

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

PROIECT BAZE DE DATE

PROFESOR COORDONATOR:
VASILE SILVIU-LAURENTIU

STUDENT:
IOJA DENIS

BUCUREȘTI
2023

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

SALĂ DE FITNESS

PROFESOR COORDONATOR:
VASILE SILVIU-LAURENTIU

STUDENT:
IOJA DENIS

BUCUREȘTI
2023

Cuprins

1. Prezentarea modelului
 - 1.1 Regulile modelului
2. Diagrama entitate-relatie
 - 2.1 Reprezentare
 - 2.2 Descrierea entitatilor, atributelor, cheilor și a cardinalitatilor
3. Diagrama conceptuală
 - 3.1 Reprezentare
 - 3.2 Descrierea entităților, atributelor, cheilor (tabelele asociative)
 - 3.3 Constrângerile de integritate
 - 3.4 Scheme relaționale
4. Implementarea în Oracle
 - 4.1 Tabeleși a constrângeri
 - 4.2 Date

1. Prezentarea modelului

În contextul zilelor noastre oamenii devin din ce în ce mai interesați să facă mișcare.

Un motiv pentru care se întâmplă asta este acela că încep să fie tot mai preocupați de sănătatea lor. Acest lucru este datorat atât stilului de viață sedentar cât și de tendința către obezitate.

Alt factor care explică acest fenomen al nevoii de mișcare este nevoia de validare socială care provine din modul în care arată, astfel că a fi cât mai în formă este ceva ce tanjesc mulți.

Din aceste motive au început să apară tot mai multe săli de fitness.

Prin proiectul meu îmi propun să vin în ajutorul proprietarului unei noi deschise care dorește să își gestioneze datele cu care lucrează, după cum urmează: clienții și abonamentele pe care le au, cât și produsele pe care le cumpără, angajații cu funcția lor și în funcție de aceasta vânzările pe care le procesează și abonamentele pe care le încheie sau clienții antrenați, produsele vândute cu categoria proprie și cu furnizorul aferent.

Precizări:

- O persoană este considerată client dacă deține un abonament sau dacă cumpără un produs.
- Un angajat poate să ocupe doar un singur post.
- Dacă un tip de abonament are câmpul "Data final" NOTNULL, atunci acesta a expirat.
- Datele despre clienți pot să lipsească dacă acesta doar cumpără produse.

1.1 Regulile modelului

1. Un client poate să nu aibă niciun tip de abonament (Acesta doar a cumpărat produse)

2. Fiecare client cu abonament are alocat câte un vestiar.

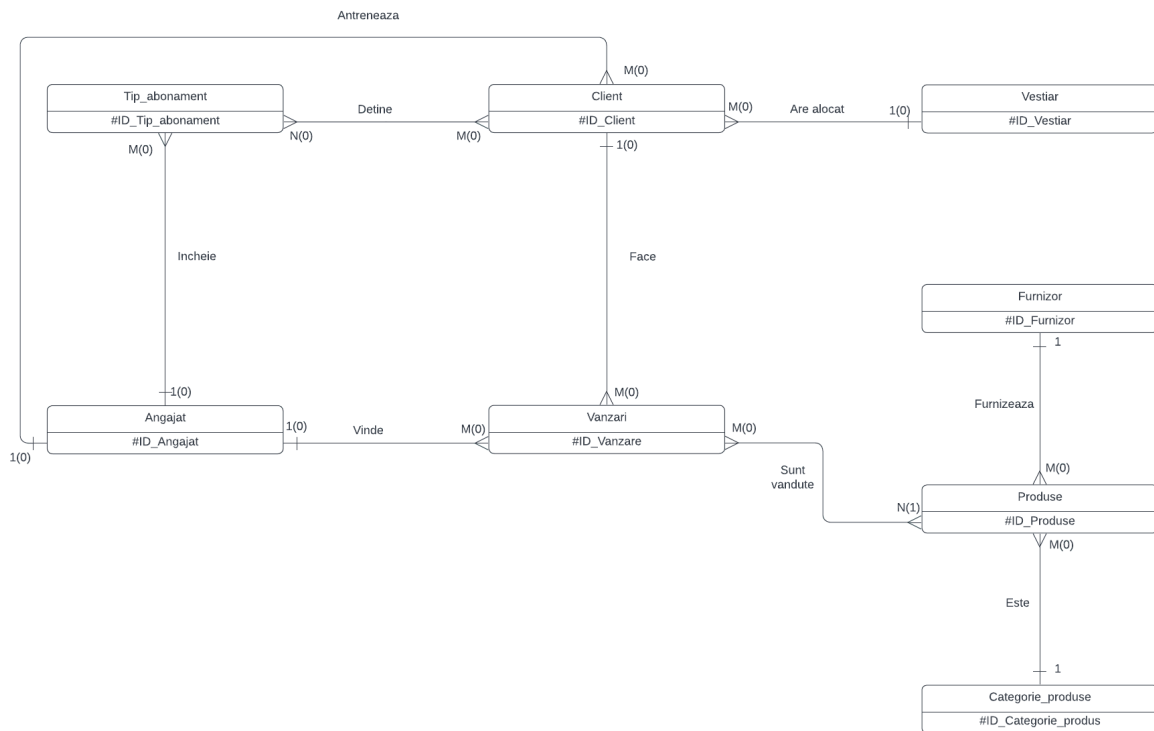
3. Un client poate să nu cumpere niciun produs. (Acesta deține un abonament)

4. Un client poate avea mai multe tipuri de abonamente, iar un tip de abonament poate să fie deținut de mai mulți clienți.
5. Un singur angajat încheie un abonament. (Abonamentul nu poate să fie încheiat de mai mulți angajați)
6. Un abonament poate să includă ambele tipuri de facilități. (Să includă și aerobic și saună)
7. Un singur angajat poate să antreneze un client. (Clientul nu poate să fie antrenat de mai mulți angajați)
8. Nu toți angajații antrenează clienți. (Aceștia pot să fie receptioneri sau îngrijitori)
9. Doar receptionerii încheie abonamente și vând produse.
10. Un produs poate face parte doar dintr-o categorie. (Nu poate face parte din mai multe categorii)
11. Pot exista produse care nu sunt cumparate de niciun client.
12. Pot să existe categorii de produse care să nu conțină niciun produs.
13. Un furnizor poate furniza mai multe produse sau niciunul.
14. Un produs poate să fie furnizat doar de un singur furnizor.
15. La o vânzare pot fi vândute mai multe produse, minim unul. (Nu există vânzare dacă nu se vinde minim un produs)
16. Un vestiar este alocat mai multor clienți. (În cazul în care un client își anulează abonamentul sau acesta expiră, vestiarul este alocat altui client)
17. Un tip de abonament poate să nu fie deținut de niciun client.

