UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

PROIECT BAZE DE DATE

STUDENT,

ȘTEICĂ MĂLINA-ALEXA, GRUPA 261

BUCUREȘTI

2023

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

BAZĂ DE DATE PENTRU GESTIUNEA UNUI LANȚ DE MAGAZINE DE HAINE

STUDENT,

ȘTEICĂ MĂLINA-ALEXA, GRUPA 261

BUCUREȘTI

2023

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

**CUPRINS**

1. Introducere
2. Diagrama entitate-relație
3. Diagrama conceptuală
   1. Tabelul „LOCATIE”
   2. Tabelul „MAGAZIN”
   3. Tabelul „ANGAJAT”
   4. Tabelul „CONTRACT”
   5. Tabelul „DISTRIBUITOR”
   6. Tabelul „APROV\_MAGAZIN”
   7. Tabelul „PRODUS”
   8. Tabelul „STOC\_MAGAZIN”
   9. Tabelul „COMANDA”
   10. Tabelul „DETALII\_COMANDA”
   11. Tabelul „CLIENT”

4. Codul sursă

**1. Introducere**

Baza de date realizată are rolul de a păstra informațiile unui lanț de magazine de haine internațional. Tabelele stochează date atât despre angajați și distribuitori, cât și despre tipurile de produse comercializate și despre clienții care le achiziționează.

Modelul oferă informații despre locația fiecărui magazin și tipul de produse care poate fi comandat de către client din magazinul respectiv, iar angajaților le corespund mai multe detalii legate de contractele incheiate și de magazinul unde își desfășoară activitatea.

Legătura dintre clienți și magazinele de unde achiziționează produsele se face printr-o comandă, a cărei valoare este calculată în funție de produs și numărul de bucăți.

Regulile bazei de date sunt intuitive, aceasta putând fi utilizată în lumea reală și regulile reies din interacțiunea entităților pe baza cărora a fost construită:

* Fiecare magazin are un stoc de produse, existând magazine care sunt specilizate în comercializarea unui singur tip de produs (magazin doar cu rochii);
* Unele magazine nu vor comercializa toate tipurile de produse, existând magazine unde se găsesc bluze, pantaloni, rochii, dar nu și geci sau alte accesorii;
* Un magazin poate avea o locație unică, însă aceasta poate lipsi, deoacere pot exista magazine strict digitale, care doar livreaza marfa de la distribuitor la client;
* Livrarea produselor se va considera făcută de o firma separată de curierat, motiv pentru care nu este discutată;
* Stocul magazinelor este considerat infinit;
* Exista angajați care pot avea mai multe roluri și, deci, mai multe contracte în cadrul unui magazin;
* Un astfel de sistem ajută clienții să vizualizeze mai usor ce cumpărături au făcut și cât au cheltuit. Clientul poate accesa comanda sa pentru a vedea valoarea totală, aceasta fiind calculată în functie de ce produse a comandat;

**2. Diagrama entitate-relație**

O diagramă entitate relație este o tehnică de modelare a datelor care ilustrează grafic entitățile unui sistem informațional și relațiile dintre aceste entități. Ea este un model conceptual și reprezentativ de date utilizate pentru a reprezenta infrastructura cadru a entității.

