (PK, INT): cheia primară a tabelului DESIGNERI
- *id\_angajat* (FK, INT): cheie străină către tabelul ANGAJATI, obligatorie (nu poate fi null), face parte din cheia primară

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_DESIGNER		NUMBER		No	(null)		1		(null)
2	ID_ANGAJAT		NUMBER		No	(null)		2		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE
1	FK_DESIGNER_ANGAJATI		Foreign_Key	(null)		DANIEL		ANGAJATI		SYS_C008347		NO ACTION		ENABLED		NOT DEFERRABLE
2	SYS_C008352		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE
3	UNIQUE_ID_DESIGNER		Unique	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE

## 11) Tabelul TIPARE

- *id\_tipar* (PK, INT): cheia primară a tabelului TIPARE
- *lungime* (INT): lungimea tiparului, reținută ca număr întreg pozitiv (număr natural)
- *lățime* (INT): lățimea tiparului, reținută ca număr întreg pozitiv (număr natural)
- *id\_designer* (FK, INT): cheie străină către tabelul DESIGNERI, opțională (poate fi null), nu face parte din cheia primară
- *id\_produs* (FK, INT): cheie străină către tabelul PRODUSE, opțională (poate fi null), nu face parte din cheia primară
- *id\_material* (FK, INT): cheie străină către tabelul MATERIALE, opțională (poate fi null), nu face parte din cheia primară

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_TIPAR		NUMBER		No	(null)		1		(null)
2	LUNGIME		NUMBER(38,0)		Yes	(null)		2		(null)
3	LATIME		NUMBER(38,0)		Yes	(null)		3		(null)
4	ID_DESIGNER		NUMBER		Yes	(null)		4		(null)
5	ID_PRODUS		NUMBER		Yes	(null)		5		(null)
6	ID_MATERIAL		NUMBER		Yes	(null)		6		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE
1	FK_TIPARE_DESIGNERI		Foreign_Key	(null)		DANIEL		DESIGNERI		UNIQUE_ID_DESIGNER		NO ACTION		ENABLED		NOT DEFERRABLE
2	FK_TIPARE_MATERIALE		Foreign_Key	(null)		DANIEL		MATERIALE		SYS_C008362		NO ACTION		ENABLED		NOT DEFERRABLE
3	FK_TIPARE_PRODUSE		Foreign_Key	(null)		DANIEL		PRODUSE		SYS_C008342		NO ACTION		ENABLED		NOT DEFERRABLE
4	SYS_C008363		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE

## 12) Tabelul MATERIALE

- *id\_material* (PK, INT): cheia primară a tabelului MATERIALE
- *nume* (VARCHAR2(30)): reprezintă numele materialului, reținut ca șir de cel mult 30 de caractere
- *tip* (VARCHAR2(30)): reprezintă tipul materialului, reținut ca șir de cel mult 30 de caractere
- *compozitie* (VARCHAR2(100)): reprezintă compoziția materialului, reținută ca șir de cel mult 100 de caractere

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_MATERIAL		NUMBER		No	(null)		1		(null)
2	NUME		VARCHAR2(30 BYTE)		Yes	(null)		2		(null)
3	TIP		VARCHAR2(30 BYTE)		Yes	(null)		3		(null)
4	COMPOZITIE		VARCHAR2(100 BYTE)		Yes	(null)		4		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE
1	SYS_C008362		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE

## 13) Tabelul CONTRACTE

- *id\_contract* (PK, INT): cheia primară a tabelului CONTRACTE
- *data* trebuie să fie o dată calendaristică
- *id\_furnizor* (FK, INT): cheie străină către tabelul FURNIZORI, obligatorie (nu poate fi null), face parte din cheia primară
- *id\_material* (FK, INT): cheie străină către tabelul MATERIALE, obligatorie (nu poate fi null), face parte din cheia primară

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_CONTRACT		NUMBER		No	(null)		1		(null)
2	DATA		DATE		Yes	(null)		2		(null)
3	ID_FURNIZOR		NUMBER		No	(null)		3		(null)
4	ID_MATERIAL		NUMBER		No	(null)		4		(null)

❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE
1	FK_CONTRACTE_FURNIZORI		Foreign_Key	(null)		DANIEL		FURNIZORI		SYS_C008368		NO ACTION		ENABLED		NOT DEFERRABLE
2	FK_CONTRACTE_MATERIALE		Foreign_Key	(null)		DANIEL		MATERIALE		SYS_C008362		NO ACTION		ENABLED		NOT DEFERRABLE
3	SYS_C008369		Check	"ID_CONTRACT" IS NOT NULL		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE
4	SYS_C008370		Check	"ID_FURNIZOR" IS NOT NULL		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE
5	SYS_C008371		Check	"ID_MATERIAL" IS NOT NULL		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE
6	SYS_C008372		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT DEFERRABLE

#### 14) Tabelul FURNIZORI

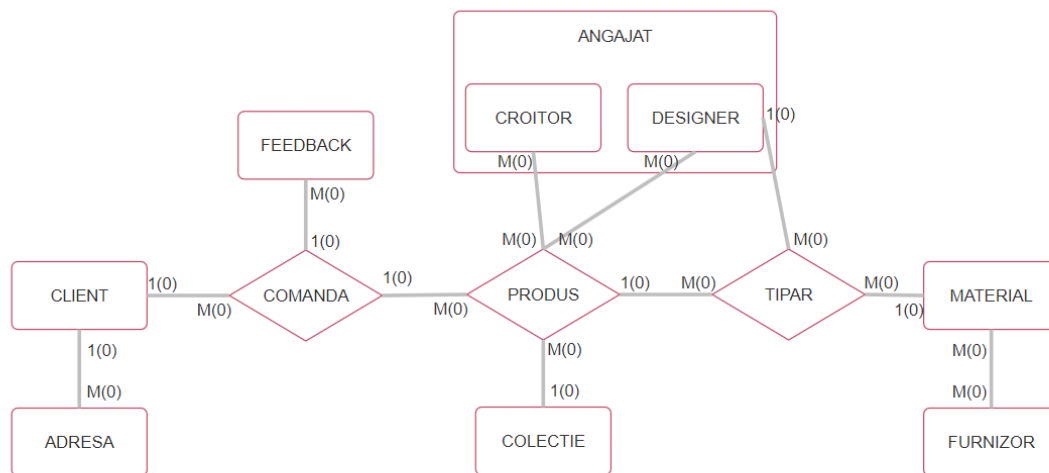
- *id\_furnizor* (PK, INT): cheia primară a tabelului FURNIZORI
- *nume* (VARCHAR2(50)): reprezintă numele furnizorului, reținut ca șir de cel mult 50 de caractere
- *cod\_fiscal* (INT): codul fiscal al furnizorului, reținut ca număr întreg pozitiv (număr natural)

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS	
1	ID_FURNIZOR		NUMBER		No	(null)		1		(null)	
2	NUME		VARCHAR2(50 BYTE)		Yes	(null)		2		(null)	
3	COD_FISCAL		NUMBER(38,0)		Yes	(null)		3		(null)	

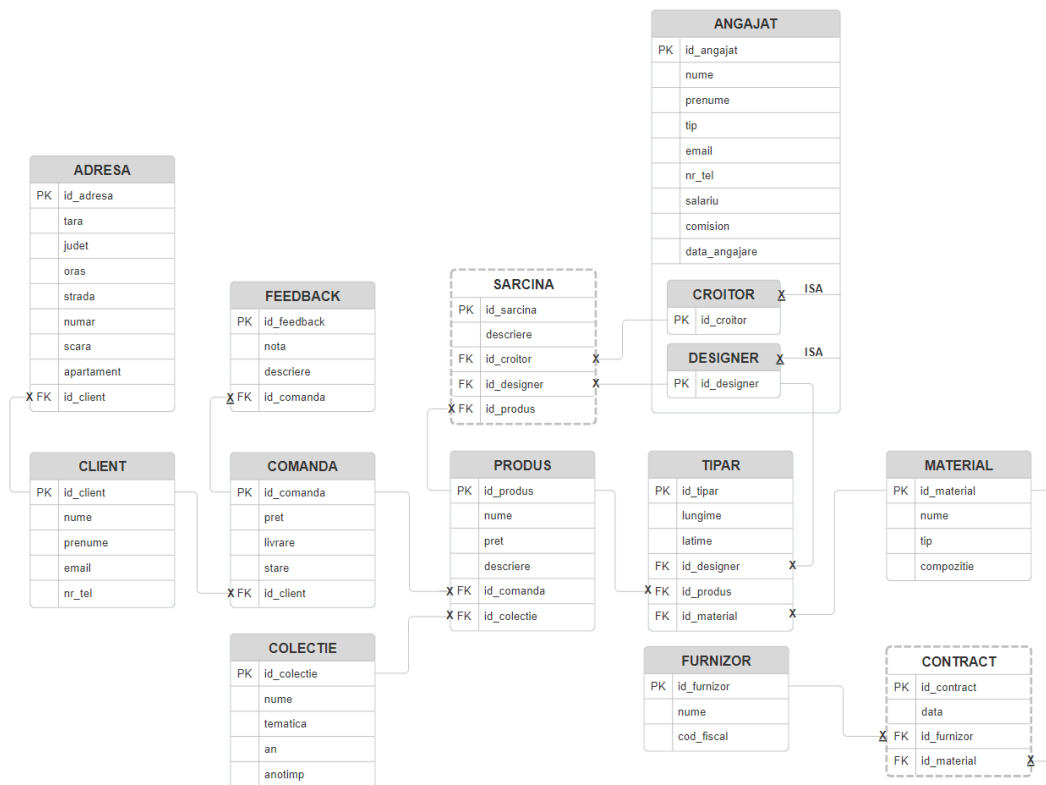
❖	CONSTRAINT_NAME	❖	CONSTRAINT_TYPE	SEARCH_CONDITION	❖	R_OWNER	❖	R_TABLE_NAME	❖	R_CONSTRAINT_NAME	❖	DELETE_RULE	❖	STATUS	❖	DEFERRABLE	❖	VALID
1	SYS_C008367		Check	COD_FISCAL > 0		(null)		(null)		(null)		(null)		ENABLED		NOT	DEFERRABLE	VALID
2	SYS_C008368		Primary_Key	(null)		(null)		(null)		(null)		(null)		ENABLED		NOT	DEFERRABLE	VALID

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





7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



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

- Enumerarea schemelor relaționale:

- 1) ADRESA (id\_adresa, tara, judet, oras, strada, numar, scara, apartament, id\_client)
- 2) CLIENT (id\_client, nume, prenume, email, nr\_tel)
- 3) COMANDA (id\_comanda, pret, livrare, stare, id\_client)
- 4) FEEDBACK (id\_feedback, nota, descriere, id\_comanda)
- 5) PRODUS (id\_produc, nume, tip, pret, descriere, id\_comanda, id\_colectie)
- 6) COLECTIE (id\_colectie, nume, tematica, an, anotimp)
- 7) SARCINA (id\_sarcina, descriere, id\_croitor, id\_designer, id\_produc)
- 8) ANGAJAT (id\_angajat, nume, prenume, tip, email, nr\_tel, salariu, commission, data\_angajare)
- 9) CROITOR (id\_croitor, id\_angajat)
- 10) DESIGNER (id\_designer, id\_angajat)
- 11) TIPAR (id\_tipar, lungime, latime)
- 12) MATERIAL (id\_material, nume, tip, compozitie)
- 13) CONTRACT (id\_contract, id\_furnizor, id\_material, data)
- 14) FURNIZOR (id\_furnizor, nume, cod\_fiscal)

\*mențiune: dacă o cheie străină a fost scrisă imediat după cheia primară, înseamnă că aceasta face parte din cheia primară, iar dacă a fost scrisă la final, înseamnă că nu face parte din cheia primară

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

### Forma normală 1

O entitate se găsește în prima formă normală dacă și numai dacă nu există attribute cu valori multiple și nu există attribute sau grupuri de attribute care se repetă.

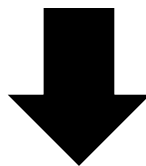
Exemplu pentru NON-FN1: relația MATERIALE – FURNIZORI

Inițial, tabelul MATERIALE conținea o coloană unde erau stocați furnizorii corespunzătoru, iar tabelul FURNIZORI conținea o coloană unde erau stocate materialele corespunzătoare.

Pentru rezolvarea problemei, se va crea tabelul CONTRACTE, ce reprezintă intersecția celor două tabele. Transformarea în FN1 se va face astfel:

MATERIALE	
id_material	id_furnizor
1	2, 4
2	1, 4

FURNIZORI	
id_furnizor	id_material
1	2
2	1
4	1, 2



MATERIALE	
id_material	
1	
2	

CONTRACTE		
id_contract	id_material (FK)	id_furnizor (FK)
1	1	2
2	1	4
3	2	1
4	2	4

FURNIZORI
id_furnizor
1
2
4

În acest mod, prin crearea tabelului CONTRACTE, care este o listă de asocieri între MATERIALE și FURNIZORI, se va rezolva problematica atributelor multiple. Cheia primară a acestui tabel va fi alcătuită din id\_contract și cele două chei străine: id\_material și id\_furnizor.

### Forma normală 2

O entitate se găsește în a doua formă normală dacă și numai dacă se găsește în prima formă normală (FN1) și, în plus, orice atribut care nu face parte din cheia primară va depinde de întreaga cheie primară, nu doar de o parte a acesteia.

Exemplu pentru NON-FN2: tabelul PRODUSE.

Inițial, tabelul produs conținea două chei primare: id\_produș și id\_colecție, cu diverse atribute. O parte din acestea (de exemplu, numele, tematica, anul și anotimpul colecției) nu depind de întreaga cheie primară, ci doar de id\_colecție.

Tabelul se află totuși în FN1, deoarece respectă regulile acestei forme normale, neavând atribute multiple.

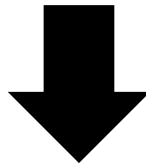
Pentru a trece entitatea în FN2, orice atribut care nu face parte din cheia primară trebuie să depindă de întreaga cheie primară.

Voi separa attributele menționate în paranteză anterior, alături de id\_colecție, și voi crea cu acestea un nou tabel: COLECTII, pe care îl voi conecta de tabelul PRODUSE prin adăugarea unei chei străine în tabelul PRODUSE, denumită id\_colecție, care nu va face parte din cheia primară.

Pentru a simplifica exemplul, voi scrie doar un atribut corespunzător integral tabelului produs (în afară de id): nume.

Transformarea în FN2 se face astfel:

PRODUSE						
id_produș	id_colecție	nume_produș	nume_colecție	tematica	an	anotimp
1	3	Valuri	Natura	Abstract	2020	Vara
2	4	Agata	Urban	Streetwear	2023	Iarna



PRODUSE		
id_produș	nume	id_colecție
1	Valuri	3
2	Agata	4

COLECTII				
id_colecție	nume	tematica	an	anotimp
3	Natura	Abstract	2020	Vara
4	Urban	Streetwear	2023	Iarna

În acest mod, prin crearea tabelului COLECTIE, va fi rezolvată problema dependențelor parțiale față de cheia primară în tabelul PRODUSE.

### Forma normală 3

O entitate se găsește în a treia formă normală dacă și numai dacă se găsește în a doua formă normală și în plus niciun atribut care nu este parte a cheii primare nu depinde de un alt atribut care nu este cheie primară. Cu alte cuvinte, nu se acceptă dependențe tranzitive, adică un atribut să depindă de cheia primară în mod indirect.

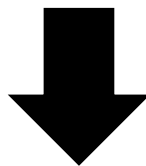
Exemplu pentru NON-FN3: tabelul PRODUSE.

Inițial (după modificările FN2), tabelul PRODUSE conținea, pe lângă cheia primară *id\_produș*, și atribute specifice, atributele (coloanele) „sarcina” și „descriere\_sarcina”. Totuși, *descriere\_sarcina* depinde de *sarcina*, nu de *id\_produș*, astfel că nu se încadrează în FN3, deoarece între *descriere\_sarcina* și *id\_produș* există o dependență tranzitivă, prin coloana *sarcina*. (coloana *sarcina* nu face parte din cheia primară)

Astfel, voi crea un nou tabel, SARCINI, care va conține descrierea acestuia, unde înregistrările vor fi identificate unic prin id\_sarcina. Aceasta va conține și o cheie străină către tabelul PRODUSE, pentru a păstra legătura cu acel tabel.

Pentru a simplifica exemplul, voi scrie doar un atribut corespunzător integral tabelului produs (în afară de id): nume.

PRODUSE			
id_produș	nume	sarcina	descriere_sarcina
1	Valuri	1	Coasere tiv circular
2	Agata	2	Decupare tipare pentru masurare



PRODUSE	
id_produș	nume
1	Valuri
2	Agata

SARCINI		
id_sarcina	descriere	id_produș
1	Coasere tiv circular	1
2	Decupare tipare pentru masurare	2

În acest mod, prin crearea tabelului SARCINI, sunt eliminate dependențele tranzitive, și implicit tabelele se află în FN3.

