### UNIVERSITATEA DIN BUCUREȘTI

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**PROIECT**

# Baze de date

### Student

**Georgescu Andrei**

**Profesor**

**Lect. dr. Iulia Banu – Demergianu**

**Anul I**

**Semestrul 2**

**Grupa 1**

**București, iunie 2024**

#### Tema

Proiectarea unei baze de date relaţională (SQL) pentru site-ul comercial de electronice GadgetHaven – acest site reprezintă proiectul personal pentru materia Tehnici Web.

#### Tehnologii utilizate

Pentru realizarea acestui proiect am folosit următoarele tehnologii:

**Docker** - v24.0.5 ( pentru containerizarea bazei de date )

**OracleDB** – v19.3 ( baza de date SQL )

**SQL Developer** - v23.1.1 ( client pentru baza de date – pentru inserare date/interogare )

sqlplus sys/oracle19\_3 as sysdba

sqlplus proiect/proiect@localhost:1521/ORCLPDB1

ORCLPDB1

**Admin ( SYSDBA )**

Username: sys

Password: oracle19\_3

**User ( default )**

Username: proiect

Password: proiect

Cuprins

[UNIVERSITATEA DIN BUCUREȘTI 1](#_Toc168862851)

[Baze de date 1](#_Toc168862852)

[Student 1](#_Toc168862853)

[1. Etapa 1 5](#_Toc168862854)

[2. Etapa 2 5](#_Toc168862855)

[3. Etapa 3 6](#_Toc168862856)

[4. Etapa 4 6](#_Toc168862857)

[5. Etapa 5 7](#_Toc168862858)

[6. Etapa 6 11](#_Toc168862859)

[7. Etapa 7 11](#_Toc168862860)

[8. Etapa 8 11](#_Toc168862861)

[9. Etapa 9 12](#_Toc168862862)

[10. Etapa 10 12](#_Toc168862863)

[11. Etapa 11 13](#_Toc168862864)

[12. Etapa 12 24](#_Toc168862865)

[13. Etapa 13 27](#_Toc168862866)

[14. Etapa 14 28](#_Toc168862867)

[15. Etapa 15 29](#_Toc168862868)

[16. Etapa 16 30](#_Toc168862869)

[17. Etapa 17 30](#_Toc168862870)

[Bibliografie 32](#_Toc168862871)

### Etapa 1

**Descrierea modelului real, a utilităţii acestuia şi a regulilor de funcţionare.**

**Modelul real:** GadgetHaven este un site de comerț specializat în vânzarea de electronice și accesorii electronice. Baza de date a acestui site trebuie să gestioneze informațiile despre produse, clienți, comenzi, stocuri și alte aspecte necesare pentru funcționarea magazinului online.

**Utilitatea:** Baza de date va permite gestionarea eficientă a stocurilor, urmărirea comenzilor, administrarea informațiilor despre clienți și furnizarea de rapoarte pentru analiza vânzărilor și a performanțelor produselor, de asemenea gestionarea problemelor cu ajutorul tichetelor create de către clienţi.

**Reguli de funcționare:**

* Fiecare stoc trebuie să aibă un produs asociat.
* Fiecare comandă trebuie să fie asociată unui client.
* Fiecare detaliu de comandă trebuie sa aibă asociat un produs si o comanda.
* Fiecare aprovizionare trebuie sa aibă asociată un produs si un furnizor.
* Prețurile și descrierile produselor trebuie să fie precise și actualizate.
* Fiecare tichet trebuie sa aibă o problemă asociată

### Etapa 2

**Prezentarea constrângerilor ( restricţii, reguli ) impuse asupra modelului.**

**Constrangeri:**

* Un admin poate **rezolva** una sau mai multe probleme.
* Un tichet poate **include** una sau mai multe probleme.
* Un client poate **crea** zero sau mai multe tichete.
* Un client poate **scrie** zero sau mai multe recenzii.
* Un client poate **crea** zero sau mai multe wishlisturi.
* Un client poate **plasa** zero sau mai multe comenzi.
* O comanda poate **include** doar un detaliu\_comandă.
* Un wishlist poate **conţine** unul sau mai multe produse.
* Un produs poate **avea** zero sau mai multe recenzii.
* Un produs poate **face** parte dintr-un singur stoc.
* Un produs pote **avea** una sau mai multe aprovizionări.
* Un furnizor poate **face** una sau mai multe aprovizionări.
* Un detaliu\_comandă poate **avea** unul sau mai multe produse.
* Chei primare unice pentru identificarea clară a fiecărei entități.
* Chei străine pentru menținerea integrității referențiale între tabele.
* Constrângeri de unicitate pentru anumite câmpuri (de exemplu, adresele de e-mail ale clienților).
* Constrângeri CHECK pentru validarea datelor (de exemplu, prețuri pozitive).
* Valori implicite pentru anumite câmpuri (de exemplu, data comenzii implicită să fie data curentă).

### Etapa 3

**Descrierea entităţilor, incluzând precizarea cheii primare.**

|  |  |  |
| --- | --- | --- |
| **ENTITATE** | **CHEIE PRIMARĂ** | **OBSERVAŢII** |
| Client | client\_id | Informaţii despre client ( nume, prenume,email) |
| Produs | produs\_id | Informaţii despre produs ( descriere, preţ, categorie ) |
| Categorie | categorie\_id | Numele categoriei |
| Furnizor | furnizor\_id | Informaţii despre furnizor ( nume, adresa/contact ) |
| Stoc | stoc\_id | Informaţii despre stoc ( cantitate, data ultimei aprovizionări ) |
| Comanda | comanda\_id | Informaţii despre comandă ( data, total ) |
| Detalii\_Comanda | detalii\_id | Detalii legate de comandă ( cantitate, preţ ) |
| Aprovizionare | aprovizionare\_id | Informaţii despre aprovizionari ( cantitate, data ) |
| Admin | admin\_id | Informaţii despre admini ( nume, prenume) |
| Tichet | tichet\_id | Informaţii legate de tichet ( status, severitate, date de creare şi închidere ) |
| Wishlist | wishlist\_id | Informaţii legate de wishlist ( produsul, notificare ) |

### Etapa 4

**Descrierea relaţiilor, incluzând precizarea cardinalităţii acestora.**

|  |  |  |
| --- | --- | --- |
| RELATIE | CARDINALITATE | OBSERVATII |
| are | Produs-Recenzie ( one-to-many )  Produs-Aprovizionare ( one-to-many)  Detaliu\_comanda-Produs ( one-to-many ) | Un produs poate avea zero sau mai multe recenzii  Un produs pote avea una sau mai multe aprovizionări  Un detaliu\_comandă poate avea unul sau mai multe produse |
| conţine | Wishlist-Produs | Un wishlist poate conţine unul sau mai multe produse. |
| rezolvă | Admin-Problema | Un admin poate rezolva una sau mai multe probleme. |
| include | Tichet-Problema  Comanda-Detaliu\_Comanda | Un tichet poate include una sau mai multe probleme.  O comanda poate include doar un detaliu\_comandă. |
| crează | Client-Tichet  Client-Wishlist | Un client poate crea zero sau mai multe tichete.  Un client poate crea zero sau mai multe wishlisturi. |
| scrie | Client-Recenzie | Un client poate scrie zero sau mai multe recenzii. |
| plasează | Client-Comanda | Un client poate plasa zero sau mai multe comenzi. |
| face | Produs-Stoc  Furnizor-Aprovizionare | Un produs poate face parte dintr-un singur stoc.  Un furnizor poate face una sau mai multe aprovizionări. |

### Etapa 5

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

**ENTITATE: CLIENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **client\_id** | **NUMBER** |  |  | **Cheia de identificare unică a entităţii „Client”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume\_client | VARCHAR | 15 |  | Numele clientului, NOT NULL |
| prenume\_client | VARCHAR | 15 |  | Prenumele clientului, NOT NULL |
| email | VARCHAR | 35 |  | Emailul clientului, NOT NULL, UNIQUE |
| telefon | VARCHAR | 13 |  | Numarul de telefon al clientului, NOT NULL |
| adresa | VARCHAR | 50 |  | Adresa clientului, NOT NULL |
| cont\_premium | VARCHAR | 2 | Valoare posibilă ‘da’ / Valoare default ‚nu’ | Tipul de cont al clientului |

**ENTITATE: ADMIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **admin\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „ADMIN”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume\_admin | VARCHAR | 15 |  | NOT NULL |
| prenume\_admin | VARCHAR | 15 |  | NOT NULL |

**RELATIE: PROBLEMA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| tichet\_id | NUMBER |  |  | FOREIGN KEY –  referinta Tichet(ticet\_id) |
| admin\_id | NUMBER |  |  | FOREIGN KEY – referinta Admin(admin\_id) |
| descriere\_problema | VARCHAR | 200 |  | NOT NULL |
| categorie\_problema | VARCHAR | 25 |  | NOT NULL |

**RELATIE: RECENZIE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| produs\_id | NUMBER |  |  | FOREIGN KEY – referinta Produs(produs\_id) |
| client\_id | NUMBER |  |  | FOREIGN KEY – referinta Client(client\_id) |
| tip\_recenzie | VARCHAR | 8 | pozitiv/negativ |  |
| continut\_recenzie | VARCHAR | 200 |  | NOT NULL |
| produs\_cumparat | VARCHAR | 2 | da/nu | NOT NULL |
| aprecieri | NUMBER |  |  |  |

**ENTITATE: TICHET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| tichet\_id | NUMBER |  |  | PRIMARY KEY – autoincrement ( identity ) |
| client\_id | NUMBER |  |  | FOREIGN KEY – referinta Client(client\_id) |
| status\_tichet | VARCHAR | 25 | deschis/în progres/ închis | NOT NULL |
| severitate | VARCHAR | 15 | critic/major/minor | NOT NULL |
| data\_creare\_tichet | DATE |  |  | NOT NULL |
| data\_inchidere\_tichet | DATE |  |  |  |

**ENTITATE: WISHLIST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| wishlist\_id | NUMBER |  |  | PRIMARY KEY – autoincrement (identity) |
| produs\_id | NUMBER |  |  | FOREIGN KEY – referinta Produs(produs\_id) |
| client\_id | NUMBER |  |  | FOREIGN KEY – referinta Client(client\_id) |
| notificare | VARCHAR | 2 | default – da  posibil - nu |  |

**ENTITATE: PRODUS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **produs\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Produs”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume\_produs | VARCHAR | 40 |  | Numele produsului, NOT NULL |
| descriere\_produs | VARCHAR | 100 |  | Descrierea produsului, NULL |
| pret\_produs | NUMBER |  |  | Pretul produsului, NOT NULL |
| categorie\_produs | NUMBER | 25 |  | ID-ul categoriei asociat produsului, FOREIGN KEY, NOT NULL |
| oferta | VARCHAR | 2 | Valoare posibila ‚da’ / Valoare default ‚nu’ | Daca produsul este la oferta sau nu |

**ENTITATE: FURNIZOR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **furnizor\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Furnizori”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume\_furnizor | VARCHAR | 30 |  | Numele furnizorului, NOT NULL |
| adresa\_furnizor | VARCHAR | 30 |  | Adresa furnizorului, NOT NULL |
| email\_furnizor | VARCHAR | 30 |  | NOT NULL |
| fax\_furnizor | VARCHAR | 25 |  |  |
| produs\_furnizat | VARCHAR | 25 |  | NOT NULL |