18. Un produs poate să nu fie cumparat de niciun client.

2. Diagramă entitate-relație

2.1 Reprezentare



2.2 Descrierea entităților, atributelor, cheilor și a cardinalităților

Descrierea entitatilor, atributelor, cheilor

- Entitatea Client: Stochează informații despre clientul sălii.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Client	ID-ul clientului	int
	Nume	Numele clientului	varchar(20)
	Prenume	Prenumele clientului	varchar(20)

	Varsta	Vârsta clientului	int
	Adresa	Adresa clientului	varchar(30)
FK	ID_Vestiar	ID-ul vestiarului care îi este alocat	int
FK	ID_Angajat	ID-ul angajatului care îl antrenează/îi încheie abonamentul sau îi vinde produse	int
	Data_nastere	Data nașterii clientului	date

- Entitatea Vestiar: Stochează informații despre vestiarul care este atribuit fiecărui client care are abonament.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Vestiar	ID-ul vestiarului	int
	Numar_Chei	Numărul de chei de la vestiar	int
	Dimensiune	Dimensiunea în cm ² a vestiarului	varchar(20)

- Entitatea Tip abonament: Stochează informații despre tipul de abonament pe care îl deține un client.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Tip_abonament	ID-ul tipului de abonament	int
	Denumire	Denumirea tipului de	varchar(30)

		abonament	
	Pret	Prețul tipului de abonament	number(7,2)
	Include_sauna	Indicator pentru a verifica dacă abonamentul include saună	char(2)
	Include_aerobic	Indicator pentru a verifica dacă abonamentul include aerobic	char(2)

- Entitatea Angajat: Stochează informații despre angajații sălii.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Angajat	ID-ul angajatului	int
	Nume	Numele angajatului	varchar(20)
	Prenume	Prenumele angajatului	varchar(20)
	Varsta	Vârsta angajatului	int
	Post	Postul pe care lucrează angajatul	varchar(20)
	Salariu	Salariul angajatului	number(7,2)
	Data_angajare	Data la care a fost angajat	date
	Vanzari	Numărul de produse vandute	int

	Nr_clienti_antrenati	Numărul de clienți antrenati	int
--	----------------------	------------------------------	-----

- Entitatea Vanzari: Stochează informații despre vânzarile efectuate la sală.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Vanzare	ID-ul asociat vanzarii	int
FK	ID_Client	ID-ul clientului căruia i-se efectuează vânzarea	int
	Data	Data la care se face vanzarea	date
FK	ID_Angajat	ID-ul angajatului care efectuează vânzarea	int

- Entitatea Produse: Stochează informații despre produsele vândute la sală.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Produs	ID-ul produsului	int
	Denumire_produs	Denumirea produsului	varchar(20)
	ID_Furnizor	ID-ul furnizorului produsului	int
	ID_Categorie	ID-ul categoriei din care face parte produsul	int
	Pret_vanzare	Prețul cu care se vinde produsul	number(7,2)

	Pret_cumparare	Prețul cu care se cumpără produsul de la furnizor	number(7,2)
	Stoc	Stocul din produsul respectiv	int
	Cantitate_vanduta	Cantitatea vânduta din produsul respectiv	int

- Entitate Furnizor: Stochează date despre furnizorul care aprovizionează sala cu produse.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Furnizor	ID-ul furnizorului	int
	Nume	Numele furnizorului	varchar(20)
	Adresa	Adresa furnizorului	varchar(30)

- Entitatea Categorie_produc: Stochează informații despre categoria din care face parte fiecare produs.

Cheie	Atribut	Descriere	Tip de data
PK	ID_Categorie_produc	ID-ul categoriei din care face parte un produs	int
	Nume	Numele categoriei din care face parte un produs	varchar(20)

Descrierea relațiilor și a cardinalitatilor:

1. Client-Tip_abonament

Aceasta este o relație de tip many-to-many astfel că este implementată prin intermediul diagramei “Abonamente” din diagrama conceptuală.

Cardinalitate minimă:

Un client poate deține 0 abonamente.

Un tip de abonament poate să fie deținut de 0 clienți.

Cardinalitate maximă:

Un client poate deține mai multe abonamente.

Un tip de abonament poate fi deținut de mai mulți clienți.

2. Client-Vestiar

Cardinalitate minimă:

Un client poate avea alocate 0 vestiare.

Un vestiar poate să fie alocat pentru 0 clienți.

Cardinalitate maximă:

Un client poate avea alocat un vestiar.

Un vestiar poate să fie alocat mai multor clienți.

3. Client-Angajat

Cardinalitate minimă:

Un client poate fi antrenat de 0 angajați.

Un angajat poate antrena 0 clienți.

Cardinalitate maximă:

Un client poate fi antrenat de un angajat.

Un angajat poate antrena mai mulți clienți.

4. Client-Vanzari

Cardinalitate minimă:

Unui client îi pot fi vândute 0 produse.

Un produs poate fi vândut pentru 0 client.

Cardinalitate maximă:

Unui client îi pot fi vândute mai multe produse.

Un produs poate fi vândut pentru un client.

5. Angajat-Vanzari

Cardinalitate minimă:

Un angajat poate să vândă 0 produse.

Un produs poate fi vândut de 0 angajați.

Cardinalitate maximă:

Un angajat poate să vândă mai multe produse.

Un produs poate fi vândut de un angajat.

6. Vanzari-Produse

Aceasta este o relație de tip many-to-many astfel ca este implementată prin intermediul diagramei “Produse_vandute” din diagrama conceptuală.

Cardinalitate minimă:

La o vânzare se poate vinde un produs.

Un produs poate să facă parte din 0 vânzări.

Cardinalitate maximă:

La o vânzare se pot vinde mai multe produse.

Un produs poate face parte din mai multe vânzări.

7. Produse-Furnizor

Cardinalitate minimă:

Un produs poate proveni de la un furnizor.

Un furnizor poate furniza 0 produse.

Cardinalitate maximă:

Un produs poate proveni de la un furnizor.

Un furnizor poate furniza mai multe produse.

8. Produse-Categorie_produs

Cardinalitate minimă:

Un produs poate face parte dintr-o categorie de produse.

O categorie de produse poate să aibă 0 produse.