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

▪ SQL:

```
CREATE SEQUENCE SEQ_ADRESE START WITH 1;
```

```
CREATE SEQUENCE SEQ_CLIENTI START WITH 1;
```

```
CREATE SEQUENCE SEQ_COMENZI START WITH 1;
```

```
CREATE SEQUENCE SEQ_FEEDBACKURI START WITH 1;
```

```
CREATE SEQUENCE SEQ_PRODUSE START WITH 1;
```

```
CREATE SEQUENCE SEQ_COLECTII START WITH 1;
```

```
CREATE SEQUENCE SEQ_SARCINI START WITH 1;
```

```
CREATE SEQUENCE SEQ_ANGAJATI START WITH 1;
```

```
CREATE SEQUENCE SEQ_CROITORI START WITH 1;
```

```
CREATE SEQUENCE SEQ_DESIGNERI START WITH 1;
```

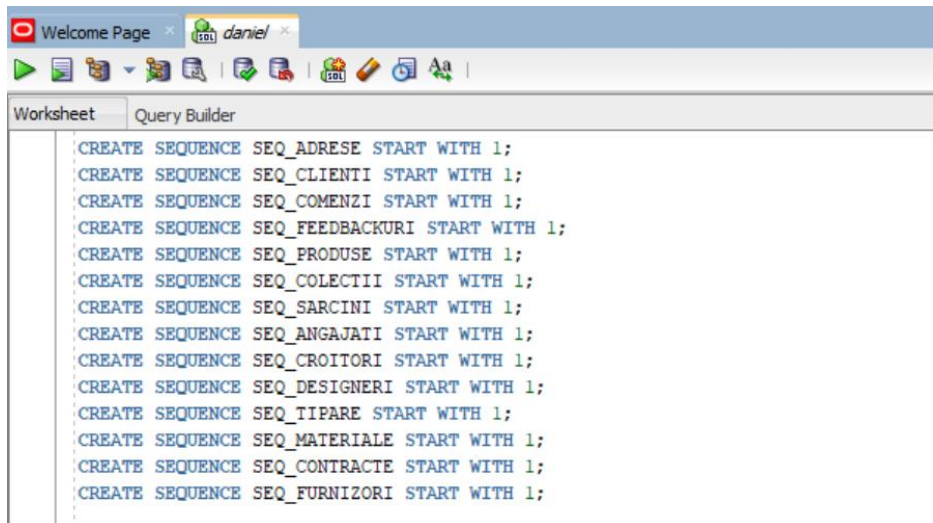
```
CREATE SEQUENCE SEQ_TIPARE START WITH 1;
```

```
CREATE SEQUENCE SEQ_MATERIALE START WITH 1;
```

```
CREATE SEQUENCE SEQ_CONTRACTE START WITH 1;
```

```
CREATE SEQUENCE SEQ_FURNIZORI START WITH 1;
```

- Codul sursă:



The screenshot shows a database query editor window with a toolbar and a 'Query Builder' tab. The query text is as follows:

```
CREATE SEQUENCE SEQ_ADRESE START WITH 1;  
CREATE SEQUENCE SEQ_CLIENTII START WITH 1;  
CREATE SEQUENCE SEQ_COMENZI START WITH 1;  
CREATE SEQUENCE SEQ_FEEDBACKURI START WITH 1;  
CREATE SEQUENCE SEQ_PRODUSE START WITH 1;  
CREATE SEQUENCE SEQ_COLECTII START WITH 1;  
CREATE SEQUENCE SEQ_SARCINI START WITH 1;  
CREATE SEQUENCE SEQ_ANGAJATI START WITH 1;  
CREATE SEQUENCE SEQ_CROITORI START WITH 1;  
CREATE SEQUENCE SEQ_DESIGNERI START WITH 1;  
CREATE SEQUENCE SEQ_TIPARE START WITH 1;  
CREATE SEQUENCE SEQ_MATERIALE START WITH 1;  
CREATE SEQUENCE SEQ_CONTRACTE START WITH 1;  
CREATE SEQUENCE SEQ_FURNIZORI START WITH 1;
```

- Execuție cod sursă:

Sequence SEQ\_ADRESE created.

Sequence SEQ\_CLIENTII created.

Sequence SEQ\_COMENZI created.

Sequence SEQ\_FEEDBACKURI created.

Sequence SEQ\_PRODUSE created.

Sequence SEQ\_COLECTII created.

Sequence SEQ\_SARCINI created.

Sequence SEQ\_ANGAJATI created.

Sequence SEQ\_CROITORI created.

Sequence SEQ\_DESIGNERI created.

Sequence SEQ\_TIPARE created.

Sequence SEQ\_MATERIALE created.

Sequence SEQ\_CONTRACTE created.

Sequence SEQ\_FURNIZORI created.



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

\* pentru a nu îngreuna codul, tabelele au fost introduse în altă ordine față de cea în care au fost descrise, pentru a nu necesita ALTER

1) Tabelul CLIENTI:

▪ SQL:

```
CREATE TABLE CLIENTI (  
    ID_CLIENT NUMBER PRIMARY KEY,  
    NUME VARCHAR2(30),  
    PRENUME VARCHAR2(30),  
    EMAIL VARCHAR2(40),  
    NR_TEL VARCHAR2(15)  
);  
  
INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, EMAIL, NR_TEL)  
VALUES (SEQ_CLIENTI.NEXTVAL, 'Stoivici', 'Alexandra', 'stoivici.alexandra@gmail.com',  
'+40711234567');  
  
INSERT INTO CLIENTI  
VALUES (SEQ_CLIENTI.NEXTVAL, 'Ionescu', 'Lidia', 'ionescu.lidia@gmail.com',  
'+40730097890');  
  
INSERT INTO CLIENTI  
VALUES (SEQ_CLIENTI.NEXTVAL, 'Popescu', 'Florin', 'popescu.florin@gmail.com',  
'+40799221443');  
  
INSERT INTO CLIENTI  
VALUES (SEQ_CLIENTI.NEXTVAL, 'Vasile', 'Cristian', 'vasile.cristian@gmail.com',  
'+40765222437');  
  
INSERT INTO CLIENTI  
VALUES (SEQ_CLIENTI.NEXTVAL, 'Marin', 'Mihaela', 'marin.mihaela@gmail.com',  
'+40766908123');
```

INSERT INTO CLIENTI

VALUES (SEQ\_CLIENTI.NEXTVAL, 'Rusu', 'Ana-Maria', 'rusu.anamaria@gmail.com',  
'+40711654890');

INSERT INTO CLIENTI

VALUES (SEQ\_CLIENTI.NEXTVAL, 'Vacaru', 'Marta', 'vacaru.marta@gmail.com',  
'+40760676248');

INSERT INTO CLIENTI

VALUES (SEQ\_CLIENTI.NEXTVAL, 'Nicolae', 'Vladimir', 'nicolae.vladimir@gmail.com',  
'+40780091421');

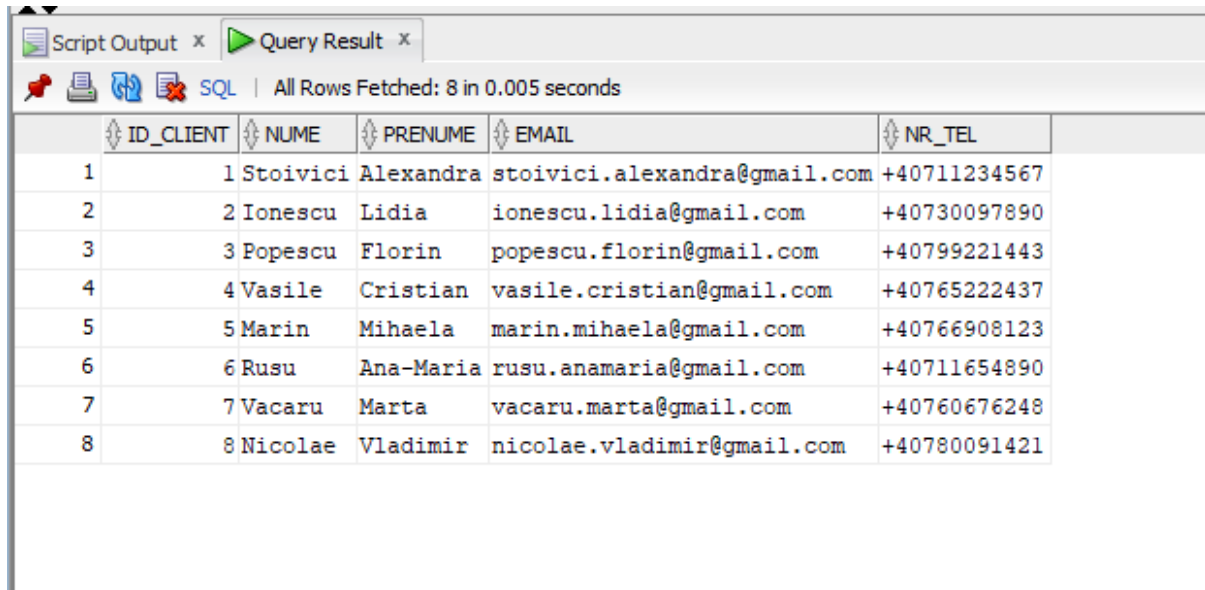
SELECT \*

FROM CLIENTI;

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE CLIENTI (
2     ID_CLIENT NUMBER PRIMARY KEY,
3     NUME VARCHAR2(30),
4     PRENUME VARCHAR2(30),
5     EMAIL VARCHAR2(40),
6     NR_TEL VARCHAR2(15)
7 );
8
9 INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, EMAIL, NR_TEL)
10 VALUES (SEQ_CLIENTI.NEXTVAL, 'Stoivici', 'Alexandra', 'stoivici.alexandra@gmail.com', '+40711234567');
11
12 INSERT INTO CLIENTI
13 VALUES (SEQ_CLIENTI.NEXTVAL, 'Ionescu', 'Lidia', 'ionescu.lidia@gmail.com', '+40730097890');
14
15 INSERT INTO CLIENTI
16 VALUES (SEQ_CLIENTI.NEXTVAL, 'Popescu', 'Florin', 'popescu.florin@gmail.com', '+40799221443');
17
18 INSERT INTO CLIENTI
19 VALUES (SEQ_CLIENTI.NEXTVAL, 'Vasile', 'Cristian', 'vasile.cristian@gmail.com', '+40765222437');
20
21 INSERT INTO CLIENTI
22 VALUES (SEQ_CLIENTI.NEXTVAL, 'Marin', 'Mihaela', 'marin.mihaela@gmail.com', '+40766908123');
23
24 INSERT INTO CLIENTI
25 VALUES (SEQ_CLIENTI.NEXTVAL, 'Rusu', 'Ana-Maria', 'rusu.anamaria@gmail.com', '+40711654890');
26
27 INSERT INTO CLIENTI
28 VALUES (SEQ_CLIENTI.NEXTVAL, 'Vacaru', 'Marta', 'vacaru.marta@gmail.com', '+40760676248');
29
30 INSERT INTO CLIENTI
31 VALUES (SEQ_CLIENTI.NEXTVAL, 'Nicolae', 'Vladimir', 'nicolae.vladimir@gmail.com', '+40780091421');
32
33 SELECT *
34 FROM CLIENTI;
35
```

- Execuție cod sursă:



The screenshot shows a SQL query result window with the title 'Query Result'. It displays 8 rows of data from a table. The columns are ID\_CLIENT, NUME, PRENUME, EMAIL, and NR\_TEL. The data is as follows:

ID_CLIENT	NUME	PRENUME	EMAIL	NR_TEL
1	Stoivici	Alexandra	stoivici.alexandra@gmail.com	+40711234567
2	Ionescu	Lidia	ionescu.lidia@gmail.com	+40730097890
3	Popescu	Florin	popescu.florin@gmail.com	+40799221443
4	Vasile	Cristian	vasile.cristian@gmail.com	+40765222437
5	Marin	Mihaela	marin.mihaela@gmail.com	+40766908123
6	Rusu	Ana-Maria	rusu.anamaria@gmail.com	+40711654890
7	Vacaru	Marta	vacaru.marta@gmail.com	+40760676248
8	Nicolae	Vladimir	nicolae.vladimir@gmail.com	+40780091421

## 2) Tabelul ADRESE:

- SQL:

```
CREATE TABLE ADRESE (  
    ID_ADRESA NUMBER PRIMARY KEY,  
    TARA VARCHAR2(40),  
    JUDET VARCHAR2(40),  
    ORAS VARCHAR2(40),  
    STRADA VARCHAR2(40),  
    NUMAR NUMBER CHECK (NUMAR > 0),  
    SCARA NUMBER,  
    APARTAMENT NUMBER,  
    ID_CLIENT NUMBER,  
    CONSTRAINT FK_ADRESE_CLIENTI FOREIGN KEY (ID_CLIENT) REFERENCES  
    CLIENTI(ID_CLIENT)  
);
```

```
INSERT INTO ADRESE (ID_ADRESA, TARA, JUDET, ORAS, STRADA, NUMAR, SCARA,  
APARTAMENT, id_client)
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Bucuresti', 'Sector 4', 'Strada Micleni', 123,  
NULL, NULL, 1);
```

```
INSERT INTO ADRESE
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Calarasi', 'Pucioasa', 'Strada Uverturii', 22,  
NULL, NULL, 2);
```

```
INSERT INTO ADRESE
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Ilfov', 'Bragadiru', 'Strada Merilor', 26, 3, 21, 2);
```

```
INSERT INTO ADRESE
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Sibiu', 'Sibiu', 'Strada Olfac', 3, 1, 2, 6);
```

```
INSERT INTO ADRESE
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Alba', 'Alba Iulia', 'Calea Voiaj', 41, NULL,  
NULL, 4);
```

```
INSERT INTO ADRESE
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'Bulgaria', 'Vidin', 'Vidin', 'Straevskol', 6, NULL, NULL, 7);
```

```
INSERT INTO ADRESE
```

```
VALUES (SEQ_ADRESE.NEXTVAL, 'United Kingdom', 'Wales', 'Swansea', 4, 2, 4, 5);
```

```
SELECT *
```

```
FROM ADRESE;
```

▪ Codul sursă:

```

Worksheet  Query Builder
1 CREATE TABLE ADRESE (
2     ID_ADRESA NUMBER PRIMARY KEY,
3     TARA VARCHAR2(40),
4     JUDET VARCHAR2(40),
5     ORAS VARCHAR2(40),
6     STRADA VARCHAR2(40),
7     NUMAR NUMBER CHECK (NUMAR > 0),
8     SCARA NUMBER,
9     APARTAMENT NUMBER,
10    ID_CLIENT NUMBER,
11    CONSTRAINT FK_ADRESE_CLIENTI FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT)
12 );
13
14
15 INSERT INTO ADRESE (ID_ADRESA, TARA, JUDET, ORAS, STRADA, NUMAR, SCARA, APARTAMENT, id_client)
16 VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Bucuresti', 'Sector 4', 'Strada Micleni', 123, NULL, NULL, 1);
17
18 INSERT INTO ADRESE
19 VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Calarasi', 'Pucioasa', 'Strada Uverturii', 22, NULL, NULL, 2);
20
21 INSERT INTO ADRESE
22 VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Ilfov', 'Bragadiru', 'Strada Merilor', 26, 3, 21, 2);
23
24 INSERT INTO ADRESE
25 VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Sibiu', 'Sibiu', 'Strada Olfac', 3, 1, 2, 6);
26
27 INSERT INTO ADRESE
28 VALUES (SEQ_ADRESE.NEXTVAL, 'Romania', 'Alba', 'Alba Iulia', 'Calea Voiaj', 41, NULL, NULL, 4);
29
30 INSERT INTO ADRESE
31 VALUES (SEQ_ADRESE.NEXTVAL, 'Bulgaria', 'Vidin', 'Vidin', 'Straevskol', 6, NULL, NULL, 7);
32
33 INSERT INTO ADRESE
34 VALUES (SEQ_ADRESE.NEXTVAL, 'United Kingdom', 'Wales', 'Swansea', 'Kindom', 4, 2, 4, 5);
35
36 SELECT *
37 FROM ADRESE;
38

```

▪ Execuție cod sursă:

ID_ADRESA	TARA	JUDET	ORAS	STRADA	NUMAR	SCARA	APARTAMENT	ID_CLIENT
1	Romania	Bucuresti	Sector 4	Strada Micleni	123	(null)	(null)	1
2	Romania	Calarasi	Pucioasa	Strada Uverturii	22	(null)	(null)	2
3	Romania	Ilfov	Bragadiru	Strada Merilor	26	3	21	2
4	Romania	Sibiu	Sibiu	Strada Olfac	3	1	2	6
5	Romania	Alba	Alba Iulia	Calea Voiaj	41	(null)	(null)	4
6	Bulgaria	Vidin	Vidin	Straevskol	6	(null)	(null)	7
7	United Kingdom	Wales	Swansea	Kindom	4	2	4	5

### 3) Tabelul COMENZI:

- SQL:


```
CREATE TABLE COMENZI (  
    ID_COMANDA NUMBER PRIMARY KEY,  
    PRET DECIMAL(10, 2),  
    LIVRARE VARCHAR2(10) CHECK (UPPER(LIVRARE) IN ('CURIER', 'POSTA',  
'PERSONAL')),  
    STARE VARCHAR2(10) CHECK (UPPER(STARE) IN ('PRELUATA', 'IN LUCRU',  
'FINALIZATA', 'LIVRATA')),  
    ID_CLIENT NUMBER NOT NULL,  
    CONSTRAINT FK_COMENZI_CLIENTI FOREIGN KEY (ID_CLIENT) REFERENCES  
CLIENTI(ID_CLIENT)  
);  
  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'PRELUATA', 1);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'PERSONAL', 'IN LUCRU', 3);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'POSTA', 'LIVRATA', 4);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'FINALIZATA', 2);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'POSTA', 'PRELUATA', 1);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'IN LUCRU', 7);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'PERSONAL', 'FINALIZATA', 6);  
INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)  
VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'PRELUATA', 7);  
  
SELECT *  
FROM COMENZI;
```

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE COMENZI (
2     ID_COMANDA NUMBER PRIMARY KEY,
3     PRET DECIMAL(10, 2),
4     LIVRARE VARCHAR2(10) CHECK (UPPER(LIVRARE) IN ('CURIER', 'POSTA', 'PERSONAL')),
5     STARE VARCHAR2(10) CHECK (UPPER(STARE) IN ('PRELUATA', 'IN LUCRU', 'FINALIZATA', 'LIVRATA')),
6     ID_CLIENT NUMBER NOT NULL,
7     CONSTRAINT FK_COMENZI_CLIENTI FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT)
8 );
9
10
11 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
12 VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'PRELUATA', 1);
13
14 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
15 VALUES (SEQ_COMENZI.NEXTVAL, 'PERSONAL', 'IN LUCRU', 3);
16
17 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
18 VALUES (SEQ_COMENZI.NEXTVAL, 'POSTA', 'LIVRATA', 4);
19
20 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
21 VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'FINALIZATA', 2);
22
23 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
24 VALUES (SEQ_COMENZI.NEXTVAL, 'POSTA', 'PRELUATA', 1);
25
26 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
27 VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'IN LUCRU', 7);
28
29 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
30 VALUES (SEQ_COMENZI.NEXTVAL, 'PERSONAL', 'FINALIZATA', 6);
31
32 INSERT INTO COMENZI (ID_COMANDA, LIVRARE, STARE, ID_CLIENT)
33 VALUES (SEQ_COMENZI.NEXTVAL, 'CURIER', 'PRELUATA', 7);
34
35 SELECT *
36 FROM COMENZI;
37
38
```