**ENTITATE: STOC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **stoc\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Stoc”, PRIMARY KEY, AUTO\_INCREMENT** |
| produs\_id | NUMBER |  |  | ID-ul produsului asociat stocului, FOREIGN KEY, NOT NULL |
| cantitate | NUMBER |  |  | Cantiatea stocului, NOT NULL |
| data\_ultima\_aprovizionare | DATE |  |  | Data ultimei aprovizionari, NOT NULL |

**ENTITATE: COMANDA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **comanda\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Comanda”, PRIMARY KEY, AUTO\_INCREMENT** |
| client\_id | NUMBER |  |  | ID-ul clientului asociat comenzii, FOREIGN KEY, NOT NULL |
| data\_comanda | DATE |  |  | Data cand s-a facut comanda, NOT NULL |
| pret\_total | NUMBER |  |  | Totalul de plata al comenzii, NOT NULL |
| mod\_plata | VARCHAR | 25 | Cash, PayPal, Revolut, Visa, Mastercard | NOT NULL |

**ENTITATE: DETALII\_COMANDA**

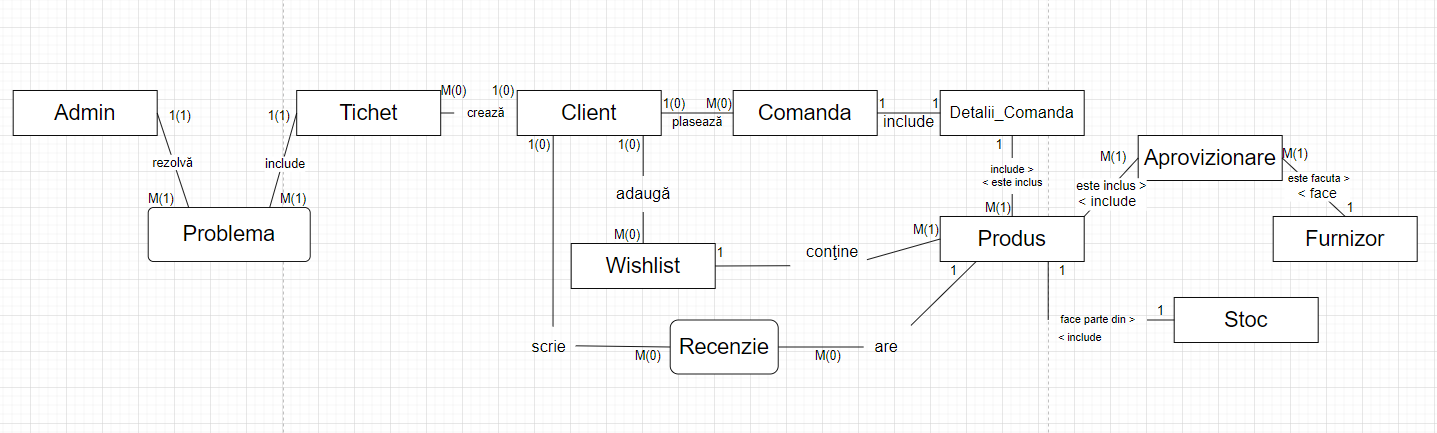
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **detalii\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „DetaliiComanda”, PRIMARY KEY, AUTO\_INCREMENT** |
| comanda\_id | NUMBER |  |  | FOREIN KEY, NOT NULL |
| produs\_id | NUMBER |  |  | FOREIN KEY, NOT NULL |
| cantitate | NUMBER |  |  | NOT NULL |
| pret\_unitar | NUMBER |  |  | NOT NULL |
| pret\_fara\_tva | DECIMAL |  |  | NOT NULL |
| pret\_cu\_tva | DECIMAL |  |  | NOT NULL |
| tva | DECIMAL |  |  | NOT NULL |
| cod\_promotional | VARCHAR | 15 |  |  |

**ENTITATE: APROVIZIONARE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile şi valori default** | **Observaţii, obligatoriu/opţional** |
| **aprovizionare\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Aprovizionare”, PRIMARY KEY, AUTO\_INCREMENT** |
| produs\_id | NUMBER |  |  | FOREIGN KEY, NOT NULL |
| furnizor\_id | NUMBER |  |  | FOREIGN KEY, NOT NULL |
| cantiate | NUMBER |  |  | NOT NULL |
| data\_aprovizionare | DATE |  |  | NOT NULL |

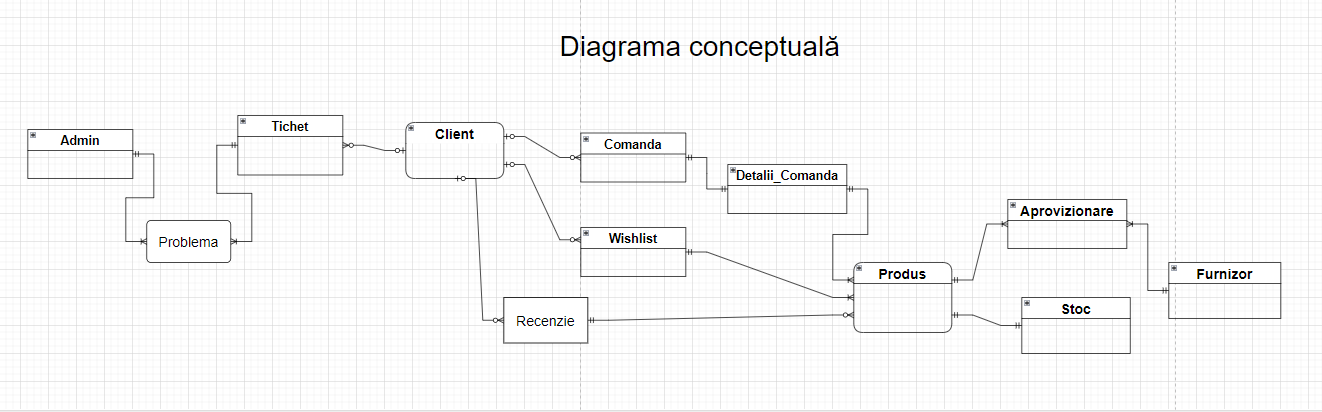
### Etapa 6

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



### Etapa 7

**Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relaţie proiectate la punctul 6. Diagrama conceptuală obţinută treubie să conţină minim 7 tabele ( fără considerarea subentităţilor), dintre cel puţin un table asociativ.**

****

### Etapa 8

**Enumerarea schemelor relaţionale corespunzătoare diagramei conceptuale proiectate la punctul 7.**

**Client**(client\_id#, nume\_client, prenume\_client, parola, email, telefon, adresa, cont\_premium)

**Produs**(produs\_id#, nume\_produs, descriere\_produs, pret\_produs, categorie\_produs, oferta)

**Furnizor**(furnizor\_id#, nume\_furnizor, adresa\_furnizor, email\_furnizor, fax\_furnizor, produs\_furnizat)

**Comanda**(comanda\_id#, client\_id#, data\_comanda, data\_scadenta, pret\_total, mod\_plata)

**Detalii\_Comanda**(detalii\_id#, comanda\_id#, produs\_id#, cantitate, pret\_unitar, pret\_fara\_tva, pret\_cu\_tva, tva, cod\_promotional)

**Stoc**(stoc\_id#, produs\_id#, cantitate, data\_ulitma\_aprovizionare)

**Aprovizionare**(aprovizionare\_id#, produs\_id#, furnizor\_id#, cantitate, data\_aprovizionare)

**Tichet**(tichet\_id#, client\_id#, status\_tichet, severitate, data\_creare\_tichet, data\_închidere\_tichet)

**Admin(**admin\_id#, nume\_admin, prenume\_admin)

**Problema**(tichet\_id#, admin\_id#, descriere\_problema, categorie\_problema)

**Wishlist**(wishlist\_id#, produs\_id#,client\_id#, notificare)

**Recenzie**(produs\_id#, client\_id#, tip\_recenzie, continut\_recenzie, produs\_cumparat, aprecieri)

### Etapa 9

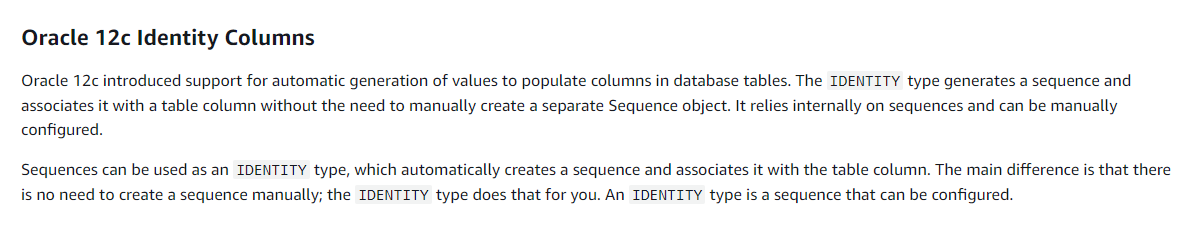
**Realizarea normalizării până la forma normală 3 (FN1-FN3).**

1. **FN1**: Toate tabelele sunt în FN1 deoarece fiecare atribut conține valori atomice.
2. **FN2**: Toate tabelele sunt în FN2 deoarece toate atributele non-cheie depind complet de cheia primară.
3. **FN3**: Toate tabelele sunt în FN3 deoarece nu există dependențe tranzitive între atributele non-cheie.

### Etapa 10

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

Pentru secvenţe am folosit **IDENTITY** – genereaza o secvenţă asociată coloanei fără a mai fi nevoie sa fie creat un obiect secvenţă separat.



CREATE TABLE Client (

    client\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Produs (

    produs\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Furnizor (

    furnizor\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Stoc (

    stoc\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Comanda (

    comanda\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Detalii\_Comanda (

    detalii\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Aprovizionare (

    aprovizionare\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Admin (

    admin\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Tichet (

    tichet\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Wishlist (

    wishlist\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

### Etapa 11

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

**CREAZĂ TABELE**

CREATE TABLE Client (

    client\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY

    (START WITH 1 INCREMENT BY 1 MINVALUE 1),

    nume\_client VARCHAR(15) NOT NULL,

    prenume\_client VARCHAR(15) NOT NULL,

    parola VARCHAR(35) NOT NULL,

    email VARCHAR(35) NOT NULL UNIQUE,

    telefon VARCHAR(13) NOT NULL,

    adresa VARCHAR(50) NOT NULL,

    cont\_premium VARCHAR(2) DEFAULT 'nu',

    PRIMARY KEY (client\_id)

);

CREATE TABLE Produs (

    produs\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    nume\_produs VARCHAR(40) NOT NULL,

    descriere\_produs VARCHAR(100),

    pret\_produs NUMBER NOT NULL,

    categorie\_produs VARCHAR(25) NOT NULL,

    oferta VARCHAR(2) DEFAULT 'nu',

    PRIMARY KEY (produs\_id)

);

CREATE TABLE Furnizor (

    furnizor\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    nume\_furnizor VARCHAR(30) NOT NULL,

    adresa\_furnizor VARCHAR(30) NOT NULL,

    email\_furnizor VARCHAR(30) NOT NULL,

    fax\_furnizor VARCHAR(25),

    produs\_furnizat VARCHAR(25) NOT NULL,

    PRIMARY KEY (furnizor\_id)

);

CREATE TABLE Stoc (

    stoc\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    produs\_id NUMBER,

    cantitate NUMBER NOT NULL,

    data\_ultima\_aprovizionare DATE NOT NULL,

    PRIMARY KEY (stoc\_id),

    FOREIGN KEY(produs\_id) REFERENCES Produs(produs\_id)

);