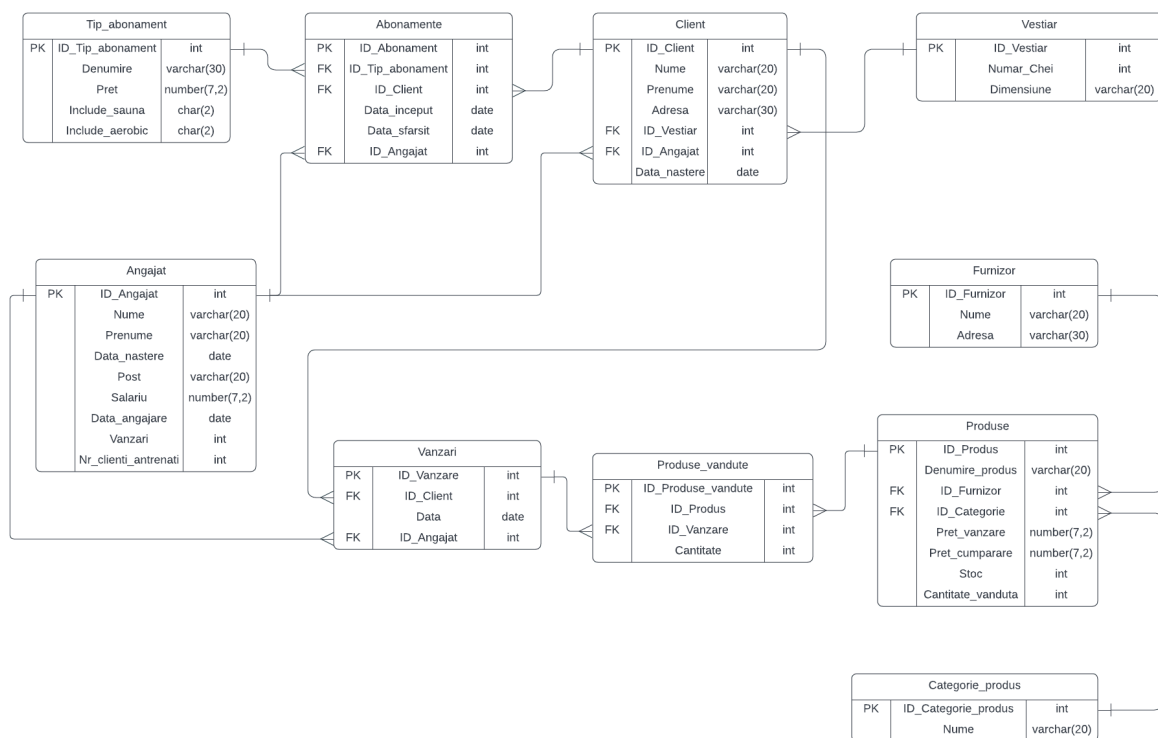
Cardinalitate maximă:

Un produs poate face parte dintr-o categorie de produse.

O categorie de produse poate să aibă mai multe produse.

3. Diagrama conceptuală

3.1 Reprezentare



3.2 Descrierea entităților, atributelor, cheilor (tabelele asociative)

- Entitatea Abonamente (Relație many-to-many între Client și Tip_Abonament)

Cheie	Nume atribut	Descriere	Tip de data
PK	ID_Abonament	ID-ul abonamentului	int
FK	ID_Tip_abonament	ID-ul tipului de abonament	int
FK	ID_Client	ID-ul clientului care își face abonament	int
	Data_inceput	Data la care începe abonamentul	date
	Data_sfarsit	Data la care se termină abonamentul	date
FK	ID_Angajat	ID-ul angajatului care încheie abonamentul	int

- Entitatea Produse_vandute (Relație many-to-many între Vanzari și Produse)

Cheie	Nume atribut	Descriere	Tip de data
PK	ID_Produse_vandute	ID-ul produsului vândut	int
FK	ID_Produs	ID-ul produsului	int
FK	ID_Vanazare	ID-ul vânzării	int
	Cantitate	Cantitatea de produse vândute o singura data	int

3.3 Constrângerile de integritate

- Tabelul Client

Constrângeri Primary Key:

- ☐ Client_ID_Client_PK - atributul ID_Client va fi PK

Constrângeri Foreign Key:

- ☐ Client_ID_Vestiar_FK - atributul ID_Vestiar va fi FK
(Face referire la atributul ID_Vestiar din tabelul Vestiar)
- ☐ Client_ID_Angajat_FK - atributul ID_Angajat va fi FK
(Face referire la atributul ID_Angajat din tabelul Angajat)

- Tabelul Abonamente - tabel asociativ

Constrângeri Primary Key:

- ☐ Abonamente_ID_Abonament_PK - atributul ID Abonament va fi PK

Constrângeri Foreign Key:

- ☐ Abonamente_ID_Tip_abonament_FK - atributul ID_Tip_abonament va fi FK
(Face referire la atributul ID_Tip_abonament din tabelul Tip_abonament)
- ☐ Abonamente_ID_Client_FK - atributul ID_Client va fi FK
(Face referire la atributul ID_Client din tabelul Client)
- ☐ Abonamete_ID_Angajat_FK - atributul ID_Angajat va fi FK
(Face referire la atributul ID_Angajat din tabelul Angajat)

Constrângeri NOT NULL:

- ☐ Abonamente_ID_Tip_abonament_NN - atributul ID_Tip_abonament va fi NN
(Orice abonament trebuie să fie de un tip)
- ☐ Abonamente_ID_Client_NN - atributul ID_Client va fi NN
(Orice abonament va avea un client)
- ☐ Abonamente_ID_Angajat_NN - atributul ID_Angajat va fi NN
(Orice abonament este încheiat de un client)
- ☐ Abonamente_Data_inceput_NN - atributul Data_inceput va fi NN
(Orice abonament are o dată de început)
- ☐ Abonamente_Data_sfarsit_NN - atributul Data_sfarsit va fi NN
(Orice abonament are o dată de sfârșit)

Constrângeri Default:

- ☐ Data_inceput - atributul Data_inceput va fi inițializat by default cu valoarea TRUNC(SYSDATE) în cazul în care nu se introduc date

● Tabelul Tip_abonament

Constrangeri Primary Key:

- ☐ Tip_abonament_ID_Tip_abonament_PK - atributul ID_Tip_abonament va fi PK

Constrângeri NN:

- ☐ Tip_abonament_Denumire_NN - atributul Denumire va fi NN
(Orice tip de abonament trebuie să aibă o denumire)
- ☐ Tip_abonament_Pret_NN - atributul Pret va fi NN
(Orice a tip de abonament are un preț)
- ☐ Tip_abonament_Include_sauna_NN - atributul Include_sauna va fi NN