▪ Execuție cod sursă:

Script Output x Query Result x

 | All Rows Fetched: 8 in 0.006 seconds

	ID_COMANDA	PRET	LIVRARE	STARE	ID_CLIENT
1	1	(null)	CURIER	PRELUATA	1
2	2	(null)	PERSONAL	IN LUCRU	3
3	3	(null)	POSTA	LIVRATA	4
4	4	(null)	CURIER	FINALIZATA	2
5	5	(null)	POSTA	PRELUATA	1
6	6	(null)	CURIER	IN LUCRU	7
7	7	(null)	PERSONAL	FINALIZATA	6
8	8	(null)	CURIER	PRELUATA	7

4) Tabelul FEEDBACKURI:

▪ SQL:

```
CREATE TABLE FEEDBACKURI (  
    ID_FEEDBACK NUMBER PRIMARY KEY,  
    NOTA NUMBER CHECK (NOTA > 0 AND NOTA <= 10),  
    DESCRIERE VARCHAR2(500),  
    ID_COMANDA NUMBER NOT NULL,  
    CONSTRAINT FK_FEEDBACKURI_COMENZI FOREIGN KEY (ID_COMANDA)  
REFERENCES COMENZI(ID_COMANDA)  
);  
  
INSERT INTO FEEDBACKURI (id_feedback, nota, descriere, id_comanda)  
VALUES (SEQ_FEEDBACKURI.NEXTVAL, 10, 'Prompt si calitativ', 3);  
  
INSERT INTO FEEDBACKURI  
VALUES (SEQ_FEEDBACKURI.NEXTVAL, 8, NULL, 2);  
  
INSERT INTO FEEDBACKURI  
VALUES (SEQ_FEEDBACKURI.NEXTVAL, 9, 'Calitate si personalizare', 1);  
  
INSERT INTO FEEDBACKURI  
VALUES (SEQ_FEEDBACKURI.NEXTVAL, 7, 'Ok', 4);  
  
INSERT INTO FEEDBACKURI  
VALUES (SEQ_FEEDBACKURI.NEXTVAL, 8, 'Foarte ok, recomand', 7);  
  
INSERT INTO FEEDBACKURI  
VALUES (SEQ_FEEDBACKURI.NEXTVAL, 9, 'De incredere', 8);  
  
SELECT *  
FROM FEEDBACKURI;
```




▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE FEEDBACKURI (
2     ID_FEEDBACK NUMBER PRIMARY KEY,
3     NOTA NUMBER CHECK (NOTA > 0 AND NOTA <= 10),
4     DESCRIERE VARCHAR2(500),
5     ID_COMANDA NUMBER NOT NULL,
6     CONSTRAINT FK_FEEDBACKURI_COMENZI FOREIGN KEY (ID_COMANDA) REFERENCES COMENZI (ID_COMANDA)
7 );
8
9
10 INSERT INTO FEEDBACKURI (id_feedback, nota, descriere, id_comanda)
11 VALUES (SEQ_FEEDBACKURI.NEXTVAL, 10, 'Prompt si calitativ', 3);
12
13 INSERT INTO FEEDBACKURI
14 VALUES (SEQ_FEEDBACKURI.NEXTVAL, 8, NULL, 2);
15
16 INSERT INTO FEEDBACKURI
17 VALUES (SEQ_FEEDBACKURI.NEXTVAL, 9, 'Calitate si personalizare', 1);
18
19 INSERT INTO FEEDBACKURI
20 VALUES (SEQ_FEEDBACKURI.NEXTVAL, 7, 'Ok', 4);
21
22 INSERT INTO FEEDBACKURI
23 VALUES (SEQ_FEEDBACKURI.NEXTVAL, 8, 'Foarte ok, recomand', 7);
24
25 INSERT INTO FEEDBACKURI
26 VALUES (SEQ_FEEDBACKURI.NEXTVAL, 9, 'De incredere', 8);
27
28 SELECT *
29 FROM FEEDBACKURI;
30
31
```

▪ Execuție cod sursă:

Script Output x Query Result x

 | All Rows Fetched: 6 in 0.007 seconds

	ID_FEEDBACK	NOTA	DESCRIERE	ID_COMANDA
1	1	10	Prompt si calitativ	3
2	2	8 (null)		2
3	3	9	Calitate si personalizare	1
4	4	7	Ok	4
5	5	8	Foarte ok, recomand	7
6	6	9	De incredere	8

5) Tabelul COLECTII:

▪ SQL:


```
CREATE TABLE COLECTII (  
    ID_COLECTIE NUMBER PRIMARY KEY,  
    NUME VARCHAR2(60),  
    TEMATICA VARCHAR2(60),  
    AN NUMBER CHECK (an > 1999),  
    ANOTIMP VARCHAR2(30)  
);  
  
INSERT INTO COLECTII (id_colectie, nume, tematica, an, anotimp)  
VALUES (SEQ_COLECTII.NEXTVAL, 'Prima Verra', 'Florală', 2022, 'Primavara');  
  
INSERT INTO COLECTII  
VALUES (SEQ_COLECTII.NEXTVAL, 'Summertime', 'Diafana', 2023, 'Vara');  
  
INSERT INTO COLECTII (id_colectie, nume, an, anotimp)  
VALUES (SEQ_COLECTII.NEXTVAL, 'Cozy', 2020, 'Iarna');  
  
INSERT INTO COLECTII  
VALUES (SEQ_COLECTII.NEXTVAL, 'Queen Gambit', 'Rosu si negru', 2023, 'Toamna-Iarna');  
  
INSERT INTO COLECTII  
VALUES (SEQ_COLECTII.NEXTVAL, 'Sweet Autumn', NULL, 2019, 'Toamna');  
  
INSERT INTO COLECTII  
VALUES (SEQ_COLECTII.NEXTVAL, 'Pastel', 'Culori pastelate', 2023, 'Primavara');  
  
INSERT INTO COLECTII  
VALUES (SEQ_COLECTII.NEXTVAL, 'Jewels', 'Pietre pretioase', 2020, 'Iarna');  
  
INSERT INTO COLECTII  
VALUES (SEQ_COLECTII.NEXTVAL, 'Kaleidoscope', 'Multicolor', 2021, 'Vara');  
  
SELECT *  
FROM COLECTII;
```

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE COLECTII (
2     ID_COLECTIE NUMBER PRIMARY KEY,
3     NUME VARCHAR2(60),
4     TEMATICA VARCHAR2(60),
5     AN NUMBER CHECK (an > 1999),
6     ANOTIMP VARCHAR2(30)
7 );
8
9 INSERT INTO COLECTII (id_collectie, nume, tematica, an, anotimp)
10 VALUES (SEQ_COLECTII.NEXTVAL, 'Prima Verra', 'Florală', 2022, 'Primavara');
11
12 INSERT INTO COLECTII
13 VALUES (SEQ_COLECTII.NEXTVAL, 'Summertime', 'Diafana', 2023, 'Vara');
14
15 INSERT INTO COLECTII (id_collectie, nume, an, anotimp)
16 VALUES (SEQ_COLECTII.NEXTVAL, 'Cozy', 2020, 'Iarna');
17
18 INSERT INTO COLECTII
19 VALUES (SEQ_COLECTII.NEXTVAL, 'Queen Gambit', 'Rosu si negru', 2023, 'Toamna-Iarna');
20
21 INSERT INTO COLECTII
22 VALUES (SEQ_COLECTII.NEXTVAL, 'Sweet Autumn', NULL, 2019, 'Toamna');
23
24 INSERT INTO COLECTII
25 VALUES (SEQ_COLECTII.NEXTVAL, 'Pastel', 'Culori pastelate', 2023, 'Primavara');
26
27 INSERT INTO COLECTII
28 VALUES (SEQ_COLECTII.NEXTVAL, 'Jewels', 'Pietre pretioase', 2020, 'Iarna');
29
30 INSERT INTO COLECTII
31 VALUES (SEQ_COLECTII.NEXTVAL, 'Kaleidoscope', 'Multicolor', 2021, 'Vara');
32
33 SELECT *
34 FROM COLECTII;
35
36
```

▪ Execuție cod sursă:

Script Output x Query Result x

 All Rows Fetched: 8 in 0.002 seconds

ID_COLECTIE	NUME	TEMATICA	AN	ANOTIMP
1	1 Prima Verra	Florală	2022	Primavara
2	2 Summertime	Diafana	2023	Vara
3	3 Cozy	(null)	2020	Iarna
4	4 Queen Gambit	Rosu si negru	2023	Toamna-Iarna
5	5 Sweet Autumn	(null)	2019	Toamna
6	6 Pastel	Culori pastelate	2023	Primavara
7	7 Jewels	Pietre pretioase	2020	Iarna
8	8 Kaleidoscope	Multicolor	2021	Vara

6) Tabelul PRODUSE:

▪ SQL:

```
CREATE TABLE PRODUSE (  
    ID_PRODUS NUMBER PRIMARY KEY,  
    NUME VARCHAR2(30),  
    PRET DECIMAL(10, 2) CHECK (PRET > 0),  
    DESCRIERE VARCHAR2(100),  
    ID_COMANDA NUMBER,  
    ID_COLECTIE NUMBER,  
    CONSTRAINT FK_PRODUSE_COMENZI FOREIGN KEY (ID_COMANDA) REFERENCES  
    COMENZI(ID_COMANDA),  
    CONSTRAINT fk_PRODUSE_COLECTII FOREIGN KEY (ID_COLECTIE) REFERENCES  
    COLECTII(ID_COLECTIE)  
);  
  
INSERT INTO PRODUSE (ID_PRODUS, NUME, PRET, DESCRIERE, ID_COMANDA,  
ID_COLECTIE)  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Bluza Valurile Marii', 700.00, 'Bluza albastra cu valuri',  
NULL, 1);  
  
INSERT INTO PRODUSE  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Bluza Agata', 600.00, 2, 7);  
  
INSERT INTO PRODUSE  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Cersei Monstera', 30.00, 'Cersei in forma de frunze de  
monstera', NULL, 5);  
  
INSERT INTO PRODUSE  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Cersei Mac', 30.00, 'Cersei in forma de flori de mac', 4, 2);  
  
INSERT INTO PRODUSE  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Rochia Breeze', 2000.00, 'Bluza lunga ajurata', 6, 6);  
  
INSERT INTO PRODUSE  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Fusta Maxima', 400.00, 'Fusta cloche in degrade', 1, 7);  
  
INSERT INTO PRODUSE  
VALUES (SEQ_PRODUSE.NEXTVAL, 'Cardiganul Kilipirro', 800.00, 'Cardigan cu nasturi scurt', 2,  
NULL);
```

INSERT INTO PRODUSE

VALUES (SEQ\_PRODUSE.NEXTVAL, 'Cardiganul Smarald', 900.00, 'Cardigan pana in talie verde', NULL, 7);

INSERT INTO PRODUSE

VALUES (SEQ\_PRODUSE.NEXTVAL, 'Cercei Picatura', 30.00, 'Cercei in forma de picatura de ploaie', NULL, NULL);

INSERT INTO PRODUSE

VALUES (SEQ\_PRODUSE.NEXTVAL, 'Puloverul Benovelle', 1200.00, 'Pulover de casmir cu guler pe gat', 4, NULL);

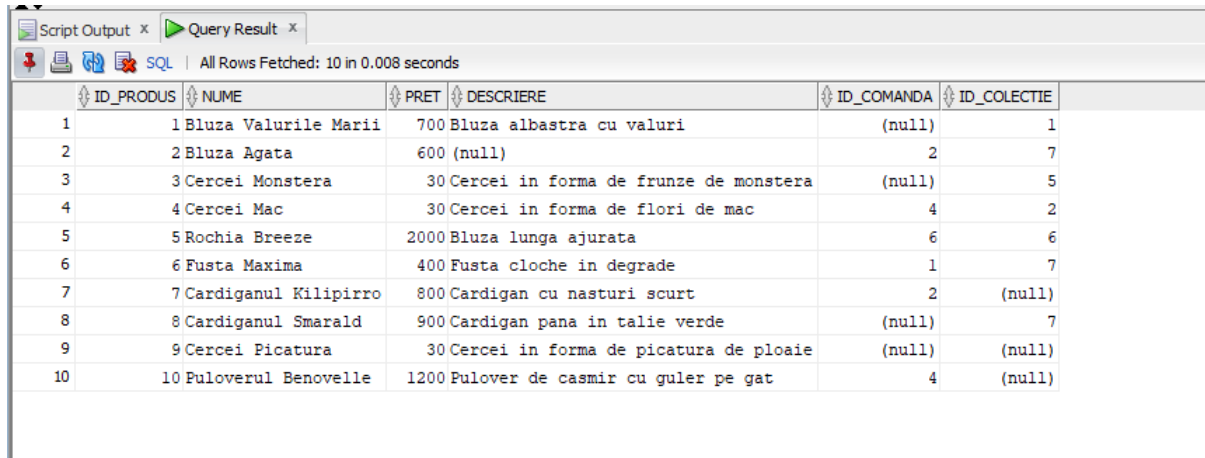
SELECT \*

FROM PRODUSE;

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE PRODUSE (
2     ID_PRODUS NUMBER PRIMARY KEY,
3     NUME VARCHAR2(30),
4     PRET DECIMAL(10, 2) CHECK (PRET > 0),
5     DESCRIERE VARCHAR2(100),
6     ID_COMANDA NUMBER,
7     ID_COLECTIE NUMBER,
8     CONSTRAINT FK_PRODUSE_COMENZI FOREIGN KEY (ID_COMANDA) REFERENCES COMENZI(ID_COMANDA),
9     CONSTRAINT fk_PRODUSE_COLECTII FOREIGN KEY (ID_COLECTIE) REFERENCES COLECTII(ID_COLECTIE)
10 );
11
12 INSERT INTO PRODUSE (ID_PRODUS, NUME, PRET, DESCRIERE, ID_COMANDA, ID_COLECTIE)
13 VALUES (SEQ_PRODUSE.NEXTVAL, 'Bluza Valurile Marii', 700.00, 'Bluza albastra cu valuri', NULL, 1);
14
15 INSERT INTO PRODUSE
16 VALUES (SEQ_PRODUSE.NEXTVAL, 'Bluza Agata', 600.00, NULL, 2, 7);
17
18 INSERT INTO PRODUSE
19 VALUES (SEQ_PRODUSE.NEXTVAL, 'Cercei Monstera', 30.00, 'Cercei in forma de frunze de monstera', NULL, 5);
20
21 INSERT INTO PRODUSE
22 VALUES (SEQ_PRODUSE.NEXTVAL, 'Cercei Mac', 30.00, 'Cercei in forma de flori de mac', 4, 2);
23
24 INSERT INTO PRODUSE
25 VALUES (SEQ_PRODUSE.NEXTVAL, 'Rochia Breeze', 2000.00, 'Bluza lunga ajurata', 6, 6);
26
27 INSERT INTO PRODUSE
28 VALUES (SEQ_PRODUSE.NEXTVAL, 'Fusta Maxima', 400.00, 'Fusta cloche in degrade', 1, 7);
29
30 INSERT INTO PRODUSE
31 VALUES (SEQ_PRODUSE.NEXTVAL, 'Cardiganul Kilipirro', 800.00, 'Cardigan cu nasturi scurt', 2, NULL);
32
33 INSERT INTO PRODUSE
34 VALUES (SEQ_PRODUSE.NEXTVAL, 'Cardiganul Smarald', 900.00, 'Cardigan pana in talie verde', NULL, 7);
35
36 INSERT INTO PRODUSE
37 VALUES (SEQ_PRODUSE.NEXTVAL, 'Cercei Picatura', 30.00, 'Cercei in forma de picatura de ploaie', NULL, NULL);
38
39 INSERT INTO PRODUSE
40 VALUES (SEQ_PRODUSE.NEXTVAL, 'Puloverul Benovelle', 1200.00, 'Pulover de casmir cu guler pe gat', 4, NULL);
41
42 SELECT *
43 FROM PRODUSE;
44
45
46
```

▪ Execuție cod sursă:



The screenshot shows a SQL query result with 10 rows. The columns are ID\_PRODUS, NUME, PRET, DESCRIERE, ID\_COMANDA, and ID\_COLECTIE. The data is as follows:

ID_PRODUS	NUME	PRET	DESCRIERE	ID_COMANDA	ID_COLECTIE
1	1 Bluza Valurile Marii	700	Bluza albastra cu valuri	(null)	1
2	2 Bluza Agata	600	(null)	2	7
3	3 Cercei Monstera	30	Cercei in forma de frunze de monstera	(null)	5
4	4 Cercei Mac	30	Cercei in forma de flori de mac	4	2
5	5 Rochia Breeze	2000	Bluza lunga ajurata	6	6
6	6 Fusta Maxima	400	Fusta cloche in degrade	1	7
7	7 Cardiganul Kilipirro	800	Cardigan cu nasturi scurt	2	(null)
8	8 Cardiganul Smarald	900	Cardigan pana in talie verde	(null)	7
9	9 Cercei Picatura	30	Cercei in forma de picatura de ploaie	(null)	(null)
10	10 Puloverul Benovelle	1200	Pulover de casmir cu guler pe gat	4	(null)

7) Tabelul ANGAJATI:

▪ SQL:

```
CREATE TABLE ANGAJATI (
    ID_ANGAJAT NUMBER PRIMARY KEY,
    NUME VARCHAR2(30),
    PRENUME VARCHAR2(30),
    TIP VARCHAR2(30),
    EMAIL VARCHAR2(40),
    NR_TEL VARCHAR2(15),
    SALARIU INT CHECK (SALARIU > 0),
    COMISION DECIMAL(10, 2) CHECK (COMISION >= 0 AND COMISION <=100),
    DATA_ANGAJARE DATE
);