CREATE TABLE Comanda (

    comanda\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    client\_id NUMBER,

    data\_comanda DATE NOT NULL,

    data\_scadenta DATE NOT NULL,

    pret\_total NUMBER NOT NULL,

    mod\_plata VARCHAR(25) NOT NULL,

    PRIMARY KEY (comanda\_id),

    FOREIGN KEY(client\_id) REFERENCES Client(client\_id)

);

CREATE TABLE Detalii\_Comanda (

    detalii\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    comanda\_id NUMBER,

    produs\_id NUMBER,

    cantitate NUMBER NOT NULL,

    pret\_unitar DECIMAL(10, 2) NOT NULL,

    pret\_fara\_tva DECIMAL(10, 2) NOT NULL,

    pret\_cu\_tva DECIMAL(10, 2) NOT NULL,

    tva DECIMAL(5, 2) NOT NULL CHECK (tva >= 0 AND tva <= 100),

    cod\_promotional VARCHAR(15),

    PRIMARY KEY (detalii\_id),

    FOREIGN KEY(comanda\_id) REFERENCES Comanda(comanda\_id),

    FOREIGN KEY(produs\_id) REFERENCES Produs(produs\_id)

);

CREATE TABLE Aprovizionare (

    aprovizionare\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    produs\_id NUMBER,

    furnizor\_id NUMBER,

    cantitate NUMBER NOT NULL,

    data\_aprovizionare DATE NOT NULL,

    PRIMARY KEY (aprovizionare\_id),

    FOREIGN KEY(produs\_id) REFERENCES Produs(produs\_id),

    FOREIGN KEY(furnizor\_id) REFERENCES Furnizor(furnizor\_id)

);

CREATE TABLE Admin (

    admin\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    nume\_admin VARCHAR(15) NOT NULL,

    prenume\_admin VARCHAR(15) NOT NULL,

    PRIMARY KEY (admin\_id)

);

CREATE TABLE Tichet (

    tichet\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    client\_id NUMBER,

    status\_tichet VARCHAR(25) NOT NULL,

    severitate VARCHAR(15) NOT NULL,

    data\_creare\_tichet DATE NOT NULL,

    data\_inchidere\_tichet DATE,

    PRIMARY KEY (tichet\_id),

    FOREIGN KEY(client\_id) REFERENCES Client(client\_id)

);

CREATE TABLE Problema (

    tichet\_id NUMBER,

    admin\_id NUMBER,

    descriere\_problema VARCHAR(200) NOT NULL,

    categorie\_problema VARCHAR(25) NOT NULL,

    FOREIGN KEY(tichet\_id) REFERENCES Tichet(tichet\_id),

    FOREIGN KEY(admin\_id) REFERENCES Admin(admin\_id)

);

CREATE TABLE Wishlist (

    wishlist\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

    produs\_id NUMBER,

    client\_id NUMBER,

    notificare VARCHAR(2) DEFAULT 'da',

    PRIMARY KEY (wishlist\_id),

    FOREIGN KEY(produs\_id) REFERENCES Produs(produs\_id),

    FOREIGN KEY(client\_id) REFERENCES Client(client\_id)

);

CREATE TABLE Recenzie (

    produs\_id NUMBER NOT NULL,

    client\_id NUMBER NOT NULL,

    tip\_recenzie VARCHAR(8),

    continut\_recenzie VARCHAR(200),

    produs\_cumparat VARCHAR(2) NOT NULL,

    aprecieri NUMBER,

    FOREIGN KEY(produs\_id) REFERENCES Produs(produs\_id),

    FOREIGN KEY(client\_id) REFERENCES Client(client\_id)

);

**INSEREAZĂ DATE ÎN TABELE**

SET SCAN OFF

INSERT INTO Client(nume\_client, prenume\_client, parola, email, telefon, adresa, cont\_premium)

SELECT \*

FROM

(

    SELECT 'Popescu', 'Ion', 'somepassword', 'ion.popescu@example.com', '0712345678', 'Str. Libertatii, nr. 1', 'nu' FROM DUAL UNION ALL

    SELECT 'Ionescu', 'Maria', 'chinchilla', 'maria.ionescu@example.com', '0723456789', 'Str. Independentei, nr. 10', 'nu' FROM DUAL UNION ALL

    SELECT 'Georgescu', 'Andrei', 'masterpassword', 'andrei.georgescu@example.com', '0734567890', 'Str. Victoriei, nr. 23', 'da' FROM DUAL UNION ALL

    SELECT 'Dumitrescu', 'Elena', 'P@ssw0rd!G5hVq#1K8j2Rz&7Tc', 'elena.dumitrescu@example.com', '0745678901', 'Str. Unirii, nr. 56', 'da' FROM DUAL UNION ALL

    SELECT 'Stanescu', 'Mihai', 'X9f#T@lM5$e@uY1W7Q8g\*4rI2v', 'mihai.stanescu@example.com', '0756789012', 'Str. Revolutiei, nr. 78', 'da' FROM DUAL UNION ALL

    SELECT 'Popa', 'Ana', 'b8@Q\*1N6yT#4vWpZ5$hJ2!rX7kE', 'ana.popa@example.com', '0767890123', 'Str. Eroilor, nr. 12', 'nu' FROM DUAL UNION ALL

    SELECT 'Tudor', 'Florin', 'Y\*8hT1@eG2$9Rz6wQ#5@v3LpI4U', 'florin.tudor@example.com', '0778901234', 'Str. Primaverii, nr. 34', 'nu' FROM DUAL UNION ALL

    SELECT 'Marin', 'Alina', 'm6@J#1R2\*8yG$7L5W!4fQz3P9Vk', 'alina.marin@example.com', '0789012345', 'Str. Sperantei, nr. 22', 'da' FROM DUAL UNION ALL

    SELECT 'Radu', 'George', 'r1$Z@4G8!k#3Q2L7V6\*9Tp5eWyU', 'george.radu@example.com', '0790123456', 'Str. Pacii, nr. 19', 'nu' FROM DUAL UNION ALL

    SELECT 'Iliescu', 'Simona', 'H5$3@Q1!Rz4v\*2Y6K7pX8T#9gLf', 'simona.iliescu@example.com', '0701234567', 'Str. Prieteniei, nr. 44', 'nu' FROM DUAL UNION ALL

    SELECT 'Vasilescu', 'Lucian', 'v8\*9Q4@2W1#T5$yR3!Lz6p7GmXk', 'lucian.vasilescu@example.com', '0712345679', 'Str. Pacea, nr. 11', 'nu' FROM DUAL UNION ALL

    SELECT 'Petrescu', 'Ioana', 'Q3$z@6R1!L4p\*8G7V5T#2y9kXWe', 'ioana.petrescu@example.com', '0723456790', 'Str. Libertatii, nr. 20', 'nu' FROM DUAL UNION ALL

    SELECT 'Munteanu', 'Stefan', 'Rz6\*4Q1@T5!9L2#3y7G$8vWkpXm', 'stefan.munteanu@example.com', '0734567901', 'Str. Florilor, nr. 33', 'nu' FROM DUAL UNION ALL

    SELECT 'Constantin', 'Laura', 'V5@8Rz4#3Q2!9G1y6L\*7T$pWkXm', 'laura.constantin@example.com', '0745679012', 'Str. Crinului, nr. 45', 'nu' FROM DUAL UNION ALL

    SELECT 'Dragomir', 'Catalin', '2Q5@T1#8v3Rz!4L\*7$9yG6pXWk', 'catalin.dragomir@example.com', '0756789123', 'Str. Bucegi, nr. 66', 'nu' FROM DUAL

);

INSERT INTO Produs (nume\_produs, descriere\_produs, pret\_produs, categorie\_produs, oferta)

SELECT \*

FROM (

    SELECT 'Laptop HP Spectre x360', 'Laptop ultrabook convertibil cu ecran tactil', 6500, 'Electronice', 'nu' FROM DUAL UNION ALL

    SELECT 'iPhone 14 Pro Max', 'Smartphone de ultimă generație cu cameră triplă', 7500, 'Electronice', 'da' FROM DUAL UNION ALL

    SELECT 'Samsung Galaxy Tab S8', 'Tableta performantă cu ecran AMOLED', 4200, 'Electronice', 'nu' FROM DUAL UNION ALL

    SELECT 'Sony WH-1000XM4', 'Căști wireless cu anulare activă a zgomotului', 1200, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'GoPro HERO10', 'Cameră de acțiune cu rezoluție 5K', 3000, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Kindle Paperwhite', 'E-book reader cu ecran iluminat și anti-reflexie', 600, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Apple Watch Series 7', 'Ceas inteligent cu funcții avansate de sănătate', 2200, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Roku Streaming Stick 4K', 'Dispozitiv de streaming media 4K', 400, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'DJI Mavic Air 2', 'Dronă pliabilă cu cameră 4K', 4800, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Logitech MX Master 3', 'Mouse wireless ergonomic pentru productivitate', 450, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Anker PowerCore 20000', 'Baterie externă portabilă cu capacitate mare', 300, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Samsung T7 Touch', 'SSD extern cu cititor de amprentă', 900, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Fitbit Charge 5', 'Brățară fitness cu monitorizare a ritmului cardiac', 600, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'JBL Charge 5', 'Boxă portabilă cu bluetooth și funcție powerbank', 700, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Oculus Quest 2', 'Set VR autonom cu controlere touch', 2500, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Laptop Lenovo X24', 'Laptop ultrabook convertibil cu ecran tactil', 6500, 'Electronice', 'nu' FROM DUAL UNION ALL

    SELECT 'Laptop Dell', 'Laptop ultrabook convertibil cu ecran tactil', 7500, 'Electronice', 'da' FROM DUAL

);

INSERT INTO Stoc (produs\_id, cantitate, data\_ultima\_aprovizionare)

SELECT \*

FROM (

    SELECT 1, 20, TO\_DATE('2024-03-01', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 2, 30, TO\_DATE('2024-03-02', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 3, 25, TO\_DATE('2024-03-03', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 4, 40, TO\_DATE('2024-03-04', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 5, 15, TO\_DATE('2024-03-05', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 6, 35, TO\_DATE('2024-03-06', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 7, 20, TO\_DATE('2024-03-07', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 8, 50, TO\_DATE('2024-03-08', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 9, 10, TO\_DATE('2024-03-09', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 10, 25, TO\_DATE('2024-03-10', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 11, 45, TO\_DATE('2024-03-11', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 12, 30, TO\_DATE('2024-03-12', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 13, 40, TO\_DATE('2024-03-13', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 14, 50, TO\_DATE('2024-03-14', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 15, 20, TO\_DATE('2024-03-15', 'YYYY-MM-DD') FROM DUAL

);

INSERT INTO Comanda (client\_id, data\_comanda, data\_scadenta, pret\_total, mod\_plata)

SELECT \*

FROM (

    SELECT 1, TO\_DATE('2022-04-01', 'YYYY-MM-DD'), TO\_DATE('2022-05-01', 'YYYY-MM-DD'), 6500, 'Visa' FROM DUAL UNION ALL

    SELECT 2, TO\_DATE('2022-04-02', 'YYYY-MM-DD'), TO\_DATE('2022-05-02', 'YYYY-MM-DD'), 7500, 'Mastercard' FROM DUAL UNION ALL