(Pentru orice tip de abonament este necesar să știm dacă acesta include sau nu facilitatea de a merge la sauna, deoarece acest aspect diferențiază abonamentele)

- ☐ Tip_abonament_Include_aerobic_NN - atributul Include_aerobic va fi NN

(Pentru orice tip de abonament este necesar să știm dacă acesta include sau nu facilitatea de a putea face aerobic, deoarece acest aspect diferențiază abonamentele)

Constrângeri Check:

- ☐ Tip_abonament_Include_aerobic - atributul Include_aerobic poate avea doar valorile Da sau Nu
- ☐ Tip_abonament_Include_sauna_CK - atributul Include_sauna poate avea doar valorile Da sau Nu

- Tabelul Vestiar

Constrângeri Primary Key:

- ☐ Vestiar_ID_Vestiar_PK - atributul ID_Vestiar va fi PK

Constrângeri NOT NULL:

- ☐ Vestiar_Numar_chei_NN - atributul Numar_chei va fi NN
(Fiecare vestiar are un numar de chei)
- ☐ Vestiar_Dimensiune_NN - atributul Dimensiune va fi NN
(Fiecare vestiar are o dimensiune)

- Tabelul Angajat

Constrângeri Primary Key:

- ☐ Angajat_ID_Angajat_PK - atributul ID_Angajat va fi PK

Constrângeri NOT NULL:

- ☐ Angajat_Nume_NN - atributul Nume va fi NN
(Fiecare angajat are un nume)
- ☐ Angajat_Prenume_NN - atributul Prenume va fi NN
(Fiecare angajat are un prenume)
- ☐ Angajat_Data_nastere_NN - atributul Data_nastere va fi NN
(Fiecare angajat are o dată de naștere)
- ☐ Angajat_Post_NN - atributul Post va fi NN
(Fiecare angajat lucrează pe un post)
- ☐ Angajat_Salariu_NN - atributul Salriu va fi NN
(Fiecare angajat are un salariu)
- ☐ Angajat_Data_angajare_NN - atributul Data_angajare va fi NN
(Fiecare angajat are o dată de angajare)

Constrângeri Default:

- ☐ Data_angajare - atributul Data_angajare va fi inițializat by default cu valoarea TRUNC(SYSDATE) în cazul în care nu se introduc date

Constrângeri Check:

- ☐ Angajat_Salariu_CK - atributul Salariu va avea valoarea mai mare sau egală cu 1000

- Tabelul Vanzari

Constrângeri Primary Key:

- ☐ Vanzari_ID_Vanzare_PK - atributul ID_Vanzare va fi PK

Constrângeri Foreign Key:

- ☐ Vanzari_ID_Client_FK - atributul ID_Client va fi FK
(Face referire la atributul ID_Client din tableul Client)

- ☐ Vanzari_ID_Angajat_FK - atributul ID_Angajat va fi FK
(Face referire la atributul ID_Angajat din tabelul Angajat)

Constrângeri NOT NULL:

- ☐ Vanzari_Data_NN - atributul Data va fi NN
(Fiecare vânzare trebuie să aibă o dată)
- ☐ Vanzari_ID_Client_NN - atributul ID_Client va fi NN
(Orice vânzare trebuie să fie făcută către un client)
- ☐ Vazari_ID_Angajat_NN - atribut ID_Angajat va fi NN
(Orice vanzare trebuie făcută de către un angajat)

Constrângeri Default:

- ☐ Data - atributul Data va fi inițializat by default cu valoarea
TRUNC(SYSDATE) în cazul în care nu se introduc date

- Tabelul Produse_vandute

Constrângeri Primary Key:

- ☐ Produse_vandute_ID_Produse_vandute_PK - atributul
ID_Produse_vandute va fi PK

Constrângeri Foreign Key:

- ☐ Produse_vandute_ID_Produs_FK - atributul ID_Produs va fi FK
(Face referire la atributul ID_Produs din tabelul Produse)
- ☐ Produse_vandute_ID_Vanzare_FK - atributul ID_Vanzare va fi FK
(Face referire la atributul ID_Vanzare din tabelul Vanzari)

Constrângeri NOT NULL:

- ☐ Produse_vandute_ID_Produs_NN - atributul ID_Produs va fi NN
(Orice produs vandut este un produs)
- ☐ Produse_vandute_ID_Vanzare_NN - atributul ID_Vanzare va fi NN

(Orice produs vândut face parte dintr-o vânzare)

- ☐ Produse_vandute_Cantitate_NN - atributul Cantitate va fi NN
(Din orice produs vândut trebuie să existe o cantitate)

Constrângeri Default:

- ☐ Produse_vandute_Cantitate_DF - atributul Cantitate va fi inițializat by default cu valoarea 1 în caz ca nu se introduc date

Constrângeri Check:

- ☐ Produse_vandute_Cantitate_CK - atributul Cantitate trebuie să aibă o valoare mai mare sau egală cu 1, o vânzare nu poate să fie mai mică decât 1

- Tabelul Produse

Constrângeri Primary Key:

- ☐ Produse_ID_Produs_PK - atributul ID_Produs va fi PK

Constrângeri Foreign Key:

- ☐ Produse_ID_Furnizor_FK - atributul ID_Furnizor va fi FK
(Face referire la atributul ID_Furnizor din tabelul Furnizor)
- ☐ Produse_ID_Categorie_FK - atributul ID_Categorie va fi FK
(Face referire la atributul ID_Categorie din tabelul Categorie_produs)

Constrângeri NOT NULL:

- ☐ Produse_Denumire_produs_NN - atributul Denumire_produs va fi NN
(Fiecare produs trebuie să aibă o denumire)
- ☐ Produse_ID_Furnizor_NN - atributul ID_Furnizor va fi NN
(Fiecare produs are un furnizor)
- ☐ Produse_ID_Categorie_NN - atributul ID_Categorie va fi NN

(Fiecare Produs face parte dintr-o categorie)

- ☐ Produse_Pret_vanzare_NN - atributul Pret_vanzare va fi NN
(Fiecare produs are un preț de vanzare)
- ☐ Produse_Pret_cumparare_NN - atributul Pret_cumparare va fi NN
(Fiecare produs are un preț de cumpărare)
- ☐ Produse_Stoc_NN - atribut Stoc va fi NN
(Din fiecare produs există un anumit stoc)
- ☐ Produse_Cantitate_vanduta_NN - atributul Cantitate_vanduta va fi NN
(Din fiecare produs există o anumită cantitate vândută)