INSERT INTO ANGAJATI (ID_ANGAJAT, NUME, PRENUME, TIP, EMAIL, NR_TEL,
SALARIU, COMISION, DATA_ANGAJARE)
VALUES (SEQ_ANGAJATI.NEXTVAL, 'Nedelcu', 'Daniel', 'fondator', 'daniel.nedelcu@nidle.com',
'+40712345678', 6000, 50.00, TO_DATE('2020-01-01', 'YYYY-MM-DD'));
```

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Ionescu', 'Diana', 'fondator', 'diana.ionescu@nidle.com',  
'+40722345678', 6000, 50.00, TO\_DATE('2020-01-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Georgescu', 'Ana', 'angajat', 'ana.georgescu@nidle.com',  
'+40732345678', 4000, 15.00, TO\_DATE('2021-03-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Dumitrescu', 'Vasile', 'angajat',  
'vasile.dumitrescu@nidle.com', '+40742345678', 3500, 25.00, TO\_DATE('2022-04-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Mihai', 'Elena', 'colaborator', 'elena.mihai@nidle.com',  
'+40752345678', 3000, 75.00, TO\_DATE('2024-01-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Petrescu', 'George', 'angajat', 'george.petrescu@nidle.com',  
'+40762345678', 3200, 60.00, TO\_DATE('2024-01-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Popa', 'Gabriela', 'analizator', 'gabriela.popa@nidle.com',  
'+40772345678', 4500, 0.00, TO\_DATE('2023-07-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Stoica', 'Marius', 'colaborator', 'marius.stoica@nidle.com',  
'+40782345678', 4800, 5.00, TO\_DATE('2023-08-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Tudor', 'Irina', 'angajat', 'irina.tudor@nidle.com',  
'+40792345678', 3700, 80.00, TO\_DATE('2023-09-01', 'YYYY-MM-DD'));

INSERT INTO ANGAJATI

VALUES (SEQ\_ANGAJATI.NEXTVAL, 'Diaconu', 'Adrian', 'angajat', 'adrian.diaconu@nidle.com',  
'+40702345678', 4200, 90.00, TO\_DATE('2023-10-01', 'YYYY-MM-DD'));

SELECT \*

FROM ANGAJATI;

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE ANGAJATI (
2     ID_ANGAJAT NUMBER PRIMARY KEY,
3     NUME VARCHAR2(30),
4     PRENUME VARCHAR2(30),
5     TIP VARCHAR2(30),
6     EMAIL VARCHAR2(40),
7     NR_TEL VARCHAR2(15),
8     SALARIU INT CHECK (SALARIU > 0),
9     COMISION DECIMAL(10, 2) CHECK (COMISION >= 0 AND COMISION <=100),
10    DATA_ANGAJARE DATE
11 );
12
13 INSERT INTO ANGAJATI (ID_ANGAJAT, NUME, PRENUME, TIP, EMAIL, NR_TEL, SALARIU, COMISION, DATA_ANGAJARE)
14 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Nedelcu', 'Daniel', 'fondator', 'daniel.nedelcu@nidle.com', '+40712345678', 6000, 50.00, TO_DATE('2020-01-01', 'YYYY-MM-DD'));
15
16 INSERT INTO ANGAJATI
17 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Ionescu', 'Diana', 'fondator', 'diana.ionescu@nidle.com', '+40722345678', 6000, 50.00, TO_DATE('2020-01-01', 'YYYY-MM-DD'));
18
19 INSERT INTO ANGAJATI
20 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Georgescu', 'Ana', 'angajat', 'ana.georgescu@nidle.com', '+40732345678', 4000, 15.00, TO_DATE('2021-03-01', 'YYYY-MM-DD'));
21
22 INSERT INTO ANGAJATI
23 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Dumitrescu', 'Vasile', 'angajat', 'vasile.dumitrescu@nidle.com', '+40742345678', 3500, 25.00, TO_DATE('2022-04-01', 'YYYY-MM-DD'));
24
25 INSERT INTO ANGAJATI
26 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Mihai', 'Elena', 'colaborator', 'elena.mihai@nidle.com', '+40752345678', 3000, 75.00, TO_DATE('2024-01-01', 'YYYY-MM-DD'));
27
28 INSERT INTO ANGAJATI
29 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Petrescu', 'George', 'angajat', 'george.petrescu@nidle.com', '+40762345678', 3200, 60.00, TO_DATE('2024-01-01', 'YYYY-MM-DD'));
30
31 INSERT INTO ANGAJATI
32 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Popa', 'Gabriela', 'analizator', 'gabriela.popa@nidle.com', '+40772345678', 4500, 0.00, TO_DATE('2023-07-01', 'YYYY-MM-DD'));
33
34 INSERT INTO ANGAJATI
35 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Stoica', 'Marius', 'colaborator', 'marius.stoica@nidle.com', '+40782345678', 4800, 5.00, TO_DATE('2023-08-01', 'YYYY-MM-DD'));
36
37 INSERT INTO ANGAJATI
38 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Tudor', 'Irina', 'angajat', 'irina.tudor@nidle.com', '+40792345678', 3700, 80.00, TO_DATE('2023-09-01', 'YYYY-MM-DD'));
39
40 INSERT INTO ANGAJATI
41 VALUES (SEQ_ANGAJATI.NEXTVAL, 'Diaconu', 'Adrian', 'angajat', 'adrian.diaconu@nidle.com', '+40702345678', 4200, 90.00, TO_DATE('2023-10-01', 'YYYY-MM-DD'));
42
43 SELECT *
44 FROM ANGAJATI;
45
46
47
48
```

▪ Execuție cod sursă:

ID_ANGAJAT	NUME	PRENUME	TIP	EMAIL	NR_TEL	SALARIU	COMISION	DATA_ANGAJARE
1	Nedelcu	Daniel	fondator	daniel.nedelcu@nidle.com	+40712345678	6000	50	01/JAN/20
2	Ionescu	Diana	fondator	diana.ionescu@nidle.com	+40722345678	6000	50	01/JAN/20
3	Georgescu	Ana	angajat	ana.georgescu@nidle.com	+40732345678	4000	15	01/MAR/21
4	Dumitrescu	Vasile	angajat	vasile.dumitrescu@nidle.com	+40742345678	3500	25	01/APR/22
5	Mihai	Elena	colaborator	elena.mihai@nidle.com	+40752345678	3000	75	01/JAN/24
6	Petrescu	George	angajat	george.petrescu@nidle.com	+40762345678	3200	60	01/JAN/24
7	Popa	Gabriela	analizator	gabriela.popa@nidle.com	+40772345678	4500	0	01/JUL/23
8	Stoica	Marius	colaborator	marius.stoica@nidle.com	+40782345678	4800	5	01/AUG/23
9	Tudor	Irina	angajat	irina.tudor@nidle.com	+40792345678	3700	80	01/SEP/23
10	Diaconu	Adrian	angajat	adrian.diaconu@nidle.com	+40702345678	4200	90	01/OCT/23



8) Tabelul CROITORI:

▪ SQL:

```
CREATE TABLE CROITORI (  
    ID_CROITOR NUMBER,  
    ID_ANGAJAT NUMBER,  
    PRIMARY KEY (ID_CROITOR, ID_ANGAJAT),  
    CONSTRAINT FK_CROITORI_ANGAJATI FOREIGN KEY (ID_ANGAJAT) REFERENCES  
    ANGAJATI(ID_ANGAJAT),  
    CONSTRAINT UNIQUE_ID_CROITOR UNIQUE (ID_CROITOR)  
);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 1);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 3);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 4);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 5);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 9);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 10);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 6);  
  
INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)  
VALUES (SEQ_CROITORI.NEXTVAL, 8);  
  
SELECT *  
FROM CROITORI;
```

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE CROITORI (
2     ID_CROITOR NUMBER,
3     ID_ANGAJAT NUMBER,
4     PRIMARY KEY (ID_CROITOR, ID_ANGAJAT),
5     CONSTRAINT FK_CROITORI_ANGAJATI FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI (ID_ANGAJAT),
6     CONSTRAINT UNIQUE_ID_CROITOR UNIQUE (ID_CROITOR)
7 );
8
9 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
10 VALUES (SEQ_CROITORI.NEXTVAL, 1);
11
12 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
13 VALUES (SEQ_CROITORI.NEXTVAL, 3);
14
15 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
16 VALUES (SEQ_CROITORI.NEXTVAL, 4);
17
18 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
19 VALUES (SEQ_CROITORI.NEXTVAL, 5);
20
21 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
22 VALUES (SEQ_CROITORI.NEXTVAL, 9);
23
24 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
25 VALUES (SEQ_CROITORI.NEXTVAL, 10);
26
27 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
28 VALUES (SEQ_CROITORI.NEXTVAL, 6);
29
30 INSERT INTO CROITORI (ID_CROITOR, ID_ANGAJAT)
31 VALUES (SEQ_CROITORI.NEXTVAL, 8);
32
33 SELECT *
34 FROM CROITORI;
```

▪ Execuție cod sursă:

Script Output x Query Result x Query Result 1 x

SQL All Rows Fetched: 8 in 0.006 seconds

	ID_CROITOR	ID_ANGAJAT
1	1	1
2	2	3
3	3	4
4	4	5
5	5	9
6	6	10
7	7	6
8	9	8

9) Tabelul DESIGNERI:

▪ SQL:

```
CREATE TABLE DESIGNERI (  
    ID_DESIGNER NUMBER,  
    ID_ANGAJAT NUMBER,  
    PRIMARY KEY (ID_DESIGNER, ID_ANGAJAT),  
    CONSTRAINT FK_DESIGNERI_ANGAJATI FOREIGN KEY (ID_ANGAJAT) REFERENCES  
    ANGAJATI(ID_ANGAJAT),  
    CONSTRAINT UNIQUE_ID_DESIGNER UNIQUE (ID_DESIGNER)  
);  
  
INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)  
VALUES (SEQ_DESIGNERI.NEXTVAL, 1);  
  
INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)  
VALUES (SEQ_DESIGNERI.NEXTVAL, 2);  
  
INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)  
VALUES (SEQ_DESIGNERI.NEXTVAL, 7);  
  
INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)  
VALUES (SEQ_DESIGNERI.NEXTVAL, 3);  
  
INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)  
VALUES (SEQ_DESIGNERI.NEXTVAL, 8);  
  
INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)  
VALUES (SEQ_DESIGNERI.NEXTVAL, 4);  
  
SELECT *  
FROM DESIGNERI;
```

▪ Codul sursă:

```
Worksheet Query Builder
1 CREATE TABLE DESIGNERI (
2     ID_DESIGNER NUMBER,
3     ID_ANGAJAT NUMBER,
4     PRIMARY KEY (ID_DESIGNER, ID_ANGAJAT),
5     CONSTRAINT FK_DESIGNERI_ANGAJATI FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI (ID_ANGAJAT),
6     CONSTRAINT UNIQUE_ID_DESIGNER UNIQUE (ID_DESIGNER)
7 );
8
9 INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)
10 VALUES (SEQ_DESIGNERI.NEXTVAL, 1);
11
12 INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)
13 VALUES (SEQ_DESIGNERI.NEXTVAL, 2);
14
15 INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)
16 VALUES (SEQ_DESIGNERI.NEXTVAL, 7);
17
18 INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)
19 VALUES (SEQ_DESIGNERI.NEXTVAL, 3);
20
21 INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)
22 VALUES (SEQ_DESIGNERI.NEXTVAL, 8);
23
24 INSERT INTO DESIGNERI (ID_DESIGNER, ID_ANGAJAT)
25 VALUES (SEQ_DESIGNERI.NEXTVAL, 4);
26
27 SELECT *
28 FROM DESIGNERI;
29
30
31
32
```

▪ Execuție cod sursă:

Script Output x Query Result x Query Result 1 x

SQL | All Rows Fetched: 6 in 0.006 seconds

ID_DESIGNER	ID_ANGAJAT
1	1
2	2
3	7
4	3
5	8
6	4

10) Tabelul SARCINI:

▪ SQL:

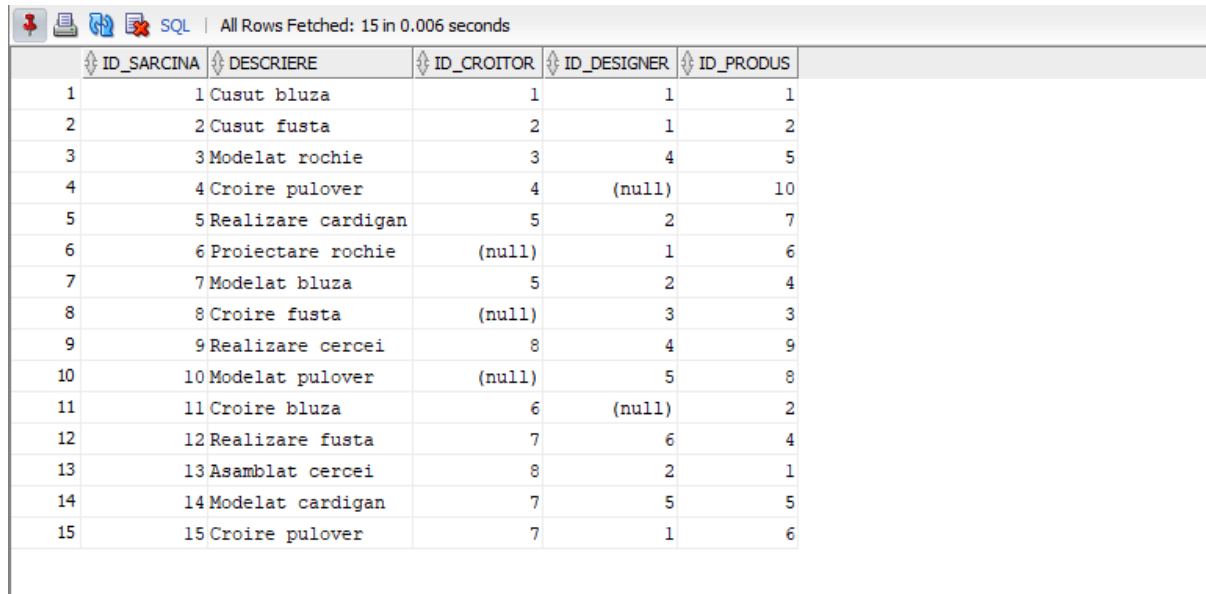
```
CREATE TABLE SARCINI (  
    ID_SARCINA NUMBER PRIMARY KEY,  
    DESCRIERE VARCHAR2(100),  
    ID_CROITOR NUMBER,  
    ID_DESIGNER NUMBER,  
    ID_PRODUS NUMBER,  
    CONSTRAINT FK_SARCINI_CROITORI FOREIGN KEY (ID_CROITOR) REFERENCES  
CROITORI(ID_CROITOR),  
    CONSTRAINT FK_SARCINI_DESIGNERI FOREIGN KEY (ID_DESIGNER) REFERENCES  
DESIGNERI(ID_DESIGNER),  
    CONSTRAINT FK_SARCINI_PRODUSE FOREIGN KEY (ID_PRODUS) REFERENCES  
PRODUSE(ID_PRODUS)  
);  
  
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,  
ID_PRODUS)  
VALUES (SEQ_SARCINI.NEXTVAL, 'Cusut bluza', 1, 1, 1);  
  
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,  
ID_PRODUS)  
VALUES (SEQ_SARCINI.NEXTVAL, 'Cusut fusta', 2, 1, 2);  
  
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,  
ID_PRODUS)  
VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat rochie', 3, 4, 5);  
  
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,  
ID_PRODUS)  
VALUES (SEQ_SARCINI.NEXTVAL, 'Croire pulover', 4, NULL, 10);  
  
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,  
ID_PRODUS)  
VALUES (SEQ_SARCINI.NEXTVAL, 'Realizare cardigan', 5, 2, 7);  
  
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,  
ID_PRODUS)  
VALUES (SEQ_SARCINI.NEXTVAL, 'Proiectare rochie', NULL, 1, 6);
```