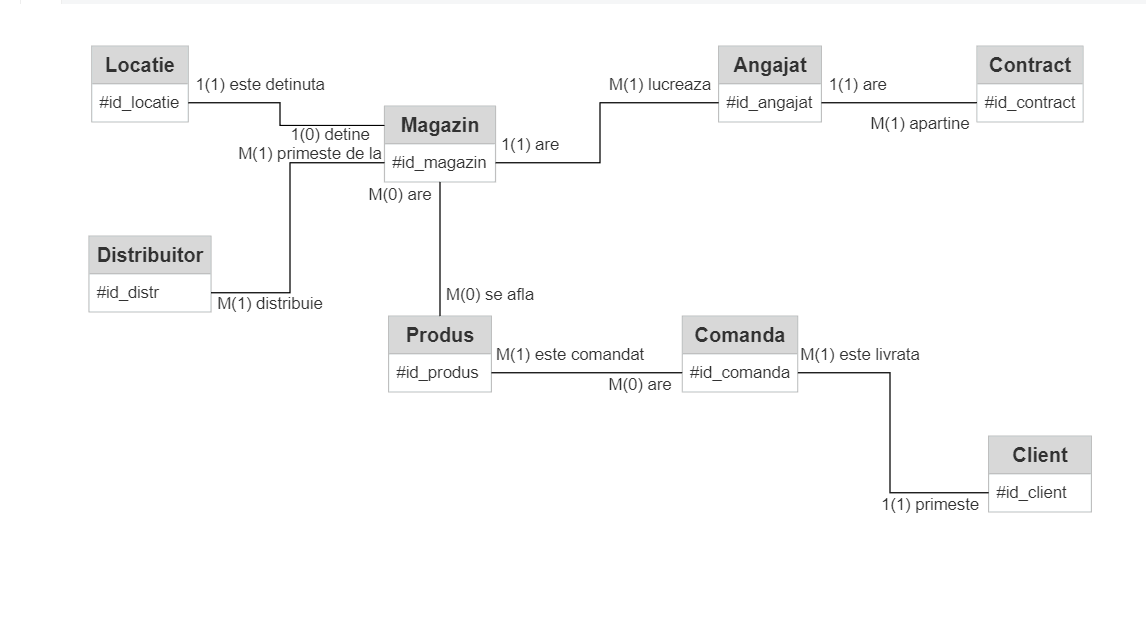


Figura 1 – Diagrama entitate-relație a bazei de date

Diagrama ER este alcătuită din entitățile: LOCATIE, MAGAZIN, DISTRIBUITOR, ANGAJAT, CONTRACT, PRODUS, COMANDA și CLIENT, fiecare având o cheie primară de forma „#id”. Aceasta asigură stocarea datelor într-un mod unic și menținerea relațiilor între entități.

Entitatea „LOCATIE” reprezintă punctul în care se află un magazin. Fiecare locație este deținută de un magazin unic, nu există o locație care nu să aibă un magazin corespunzător. Astfel, relația one-to-one se traduce prin faptul că un magazin poate deține o locație sau niciuna, dar fiecare locație este deținută de un magazin. Cheia primară este „id\_locatie”.

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| LOCATIE-MAGAZIN | „este deținută” | One-to-one | 1/1 |

Entitatea „MAGAZIN” reprezintă „brand”-ul, firma care comercializează produse primite de la distribuitori. Am definit anterior relația dintre un magazin si locația acestuia ca fiind una one-to-one, unde un magazin poate avea o locație sau niciuna, existând magazine digitale. În relație cu distribuitorul, un magazin primeste marfa de la unul sau mai mulți distribuitori, iar un distribuitor are rolul de a asigura marfa unuia sau mai multor magazine. Astfel, relația este una many-to-many. În relație cu angajații, într-un magazin lucrează unul sau mai mulți angajați, fiecare angajat având măcar un magazin la care lucrează, iar fiecare magazin având măcar un angajat. Un magazin poate avea mai multe tipuri de produse sau niciunul, iar un produse se poate afla în mai multe magazine sau în niciunul. Cheia primară este „id\_magazin”

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| MAGAZIN- LOCATIE | „deține” | One-to-one | 1/0 |
| MAGAZIN-DISTRIBUITOR | „primește (de la)” | Many-to-many | Mai mulți/0 |
| MAGAZIN-PRODUS | „are” | Many-to-many | Mai multe/0 |
| MAGAZIN-ANGAJAT | „are” | One-to-many | Mai mulți/1 |

Entitatea „DISTRIBUITOR” reprezintă firma ce asigură marfa care va ajunge pe rafturile magazinelor si, ulterior, la clienți. Un distribuitor se află în relație cu unul sau mai multe magazine, iar un magazin va primi marfa de la unul sau mai mulți distribuitori. Cheia primară este „id\_distr”.