    SELECT 3, TO\_DATE('2023-04-03', 'YYYY-MM-DD'), TO\_DATE('2023-05-03', 'YYYY-MM-DD'), 4200, 'Cash' FROM DUAL UNION ALL

    SELECT 4, TO\_DATE('2024-04-04', 'YYYY-MM-DD'), TO\_DATE('2024-05-04', 'YYYY-MM-DD'), 1200, 'Cash' FROM DUAL UNION ALL

    SELECT 5, TO\_DATE('2024-04-05', 'YYYY-MM-DD'), TO\_DATE('2024-05-05', 'YYYY-MM-DD'), 3000, 'Cash' FROM DUAL UNION ALL

    SELECT 6, TO\_DATE('2024-04-06', 'YYYY-MM-DD'), TO\_DATE('2024-05-06', 'YYYY-MM-DD'), 600, 'Revolut' FROM DUAL UNION ALL

    SELECT 7, TO\_DATE('2024-04-07', 'YYYY-MM-DD'), TO\_DATE('2024-05-07', 'YYYY-MM-DD'), 2200, 'PayPal' FROM DUAL UNION ALL

    SELECT 8, TO\_DATE('2024-04-08', 'YYYY-MM-DD'), TO\_DATE('2024-05-08', 'YYYY-MM-DD'), 400, 'Cash' FROM DUAL UNION ALL

    SELECT 9, TO\_DATE('2024-04-09', 'YYYY-MM-DD'), TO\_DATE('2024-05-09', 'YYYY-MM-DD'), 4800, 'Visa' FROM DUAL UNION ALL

    SELECT 10, TO\_DATE('2024-04-10', 'YYYY-MM-DD'), TO\_DATE('2024-05-10', 'YYYY-MM-DD'), 450, 'Cash' FROM DUAL UNION ALL

    SELECT 11, TO\_DATE('2024-04-11', 'YYYY-MM-DD'), TO\_DATE('2024-05-11', 'YYYY-MM-DD'), 300, 'Visa' FROM DUAL UNION ALL

    SELECT 12, TO\_DATE('2024-04-12', 'YYYY-MM-DD'), TO\_DATE('2024-05-12', 'YYYY-MM-DD'), 900, 'Cash' FROM DUAL UNION ALL

    SELECT 13, TO\_DATE('2024-04-13', 'YYYY-MM-DD'), TO\_DATE('2024-05-13', 'YYYY-MM-DD'), 700, 'Mastercard' FROM DUAL UNION ALL

    SELECT 14, TO\_DATE('2024-04-14', 'YYYY-MM-DD'), TO\_DATE('2024-05-14', 'YYYY-MM-DD'), 2500, 'Cash' FROM DUAL UNION ALL

    SELECT 15, TO\_DATE('2024-04-15', 'YYYY-MM-DD'), TO\_DATE('2024-05-15', 'YYYY-MM-DD'), 2500, 'Visa' FROM DUAL UNION ALL

    SELECT 10, TO\_DATE('2024-04-16', 'YYYY-MM-DD'), TO\_DATE('2024-05-16', 'YYYY-MM-DD'), 2500, 'Cash' FROM DUAL UNION ALL

    SELECT 3, TO\_DATE('2023-04-17', 'YYYY-MM-DD'), TO\_DATE('2023-05-17', 'YYYY-MM-DD'), 7500, 'Cash' FROM DUAL UNION ALL

    SELECT 3, TO\_DATE('2023-04-17', 'YYYY-MM-DD'), TO\_DATE('2023-05-17', 'YYYY-MM-DD'), 7500, 'Cash' FROM DUAL

);

INSERT INTO Detalii\_Comanda (comanda\_id, produs\_id, cantitate, pret\_unitar, pret\_fara\_tva, pret\_cu\_tva, tva, cod\_promotional)

SELECT \*

FROM (

    SELECT 1 AS comanda\_id, 1 AS produs\_id, 1 AS cantitate, 6500 AS pret\_unitar, 5416.67 AS pret\_fara\_tva, 6500 AS pret\_cu\_tva, 20 AS tva, 'ELEC20OFF' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 1 AS comanda\_id, 4 AS produs\_id, 1 AS cantitate, 1200 AS pret\_unitar, 960 AS pret\_fara\_tva, 1200 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 1 AS comanda\_id, 2 AS produs\_id, 2 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 2 AS comanda\_id, 2 AS produs\_id, 1 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 3 AS comanda\_id, 3 AS produs\_id, 1 AS cantitate, 4200 AS pret\_unitar, 3500 AS pret\_fara\_tva, 4200 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 4 AS comanda\_id, 4 AS produs\_id, 1 AS cantitate, 1200 AS pret\_unitar, 1000 AS pret\_fara\_tva, 1200 AS pret\_cu\_tva, 20 AS tva, 'SMARTDEALS10' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 5 AS comanda\_id, 5 AS produs\_id, 1 AS cantitate, 3000 AS pret\_unitar, 2500 AS pret\_fara\_tva, 3000 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 6 AS comanda\_id, 6 AS produs\_id, 1 AS cantitate, 600 AS pret\_unitar, 500 AS pret\_fara\_tva, 600 AS pret\_cu\_tva, 20 AS tva, 'TECHSALE30' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 7 AS comanda\_id, 7 AS produs\_id, 1 AS cantitate, 2200 AS pret\_unitar, 1833.33 AS pret\_fara\_tva, 2200 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 8 AS comanda\_id, 8 AS produs\_id, 1 AS cantitate, 400 AS pret\_unitar, 333.33 AS pret\_fara\_tva, 400 AS pret\_cu\_tva, 20 AS tva, 'SMARTDEALS10' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 9 AS comanda\_id, 9 AS produs\_id, 1 AS cantitate, 4800 AS pret\_unitar, 4000 AS pret\_fara\_tva, 4800 AS pret\_cu\_tva, 20 AS tva, 'SMARTDEALS10' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 10 AS comanda\_id, 10 AS produs\_id, 1 AS cantitate, 450 AS pret\_unitar, 375 AS pret\_fara\_tva, 450 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 10 AS comanda\_id, 2 AS produs\_id, 1 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 11 AS comanda\_id, 11 AS produs\_id, 1 AS cantitate, 300 AS pret\_unitar, 250 AS pret\_fara\_tva, 300 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 12 AS comanda\_id, 12 AS produs\_id, 1 AS cantitate, 900 AS pret\_unitar, 750 AS pret\_fara\_tva, 900 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 13 AS comanda\_id, 13 AS produs\_id, 1 AS cantitate, 600 AS pret\_unitar, 500 AS pret\_fara\_tva, 600 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 14 AS comanda\_id, 14 AS produs\_id, 1 AS cantitate, 700 AS pret\_unitar, 583.33 AS pret\_fara\_tva, 700 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 15 AS comanda\_id, 15 AS produs\_id, 1 AS cantitate, 2500 AS pret\_unitar, 2083.33 AS pret\_fara\_tva, 2500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 16 AS comanda\_id, 10 AS produs\_id, 1 AS cantitate, 2500 AS pret\_unitar, 2083.33 AS pret\_fara\_tva, 2500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 17 AS comanda\_id, 1 AS produs\_id, 1 AS cantitate, 6500 AS pret\_unitar, 5416.67 AS pret\_fara\_tva, 6500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 18 AS comanda\_id, 2 AS produs\_id, 1 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, '' AS cod\_promotional FROM DUAL

);

INSERT INTO Furnizor (nume\_furnizor, adresa\_furnizor, email\_furnizor, fax\_furnizor, produs\_furnizat)

SELECT \*

FROM (

    SELECT 'TechDistribution', 'Str. Tehnologiei, nr. 12', 'contact@techdistribution.com', 0211234567, 'Laptopuri' FROM DUAL UNION ALL

    SELECT 'MobileWorld', 'Str. Mobilei, nr. 8', 'sales@mobileworld.com', 0212345678, 'Smartphone-uri' FROM DUAL UNION ALL

    SELECT 'TabletHub', 'Str. Tabletelor, nr. 22', 'info@tablethub.com', 0213456789, 'Tablete' FROM DUAL UNION ALL

    SELECT 'AudioMasters', 'Str. Audio, nr. 34', 'support@audiomasters.com', 0214567890, 'Căști' FROM DUAL UNION ALL

    SELECT 'CameraExperts', 'Str. Camerei, nr. 18', 'contact@cameraexperts.com', 0215678901, 'Camere de acțiune' FROM DUAL UNION ALL

    SELECT 'EbookReaders', 'Str. Ebook, nr. 45', 'info@ebookreaders.com', 0216789012, 'E-book readere' FROM DUAL UNION ALL

    SELECT 'SmartWatches', 'Str. Ceasului, nr. 3', 'sales@smartwatches.com', 0217890123, 'Ceasuri inteligente' FROM DUAL UNION ALL

    SELECT 'StreamingDevices', 'Str. Streaming, nr. 14', 'support@streamingdevices.com', 0218901234, 'Dispozitive de streaming' FROM DUAL UNION ALL

    SELECT 'DroneZone', 'Str. Dronelor, nr. 27', 'info@dronezone.com', 0219012345, 'Drone' FROM DUAL UNION ALL

    SELECT 'AccessoryCenter', 'Str. Accesoriilor, nr. 9', 'contact@accessorycenter.com', 0210123456, 'Mouse-uri' FROM DUAL UNION ALL

    SELECT 'PowerSolutions', 'Str. Puterii, nr. 5', 'support@powersolutions.com', 0211234568, 'Baterii externe' FROM DUAL UNION ALL

    SELECT 'StoragePro', 'Str. Stocării, nr. 11', 'sales@storagepro.com', 0212345679, 'SSD-uri' FROM DUAL UNION ALL

    SELECT 'FitnessGear', 'Str. Fitness, nr. 7', 'info@fitnessgear.com', 0213456780, 'Brățări fitness' FROM DUAL UNION ALL

    SELECT 'SoundWave', 'Str. Sunetului, nr. 6', 'contact@soundwave.com', 0214567891, 'Boxe portabile' FROM DUAL UNION ALL

    SELECT 'VRInnovations', 'Str. Virtuală, nr. 10', 'sales@vrinnovations.com', 0215678902, 'Seturi VR' FROM DUAL

);

INSERT INTO Aprovizionare (produs\_id, furnizor\_id, cantitate, data\_aprovizionare)

SELECT \*

FROM (

    SELECT 1 AS produs\_id, 1 AS furnizor\_id, 20 AS cantitate, TO\_DATE('2024-03-01', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 2 AS produs\_id, 2 AS furnizor\_id, 30 AS cantitate, TO\_DATE('2024-03-02', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 3 AS produs\_id, 3 AS furnizor\_id, 25 AS cantitate, TO\_DATE('2024-03-03', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 4 AS produs\_id, 4 AS furnizor\_id, 40 AS cantitate, TO\_DATE('2024-03-04', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 5 AS produs\_id, 5 AS furnizor\_id, 15 AS cantitate, TO\_DATE('2024-03-05', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 6 AS produs\_id, 6 AS furnizor\_id, 35 AS cantitate, TO\_DATE('2024-03-06', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 7 AS produs\_id, 7 AS furnizor\_id, 20 AS cantitate, TO\_DATE('2024-03-07', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 8 AS produs\_id, 8 AS furnizor\_id, 50 AS cantitate, TO\_DATE('2024-03-08', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 9 AS produs\_id, 9 AS furnizor\_id, 10 AS cantitate, TO\_DATE('2024-03-09', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 10 AS produs\_id, 10 AS furnizor\_id, 25 AS cantitate, TO\_DATE('2024-03-10', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 11 AS produs\_id, 11 AS furnizor\_id, 45 AS cantitate, TO\_DATE('2024-03-11', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 12 AS produs\_id, 12 AS furnizor\_id, 30 AS cantitate, TO\_DATE('2024-03-12', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 13 AS produs\_id, 13 AS furnizor\_id, 40 AS cantitate, TO\_DATE('2022-03-13', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 14 AS produs\_id, 14 AS furnizor\_id, 50 AS cantitate, TO\_DATE('2024-03-14', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 15 AS produs\_id, 15 AS furnizor\_id, 20 AS cantitate, TO\_DATE('2021-03-15', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL

);

INSERT INTO Tichet(client\_id, status\_tichet, severitate, data\_creare\_tichet, data\_inchidere\_tichet)

SELECT \*

FROM

(

    SELECT 1 AS client\_id, 'inchis' AS status\_tichet, 'critic' AS severitate, TO\_DATE('2024-02-01', 'YYYY-MM-DD') AS data\_creare\_tichet, TO\_DATE('2024-02-01', 'YYYY-MM-DD') AS data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 5 AS client\_id, 'inchis' AS status\_tichet, 'minor' AS severitate, TO\_DATE('2024-03-05', 'YYYY-MM-DD') AS data\_creare\_tichet, TO\_DATE('2024-03-05', 'YYYY-MM-DD') AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 7 AS client\_id, 'inchis' AS status\_tichet, 'major' AS severitate, TO\_DATE('2024-03-06', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 8 AS client\_id, 'in progres' AS status\_tichet, 'minor' AS severitate, TO\_DATE('2024-03-11', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 10 AS client\_id, 'in progres' AS status\_tichet, 'major' AS severitate, TO\_DATE('2024-03-12', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 11 AS client\_id, 'inchis' AS status\_tichet, 'critic' AS severitate, TO\_DATE('2024-03-13', 'YYYY-MM-DD') AS data\_creare\_tichet, TO\_DATE('2024-02-13', 'YYYY-MM-DD') AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 5 AS client\_id, 'deschis' AS status\_tichet, 'major' AS severitate, TO\_DATE('2024-03-13', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 5 AS client\_id, 'deschis' AS status\_tichet, 'major' AS severitate, TO\_DATE('2024-03-22', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 9 AS client\_id, 'deschis' AS status\_tichet, 'minor' AS severitate, TO\_DATE('2024-04-02', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL UNION ALL

    SELECT 12 AS client\_id, 'deschis' AS status\_tichet, 'major' AS severitate, TO\_DATE('2024-04-05', 'YYYY-MM-DD') AS data\_creare\_tichet, null AS  data\_inchidere\_tichet FROM DUAL

);

INSERT INTO Problema(tichet\_id, admin\_id, descriere\_problema, categorie\_problema)

SELECT \*

FROM

(

    SELECT '1', '5', 'Site-ul este complet indisponibil', 'functionalitate' FROM DUAL UNION ALL

    SELECT '2', '4', 'Link-ul catre pagina de ‘Politica de confidentialitate’ nu functioneaza', 'functionalitate'  FROM DUAL UNION ALL

    SELECT '3', '2', 'Site-ul se încarca foarte lent, afectand experienta utilizatorilor.', 'performanta' FROM DUAL UNION ALL

    SELECT '4', '2', 'Butonul de ‘Inregistrare’ nu este aliniat corect pe pagină.', 'UI' FROM DUAL UNION ALL

    SELECT '5', '1', 'Textul din antetul paginii este putin pixelat pe dispozitive mobile.', 'UI' FROM DUAL UNION ALL

    SELECT '8', '3', 'Link-ul către pagina de ‘Termeni si conditii’ este greu de găsit.', 'navigare' FROM DUAL UNION ALL

    SELECT '8', '4', 'Formularul de abonare la newsletter nu afiseaza mesajul de confirmare.', 'functionalitate' FROM DUAL UNION ALL

    SELECT '6', '2', 'Site-ul este redirecționat către un site phishing.', 'securitate' FROM DUAL UNION ALL

    SELECT '7', '4', 'Site-ul afisează o eroare 404 pentru anumite URL-uri.', 'navigare' FROM DUAL UNION ALL

    SELECT '9', '1', 'Imaginea de fundal nu se ajustează corect pe ecranele mai mici.', 'UI' FROM DUAL UNION ALL

    SELECT '10', '5', 'Butoanele de adăugare în coș nu funcționează pe Safari.', 'UI' FROM DUAL

);

INSERT INTO Recenzie(produs\_id, client\_id, tip\_recenzie, continut\_recenzie, produs\_cumparat, aprecieri)

SELECT \*

FROM

(

    SELECT '4', '5', 'negativ', 'Confortul poate fi îmbunătătit, mai ales pentru utilizarea pe termen lung.', 'nu', '2' FROM DUAL UNION ALL

    SELECT '5', '4', 'pozitiv', 'Rezolutia 5K este uimitoare, iar calitatea imaginilor este excelenta.', 'nu', '5' FROM DUAL UNION ALL

    SELECT '7', '2', null, 'Pretul este ridicat, iar autonomia bateriei ar putea fi mai buna.', 'nu', '25' FROM DUAL UNION ALL

    SELECT '6', '2', null, 'Ecranul iluminat și anti-reflexie face cititul confortabil în orice conditii de lumina.', 'nu', '1' FROM DUAL UNION ALL

    SELECT '9', '1', null,  'Experiența VR este captivantă, iar controlerele touch sunt precise.', 'nu', '10' FROM DUAL UNION ALL

    SELECT '7', '7', 'pozitiv', 'Funcțiile avansate de sănătate și designul elegant fac din Apple Watch Series 7 un companion excelent.', 'da', '46' FROM DUAL UNION ALL

    SELECT '8', '4', null, 'Experienta VR este captivanta, iar controlerele touch sunt precise.', 'nu', '3' FROM DUAL UNION ALL

    SELECT '8', '8', 'negativ', 'Uneori poate provoca disconfort la nivelul ochilor.', 'da', '24' FROM DUAL UNION ALL

    SELECT '3', '3', 'pozitiv', 'Tableta Samsung Galaxy Tab S8 are un ecran AMOLED vibrant și performanta rapida. Este ideală pentru consumul de continut.', 'da', '33' FROM DUAL UNION ALL

    SELECT '4', '1', 'pozitiv', 'Anularea activă a zgomotului este excelenta, iar sunetul este clar și bogat.', 'da', '100' FROM DUAL UNION ALL

    SELECT '1', '1', 'pozitiv', 'Laptopul HP Spectre x360 este o alegere excelenta pentru cei care doresc un dispozitiv versatil. Ecranul tactil este foarte receptiv, iar performanța este excelenta', 'da', '20' FROM DUAL

);

INSERT INTO Wishlist(client\_id, produs\_id, notificare)

SELECT \*

FROM

(

    SELECT 1 AS client\_id, 1 AS produs\_id, 'da' AS notificare FROM DUAL UNION ALL

    SELECT 2 AS client\_id, 2 AS produs\_id, 'da' AS notificare FROM DUAL UNION ALL

    SELECT 3 AS client\_id, 3 AS produs\_id, 'nu' AS notificare FROM DUAL UNION ALL

    SELECT 4 AS client\_id, 4 AS produs\_id, 'da' AS notificare FROM DUAL UNION ALL

    SELECT 5 AS client\_id, 5 AS produs\_id, 'nu' AS notificare FROM DUAL UNION ALL

    SELECT 6 AS client\_id, 6 AS produs\_id, 'nu' AS notificare FROM DUAL UNION ALL

    SELECT 7 AS client\_id, 7 AS produs\_id, 'nu' AS notificare FROM DUAL UNION ALL

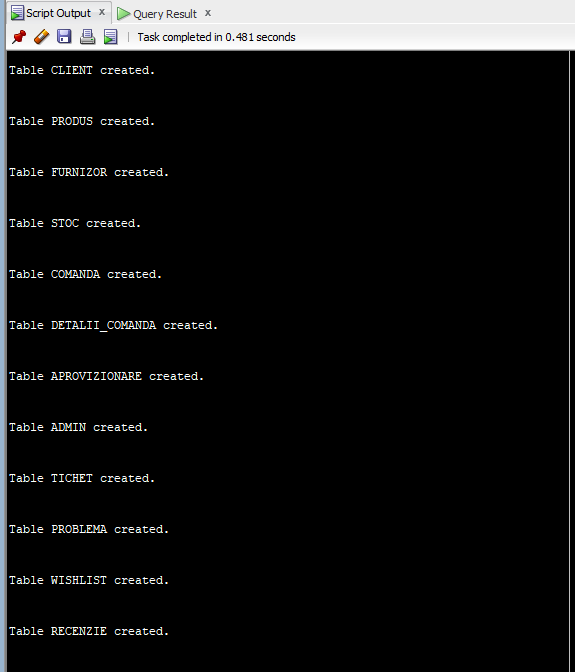
    SELECT 8 AS client\_id, 8 AS produs\_id, 'da' AS notificare FROM DUAL UNION ALL

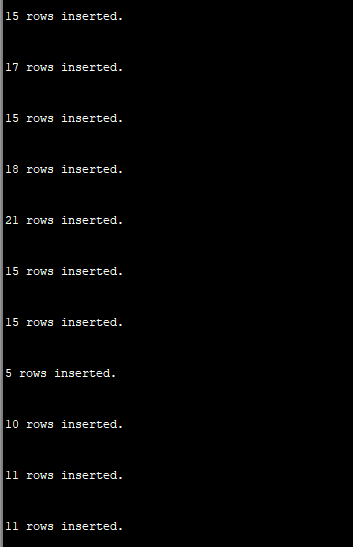
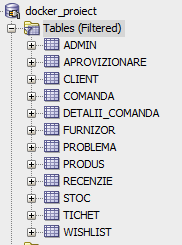
    SELECT 9 AS client\_id, 9 AS produs\_id, 'nu' AS notificare FROM DUAL UNION ALL

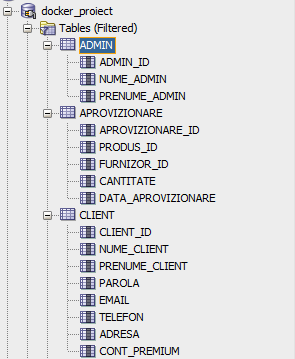
    SELECT 10 AS client\_id, 10 AS produs\_id, 'nu' AS notificare FROM DUAL

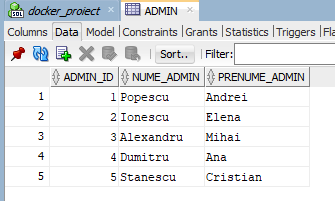
);

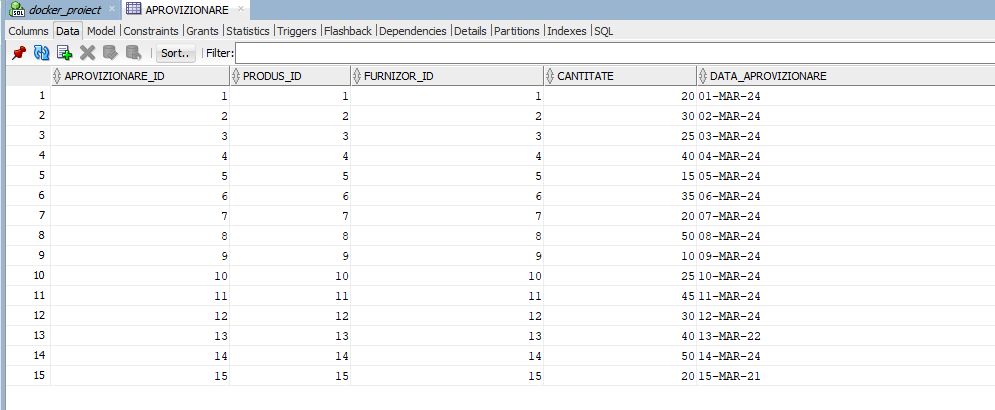
Rezultate după rularea scriptului:

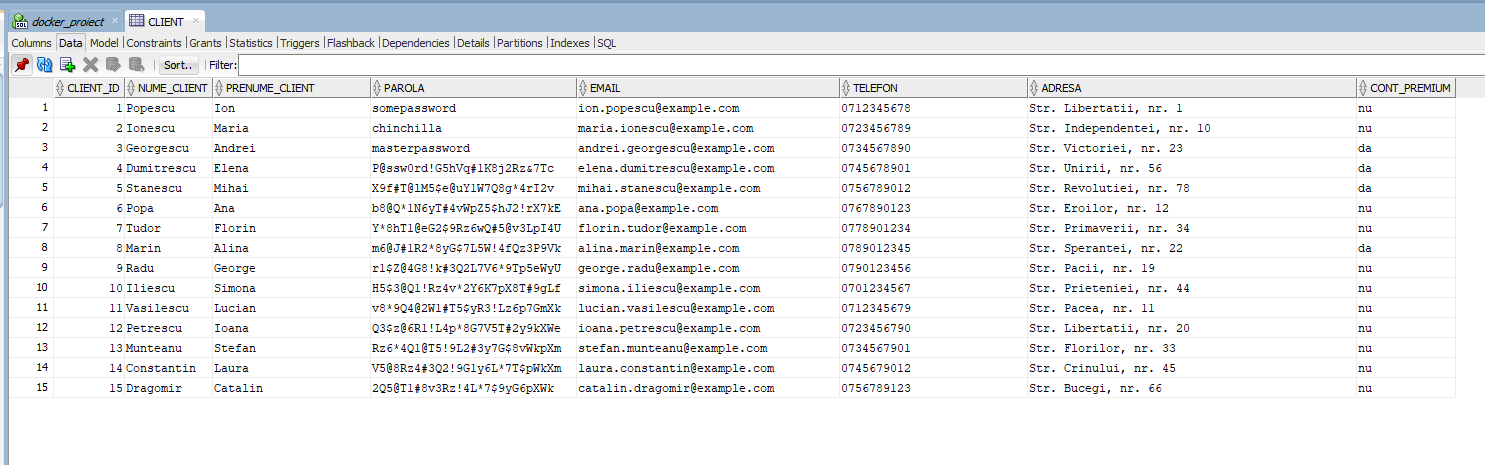


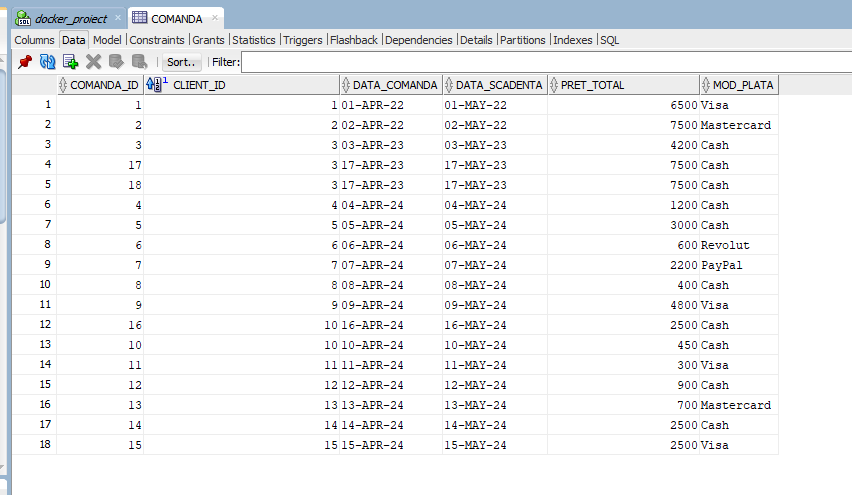
  


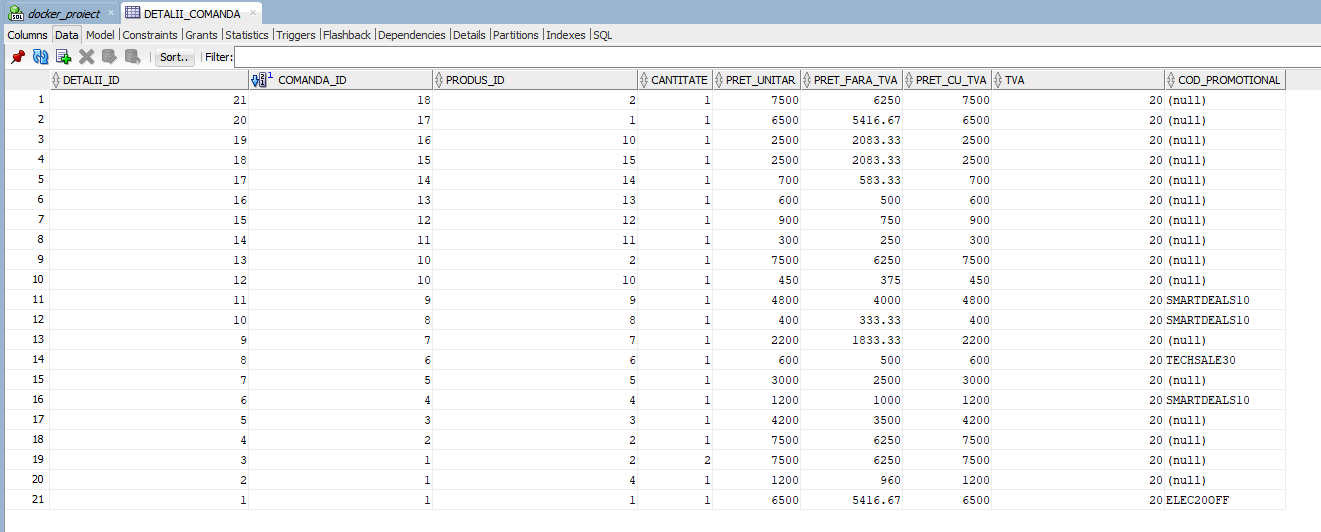


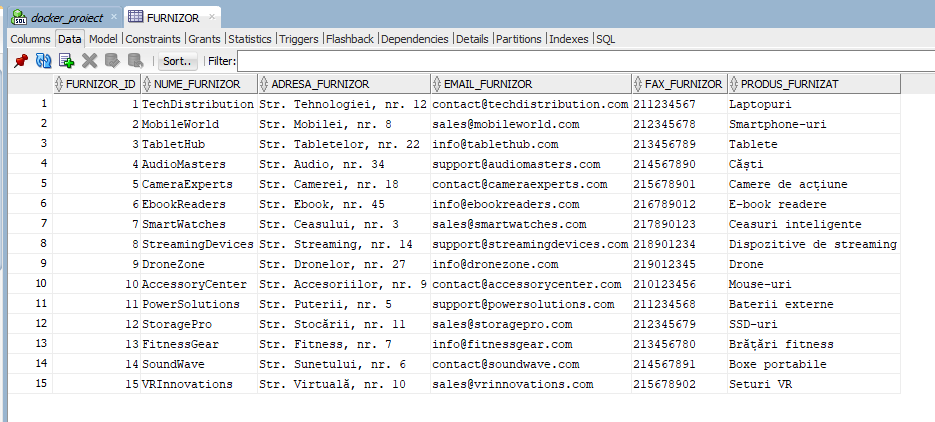


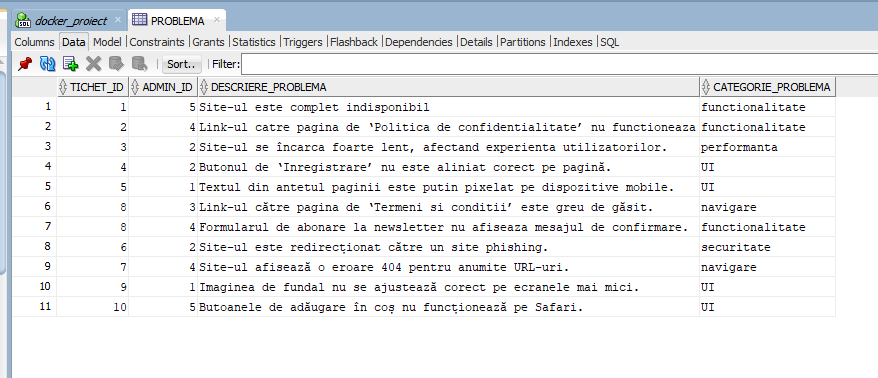


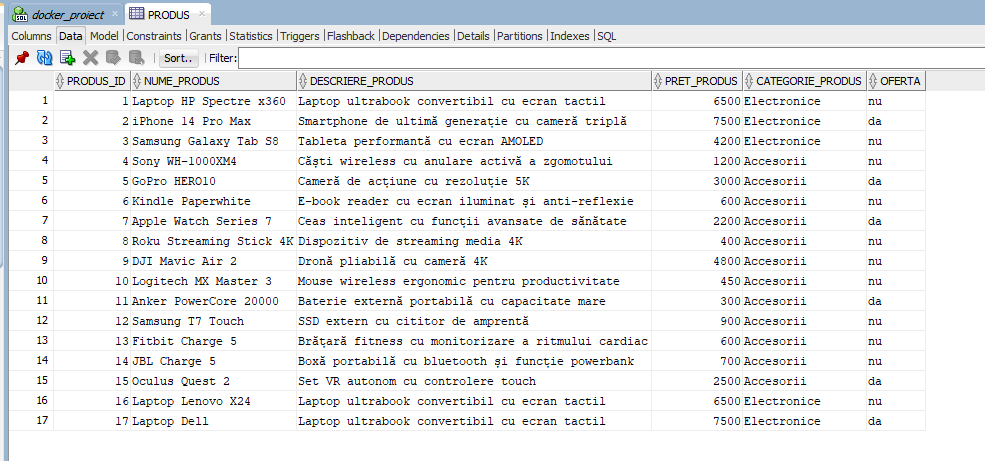


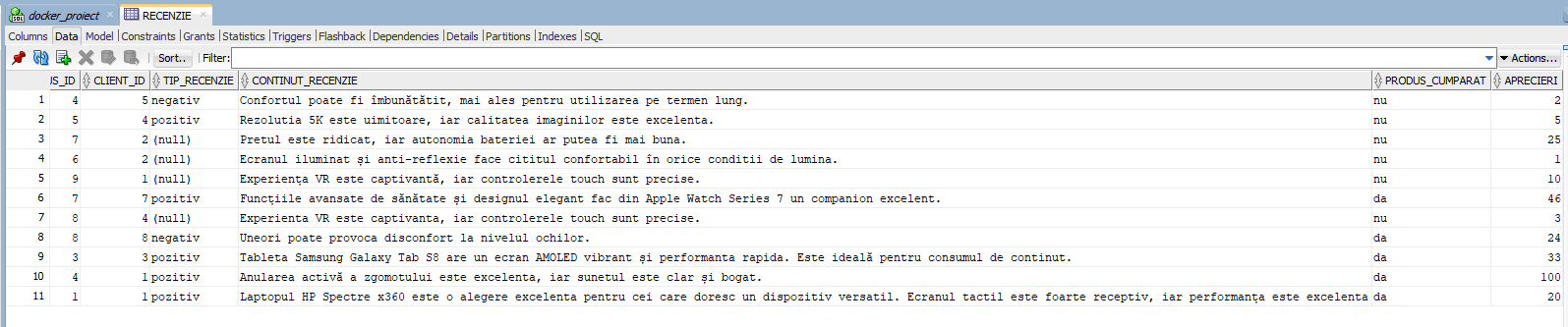


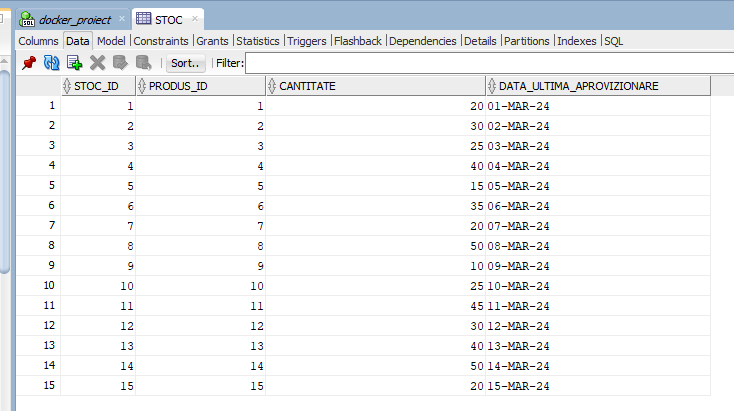


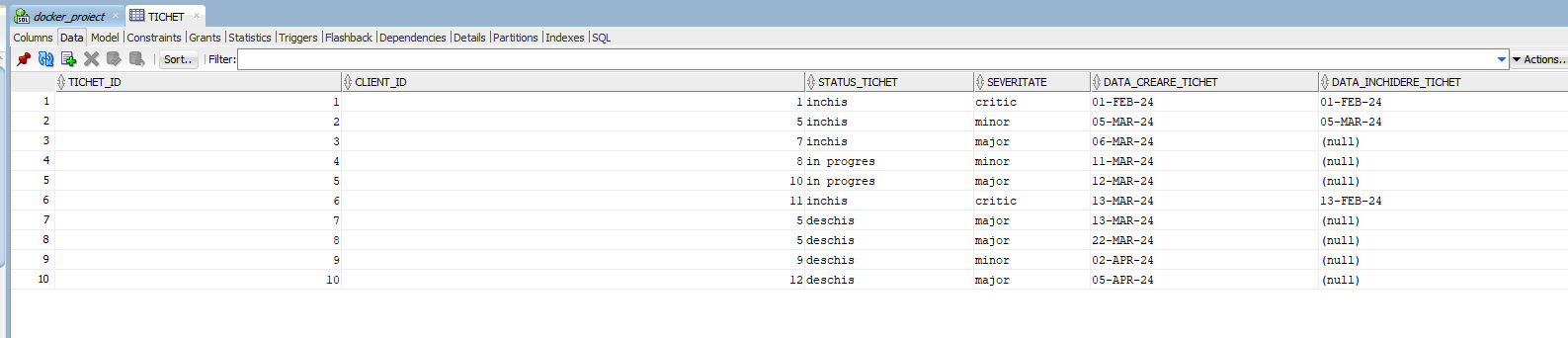


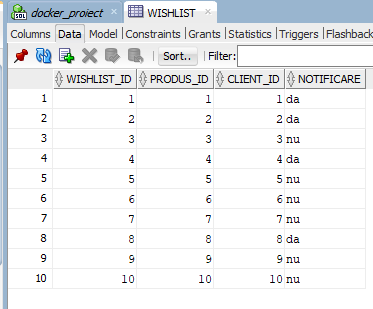












### Etapa 12

**Formulaţi în limbaj natural şi implementaţi 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

* 1. subcereri sincronizate în care intervin cel puțin 3 tabele

Afișează numele clienților și numărul total de produse comandate de fiecare dintre ei, filtrând clienții care au comandat produse din cel puțin 2 categorii diferite.

SELECT cl.nume\_client, cl.prenume\_client, COUNT(dc.produs\_id) AS numar\_produse

FROM Client cl

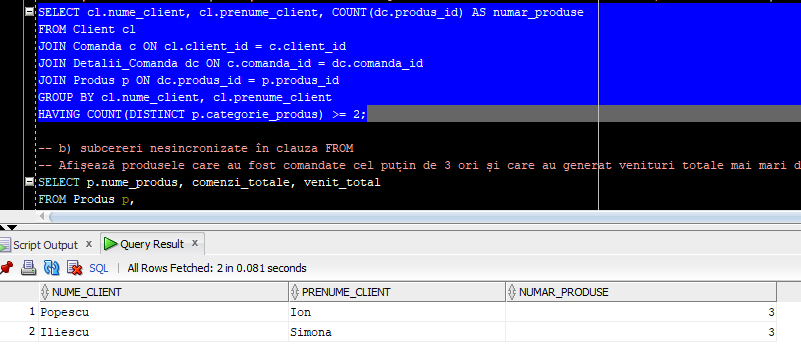
JOIN Comanda c ON cl.client\_id = c.client\_id

JOIN Detalii\_Comanda dc ON c.comanda\_id = dc.comanda\_id

JOIN Produs p ON dc.produs\_id = p.produs\_id

GROUP BY cl.nume\_client, cl.prenume\_client

HAVING COUNT(DISTINCT p.categorie\_produs) >= 2;



* 1. subcereri nesincronizate în clauza FROM

Afișează produsele care au fost comandate cel puțin de 3 ori și care au generat venituri totale mai mari de 10000.

SELECT p.nume\_produs, comenzi\_totale, venit\_total

FROM Produs p,

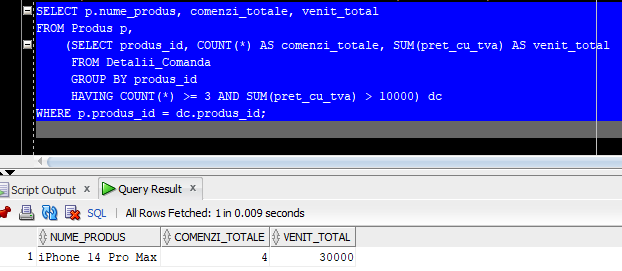
    (SELECT produs\_id, COUNT(\*) AS comenzi\_totale, SUM(pret\_cu\_tva) AS venit\_total

     FROM Detalii\_Comanda

     GROUP BY produs\_id

     HAVING COUNT(\*) >= 3 AND SUM(pret\_cu\_tva) > 10000) dc

WHERE p.produs\_id = dc.produs\_id;



* 1. grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)

Afișează categoriile de produse care au generat venituri totale mai mari de 7000, împreună cu numele clienților care au comandat din aceste categorii, și zilele de la ultima comandă pentru fiecare client.