```
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat bluza', 5, 2, 4);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Croire fusta', NULL, 3, 3);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Realizare cercei', 8, 4, 9);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat pulover', NULL, 5, 8);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Croire bluza', 6, NULL, 2);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Realizare fusta', 7, 6, 4);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Asamblat cercei', 8, 2, 1);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat cardigan', 7, 5, 5);
INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER,
ID_PRODUS)
VALUES (SEQ_SARCINI.NEXTVAL, 'Croire pulover', 7, 1, 6);
SELECT *
FROM SARCINI;
```

▪ Codul sursă:

Worksheet	Query Builder
1	CREATE TABLE SARCINI (
2	ID_SARCINA NUMBER PRIMARY KEY,
3	DESCRIERE VARCHAR2(100),
4	ID_CROITOR NUMBER,
5	ID_DESIGNER NUMBER,
6	ID_PRODUS NUMBER,
7	CONSTRAINT FK_SARCINI_CROITORI FOREIGN KEY (ID_CROITOR) REFERENCES CROITORI(ID_CROITOR),
8	CONSTRAINT FK_SARCINI_DESIGNERI FOREIGN KEY (ID_DESIGNER) REFERENCES DESIGNERI(ID_DESIGNER),
9	CONSTRAINT FK_SARCINI_PRODUSE FOREIGN KEY (ID_PRODUS) REFERENCES PRODUSE(ID_PRODUS)
10	);
11	
12	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
13	VALUES (SEQ_SARCINI.NEXTVAL, 'Cusut bluza', 1, 1, 1);
14	
15	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
16	VALUES (SEQ_SARCINI.NEXTVAL, 'Cusut fusta', 2, 1, 2);
17	
18	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
19	VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat rochie', 3, 4, 5);
20	
21	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
22	VALUES (SEQ_SARCINI.NEXTVAL, 'Croire pulover', 4, NULL, 10);
23	
24	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
25	VALUES (SEQ_SARCINI.NEXTVAL, 'Realizare cardigan', 5, 2, 7);
26	
27	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
28	VALUES (SEQ_SARCINI.NEXTVAL, 'Proiectare rochie', NULL, 1, 6);
29	
30	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
31	VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat bluza', 5, 2, 4);
32	
33	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
34	VALUES (SEQ_SARCINI.NEXTVAL, 'Croire fusta', NULL, 3, 3);
35	
36	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
37	VALUES (SEQ_SARCINI.NEXTVAL, 'Realizare cercei', 8, 4, 9);
38	
39	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
40	VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat pulover', NULL, 5, 8);
41	
42	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
43	VALUES (SEQ_SARCINI.NEXTVAL, 'Croire bluza', 6, NULL, 2);
44	
45	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
46	VALUES (SEQ_SARCINI.NEXTVAL, 'Realizare fusta', 7, 6, 4);
47	
48	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
49	VALUES (SEQ_SARCINI.NEXTVAL, 'Asamblat cercei', 8, 2, 1);
50	
51	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
52	VALUES (SEQ_SARCINI.NEXTVAL, 'Modelat cardigan', 7, 5, 5);
53	
54	INSERT INTO SARCINI (ID_SARCINA, DESCRIERE, ID_CROITOR, ID_DESIGNER, ID_PRODUS)
55	VALUES (SEQ_SARCINI.NEXTVAL, 'Croire pulover', 7, 1, 6);
56	
57	SELECT *
58	FROM SARCINI;
59	
60	

- Execuție cod sursă:



ID_SARCINA	DESCRIERE	ID_CROITOR	ID_DESIGNER	ID_PRODUS	
1	1 Cusut bluza	1	1	1	
2	2 Cusut fusta	2	1	2	
3	3 Modelat rochie	3	4	5	
4	4 Croire pulover	4	(null)	10	
5	5 Realizare cardigan	5	2	7	
6	6 Proiectare rochie	(null)	1	6	
7	7 Modelat bluza	5	2	4	
8	8 Croire fusta	(null)	3	3	
9	9 Realizare cercei	8	4	9	
10	10 Modelat pulover	(null)	5	8	
11	11 Croire bluza	6	(null)	2	
12	12 Realizare fusta	7	6	4	
13	13 Asamblat cercei	8	2	1	
14	14 Modelat cardigan	7	5	5	
15	15 Croire pulover	7	1	6	

11) Tabelul MATERIALE:

- SQL:

```
CREATE TABLE MATERIALE (  
    ID_MATERIAL NUMBER PRIMARY KEY,  
    NUME VARCHAR2(30),  
    TIP VARCHAR2(30),  
    COMPOZITIE VARCHAR2(100)  
);  
  
INSERT INTO MATERIALE (ID_MATERIAL, NUME, TIP, COMPOZITIE)  
VALUES (SEQ_MATERIALE.NEXTVAL, 'Bumbac', 'Material textil', '100% bumbac');  
  
INSERT INTO MATERIALE (ID_MATERIAL, NUME, TIP, COMPOZITIE)  
VALUES (SEQ_MATERIALE.NEXTVAL, 'Poliester', 'Material textil', '100% poliester');  
  
INSERT INTO MATERIALE (ID_MATERIAL, NUME, TIP, COMPOZITIE)  
VALUES (SEQ_MATERIALE.NEXTVAL, 'Lana', 'Material textil', '100% lana merinos');  
  
INSERT INTO MATERIALE (ID_MATERIAL, NUME, TIP, COMPOZITIE)  
VALUES (SEQ_MATERIALE.NEXTVAL, 'Matase', 'Material textil', '100% matase naturala');
```

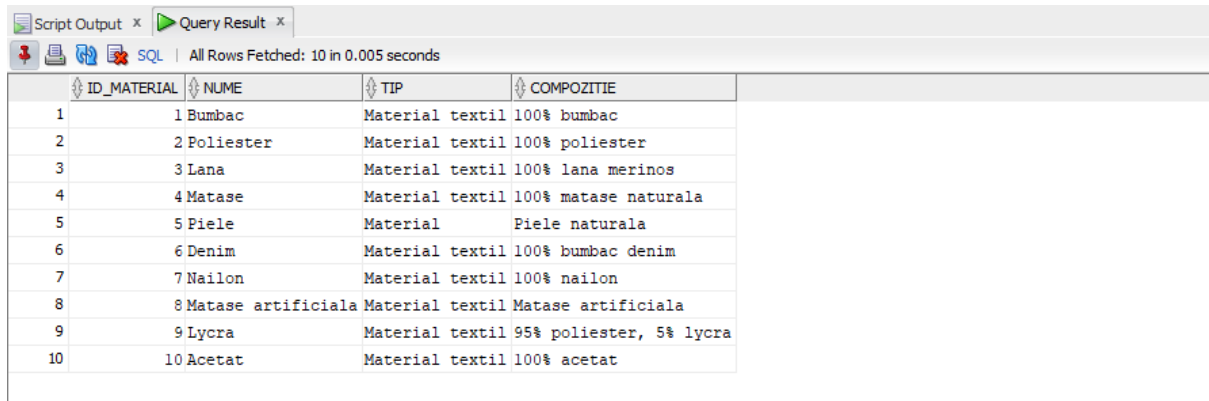


```
INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
VALUES (SEQ_MATERIALE.NEXTVAL, 'Piele', 'Material', 'Piele naturala');
INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
VALUES (SEQ_MATERIALE.NEXTVAL, 'Denim', 'Material textil', '100% bumbac denim');
INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
VALUES (SEQ_MATERIALE.NEXTVAL, 'Nailon', 'Material textil', '100% nailon');
INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
VALUES (SEQ_MATERIALE.NEXTVAL, 'Matase artificiala', 'Material textil', 'Matase artificiala');
INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
VALUES (SEQ_MATERIALE.NEXTVAL, 'Lycra', 'Material textil', '95% poliester, 5% lycra');
INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
VALUES (SEQ_MATERIALE.NEXTVAL, 'Acetat', 'Material textil', '100% acetat');
SELECT *
FROM MATERIALE;
```

▪ Codul sursă:

Worksheet	Query Builder
1	CREATE TABLE MATERIALE (
2	ID_MATERIALE NUMBER PRIMARY KEY,
3	NUME VARCHAR2(30),
4	TIP VARCHAR2(30),
5	COMPOZITIE VARCHAR2(100)
6	);
7	
8	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
9	VALUES (SEQ_MATERIALE.NEXTVAL, 'Bumbac', 'Material textil', '100% bumbac');
10	
11	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
12	VALUES (SEQ_MATERIALE.NEXTVAL, 'Poliester', 'Material textil', '100% poliester');
13	
14	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
15	VALUES (SEQ_MATERIALE.NEXTVAL, 'Lana', 'Material textil', '100% lana merinos');
16	
17	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
18	VALUES (SEQ_MATERIALE.NEXTVAL, 'Matase', 'Material textil', '100% matase naturala');
19	
20	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
21	VALUES (SEQ_MATERIALE.NEXTVAL, 'Piele', 'Material', 'Piele naturala');
22	
23	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
24	VALUES (SEQ_MATERIALE.NEXTVAL, 'Denim', 'Material textil', '100% bumbac denim');
25	
26	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
27	VALUES (SEQ_MATERIALE.NEXTVAL, 'Nailon', 'Material textil', '100% nailon');
28	
29	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
30	VALUES (SEQ_MATERIALE.NEXTVAL, 'Matase artificiala', 'Material textil', 'Matase artificiala');
31	
32	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
33	VALUES (SEQ_MATERIALE.NEXTVAL, 'Lycra', 'Material textil', '95% poliester, 5% lycra');
34	
35	INSERT INTO MATERIALE (ID_MATERIALE, NUME, TIP, COMPOZITIE)
36	VALUES (SEQ_MATERIALE.NEXTVAL, 'Acetat', 'Material textil', '100% acetat');
37	
38	SELECT *
39	FROM MATERIALE;
40	

▪ Execuție cod sursă:



The screenshot shows a database query result window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying a table with 10 rows. The table has four columns: ID\_MATERIAL, NUME, TIP, and COMPOZITIE. The data is as follows:

ID_MATERIAL	NUME	TIP	COMPOZITIE
1	1 Bumbac	Material textil	100% bumbac
2	2 Poliester	Material textil	100% poliester
3	3 Lana	Material textil	100% lana merinos
4	4 Matase	Material textil	100% matase naturala
5	5 Piele	Material	Piele naturala
6	6 Denim	Material textil	100% bumbac denim
7	7 Nailon	Material textil	100% nailon
8	8 Matase artificiala	Material textil	Matase artificiala
9	9 Lycra	Material textil	95% poliester, 5% lycra
10	10 Acetat	Material textil	100% acetat

12) Tabelul TIPARE:

▪ SQL:

```
CREATE TABLE TIPARE (  
    ID_TIPAR NUMBER PRIMARY KEY,  
    LUNGIME INT,  
    LATIME INT,  
    ID_DESIGNER NUMBER,  
    ID_PRODUS NUMBER,  
    ID_MATERIAL NUMBER,  
    CONSTRAINT FK_TIPARE_DESIGNERI FOREIGN KEY (ID_DESIGNER) REFERENCES  
    DESIGNERI(ID_DESIGNER),  
    CONSTRAINT FK_TIPARE_PRODUSE FOREIGN KEY (ID_PRODUS) REFERENCES  
    PRODUSE(ID_PRODUS),  
    CONSTRAINT FK_TIPARE_MATERIALE FOREIGN KEY (ID_MATERIAL) REFERENCES  
    MATERIALE(ID_MATERIAL)  
);  
  
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)  
VALUES (SEQ_TIPARE.NEXTVAL, 50, 30, 1, NULL, 1);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 60, 40, 2, NULL, 2);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 70, 50, 3, 1, NULL);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 80, 60, 4, 2, 3);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 90, 70, NULL, 3, 4);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 100, 80, NULL, 4, 5);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 110, 90, 5, NULL, 6);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 120, 100, 6, 5, 7);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 130, 110, NULL, 6, 8);
```

```
INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS,  
ID_MATERIAL)
```

```
VALUES (SEQ_TIPARE.NEXTVAL, 140, 120, NULL, 7, 9);
```

```
SELECT *
```

```
FROM TIPARE;
```





▪ Codul sursă:

Worksheet	Query Builder
1	CREATE TABLE TIPARE (
2	ID_TIPAR NUMBER PRIMARY KEY,
3	LUNGIME INT,
4	LATIME INT,
5	ID_DESIGNER NUMBER,
6	ID_PRODUS NUMBER,
7	ID_MATERIAL NUMBER,
8	CONSTRAINT FK_TIPARE_DESIGNER FOREIGN KEY (ID_DESIGNER) REFERENCES DESIGNER (ID_DESIGNER),
9	CONSTRAINT FK_TIPARE_PRODUSE FOREIGN KEY (ID_PRODUS) REFERENCES PRODUSE (ID_PRODUS),
10	CONSTRAINT FK_TIPARE_MATERIALE FOREIGN KEY (ID_MATERIAL) REFERENCES MATERIALE (ID_MATERIAL)
11	);
12	
13	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
14	VALUES (SEQ_TIPARE.NEXTVAL, 50, 30, 1, NULL, 1);
15	
16	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
17	VALUES (SEQ_TIPARE.NEXTVAL, 60, 40, 2, NULL, 2);
18	
19	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
20	VALUES (SEQ_TIPARE.NEXTVAL, 70, 50, 3, 1, NULL);
21	
22	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
23	VALUES (SEQ_TIPARE.NEXTVAL, 80, 60, 4, 2, 3);
24	
25	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
26	VALUES (SEQ_TIPARE.NEXTVAL, 90, 70, NULL, 3, 4);
27	
28	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
29	VALUES (SEQ_TIPARE.NEXTVAL, 100, 80, NULL, 4, 5);
30	
31	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
32	VALUES (SEQ_TIPARE.NEXTVAL, 110, 90, 5, NULL, 6);
33	
34	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
35	VALUES (SEQ_TIPARE.NEXTVAL, 120, 100, 6, 5, 7);
36	
37	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
38	VALUES (SEQ_TIPARE.NEXTVAL, 130, 110, NULL, 6, 8);
39	
40	INSERT INTO TIPARE (ID_TIPAR, LUNGIME, LATIME, ID_DESIGNER, ID_PRODUS, ID_MATERIAL)
41	VALUES (SEQ_TIPARE.NEXTVAL, 140, 120, NULL, 7, 9);
42	
43	SELECT *
44	FROM TIPARE;
45	

▪ Execuție cod sursă:

Script Output x

Query Result x

 SQL | All Rows Fetched: 10 in 0.013 seconds

ID_TIPAR	LUNGIME	LATIME	ID_DESIGNER	ID_PRODUS	ID_MATERIAL	
1	1	50	30	1	(null)	1
2	2	60	40	2	(null)	2
3	3	70	50	3	1	(null)
4	4	80	60	4	2	3
5	5	90	70	(null)	3	4
6	6	100	80	(null)	4	5
7	7	110	90	5	(null)	6
8	8	120	100	6	5	7
9	9	130	110	(null)	6	8
10	10	140	120	(null)	7	9

13) Tabelul FURNIZORI:

▪ SQL:

```
CREATE TABLE FURNIZORI (  
    ID_FURNIZOR NUMBER PRIMARY KEY,  
    NUME VARCHAR2(50),  
    COD_FISCAL INT CHECK (COD_FISCAL > 0)  
);  
  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Fabrica de textile SRL', 123456789);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Furnizorul de accesorii SRL', 987654321);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Distribuitorul de mătase SA', 456789123);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'PFA Producătorul de piele', 789123456);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Distribuitorul de dantelă SRL', 321654987);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Fabrica de mătase SA', 654987321);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'PFA Producătorul de dungi', 987321654);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Furnizorul de nasturi SA', 159753468);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Fabrica de dantelă SRL', 357951468);  
INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)  
VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Violette Andrele SA', 468135792);  
  
SELECT *  
  
FROM FURNIZORI;
```

▪ Codul sursă:

```
Worksheet | Query Builder
1 CREATE TABLE FURNIZORI (
2     ID_FURNIZOR NUMBER PRIMARY KEY,
3     NUME VARCHAR2(50),
4     COD_FISCAL INT CHECK (COD_FISCAL > 0)
5 );
6
7 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
8 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Fabrica de textile SRL', 123456789);
9
10 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
11 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Furnizorul de accesorii SRL', 987654321);
12
13 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
14 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Distribuitorul de mătase SA', 456789123);
15
16 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
17 VALUES (SEQ_FURNIZORI.NEXTVAL, 'PFA Producătorul de piele', 789123456);
18
19 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
20 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Distribuitorul de dantelă SRL', 321654987);
21
22 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
23 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Fabrica de mătase SA', 654987321);
24
25 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
26 VALUES (SEQ_FURNIZORI.NEXTVAL, 'PFA Producătorul de dungi', 987321654);
27
28 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
29 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Furnizorul de nasturi SA', 159753468);
30
31 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
32 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Fabrica de dantelă SRL', 357951468);
33
34 INSERT INTO FURNIZORI (ID_FURNIZOR, NUME, COD_FISCAL)
35 VALUES (SEQ_FURNIZORI.NEXTVAL, 'SC Violette Andrele SA', 468135792);
36
37 SELECT *
38 FROM FURNIZORI;
39
40
```

▪ Execuție cod sursă:

ID_FURNIZOR	NUME	COD_FISCAL
1	SC Fabrica de textile SRL	123456789
2	SC Furnizorul de accesorii SRL	987654321
3	SC Distribuitorul de mătase SA	456789123
4	PFA Producătorul de piele	789123456
5	SC Distribuitorul de dantelă SRL	321654987
6	SC Fabrica de mătase SA	654987321
7	PFA Producătorul de dungi	987321654
8	SC Furnizorul de nasturi SA	159753468
9	SC Fabrica de dantelă SRL	357951468
10	SC Violette Andrele SA	468135792

#### 14) Tabelul CONTRACTE

- SQL:

```
CREATE TABLE CONTRACTE (  
    ID_CONTRACT NUMBER NOT NULL,  
    DATA DATE,  
    ID_FURNIZOR NUMBER NOT NULL,  
    ID_MATERIAL NUMBER NOT NULL,  
    PRIMARY KEY (ID_CONTRACT, ID_FURNIZOR, ID_MATERIAL),  
    CONSTRAINT FK_CONTRACTE_FURNIZORI FOREIGN KEY (ID_FURNIZOR)  
REFERENCES FURNIZORI(ID_FURNIZOR),  
    CONSTRAINT FK_CONTRACTE_MATERIALE FOREIGN KEY (ID_MATERIAL)  
REFERENCES MATERIALE(ID_MATERIAL)  
);  
  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-01-15', 'YYYY-MM-DD'), 1, 1);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-02-20', 'YYYY-MM-DD'), 2, 2);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 3, 3);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-04-05', 'YYYY-MM-DD'), 4, 4);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-05-18', 'YYYY-MM-DD'), 5, 5);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-06-22', 'YYYY-MM-DD'), 6, 6);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-07-09', 'YYYY-MM-DD'), 7, 7);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-08-14', 'YYYY-MM-DD'), 8, 8);  
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)  
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-09-30', 'YYYY-MM-DD'), 9, 9);
```

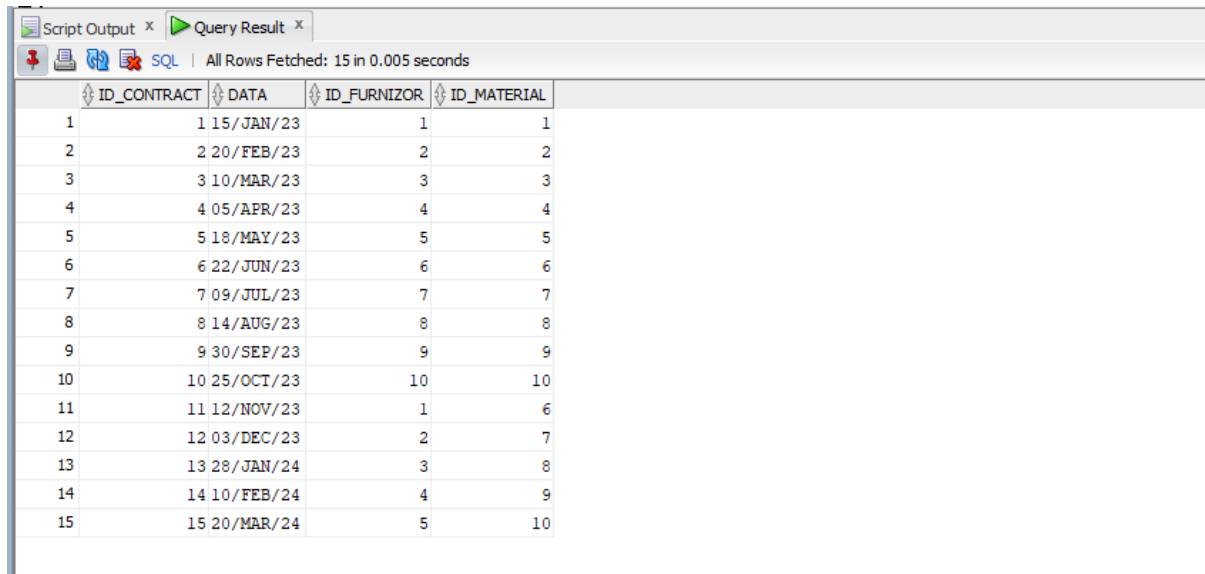
```
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-10-25', 'YYYY-MM-DD'), 10, 10);
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-11-12', 'YYYY-MM-DD'), 1, 6);
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-12-03', 'YYYY-MM-DD'), 2, 7);
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2024-01-28', 'YYYY-MM-DD'), 3, 8);
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2024-02-10', 'YYYY-MM-DD'), 4, 9);
INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)
VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2024-03-20', 'YYYY-MM-DD'), 5, 10);
SELECT *
FROM CONTRACTE;
```



▪ Codul sursă:

Worksheet	Query Builder
1	<code>CREATE TABLE CONTRACTE (</code>
2	<code>    ID_CONTRACT NUMBER NOT NULL,</code>
3	<code>    DATA DATE,</code>
4	<code>    ID_FURNIZOR NUMBER NOT NULL,</code>
5	<code>    ID_MATERIAL NUMBER NOT NULL,</code>
6	<code>    PRIMARY KEY (ID_CONTRACT, ID_FURNIZOR, ID_MATERIAL),</code>
7	<code>    CONSTRAINT FK_CONTRACTE_FURNIZORI FOREIGN KEY (ID_FURNIZOR) REFERENCES FURNIZORI(ID_FURNIZOR),</code>
8	<code>    CONSTRAINT FK_CONTRACTE_MATERIALE FOREIGN KEY (ID_MATERIAL) REFERENCES MATERIALE(ID_MATERIAL)</code>
9	<code>);</code>
10	
11	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
12	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-01-15', 'YYYY-MM-DD'), 1, 1);</code>
13	
14	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
15	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-02-20', 'YYYY-MM-DD'), 2, 2);</code>
16	
17	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
18	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 3, 3);</code>
19	
20	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
21	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-04-05', 'YYYY-MM-DD'), 4, 4);</code>
22	
23	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
24	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-05-18', 'YYYY-MM-DD'), 5, 5);</code>
25	
26	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
27	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-06-22', 'YYYY-MM-DD'), 6, 6);</code>
28	
29	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
30	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-07-09', 'YYYY-MM-DD'), 7, 7);</code>
31	
32	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
33	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-08-14', 'YYYY-MM-DD'), 8, 8);</code>
34	
35	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
36	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-09-30', 'YYYY-MM-DD'), 9, 9);</code>
37	
38	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
39	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-10-25', 'YYYY-MM-DD'), 10, 10);</code>
40	
41	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
42	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-11-12', 'YYYY-MM-DD'), 1, 6);</code>
43	
44	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
45	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2023-12-03', 'YYYY-MM-DD'), 2, 7);</code>
46	
47	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
48	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2024-01-28', 'YYYY-MM-DD'), 3, 8);</code>
49	
50	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
51	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2024-02-10', 'YYYY-MM-DD'), 4, 9);</code>
52	
53	<code>INSERT INTO CONTRACTE (ID_CONTRACT, DATA, ID_FURNIZOR, ID_MATERIAL)</code>
54	<code>VALUES (SEQ_CONTRACTE.NEXTVAL, TO_DATE('2024-03-20', 'YYYY-MM-DD'), 5, 10);</code>
55	
56	<code>SELECT *</code>
57	<code>FROM CONTRACTE;</code>
58	
59	

- Execuție cod sursă:



ID_CONTRACT	DATA	ID_FURNIZOR	ID_MATERIAL
1	1 15/JAN/23	1	1
2	2 20/FEB/23	2	2
3	3 10/MAR/23	3	3
4	4 05/APR/23	4	4
5	5 18/MAY/23	5	5
6	6 22/JUN/23	6	6
7	7 09/JUL/23	7	7
8	8 14/AUG/23	8	8
9	9 30/SEP/23	9	9
10	10 25/OCT/23	10	10
11	11 12/NOV/23	1	6
12	12 03/DEC/23	2	7
13	13 28/JAN/24	3	8
14	14 10/FEB/24	4	9
15	15 20/MAR/24	5	10

\* În final, modific coloana PRET din COMANDA, pentru a fi suma preturilor produselor corespunzătoare:

- SQL:

UPDATE COMENZI C

SET C.PRET = (

SELECT SUM(P.PRET)

FROM PRODUSE P

WHERE P.ID\_COMANDA = C.ID\_COMANDA

AND P.PRET IS NOT NULL

);

SELECT \*

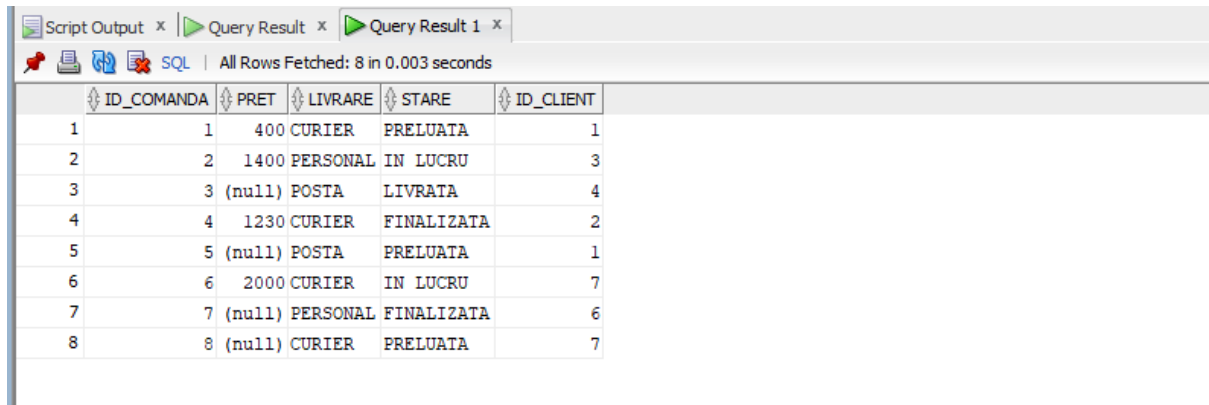
FROM COMENZI;

- Cod sursă:



```
1 UPDATE COMENZI C
2 SET C.PRET = (
3     SELECT SUM(P.PRET)
4     FROM PRODUSE P
5     WHERE P.ID_COMANDA = C.ID_COMANDA
6     AND P.PRET IS NOT NULL
7 );
8
9 SELECT *
10 FROM COMENZI;
```

▪ Rulare cod sursă:



ID_COMANDA	PRET	LIVRARE	STARE	ID_CLIENT
1	1	400 CURIER	PRELUATA	1
2	2	1400 PERSONAL	IN LUCRU	3
3	3 (null)	POSTA	LIVRATA	4
4	4	1230 CURIER	FINALIZATA	2
5	5 (null)	POSTA	PRELUATA	1
6	6	2000 CURIER	IN LUCRU	7
7	7 (null)	PERSONAL	FINALIZATA	6
8	8 (null)	CURIER	PRELUATA	7

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

- a) subcereri sincronizate în care intervin cel puțin 3 tabele
- b) subcereri nesincronizate în clauza FROM
- c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (în cadrul aceleiași cereri)
- d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
- e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
- f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Cererea 1

Afișați, pentru fiecare comandă, numele și prenumele clientului care îi corespunde, iar mai apoi tipul de livrare, prețul comenzii (pe care îl vom numi subtotal), costul de livrare (curier: 18.00, poștă: 10.00, ridicare personală: 0.00), și în final totalul de plată.

- subpuncte: f) WITH, d) ordonări, NVL, DECODE în aceeași cerere

▪ SQL:

WITH LIVRARI AS (

SELECT

C.ID\_CLIENT,

C.LIVRARE,

DECODE(UPPER(C.LIVRARE),

'CURIER', 18.00,

'POSTA', 10.00,

'PERSONAL', 0.00

) AS COST\_LIVRARE,

```
        NVL(C.PRET, 0) AS SUBTOTAL,  
        DECODE(UPPER(C.LIVRARE),  
        'CURIER', NVL(C.PRET, 0)+18.00,  
        'PERSONAL', NVL(C.PRET, 0),  
        'POSTA', NVL(C.PRET, 0)+10.0) AS TOTAL  
    FROM COMENZI C  
)  
SELECT  
    CL.NUME,  
    CL.PRENUME,  
    L.LIVRARE,  
    L.SUBTOTAL,  
    L.COST_LIVRARE,  
    L.TOTAL  
FROM LIVRARI L  
JOIN CLIENTI CL ON CL.ID_CLIENT = L.ID_CLIENT  
ORDER BY CL.NUME;
```

▪ Cod sursă:


```

Worksheet Query Builder
1 WITH LIVRARI AS (
2     SELECT
3         C.ID_CLIENT,
4         C.LIVRARE,
5         DECODE(UPPER(C.LIVRARE),
6             'CURIER', 18.00,
7             'POSTA', 10.00,
8             'PERSONAL', 0.00
9         ) AS COST_LIVRARE,
10        NVL(C.PRET, 0) AS SUBTOTAL,
11        DECODE(UPPER(C.LIVRARE),
12            'CURIER', NVL(C.PRET, 0)+18.00,
13            'PERSONAL', NVL(C.PRET, 0),
14            'POSTA', NVL(C.PRET, 0)+10.0) AS TOTAL
15    FROM COMENZI C
16 )
17 SELECT
18     CL.NUME,
19     CL.PRENUME,
20     L.LIVRARE,
21     L.SUBTOTAL,
22     L.COST_LIVRARE,
23     L.TOTAL
24 FROM LIVRARI L
25 JOIN CLIENTI CL ON CL.ID_CLIENT = L.ID_CLIENT
26 ORDER BY CL.NUME;
27
28
29
30

```

▪ Rulare cod sursă:

Script Output x Query Result x

 | All Rows Fetched: 8 in 0.002 seconds

NUME	PRENUME	LIVRARE	SUBTOTAL	COST_LIVRARE	TOTAL
1 Ionescu	Lidia	CURIER	1230	18	1248
2 Popescu	Florin	PERSONAL	1400	0	1400
3 Rusu	Ana-Maria	PERSONAL	0	0	0
4 Stoivici	Alexandra	POSTA	0	10	10
5 Stoivici	Alexandra	CURIER	400	18	418
6 Vacaru	Marta	CURIER	0	18	18
7 Vacaru	Marta	CURIER	2000	18	2018
8 Vasile	Cristian	POSTA	0	10	10

## Cererea 2

Afișați numele complet (concatenare între nume și prenume) al clientului și id-ul, prețul, și numărul de produse ale comenzii pentru toți clienții la comanda cărora a avut cel puțin o sarcină pentru cel puțin un produs un designer angajat acum mai mult de 4 ani.

- subpuncte: a) subcereri sincronizate, e) funcțiile pe șiruri de caractere și funcțiile pe date calendaristice

### ▪ SQL:

```
WITH NRPROD AS (
    SELECT
        C2.ID_COMANDA,
        COUNT(P2.ID_PRODUS) NR_PRODUSE
    FROM COMENZI C2
    JOIN PRODUSE P2 ON C2.ID_COMANDA = P2.ID_COMANDA
    GROUP BY C2.ID_COMANDA
)
SELECT
    CONCAT(CONCAT(UPPER(CL.NUME),' '), UPPER(CL.PRENUME)) AS NUME_COMPLET,
    C.ID_COMANDA,
    C.PRET,
    NP.NR_PRODUSE
FROM CLIENTI CL
JOIN COMENZI C ON C.ID_CLIENT = CL.ID_CLIENT
JOIN NRPROD NP ON NP.ID_COMANDA = C.ID_COMANDA
WHERE EXISTS (
    SELECT *
    FROM COMENZI C1
    JOIN PRODUSE P1 ON P1.ID_COMANDA = C1.ID_COMANDA
    JOIN SARCINI S1 ON S1.ID_PRODUS = P1.ID_PRODUS
    JOIN DESIGNERI D1 ON D1.ID_DESIGNER = S1.ID_DESIGNER
    JOIN ANGAJATI A1 ON A1.ID_ANGAJAT = D1.ID_ANGAJAT
```

WHERE C1.ID\_CLIENT = CL.ID\_CLIENT

AND MONTHS\_BETWEEN(ROUND(SYSDATE), A1.DATA\_ANGAJARE) > 48

)

- Cod sursă:

```
Worksheet Query Builder
1 WITH NRPROD AS (
2     SELECT
3         C2.ID_COMANDA,
4         COUNT(P2.ID_PRODUS) NR_PRODUSE
5     FROM COMENZI C2
6     JOIN PRODUSE P2 ON C2.ID_COMANDA = P2.ID_COMANDA
7     GROUP BY C2.ID_COMANDA
8 )
9 SELECT
10     CONCAT(CONCAT(UPPER(CL.NUME), ' '), UPPER(CL.PRENUME)) AS NUME_COMPLET,
11     C.ID_COMANDA,
12     C.PRET,
13     NP.NR_PRODUSE
14 FROM CLIENTI CL
15 JOIN COMENZI C ON C.ID_CLIENT = CL.ID_CLIENT
16 JOIN NRPROD NP ON NP.ID_COMANDA = C.ID_COMANDA
17 WHERE EXISTS (
18     SELECT *
19     FROM COMENZI C1
20     JOIN PRODUSE P1 ON P1.ID_COMANDA = C1.ID_COMANDA
21     JOIN SARCINI S1 ON S1.ID_PRODUS = P1.ID_PRODUS
22     JOIN DESIGNERI D1 ON D1.ID_DESIGNER = S1.ID_DESIGNER
23     JOIN ANGAJATI A1 ON A1.ID_ANGAJAT = D1.ID_ANGAJAT
24     WHERE C1.ID_CLIENT = CL.ID_CLIENT
25         AND MONTHS_BETWEEN(ROUND(SYSDATE), A1.DATA_ANGAJARE) > 48
26 );
27
28
29
```

- Rulare cod sursă:

Query Result x				
SQL   All Rows Fetched: 3 in 0.02 seconds				
	NUME_COMPLET	ID_COMANDA	PRET	NR_PRODUSE
1	STOIVICI ALEXANDRA	1	400	1
2	POPESCU FLORIN	2	1400	2
3	IONESCU LIDIA	4	1230	2



### Cererea 3

Să se afișeze numele materialelor, distribuitorul (furnizorul) și codul fiscal al acestuia, și data semnării contractului pentru toate materialele care au contracte semnate începând cu 1 ianuarie 2024.

- subpuncte: b) subcerere nesincronizată în clauza FROM

- SQL:

```
SELECT DISTINCT
```

```
TAB.NUME_MATERIAL, TAB.NUME DISTRIBUTOR, TAB.COD_FISCAL, TAB.DATA  
DATA_CONTRACT
```

```
FROM (
```

```
    SELECT DISTINCT
```

```
        M.ID_MATERIAL, M.NUME NUME_MATERIAL, F.NUME, F.COD_FISCAL, C.DATA  
    FROM MATERIALE M
```

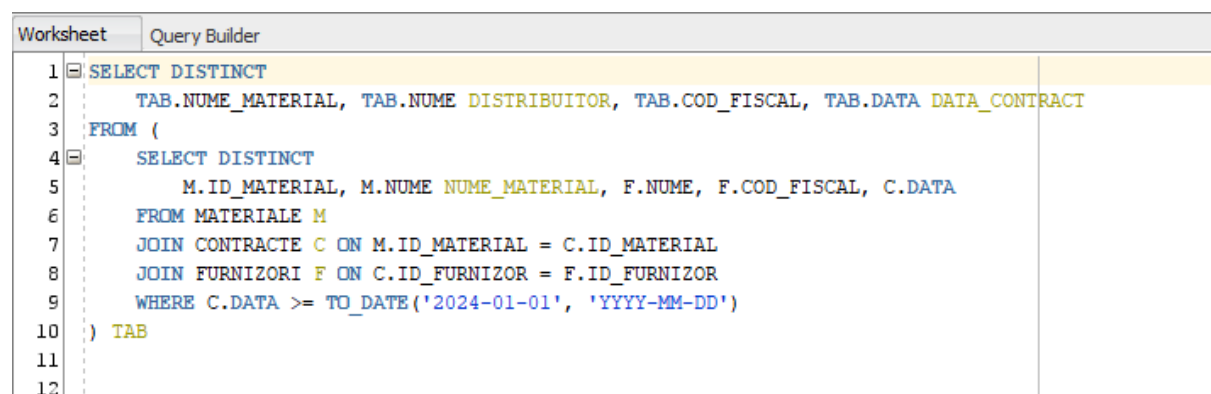
```
    JOIN CONTRACTE C ON M.ID_MATERIAL = C.ID_MATERIAL
```