Constrangeri Default:

- ☐ Produse_Stoc_DF - atributul Stoc va fi inițializat by default cu valoarea 0 în caz ca nu se introduc date
- ☐ Produse_Cantitate_vanduta_DF - atributul Cantitate_vanduta va fi inițializat by default cu valoarea 0 în caz ca nu se introduc date

Constrangeri Check:

- ☐ Produse_Pret_vanzare_CK - atributul Pret_vanzare trebuie să aibă o valoare mai mare sau egala cu atributul Pret_cumparare, deoarece nu putem vinde un produs cu un preț mai mic decat l-am cumparat
- ☐ Produse_Stoc_CK - atribut Stoc trebuie să aibă o valoare mai mare decat 0 altfel nu există produse de vândut

- Tabelul Furnizor

Constrângeri Primary Key:

- ☐ Furnizor_ID_Furnizor_PK

Constrângeri NOT NULL:

- ☐ Furnizor_Nume_NN - atributul Nume va fi NN
(Fiecare furnizor are un nume)
- ☐ Furnizor_Adresa_NN - atributul Adresa va fi NN
(Fiecare furnizor are o adresă)
- Tabelul Categorie_produc
- Constrângeri Primary Key:
 - ☐ Categorie_produc_ID_Categorie_produc_PK
- Constrângeri NOT NULL:
 - ☐ Categorie_produc_Nume_NN - atributul Nume va fi NN
(Fiecare categorie de produs are un nume)

3.4 Scheme relaționale

- Tabel Client(#ID_Client, Nume, Prenume, Adresa, ID_Vestiar (FK), ID_Angajat (FK), Data_nastere)
- Tabel Abonamente(#ID_Abonament, ID_Tip_abonament(FK), ID_Client(FK), Data_inceput, Data_sfarsit, ID_Angajat(FK))
- Tabel Tip_abonament(#ID_Tip_abonament, Denumire, Pret, Include_sauna, Include_aerobic)
- Tabel Vestiar(#ID_Vestiar, Numar_Chei, Dimensiune)
- Tabel Angajat(#ID_Angajat, Nume, Prenume, Data_nastere, Post, Salariu, Data_angajare, Vanzari, Nr_clienti_antrenati)
- Tabel Vanzari(#ID_Vanzari, ID_Client(FK), Data, ID_Angajat(FK))
- Tabel Produse_vandute(#ID_Produse_vandute, ID_Produc(FK), ID_Vanzare(FK), Cantitate)

- Tabel Produse(#ID_Produs, Denumire_denumire, ID_Furnizor(FK), ID_Categorie(FK), Pret_vanzare, Pret_cumparare, Stoc, Cantitate_vanduta)
- Tabel Furnizor(#ID_Furnizor, Nume, Adresa)
- Tabel Categorie_produs(#ID_Categorie_produs, Nume)

3.5 Descrierea constrângerilor ON DELETE CASCADE/ SET NULL

- Tabel Client
 - ☐ Client_ID_Vestiar - Set NULL
Dacă se șterge un vestiar atunci atributul ID_Vestiar va fi marcat cu valoarea NULL
 - ☐ Client_ID_Angajat - Set NULL
Dacă se șterge un angajat atunci atributul ID_Angajat ca fi marcat cu valoarea NULL
- Tabel Abonamente
 - ☐ Abonamente_ID_Tip_abonament - On delete cascade
Dacă se șterge un tip de abonament, se șterge și abonamentul pentru că nu poate exista un abonament de niciun fel
 - ☐ Abonamente_ID_Client - On delete cascade
Dacă se șterge un client, se șterge și abonamentul pentru ca nu poate un abonament atribuit nimănui
 - ☐ Abonamente_ID_Angajat - Set NULL
Dacă se șterge un angajat atunci atributul ID_Angajat ca fi marcat cu valoarea NULL
- Tabel Vanzari
 - ☐ Vanzari_ID_Client - On delete cascade

Dacă se șterge un client, se șterge și vânzarea pentru că nu se poate face o vânzare fără un client

☐ Vanzari_ID_Angajat - On delete cascade

Dacă se șterge un client, se șterge și vânzarea pentru că nu se poate face o vânzare fără un vânzător

- Tabel Produse_vandute

☐ Produse_vandute_ID_Produs - On delete cascade

Dacă se șterge un produs, se șterge și produsul vândut deoarece nu se poate efectua o vânzare fără un produs

☐ Produs_vandute_ID_Vanzare - On delete cascade

Dacă se șterge o vânzare, se șterge și produsul vândut deoarece nu poate exista un produs vândut fără o vânzare

- Tabel Produse

☐ Produse_ID_Furnizor - Set Null

Dacă se șterge un furnizor, produsele din baza de date rămâne, iar câmpul ID_Furnizor va fi marcat cu valoarea NULL

☐ Produse_ID_Categorie - On delete cascade

Dacă se șterge o categorie de produs, se vor șterge și produsele din ea, pentru că nu poate exista un produs care să nu facă parte din nicio categorie