SELECT p.categorie\_produs, cl.nume\_client, SUM(dc.pret\_cu\_tva) AS venit\_total,

       TRUNC(SYSDATE - MAX(c.data\_comanda)) AS zile\_de\_la\_ultima\_comanda,

       CASE

           WHEN TRUNC(SYSDATE - MAX(c.data\_comanda)) <= 30 THEN 'Recent'

           ELSE 'Nu este recent'

       END AS statut\_comanda

FROM Produs p

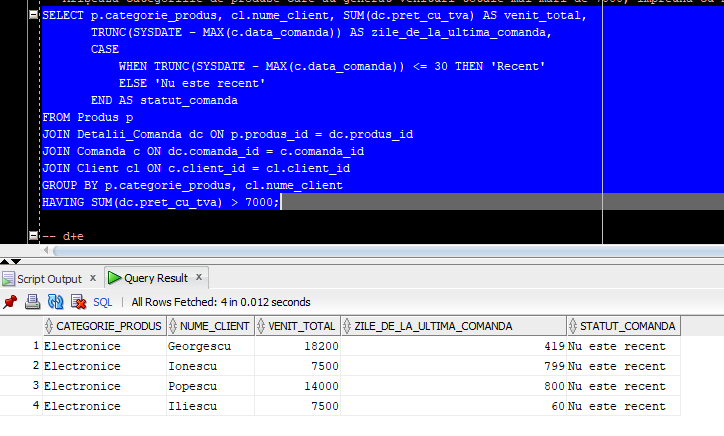
JOIN Detalii\_Comanda dc ON p.produs\_id = dc.produs\_id

JOIN Comanda c ON dc.comanda\_id = c.comanda\_id

JOIN Client cl ON c.client\_id = cl.client\_id

GROUP BY p.categorie\_produs, cl.nume\_client

HAVING SUM(dc.pret\_cu\_tva) > 7000;



* 1. ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

Afișează toate comenzile ordonate după data comenzii, folosește NVL pentru a înlocui valoarea null în câmpul cod\_promotional și DECODE pentru a traduce modurile de plată, și funcții pe șiruri de caractere pentru a formata numele clienților.

SELECT c.comanda\_id, c.data\_comanda, NVL(dc.cod\_promotional, 'FARA\_COD') AS cod\_promotional,

       DECODE(UPPER(c.mod\_plata), 'CASH', 'CASH', 'CARD') AS mod\_plata,

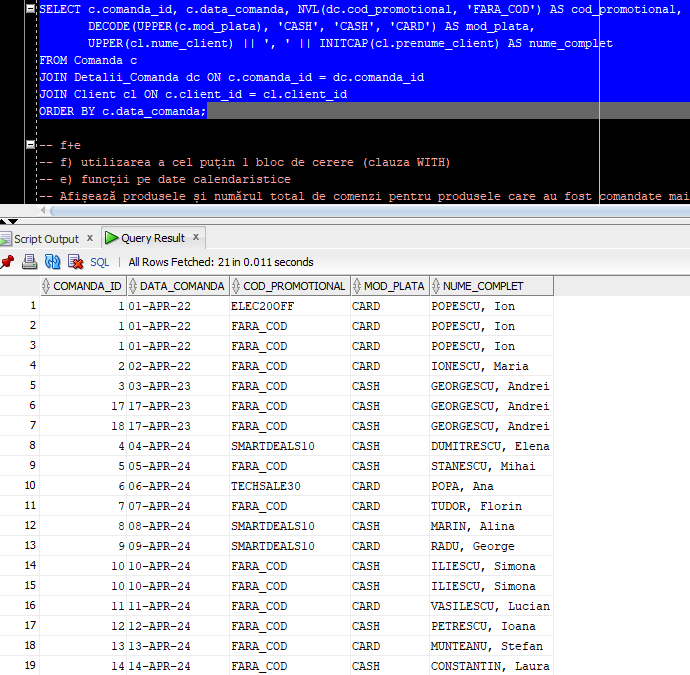
       UPPER(cl.nume\_client) || ', ' || INITCAP(cl.prenume\_client) AS nume\_complet

FROM Comanda c

JOIN Detalii\_Comanda dc ON c.comanda\_id = dc.comanda\_id

JOIN Client cl ON c.client\_id = cl.client\_id

ORDER BY c.data\_comanda;



* 1. utilizarea a cel puțin 2 funcții pe șiruri de caractere – în 4
  2. 2 funcții pe date calendaristice – în 3 și 8
  3. a cel puțin unei expresii CASE – în 3
  4. utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Afișează produsele și numărul total de comenzi pentru produsele care au fost comandate mai mult de 5 ori, folosind un bloc de cerere (WITH), și zilele trecute de la prima comandă pentru fiecare produs.