```
    JOIN FURNIZORI F ON C.ID_FURNIZOR = F.ID_FURNIZOR
```

```
    WHERE C.DATA >= TO_DATE('2024-01-01', 'YYYY-MM-DD')
```

```
) TAB;
```

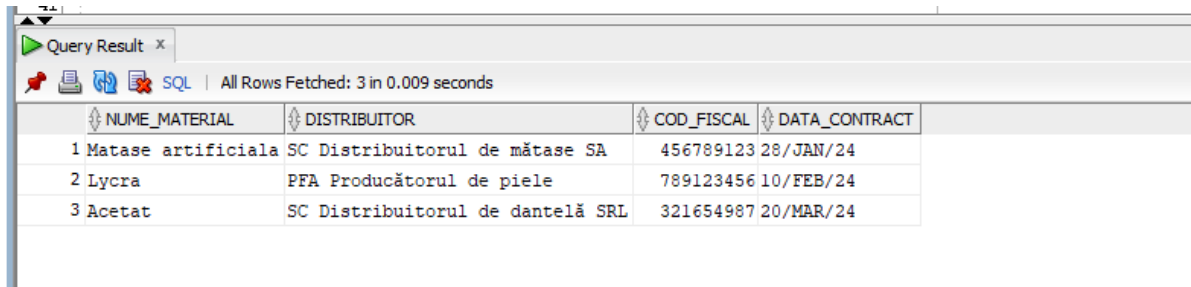
- Cod sursă:



The screenshot shows a 'Query Builder' window with a 'Worksheet' tab. The SQL query is displayed in a text area with line numbers 1 through 12. The query is as follows:

```
1 SELECT DISTINCT
2     TAB.NUME_MATERIAL, TAB.NUME DISTRIBUTOR, TAB.COD_FISCAL, TAB.DATA DATA_CONTRACT
3 FROM (
4     SELECT DISTINCT
5         M.ID_MATERIAL, M.NUME NUME_MATERIAL, F.NUME, F.COD_FISCAL, C.DATA
6     FROM MATERIALE M
7     JOIN CONTRACTE C ON M.ID_MATERIAL = C.ID_MATERIAL
8     JOIN FURNIZORI F ON C.ID_FURNIZOR = F.ID_FURNIZOR
9     WHERE C.DATA >= TO_DATE('2024-01-01', 'YYYY-MM-DD')
10 ) TAB
11
12
```

- Rulare cod sursă:



NUME_MATERIAL	DISTRIBUTOR	COD_FISCAL	DATA_CONTRACT
1 Matase artificiala	SC Distribuitorul de mătase SA	456789123	28/JAN/24
2 Lycra	PFA Producătorul de piele	789123456	10/FEB/24
3 Acetat	SC Distribuitorul de dantelă SRL	321654987	20/MAR/24

#### Cererea 4

Afișați ID-ul fiecărei comenzi, suma totală a acesteia, numărul total de tipare folosite pentru produsele ei și numărul de materiale diferite pentru aceasta, dar numai pentru comenzile care respectă următoarele reguli:

- i) Suma totală trebuie să fie mai mare decât media sumelor totale ale produselor pentru toate comenzile;
  - ii) Numărul total de tipare asociate comenzii trebuie să fie mai mare decât media numărului total de tipare pentru toate comenzile.
- subpuncte: c) complet

- SQL:

```
SELECT
    CL.NUME,
    CL.PRENUME,
    C.ID_COMANDA,
    C.PRET,
    COUNT(T.ID_TIPAR) NR_TIPARE,
    COUNT(DISTINCT M.ID_MATERIAL) NR_MATERIALE_UNICE
FROM COMENZI C
JOIN CLIENTI CL ON C.ID_CLIENT = CL.ID_CLIENT
JOIN PRODUSE P ON C.ID_COMANDA = P.ID_COMANDA
JOIN TIPARE T ON P.ID_PRODUS = T.ID_PRODUS
JOIN MATERIALE M ON T.ID_MATERIAL = M.ID_MATERIAL
```

GROUP BY C.ID\_COMANDA, C.PRET, CL.NUME, CL.PRENUME  
HAVING

```
C.PRET > (  
    SELECT AVG(SUMA_PRODUSE)  
    FROM (  
        SELECT SUM(P1.PRET) SUMA_PRODUSE  
        FROM PRODUSE P1  
        GROUP BY P1.ID_COMANDA  
    )  
)  
AND  
COUNT(T.ID_TIPAR) > (  
    SELECT AVG(NR_TIPARE)  
    FROM (  
        SELECT COUNT(T1.ID_TIPAR) AS NR_TIPARE  
        FROM TIPARE T1  
        JOIN PRODUSE P2 ON T1.ID_PRODUS = P2.ID_PRODUS  
        GROUP BY P2.ID_COMANDA  
    )  
);
```

▪ Cod sursă:

```
Worksheet Query Builder
1 SELECT
2     CL.NUME,
3     CL.PRENUME,
4     C.ID_COMANDA,
5     C.PRET,
6     COUNT(T.ID_TIPAR) NR_TIPARE,
7     COUNT(DISTINCT M.ID_MATERIAL) NR_MATERIALE_UNICE
8 FROM COMENZI C
9 JOIN CLIENTI CL ON C.ID_CLIENT = CL.ID_CLIENT
10 JOIN PRODUSE P ON C.ID_COMANDA = P.ID_COMANDA
11 JOIN TIPARE T ON P.ID_PRODUS = T.ID_PRODUS
12 JOIN MATERIALE M ON T.ID_MATERIAL = M.ID_MATERIAL
13 GROUP BY C.ID_COMANDA, C.PRET, CL.NUME, CL.PRENUME
14 HAVING
15     C.PRET > (
16         SELECT AVG(SUMA_PRODUSE)
17         FROM (
18             SELECT SUM(P1.PRET) SUMA_PRODUSE
19             FROM PRODUSE P1
20             GROUP BY P1.ID_COMANDA
21         )
22     )
23 AND
24     COUNT(T.ID_TIPAR) > (
25         SELECT AVG(NR_TIPARE)
26         FROM (
27             SELECT COUNT(T1.ID_TIPAR) AS NR_TIPARE
28             FROM TIPARE T1
29             JOIN PRODUSE P2 ON T1.ID_PRODUS = P2.ID_PRODUS
30             GROUP BY P2.ID_COMANDA
31         )
32     );
33
34
```

▪ Rulare cod sursă:

Query Result x					
All Rows Fetched: 1 in 0.037 seconds					
NUME	PRENUME	ID_COMANDA	PRET	NR_TIPARE	NR_MATERIALE_UNICE
1 Popescu Florin		2	1400	2	2

### Cererea 5

Afișați numele, prenumele, orașul (sau orașele, în linii diferite, în cazul în care clientul are mai multe adrese), calitatea ('Buna', dacă nota e mai mare decât 7, și 'Acceptabila', altfel), descrierea feedbackului, și starea comenzii, pentru clienții din România care au acordat o notă de feedback mai mare decât 5 și care au selectat ca metodă de livrare 'curier'.

- e) CASE

▪ SQL:

```
SELECT DISTINCT
    CL.NUME,
    CL.PRENUME,
    A.ORAS,
    CASE
        WHEN F.NOTA >7 THEN 'Buna'
        ELSE 'Acceptabila'
    END CALITATE,
    F.DESCRIERE,
    C.STARE
FROM FEEDBACKURI F
JOIN COMENZI C ON F.ID_COMANDA = C.ID_COMANDA
JOIN CLIENTI CL ON CL.ID_CLIENT = C.ID_CLIENT
JOIN ADRESE A ON A.ID_CLIENT = CL.ID_CLIENT
WHERE UPPER(A.TARA) LIKE 'ROMANIA'
    AND F.NOTA > 5
    AND UPPER(C.LIVRARE) LIKE 'CURIER';
```

▪ Cod sursă:

```
Worksheet | Query Builder
1 SELECT DISTINCT
2   CL.NUME,
3   CL.PRENUME,
4   A.ORAS,
5   CASE
6     WHEN F.NOTA >7 THEN 'Buna'
7     ELSE 'Acceptabila'
8   END CALITATE,
9   F.DESCRIERE,
10  C.STARE
11 FROM FEEDBACKURI F
12 JOIN COMENZI C ON F.ID_COMANDA = C.ID_COMANDA
13 JOIN CLIENTI CL ON CL.ID_CLIENT = C.ID_CLIENT
14 JOIN ADRESE A ON A.ID_CLIENT = CL.ID_CLIENT
15 WHERE UPPER(A.TARA) LIKE 'ROMANIA'
16        AND F.NOTA > 5
17        AND UPPER(C.LIVRARE) LIKE 'CURIER';
18
19
20
```

▪ Rulare cod sursă:

Query Result x						
SQL   All Rows Fetched: 3 in 0.007 seconds						
	NUME	PRENUME	ORAS	CALITATE	DESCRIERE	STARE
1	Stoivici	Alexandra	Sector 4	Buna	Calitate si personalizare	PRELUATA
2	Ionescu	Lidia	Pucioasa	Acceptabila	Ok	FINALIZATA
3	Ionescu	Lidia	Bragadiru	Acceptabila	Ok	FINALIZATA

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

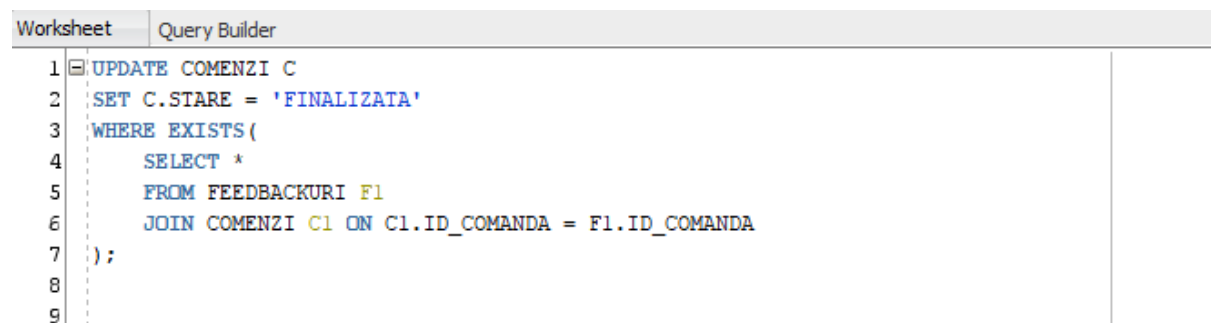
#### Operația 1

Modificați tabelul comenzi astfel încât toate comenzile pentru care există feedback să aibă starea 'FINALIZATA'.

- SQL:

```
UPDATE COMENZI C
SET C.STARE = 'FINALIZATA'
WHERE EXISTS(
    SELECT *
    FROM FEEDBACKURI F1
    JOIN COMENZI C1 ON C1.ID_COMANDA = F1.ID_COMANDA
);
```

- Cod sursă:

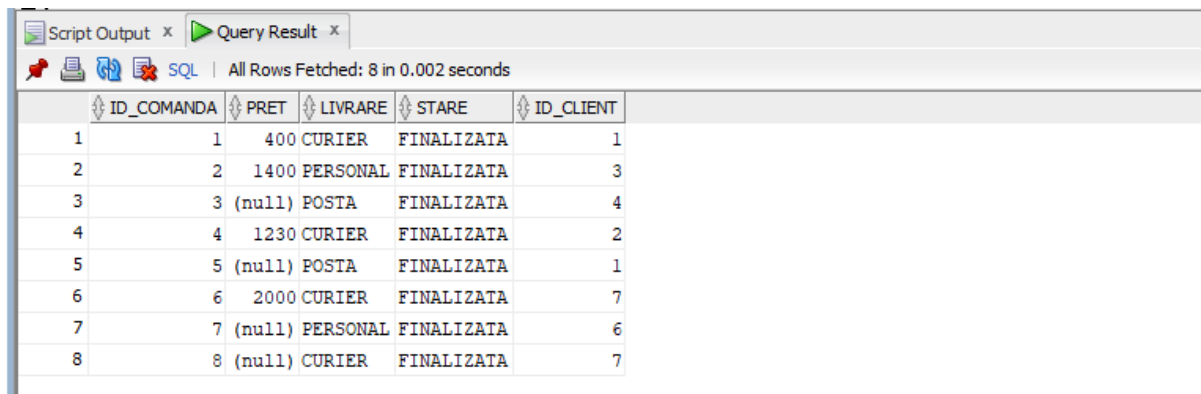


The screenshot shows a 'Query Builder' window with a 'Worksheet' tab. The SQL code is displayed in a text area with line numbers 1 through 9. The code is: UPDATE COMENZI C; SET C.STARE = 'FINALIZATA'; WHERE EXISTS( SELECT \* FROM FEEDBACKURI F1 JOIN COMENZI C1 ON C1.ID\_COMANDA = F1.ID\_COMANDA );

- Rulare cod sursă:



The screenshot shows the result of the query execution: 8 rows updated.



Script Output x Query Result x

SQL | All Rows Fetched: 8 in 0.002 seconds

ID_COMANDA	PRET	LIVRARE	STARE	ID_CLIENT
1	1	400 CURIER	FINALIZATA	1
2	2	1400 PERSONAL	FINALIZATA	3
3	3 (null)	POSTA	FINALIZATA	4
4	4	1230 CURIER	FINALIZATA	2
5	5 (null)	POSTA	FINALIZATA	1
6	6	2000 CURIER	FINALIZATA	7
7	7 (null)	PERSONAL	FINALIZATA	6
8	8 (null)	CURIER	FINALIZATA	7

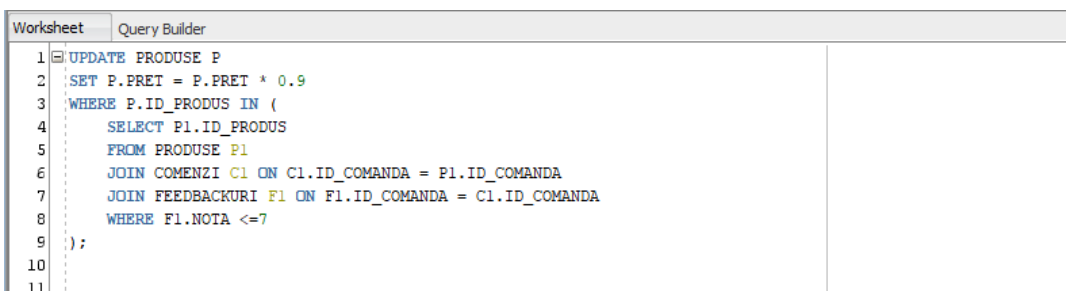
## Operația 2

Modificați tabelul produse astfel încât toate produsele care se află în comenzi cu nota feedbackului mai mică sau egală cu 7 să primească o reducere de 10%.

- SQL:

```
UPDATE PRODUSE P
SET P.PRET = P.PRET * 0.9
WHERE P.ID_PRODUS IN (
    SELECT P1.ID_PRODUS
    FROM PRODUSE P1
    JOIN COMENZI C1 ON C1.ID_COMANDA = P1.ID_COMANDA
    JOIN FEEDBACKURI F1 ON F1.ID_COMANDA = C1.ID_COMANDA
    WHERE F1.NOTA <=7
);
```

- Cod sursă:



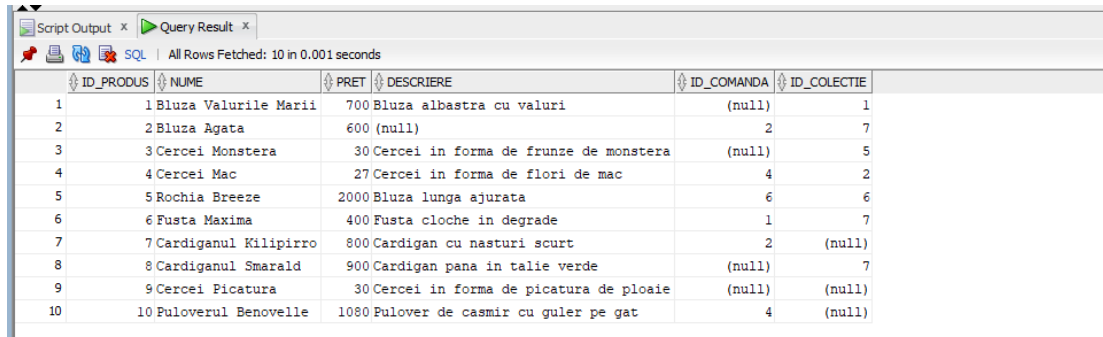
Worksheet Query Builder

```
1 UPDATE PRODUSE P
2 SET P.PRET = P.PRET * 0.9
3 WHERE P.ID_PRODUS IN (
4     SELECT P1.ID_PRODUS
5     FROM PRODUSE P1
6     JOIN COMENZI C1 ON C1.ID_COMANDA = P1.ID_COMANDA
7     JOIN FEEDBACKURI F1 ON F1.ID_COMANDA = C1.ID_COMANDA
8     WHERE F1.NOTA <=7
9 );
10
11
```



- Rulare cod sursă:

2 rows updated.



ID_PRODUS	NUME	PRET	DESCRIERE	ID_COMANDA	ID_COLECTIE
1	1 Bluza Valurile Marii	700	Bluza albastra cu valuri	(null)	1
2	2 Bluza Agata	600 (null)		2	7
3	3 Cercei Monstera	30	Cercei in forma de frunze de monstera	(null)	5
4	4 Cercei Mac	27	Cercei in forma de flori de mac	4	2
5	5 Rochia Breeze	2000	Bluza lunga ajurata	6	6
6	6 Fusta Maxima	400	Fusta cloche in degrade	1	7
7	7 Cardiganul Kilipirro	800	Cardigan cu nasturi scurt	2	(null)
8	8 Cardiganul Smarald	900	Cardigan pana in talie verde	(null)	7
9	9 Cercei Picatura	30	Cercei in forma de picatura de ploaie	(null)	(null)
10	10 Puloverul Benovelle	1080	Pulover de casmir cu guler pe gat	4	(null)

### Operația 3

Ștergeți toate contractele pentru materialele care nu sunt textile.

- SQL:

DELETE

FROM CONTRACTE C

WHERE C.ID\_CONTRACT IN (

SELECT C1.ID\_CONTRACT

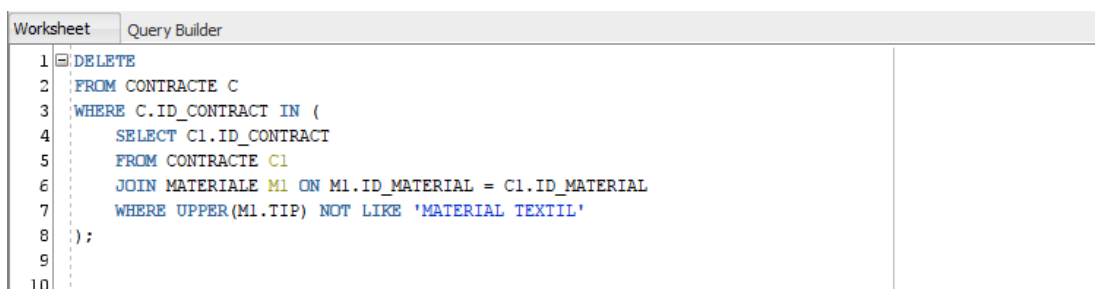
FROM CONTRACTE C1

JOIN MATERIALE M1 ON M1.ID\_MATERIAL = C1.ID\_MATERIAL

WHERE UPPER(M1.TIP) NOT LIKE 'MATERIAL TEXTIL'

);

- Cod sursă:







	Worksheet	Query Builder
1	DELETE	
2	FROM CONTRACTE C	
3	WHERE C.ID_CONTRACT IN (	
4	SELECT C1.ID_CONTRACT	
5	FROM CONTRACTE C1	
6	JOIN MATERIALE M1 ON M1.ID_MATERIAL = C1.ID_MATERIAL	
7	WHERE UPPER(M1.TIP) NOT LIKE 'MATERIAL TEXTIL'	
8	);	
9		
10		

- Rulare cod sursă:

```
1 row deleted.
```

Script Output x | Query Result x

    SQL | All Rows Fetched: 14 in 0.001 seconds

	ID_CONTRACT	DATA	ID_FURNIZOR	ID_MATERIAL
1	1	15/JAN/23	1	1
2	2	20/FEB/23	2	2
3	3	10/MAR/23	3	3
4	4	05/APR/23	4	4
5	6	22/JUN/23	6	6
6	7	09/JUL/23	7	7
7	8	14/AUG/23	8	8
8	9	30/SEP/23	9	9
9	10	25/OCT/23	10	10
10	11	12/NOV/23	1	6
11	12	03/DEC/23	2	7
12	13	28/JAN/24	3	8
13	14	10/FEB/24	4	9
14	15	20/MAR/24	5	10

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

Creare vizualizare

*Enunț:* Creați o vizualizare care, pentru fiecare comandă, prezintă ID-ul comenzii, numele și prenumele clientului care îi corespunde, iar mai apoi tipul de livrare, prețul comenzii (numit subtotal), costul de livrare (curier: 18.00, poștă: 10.00, ridicare personală: 0.00), și în final totalul de plată.

▪ SQL:

CREATE OR REPLACE VIEW

PLATI\_COMENZI

AS SELECT

C.ID\_COMANDA,

CL.NUME,

CL.PRENUME,

C.LIVRARE,

NVL(C.PRET, 0) AS SUBTOTAL,

DECODE(UPPER(C.LIVRARE),

'CURIER', 18.00,

'POSTA', 10.00,

'PERSONAL', 0.00

) AS COST\_LIVRARE,

DECODE(UPPER(C.LIVRARE),

'CURIER', NVL(C.PRET, 0) + 18.00,

'PERSONAL', NVL(C.PRET, 0),

'POSTA', NVL(C.PRET, 0) + 10.00

) AS TOTAL

FROM COMENZI C

JOIN CLIENTI CL ON CL.ID\_CLIENT = C.ID\_CLIENT  
ORDER BY CL.NUME;

▪ Cod sursă:

```
Worksheet Query Builder
1 CREATE OR REPLACE VIEW
2   PLATI_COMENZI
3 AS SELECT
4   C.ID_COMANDA,
5   CL.NUME,
6   CL.PRENUME,
7   C.LIVRARE,
8   NVL(C.PRET, 0) AS SUBTOTAL,
9   DECODE(UPPER(C.LIVRARE),
10    'CURIER', 18.00,
11    'POSTA', 10.00,
12    'PERSONAL', 0.00
13   ) AS COST_LIVRARE,
14   DECODE(UPPER(C.LIVRARE),
15    'CURIER', NVL(C.PRET, 0) + 18.00,
16    'PERSONAL', NVL(C.PRET, 0),
17    'POSTA', NVL(C.PRET, 0) + 10.00
18   ) AS TOTAL
19 FROM COMENZI C
20 JOIN CLIENTI CL ON CL.ID_CLIENT = C.ID_CLIENT
21 ORDER BY CL.NUME;
```

▪ Rulare cod sursă:

Welcome Page x daniel x PLATI_COMENZI x							
Columns Data Grants Dependencies Details Triggers SQL Errors							
	ID_COMANDA	NUME	PRENUME	LIVRARE	SUBTOTAL	COST_LIVRARE	TOTAL
1	4	Ionescu	Lidia	CURIER	1230	18	1248
2	2	Popescu	Florin	PERSONAL	1400	0	1400
3	7	Rusu	Ana-Maria	PERSONAL	0	0	0
4	5	Stoivici	Alexandra	POSTA	0	10	10
5	1	Stoivici	Alexandra	CURIER	400	18	418
6	8	Vacaru	Marta	CURIER	0	18	18
7	6	Vacaru	Marta	CURIER	2000	18	2018
8	3	Vasile	Cristian	POSTA	0	10	10

### Operație LMD permisă

*Enunț:* Modificați vizualizarea astfel încât pentru toate comenzile cu subtotal 0, modul de livrare să fie 'PERSONAL'.

- SQL:

```
UPDATE PLATI_COMENZI  
SET LIVRARE='PERSONAL'  
WHERE SUBTOTAL=0;
```

- Cod sursă:

Worksheet		Query Builder
1	UPDATE PLATI_COMENZI	
2	SET LIVRARE='PERSONAL'	
3	WHERE SUBTOTAL=0;	
4		

- Rulare cod sursă:

	ID_COMANDA	NUME	PRENUME	LIVRARE	SUBTOTAL	COST_LIVRARE	TOTAL
1	4	Ionescu	Lidia	CURIER	1230	18	1248
2	2	Popescu	Florin	PERSONAL	1400	0	1400
3	7	Rusu	Ana-Maria	PERSONAL	0	0	0
4	5	Stoivici	Alexandra	PERSONAL	0	0	0
5	1	Stoivici	Alexandra	CURIER	400	18	418
6	8	Vacaru	Marta	PERSONAL	0	0	0
7	6	Vacaru	Marta	CURIER	2000	18	2018
8	3	Vasile	Cristian	PERSONAL	0	0	0

4 rows updated.

### Operație LMD nepermisă

*Enunț:* Modificați vizualizarea astfel încât pentru comanda cu ID-ul 5, subtotalul să fie 200.

- SQL:

```
UPDATE PLATI_COMENZI  
SET SUBTOTAL = 200.00  
WHERE ID_COMANDA = 5;
```

- Cod sursă:

Worksheet	Query Builder
1	UPDATE PLATI_COMENZI
2	SET SUBTOTAL = 200.00
3	WHERE ID_COMANDA = 5;
4	
5	
6	

- Rulare cod sursă:

```
Error starting at line : 1 in command -  
UPDATE PLATI_COMENZI  
SET SUBTOTAL = 200.00  
WHERE ID_COMANDA = 5  
Error at Command Line : 2 Column : 5  
Error report -  
SQL Error: ORA-01733: virtual column not allowed here  
01733. 00000 - "virtual column not allowed here"  
*Cause:  
*Action:
```

\* deoarece coloana subtotal este obținută printr-un NVL (și deci, coloană virtuală), aceasta nu poate fi modificată de utilizator

15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

OUTER-JOIN pe minimum 4 tabele

Pentru fiecare care costă mai puțin de 1000 de lei, afișați numele, prețul, și, indiferent dacă există sau nu (dacă nu există, se va afișa null): numele și anul colecției din care face parte, numele și prenumele clientului care l-a achiziționat, și în final nota feedback-ului pentru comanda din care acesta face parte.

▪ SQL:

SELECT

P.NUME NUME\_PRODUS,

P.PRET,

COL.NUME NUME\_COLECTIE,

COL.AN AN\_COLECTIE,

CL.NUME NUME\_CLIENT,

CL.PRENUME PRENUME\_CLIENT,

F.NOTA NOTA\_FEEDBACK

FROM PRODUSE P

LEFT OUTER JOIN COLECTII COL ON COL.ID\_COLECTIE = P.ID\_COLECTIE

LEFT OUTER JOIN COMENZI COM ON COM.ID\_COMANDA = P.ID\_COMANDA

LEFT OUTER JOIN CLIENTI CL ON CL.ID\_CLIENT = COM.ID\_CLIENT

LEFT OUTER JOIN FEEDBACKURI F ON F.ID\_COMANDA = COM.ID\_COMANDA


WHERE P.PRET < 1000;

- Cod sursă:

	Worksheet	Query Builder
1	SELECT	
2	P.NUME NUME_PRODUS,	
3	P.PRET,	
4	COL.NUME NUME_COLECTIE,	
5	COL.AN AN_COLECTIE,	
6	CL.NUME NUME_CLIENT,	
7	CL.PRENUME PRENUME_CLIENT,	
8	F.NOTA NOTA_FEEDBACK	
9	FROM PRODUSE P	
10	LEFT OUTER JOIN COLECTII COL ON COL.ID_COLECTIE = P.ID_COLECTIE	
11	LEFT OUTER JOIN COMENZI COM ON COM.ID_COMANDA = P.ID_COMANDA	
12	LEFT OUTER JOIN CLIENTI CL ON CL.ID_CLIENT = COM.ID_CLIENT	
13	LEFT OUTER JOIN FEEDBACKURI F ON F.ID_COMANDA = COM.ID_COMANDA	
14	WHERE P.PRET < 1000;	
15		
16		

- Rulare cod sursă:

Script Output x Query Result x

 All Rows Fetched: 8 in 0.006 seconds

	NUME_PRODUS	PRET	NUME_COLECTIE	AN_COLECTIE	NUME_CLIENT	PRENUME_CLIENT	NOTA_FEEDBACK
1	Bluza Agata	600	Jewels	2020	Popescu	Florin	8
2	Cardiganul Kilipirro	800	(null)	(null)	Popescu	Florin	8
3	Fusta Maxima	400	Jewels	2020	Stoivici	Alexandra	9
4	Cercei Mac	30	Summertime	2023	Ionescu	Lidia	7
5	Bluza Valurile Marii	700	Prima Verra	2022	(null)	(null)	(null)
6	Cercei Monstera	30	Sweet Autumn	2019	(null)	(null)	(null)
7	Cardiganul Smarald	900	Jewels	2020	(null)	(null)	(null)
8	Cercei Picatura	30	(null)	(null)	(null)	(null)	(null)



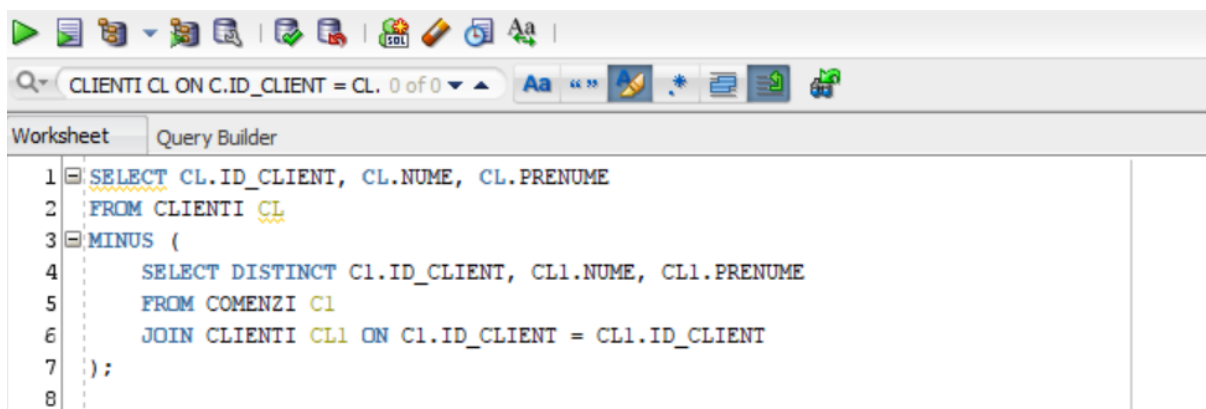
### Operația DIVISION

Să se afișeze ID-ul, numele și prenume pentru fiecare „potențial” client (clienții introduși în baza de date, dar care nu au nicio comandă).

- SQL:

```
SELECT CL.ID_CLIENT, CL.NUME, CL.PRENUME  
FROM CLIENTI CL  
MINUS (  
    SELECT DISTINCT C1.ID_CLIENT, CL1.NUME, CL1.PRENUME  
    FROM COMENZI C1  
    JOIN CLIENTI CL1 ON C1.ID_CLIENT = CL1.ID_CLIENT  
);
```

- Cod sursă:



- Rulare cod sursă:

The screenshot shows a SQL query result window titled "Query Result x". It displays the results of the query in a table format. The status bar indicates "All Rows Fetched: 2 in 0.094 seconds".

	ID_CLIENT	NUME	PRENUME
1	8	Nicolae	Vladimir
2	5	Marin	Mihaela

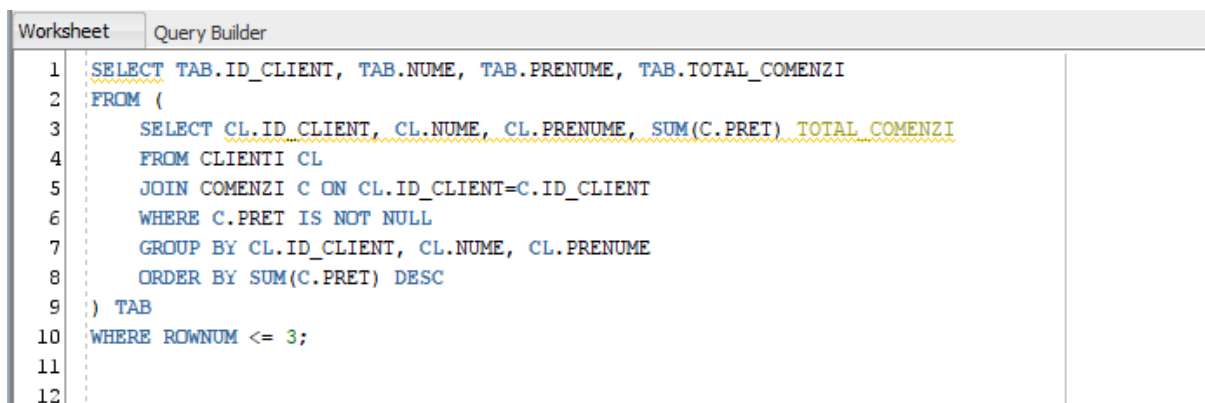
### Analiza TOP-N

Să se afișeze pentru cei 3 clienți care au cumpărat de sume cât mai mari de la magazin ID-ul, numele, prenumele, și totalul de sume plătite.

- SQL:

```
SELECT TAB.ID_CLIENT, TAB.NUME, TAB.PRENUME, TAB.TOTAL_COMENZI
FROM (
    SELECT CL.ID_CLIENT, CL.NUME, CL.PRENUME, SUM(C.PRET) TOTAL_COMENZI
    FROM CLIENTI CL
    JOIN COMENZI C ON CL.ID_CLIENT=C.ID_CLIENT
    WHERE C.PRET IS NOT NULL
    GROUP BY CL.ID_CLIENT, CL.NUME, CL.PRENUME
    ORDER BY SUM(C.PRET) DESC
) TAB
WHERE ROWNUM <= 3;
```

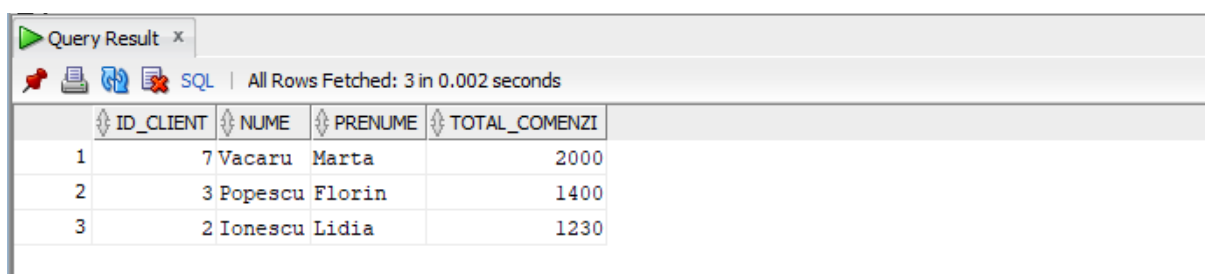
- Cod sursă:



The screenshot shows a 'Query Builder' window with a text editor containing the following SQL code:

```
1 SELECT TAB.ID_CLIENT, TAB.NUME, TAB.PRENUME, TAB.TOTAL_COMENZI
2 FROM (
3     SELECT CL.ID_CLIENT, CL.NUME, CL.PRENUME, SUM(C.PRET) TOTAL_COMENZI
4     FROM CLIENTI CL
5     JOIN COMENZI C ON CL.ID_CLIENT=C.ID_CLIENT
6     WHERE C.PRET IS NOT NULL
7     GROUP BY CL.ID_CLIENT, CL.NUME, CL.PRENUME
8     ORDER BY SUM(C.PRET) DESC
9 ) TAB
10 WHERE ROWNUM <= 3;
```

- Rulare cod sursă:



The screenshot shows a 'Query Result' window with the following data:

ID_CLIENT	NUME	PRENUME	TOTAL_COMENZI
1	7 Vacaru	Marta	2000
2	3 Popescu	Florin	1400
3	2 Ionescu	Lidia	1230