4. Implementarea in Oracle

4.1 Tabele și a constrângeri

--CREAREA TABELELOR

```
create sequence Vestiar_ID_Vestiar_Seq
    start with 1
maxvalue 99999
nocycle nocache;
```

```
create table Vestiar (
    ID_Vestiar int constraint Vestiar_ID_Vestiar_PK primary key,
    Numar_chei int constraint Vestiar_Numar_chei_NN not null,
    Dimeniune int constraint Vestiar_Dimensiune_NN not null
);
```

```
create sequence Angajat_ID_Angajat_Seq
    start with 1
maxvalue 99999
nocycle nocache;
```

```
create table Angajat (
    ID_Angajat int constraint Angajat_ID_Angajat_PK primary key,
    Nume varchar(20) constraint Angajat_Nume_NN not null,
    Prenume varchar(20) constraint Angajat_Prenume_NN not null,
    Data_nastere date constraint Angajat_Data_nastere_NN not null,
    Post varchar(20) constraint Angajat_Post_NN not null,
    Salariu number(7,2)
    constraint Angajat_Salariu_NN not null
    constraint Angajat_Salariu_CK check (Salariu >= 1000),
    Data_angajare date default trunc(sysdate)
    constraint Angajat_Data_angajare_NN not null,
    Vanzari int,
    Nr_clienti_antrenati int
);
```

```
create sequence Client_ID_Client_Seq
    start with 1
maxvalue 99999
nocycle nocache;
```

```
create table Client (
    ID_Client int constraint Client_ID_Client_PK primary key,
    Nume varchar(20),
    Prenume varchar(20),
    Adresa varchar(30),
    ID_Vestiar int constraint Client_ID_Vestiar_FK references
    Vestiar(ID_Vestiar) on delete set null,
    ID_Angajat int constraint Client_ID_Angajat_FK references
    Angajat(ID_Angajat) on delete set null,
    Data_nastere date
);
```

```

create sequence Tip_abon_ID_Tip_abonament_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Tip_abon (
    ID_Tip_abon int constraint Tip_abon_ID_Tip_abonament_PK primary key,
    Denumire varchar(30) constraint Tip_abon_Denumire_NN not null,
    Pret number(7,2) constraint Tip_abon_Pret_NN not null,
    Include_sauna char(2)
constraint Tip_abon_Include_sauna_NN not null
    constraint Tip_abon_Include_sauna_CK check ( Include_sauna in ('Da',
'Nu') ),
    Include_aerobic char(2)
constraint Tip_abon_Include_aerobic_NN not null
    constraint Tip_abon_Include_aerobic_CK check ( Include_aerobic in
('Da', 'Nu') )
);

create sequence Abonamente_ID_Abonament_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Abonamente (
    ID_Abonamente int constraint Abonamente_ID_Abonamente_PK primary
key,
    ID_Tip_Abonament int
    constraint Abonamente_ID_Tip_Abonament_NN not null
    constraint Abonamente_ID_Tip_Abonament_FK references
Tip_abon(ID_Tip_abon) on delete cascade,
    ID_Client int
    constraint Abonamente_ID_Client_NN not null
    constraint Abonamente_ID_Client_FK references Client(ID_Client) on
delete cascade,
    Data_inceput date default trunc(sysdate),
    Data_sfarsit date,
    ID_Angajat int
    constraint Abonamente_ID_Angajat_NN not null
    constraint Abonamente_ID_Angajat_FK references Angajat(ID_Angajat)
on delete set null
);

create sequence Vanzari_ID_Vanzare_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Vanzari (
    ID_Vanzare int constraint Vanzari_ID_Vanzare_PK primary key,
    ID_Client int

```

```

        constraint Vanzari_ID_Client_FK references Client(ID_Client) on
delete cascade
        constraint Vanzari_ID_Client_NN not null,
Data date default trunc(sysdate)
constraint Vanzari_Data_NN not null,
ID_Angajat int
        constraint Vazari_ID_Angajat_FK references Angajat(ID_Angajat) on
delete cascade
        constraint Vazari_ID_Angajat_NN not null
);

create sequence Furnizor_ID_Furnizor_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Furnizor (
    ID_Furnizor int constraint Furnizor_ID_Furnizor_PK primary key,
Nume varchar(20) constraint Furnizor_Nume_NN not null,
Adresa varchar(30) constraint Furnizor_Adresa_NN not null
);

create sequence Cat_produș_ID_Cat_produș_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Cat_produș (
    ID_Categorie_produș int constraint Cat_produș_ID_Cat_produș_PK
primary key,
Nume varchar(20) constraint Cat_produș_Nume_NN not null
);

create sequence Produse_ID_Produs_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Produse (
    ID_Produs int constraint Produse_ID_Produs_PK primary key,
Denumire_produș varchar(20) constraint Produse_Denumire_produș_NN not null,
ID_Furnizor int
        constraint Produse_ID_Furnizor_NN not null
        constraint Produse_ID_Furnizor_FK references Furnizor(ID_Furnizor)
on delete set null,
    ID_Categorie int
        constraint Produse_ID_Categorie_NN not null
        constraint Produse_ID_Categorie_FK references
Cat_produș(ID_Categorie_produș) on delete cascade,
Pret_vanzare number(7,2)
        constraint Produse_Pret_vanzare_NN not null,
        constraint Produse_Pret_vanzare_CK check ( Pret_vanzare >= Pret_cumparare
),

```

```

Pret_cumparare number(7,2) constraint Produse_Pret_cumparare_NN not null,
Stoc int default 0
constraint Produse_Stoc_CK check ( Stoc > 0 )
constraint Produse_Stoc_NN not null,
Cantite_vanduta int default 0
constraint Produse_Cantitate_vanduta_NN not null
);

create sequence Prod_vand_ID_Prod_vand_Seq
    start with 1
maxvalue 99999
nocycle nocache;

create table Prod_vand (
    ID_Produse_vandute int constraint Prod_vand_ID_Prod_vand_PK primary
key ,
ID_Produs int
    constraint Prod_vand_ID_Produs_FK references Produse(ID_Produs) on
delete cascade
    constraint Prod_vand_ID_Produs_NN not null,
ID_Vanzare int
    constraint Prod_vand_ID_Vanzare_FK references Vanzari(ID_Vanzare) on
delete cascade
    constraint Prod_vand_ID_Vanzare_NN not null,
Cantitate int default 1
constraint Prod_vand_Cantitate_NN not null,
constraint Prod_vand_Cantitate_CK check (Cantitate>=1)
);

--CONSTRANGERI, SEQUENCE SI STERGERE PENTRU TABELE

    drop sequence Client_ID_Client_Seq;
drop table Client cascade constraints;

drop sequence Abonamente_ID_Abonament_Seq;
drop table Abonamente;

drop sequence Tip_abon_ID_Tip_abonament_Seq;
drop table Tip_abon cascade constraints;

drop sequence Vestiar_ID_Vestiar_Seq;
drop table Vestiar cascade constraints;

drop sequence Angajat_ID_Angajat_Seq;
drop table Angajat cascade constraints;

drop sequence Vanzari_ID_Vanzare_Seq;
drop table Vanzari cascade constraints;

drop sequence Prod_vand_ID_Prod_vand_Seq;
drop table Prod_vand;

drop sequence Produse_ID_Produs_Seq;

```

```

drop table Produse cascade constraints;

drop sequence Furnizor_ID_Furnizor_Seq;
drop table Furnizor cascade constraints;

drop sequence Cat_produs_ID_Cat_produs_Seq;
drop table Cat_produs cascade constraints;

```

4.2 Date

```
--INSERARE DATE
```

```
--Vestiar
```

```

insert into Vestiar(ID_Vestiar, Numar_chei, Dimeniune)
values (Vestiar_ID_Vestiar_Seq.nextval, 2, 2500);

```

```

insert into Vestiar(ID_Vestiar, Numar_chei, Dimeniune)
values (Vestiar_ID_Vestiar_Seq.nextval, 3, 2400);