WITH Produse\_Comandate AS (

    SELECT produs\_id, COUNT(\*) AS numar\_comenzi, MIN(c.data\_comanda) AS prima\_comanda

    FROM Detalii\_Comanda dc

    JOIN Comanda c ON dc.comanda\_id = c.comanda\_id

    GROUP BY produs\_id

    HAVING COUNT(\*) > 2

)

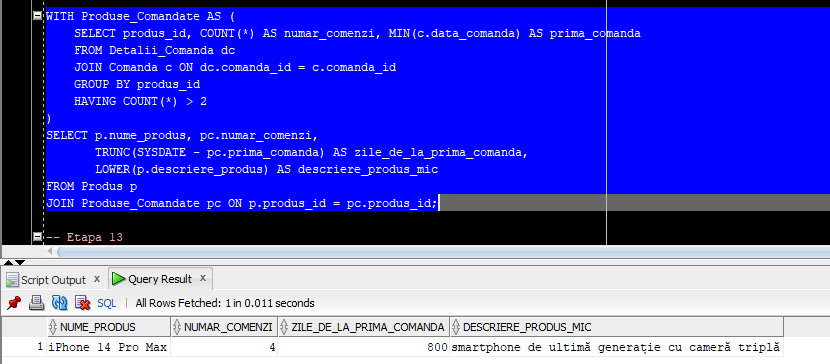
SELECT p.nume\_produs, pc.numar\_comenzi,

       TRUNC(SYSDATE - pc.prima\_comanda) AS zile\_de\_la\_prima\_comanda,

       LOWER(p.descriere\_produs) AS descriere\_produs\_mic

FROM Produs p

JOIN Produse\_Comandate pc ON p.produs\_id = pc.produs\_id;



Observaţie: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus astfel încât cele 5 cereri să cuprindă pe toate.

### Etapa 13

**Implementarea a 3 operaţii de actulizare şi de suprimare a datelor utilizând subcereri.**

**Operații ACTUALIZARE**

1) Actualizare preț produs cu cantitate < 20 în stoc ( marire cu 10% )

UPDATE Produs

SET pret\_produs = pret\_produs \* 1.1

WHERE produs\_id IN (SELECT produs\_id FROM Stoc WHERE cantitate < 20);

2) Actualizare cantitate produse din categoria ("Accesorii", din tabelul 'Stoc', marind-o cu 50)

UPDATE Stoc

SET cantitate = cantitate + 50

WHERE produs\_id IN (SELECT produs\_id FROM Produs WHERE categorie\_produs = 'Accesorii');

3) Actualizare cont\_premium client atunci când are 2 sau mai multe comenzi făcute

UPDATE Client

SET cont\_premium = 'da'

WHERE client\_id IN (

    SELECT client\_id

    FROM Comanda

    GROUP BY client\_id

    HAVING COUNT(comanda\_id) >= 2

);

**Operații SUPRIMARE**

1) Şterge comenzile care au fost plasate înainte de 1 ianuarie 2023

DELETE FROM Detalii\_Comanda

WHERE comanda\_id IN (

    SELECT comanda\_id

    FROM Comanda

    WHERE data\_comanda < TO\_DATE('2023-01-01', 'YYYY-MM-DD')

);

DELETE FROM Comanda

WHERE comanda\_id IN (SELECT comanda\_id FROM Comanda WHERE data\_comanda < TO\_DATE('2023-01-01', 'YYYY-MM-DD'));

2) Şterge produsele care nu sunt în stoc

DELETE FROM Produs

WHERE produs\_id NOT IN (SELECT produs\_id FROM Stoc);

3) Şterge furnizori inactivi

DELETE FROM Aprovizionare

WHERE furnizor\_id IN (

    SELECT furnizor\_id

    FROM Aprovizionare

    WHERE data\_aprovizionare < TO\_DATE('2023-01-01', 'YYYY-MM-DD')

);

DELETE FROM Furnizor

WHERE furnizor\_id NOT IN (SELECT DISTINCT furnizor\_id FROM Aprovizionare WHERE data\_aprovizionare > TO\_DATE('2023-01-01', 'YYYY-MM-DD'));

### Etapa 14

**Crearea unei vizualizări complexe. Daţi un exemplu de operaţie LMD permisă pe vizualizarea respectivă şi un exemplu de operaţie LMD nepermisă.**

**Crearea vizualizării**

CREATE VIEW Vizualizare\_Comenzi\_Clienti\_Produse AS

SELECT

    c.nume\_client,

    c.prenume\_client,

    com.data\_comanda,

    com.pret\_total,

    p.nume\_produs,

    dc.cantitate

FROM

    Client c

JOIN

    Comanda com ON c.client\_id = com.client\_id

JOIN

    Detalii\_Comanda dc ON com.comanda\_id = dc.comanda\_id

JOIN

    Produs p ON dc.produs\_id = p.produs\_id;

SELECT \* FROM Vizualizare\_Comenzi\_Clienti\_Produse

**Operație LMD permisă**

Aceasta actualizează cantitatea pentru o anumită comandă a clientului Iliescu pentru produsul iPhone 14 Pro Max.

UPDATE Vizualizare\_Comenzi\_Clienti\_Produse

SET cantitate = 5

WHERE nume\_client = 'Iliescu' AND nume\_produs = 'iPhone 14 Pro Max';

**Operație LMD nepermisă**

Inserarea de rânduri într-o vizualizare complexă care implică mai multe tabele nu este permisă, deoarece vizualizarea nu poate determina în ce tabele ar trebui inserate datele noi.

INSERT INTO Vizualizare\_Comenzi\_Clienti\_Produse (nume\_client, prenume\_client, data\_comanda, pret\_total, nume\_produs, cantitate)

VALUES ('Client', 'Nou', TO\_DATE('2024-06-01', 'YYYY-MM-DD'), 1000, 'Samsung Galaxy Tab S8', 2);

### Etapa 15

**Formulaţi în limbaj natural şi implementaţi în SQL: o cerere ce utilizează operaţia outer-join pe minim 4 tabele, o cerere ce utilizează operaţia division şi o cerere care implementează analiza top-n.**

**Observatie: Cele 3 cereri sunt diferite de cererile de la exerciţiul 12.**

**Operația outer-join**

Afișează toate comenzile și produsele comandate, inclusiv comenzile fără produse și produsele fără comenzi, împreună cu informațiile despre clienți și stocuri.

SELECT

    c.nume\_client,

    com.data\_comanda,

    p.nume\_produs,

    s.cantitate AS stoc\_cantitate

FROM

    Comanda com

FULL OUTER JOIN

    Client c ON com.client\_id = c.client\_id

FULL OUTER JOIN

    Detalii\_Comanda dc ON com.comanda\_id = dc.comanda\_id

FULL OUTER JOIN

    Produs p ON dc.produs\_id = p.produs\_id

FULL OUTER JOIN

    Stoc s ON p.produs\_id = s.produs\_id;

**Operația division**

Afișează clienții care au comandat toate produsele din categoria "Electronice".

SELECT nume\_client, prenume\_client

FROM Client c

WHERE NOT EXISTS (

    SELECT p.produs\_id

    FROM Produs p

    WHERE p.categorie\_produs = 'Electronice'

    MINUS

    SELECT dc.produs\_id

    FROM Comanda com

    JOIN Detalii\_Comanda dc ON com.comanda\_id = dc.comanda\_id

    WHERE com.client\_id = c.client\_id

);

**Operația top-n**

Afișează primele 5 produse cele mai comandate.

SELECT p.nume\_produs, SUM(dc.cantitate) AS total\_cantitate

FROM Produs p

JOIN Detalii\_Comanda dc ON p.produs\_id = dc.produs\_id

GROUP BY p.nume\_produs

ORDER BY total\_cantitate DESC

FETCH FIRST 5 ROWS ONLY;

### Etapa 16

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

Afișează numele și prenumele clienților care au plasat comenzi în 2024 și care au achiziționat produse din categoria "Accesorii":

SELECT c.nume\_client, c.prenume\_client

FROM Client c

JOIN Comanda com ON c.client\_id = com.client\_id

JOIN Detalii\_Comanda dc ON com.comanda\_id = dc.comanda\_id

JOIN Produs p ON dc.produs\_id = p.produs\_id

WHERE com.data\_comanda BETWEEN TO\_DATE('2024-01-01', 'YYYY-MM-DD') AND TO\_DATE('2024-12-31', 'YYYY-MM-DD')

AND p.categorie\_produs = 'Accesorii';

Optimizăm cererea prin utilizarea indecșilor și eliminarea join-urilor nenecesare.

SELECT DISTINCT c.nume\_client, c.prenume\_client

FROM Client c

WHERE EXISTS (

    SELECT 1

    FROM Comanda com

    JOIN Detalii\_Comanda dc ON com.comanda\_id = dc.comanda\_id

    JOIN Produs p ON dc.produs\_id = p.produs\_id

    WHERE com.client\_id = c.client\_id

    AND com.data\_comanda BETWEEN TO\_DATE('2024-01-01', 'YYYY-MM-DD') AND TO\_DATE('2024-12-31', 'YYYY-MM-DD')

    AND p.categorie\_produs = 'Accesorii'

);

### Etapa 17

**Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării, justificând necesitatea ac****esteia.**

Vom realiza verificarea progresivă de la BCNF -> FN4 -> FN5 şi apoi denormalizarea după caz

Vom lua ca exemplu tabelele Client şi Comanda

Normalizarea la **BCNF** - este deja în BCNF deoarece:

- Tabelele sunt deja în FN3

- Tabelul Client are cheie primară 'client\_id' şi nu există dependenţe tranzitive

- Tabelul comandă are cheie primară 'comanda\_id' şi 'clinet\_id' este cheie straină ( foreign key ).

Normalizarea la **FN4** - este deja în FN4 deoarece:

- Tabelele sunt deja în BCNF

- Nu există dependinţe multi-valued în aceste tabele

Normalizare la **FN5** - este deja FN5 deoarece:

- Tabelele sunt deja în FN4

- Nu există dependinţe join non-triviale

**Denormalizarea**

**Justificare:**

- Performanță îmbunătățită: Având toate datele într-un singur tabel, reducem numărul de join-uri necesare pentru interogările frecvente, ceea ce poate îmbunătăți semnificativ performanța.

- Complexitate redusă: Interogările devin mai simple și mai rapide de scris, ceea ce este benefic pentru dezvoltatori și administratori de baze de date.

- Cazuri specifice de utilizare: În aplicații OLAP ( Online Analytical Processing ) sau alte scenarii unde citirea rapidă este critică, denormalizarea poate fi esențială pentru a asigura răspunsuri rapide la interogări.

Crearea tabelului denormalizat ClientComanda

CREATE TABLE ClientComanda (

    client\_id INT,

    nume VARCHAR2(100),

    adresa VARCHAR2(200),

    cont\_premium VARCHAR2(3),

    comanda\_id INT,

    data\_comanda DATE,

    pret\_total NUMBER,

    PRIMARY KEY (client\_id, comanda\_id)

);

Inserarea datelor în tabelul denormalizat

INSERT INTO ClientComanda (client\_id, nume, adresa, cont\_premium, comanda\_id, data\_comanda, pret\_total)

SELECT c.client\_id, c.nume\_client, c.adresa, c.cont\_premium, com.comanda\_id, com.data\_comanda, com.pret\_total

FROM Client c

JOIN Comanda com ON c.client\_id = com.client\_id;

Exemplu de interogare pe tabelul denormalizat

Să presupunem că dorim să vedem toate comenzile și informațiile despre clienți pentru comenzile din luna aprilie 2024:

SELECT client\_id, nume, adresa, cont\_premium, comanda\_id, data\_comanda, pret\_total

FROM ClientComanda

WHERE data\_comanda BETWEEN TO\_DATE('2024-04-01', 'YYYY-MM-DD') AND TO\_DATE('2024-04-30', 'YYYY-MM-DD');

# Bibliografie

[1] Oracle Database documentation, URL: <https://docs.oracle.com/en/database/>

[2] Guru99, URL: <https://www.guru99.com/ro/>

[3] Geeksforgeeks, URL: <https://www.geeksforgeeks.org/oracle-architecture/>