```

```

insert into Vestiar(ID_Vestiar, Numar_chei, Dimeniune)
values (Vestiar_ID_Vestiar_Seq.nextval, 1, 2450);

```

```

insert into Vestiar(ID_Vestiar, Numar_chei, Dimeniune)
values (Vestiar_ID_Vestiar_Seq.nextval, 3, 2550);

```

```

insert into Vestiar(ID_Vestiar, Numar_chei, Dimeniune)
values (Vestiar_ID_Vestiar_Seq.nextval, 2, 2500);

```

```
--Angajat
```

```

insert into Angajat(ID_Angajat, Nume, Prenume, Data_nastere, Post,
Salariu, Data_angajare, Vanzari, Nr_clienti_antrenati)
values (Angajat_ID_Angajat_Seq.nextval, 'Vaidos', 'Marcel',
TO_DATE('12-01-2002', 'DD-MM-YYYY'), 'Antrenor', 3000,
TO_DATE('14-05-2020', 'DD-MM-YYYY'), null, 5);

```

```

insert into Angajat(ID_Angajat, Nume, Prenume, Data_nastere, Post, Salariu,
Data_angajare, Vanzari, Nr_clienti_antrenati)
values (Angajat_ID_Angajat_Seq.nextval, 'Husariu', 'Paul',
TO_DATE('29-11-2003', 'DD-MM-YYYY'), 'Receptioner', 2500,
TO_DATE('13-08-2019', 'DD-MM-YYYY'), 6, null);

```

```

insert into Angajat(ID_Angajat, Nume, Prenume, Data_nastere, Post, Salariu,
Data_angajare, Vanzari, Nr_clienti_antrenati)
values (Angajat_ID_Angajat_Seq.nextval, 'Spaciu', 'Daniel',
TO_DATE('23-06-2000', 'DD-MM-YYYY'), 'Antrenor', 3100,
TO_DATE('02-12-2018', 'DD-MM-YYYY'), null, 3);

```

```

insert into Angajat(ID_Angajat, Nume, Prenume, Data_nastere, Post, Salariu,
Data_angajare, Vanzari, Nr_clienti_antrenati)

```

```
values (Angajat_ID_Angajat_Seq.nextval, 'Dobre', 'Ana',
TO_DATE('30-05-1989', 'DD-MM-YYYY'), 'Ingrijitoare', 2000,
TO_DATE('18-03-2015', 'DD-MM-YYYY'), null, null);
```

```
insert into Angajat(ID_Angajat, Nume, Prenume, Data_nastere, Post, Salariu,
Data_angajare, Vanzari, Nr_clienti_antrenati)
values (Angajat_ID_Angajat_Seq.nextval, 'Tepes', 'Raluca',
TO_DATE('19-09-1999', 'DD-MM-YYYY'), 'Receptioner', 2500,
TO_DATE('04-04-2021', 'DD-MM-YYYY'), 10, null);
```

--Client

```
insert into Client(ID_Client, Nume, Prenume, Adresa, ID_Vestiar,
ID_Angajat, Data_nastere)
values (Client_ID_Client_Seq.nextval, 'Ionescu', 'Mihai', 'Calea Giulesti,
nr.69', 1, 1, TO_DATE('12-01-2003', 'DD-MM-YYYY'));
```

```
insert into Client(ID_Client, Nume, Prenume, Adresa, ID_Vestiar,
ID_Angajat, Data_nastere)
values (Client_ID_Client_Seq.nextval, null, null, null, null, null, null);
```

```
insert into Client(ID_Client, Nume, Prenume, Adresa, ID_Vestiar,
ID_Angajat, Data_nastere)
values (Client_ID_Client_Seq.nextval, 'Dumitru', 'Fabian', 'Calea
Dristorului, nr.10', 2, 3, TO_DATE('26-02-1999', 'DD-MM-YYYY'));
```

```
insert into Client(ID_Client, Nume, Prenume, Adresa, ID_Vestiar,
ID_Angajat, Data_nastere)
values (Client_ID_Client_Seq.nextval, 'Besel', 'Adrian', 'Str Baba Novac,
nr.10', 4, 2, TO_DATE('24-07-2002', 'DD-MM-YYYY'));
```

```
insert into Client(ID_Client, Nume, Prenume, Adresa, ID_Vestiar,
ID_Angajat, Data_nastere)
values (Client_ID_Client_Seq.nextval, 'Popescu', 'George', 'Str.
Camapuling, nr.1', 3, 1, TO_DATE('12-08-2002', 'DD-MM-YYYY'));
```

--Tip abonament

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o zi', 40,
'Nu', 'Nu');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o zi', 45,
'Da', 'Nu');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o zi', 45,
'Nu', 'Da');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o zi', 50,
'Da', 'Da');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o
saptamana', 60, 'Nu', 'Nu');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o
saptamana', 65, 'Da', 'Nu');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o
saptamana', 65, 'Nu', 'Da');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o
saptamana', 70, 'Da', 'Da');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o luna',
120, 'Nu', 'Nu');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o luna',
125, 'Da', 'Nu');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o luna',
125, 'Nu', 'Da');
```

```
insert into Tip_abon(ID_Tip_abon, Denumire, Pret, Include_sauna,
Include_aerobic)
values (Tip_abon_ID_Tip_abonament_Seq.nextval, 'Abonament pentru o luna',
130, 'Da', 'Da');
```

--Abonamente

```
insert into Abonamente(ID_Abonamente, ID_Tip_Abonament, ID_Client,
Data_inceput, Data_sfarsit, ID_Angajat)
values (Abonamente_ID_Abonament_Seq.nextval, 1, 1, TO_DATE('24-01-2023',
'DD-MM-YYYY'), TO_DATE('25-01-2023', 'DD-MM-YYYY'), 5);
```

```
insert into Abonamente(ID_Abonamente, ID_Tip_Abonament, ID_Client,
Data_inceput, Data_sfarsit, ID_Angajat)
values (Abonamente_ID_Abonament_Seq.nextval, 5, 4, TO_DATE('24-01-2023',
'DD-MM-YYYY'), TO_DATE('31-01-2023', 'DD-MM-YYYY'), 1);
```

```
insert into Abonamente(ID_Abonamente, ID_Tip_Abonament, ID_Client,
Data_inceput, Data_sfarsit, ID_Angajat)
values (Abonamente_ID_Abonament_Seq.nextval, 2, 1, TO_DATE('01-02-2023',
'DD-MM-YYYY'), TO_DATE('01-02-2023', 'DD-MM-YYYY'), 3);
```

```
insert into Abonamente(ID_Abonamente, ID_Tip_Abonament, ID_Client,
Data_inceput, Data_sfarsit, ID_Angajat)
values (Abonamente_ID_Abonament_Seq.nextval, 3, 4, TO_DATE('05-01-2023',
'DD-MM-YYYY'), TO_DATE('05-02-2023', 'DD-MM-YYYY'), 4);
```

```
insert into Abonamente(ID_Abonamente, ID_Tip_Abonament, ID_Client,
Data_inceput, Data_sfarsit, ID_Angajat)
values (Abonamente_ID_Abonament_Seq.nextval, 2, 4, TO_DATE('17-01-2023',
'DD-MM-YYYY'), TO_DATE('18-01-2023', 'DD-MM-YYYY'), 2);
```

--Vanzari

```
insert into Vanzari(ID_Vanzare, ID_Client, ID_Angajat)
values (Vanzari_ID_Vanzare_Seq.nextval, 1, 3);
```

```
insert into Vanzari(ID_Vanzare, ID_Client, Data, ID_Angajat)
values (Vanzari_ID_Vanzare_Seq.nextval, 3, TO_DATE('12-01-2023',
'DD-MM-YYYY'), 4);
```

```
insert into Vanzari(ID_Vanzare, ID_Client, Data, ID_Angajat)
values (Vanzari_ID_Vanzare_Seq.nextval, 5, TO_DATE('02-11-2022',
'DD-MM-YYYY'), 1);
```

```
insert into Vanzari(ID_Vanzare, ID_Client, Data, ID_Angajat)
values (Vanzari_ID_Vanzare_Seq.nextval, 4, TO_DATE('24-12-2022',
'DD-MM-YYYY'), 2);
```

```
insert into Vanzari(ID_Vanzare, ID_Client, Data, ID_Angajat)
values (Vanzari_ID_Vanzare_Seq.nextval, 2, TO_DATE('17-12-2022',
'DD-MM-YYYY'), 5);
```

--Furnizor

```
insert into Furnizor(ID_Furnizor, Nume, Adresa)
values (Furnizor_ID_Furnizor_Seq.nextval, 'GymBeam', 'Sos. Mihai Bravu,
nr.2');
```

```
insert into Furnizor(ID_Furnizor, Nume, Adresa)
values (Furnizor_ID_Furnizor_Seq.nextval, 'LR', 'Bd-ul Energeticienilor,
nr.7');
```

```
insert into Furnizor(ID_Furnizor, Nume, Adresa)
```



```
values (Furnizor_ID_Furnizor_Seq.nextval, 'Dymatize', 'Str. Nicolae Iorga, nr.1');
```

```
insert into Furnizor(ID_Furnizor, Nume, Adresa)
values (Furnizor_ID_Furnizor_Seq.nextval, 'MuscleTech', 'Splaiul Unirii, nr.12');
```

```
insert into Furnizor(ID_Furnizor, Nume, Adresa)
values (Furnizor_ID_Furnizor_Seq.nextval, 'BioTech', 'Int. Rosiori, nr.50');
```

--Categorie produs

```
insert into Cat_produc(ID_Categorie_produc, Nume)
values (Cat_produc_ID_Cat_produc_Seq.nextval, 'Proteine');
```

```
insert into Cat_produc(ID_Categorie_produc, Nume)
values (Cat_produc_ID_Cat_produc_Seq.nextval, 'Vitamine');
```

```
insert into Cat_produc(ID_Categorie_produc, Nume)
values (Cat_produc_ID_Cat_produc_Seq.nextval, 'Creatina');
```

```
insert into Cat_produc(ID_Categorie_produc, Nume)
values (Cat_produc_ID_Cat_produc_Seq.nextval, 'Gustari proteice');
```

```
insert into Cat_produc(ID_Categorie_produc, Nume)
values (Cat_produc_ID_Cat_produc_Seq.nextval, 'Bauturi');
commit;
```

--Produse

```
insert into Produse(ID_Produc, Denumire_produc, ID_Furnizor, ID_Categorie, Pret_vanzare, Pret_cumparare, Stoc, Cantitate_vanduta)
values (Produse_ID_Produc_Seq.nextval, 'Colagen', 1, 1, 50, 45, 100, 12);
```

```
insert into Produse(ID_Produc, Denumire_produc, ID_Furnizor, ID_Categorie, Pret_vanzare, Pret_cumparare, Stoc, Cantitate_vanduta)
values (Produse_ID_Produc_Seq.nextval, 'Vitamina C', 3, 2, 30, 25, 50, 2);
```

```
insert into Produse(ID_Produc, Denumire_produc, ID_Furnizor, ID_Categorie, Pret_vanzare, Pret_cumparare, Stoc, Cantitate_vanduta)
values (Produse_ID_Produc_Seq.nextval, 'Baton proteic', 5, 4, 10, 7, 321, 52) ;
```

```
insert into Produse(ID_Produc, Denumire_produc, ID_Furnizor, ID_Categorie, Pret_vanzare, Pret_cumparare, Stoc, Cantitate_vanduta)
values (Produse_ID_Produc_Seq.nextval, 'Creatina monohidrata', 4, 3, 90, 80, 12, 4);
```

```
insert into Produse(ID_Produc, Denumire_produc, ID_Furnizor, ID_Categorie, Pret_vanzare, Pret_cumparare, Stoc, Cantitate_vanduta)
values (Produse_ID_Produc_Seq.nextval, 'Bauturi izotonice', 2, 5, 40, 35, 37, 15);
```

--Produce vandute

```
insert into Prod_vand(ID_Produse_vandute, ID_Produs, ID_Vanzare, Cantitate)
values (Prod_vand_ID_Prod_vand_Seq.nextval, 1, 3, 5);
```

```
insert into Prod_vand(ID_Produse_vandute, ID_Produs, ID_Vanzare, Cantitate)
values (Prod_vand_ID_Prod_vand_Seq.nextval, 3, 2, 3);
```

```
insert into Prod_vand(ID_Produse_vandute, ID_Produs, ID_Vanzare, Cantitate)
values (Prod_vand_ID_Prod_vand_Seq.nextval, 4, 5, 1);
```

```
insert into Prod_vand(ID_Produse_vandute, ID_Produs, ID_Vanzare, Cantitate)
values (Prod_vand_ID_Prod_vand_Seq.nextval, 2, 4, 2);
```

```
insert into Prod_vand(ID_Produse_vandute, ID_Produs, ID_Vanzare, Cantitate)
values (Prod_vand_ID_Prod_vand_Seq.nextval, 5, 1, 4);
```

```
commit ;
```