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| DISTRIBUITOR-MAGAZIN | „distribuie” | Many-to-many | Mai mulți/1 |

Entitatea „Angajat” denumește angajații fiecărui magazin. De aceea, un angajat lucrează la un magazin, iar un magazin are unul sau mai mulți angajați, relatia dintre entități fiind una one-to-many. Aceeași relație se stabilește și între angajat și contract, deoarece fiecarui angajat îi corespund unul sau mai multe contracte, un contract aparținând unui singur angajat. Cheia primară este „id\_angajat”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Descriere relație | Verb | | Relație | Cardinalitate maximă și minimă |
| ANGAJAT-MAGAZIN | „lucrează” | | One-to-many | Mai mulți/1 |
| ANGAJAT-CONTRACT | | „are” | One-to-many | 1/1 |

Entitatea „Contract” se referă la datele contractuale ale angajatului. Nu există un angajat fără unul sau mai multe contracte, pentru că acest lucru nu ar avea sens în lumea reală. Cheia primară este „id\_contract”.

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| CONTRACT-ANGAJAT | „aparține” | One-to-many | Mai multe/1 |

Entitatea „PRODUS” descrie tipul de produs comercializat, acesta fiind un articol de îmbrăcăminte. Relația dintre produs și magazin este una many-to-many, deoarece un produs se poate afla, sau nu, în mai multe magazine. Un magazin, de asemenea, poate avea sau nu mai multe produse. Aceeași relatie o are si cu entitatea „COMANDA”, un produs putând să se afle, sau nu, într-o comandă. Cheia primară este „id\_produs”.

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| PRODUS-MAGAZIN | „se află” | Many-to-many | Mai multe/0 |
| PRODUS-COMANDA | „este comandat” | One-to-many | Mai multe/0 |

Entitatea „COMANDA” face legătura între client și produs. Crearea unei comenzi goale este restricționată, deoarece o comandă trebuie să aiba unul sau mai multe produse. O comandă este primită de un singur client, însă același client poate plasa mai multe comenzi, fapt pentru care relația este una one-to-many. Cheia primară este „id\_comanda”.

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| COMANDA-PRODUS | „are” | Many-to-many | Mai multe/1 |
| COMANDA-CLIENT | „este livrată” | One-to-many | Mai multe/1 |

Entitatea „CLIENT” se referă la cumpărătorul care a decis sa ia produse de la un magazin. Clientul poate plasa una sau mai multe comenzi, iar cheia primară este „id\_client”.

|  |  |  |  |
| --- | --- | --- | --- |
| Descriere relație | Verb | Relație | Cardinalitate maximă și minimă |
| CLIENT-COMANDA | „primește” | One-to-many | Mai multe/1 |

Relațiile dintre entități definesc, deci, un set de reguli dupa care a fost proiectată baza de date:

* Un magazin are unul sau mai mulți angajați.
* Un magazin poate avea mai multe produse.
* Un magazin poate deține o locație.
* Un magazin primește marfa de la unul sau mai mulți distribuitori.
* Un angajat lucrează la un magazin.
* Un angajat are unul sau mai multe contracte.
* Un produs se poate afla la mai multe magazine.
* Un produs poate fi comandat în mai multe comenzi.
* Un distribuitor distribuie unuia sau mai multor magazine.
* Un client primește una sau mai multe comenzi.
* O comandă este livrată unui client.
* O comandă are unul sau mai multe produse.
* O locație este deținută de un magazin.
* Un contract aparține unui angajat.

**3. Diagrama conceptuală**

Diagrama conceptuală este un design de nivel înalt (incluzând relațiile dintre datele întregului sistem), care descrie datele și relațiile necesare pentru execuția operațiilor necesare în proiectarea bazei de date. În acest fel va fi implementat modelul, apărând încă trei tabele asociative: „APROV\_MAGAZIN”, „STOC\_MAGAZIN” și „DETALII\_COMANDA”. Relațiile și cardinalitățile dispar, fiind introduse, în schimb, cheile străine și atributele, entitățile devenind tabele.

Schemele relaționale corespunzătoare diagramei conceptuale sunt:

LOCATIE(id\_locatie#, tara, oras, strada, numar)

MAGAZIN(id\_magazin#, id\_locatie)

DISTRIBUITOR(id\_distr#, mail, telefon)

ANGAJAT(id\_angajat#, id\_magazin, mail, telefon, nume)

CONTRACT(id\_contract#, id\_angajat, functie, salariu)

PRODUS(id\_produs#, pret)

COMANDA(id\_comanda#, id\_client, valoare\_totala)

CLIENT(id\_client#, nume, mail, telefon)

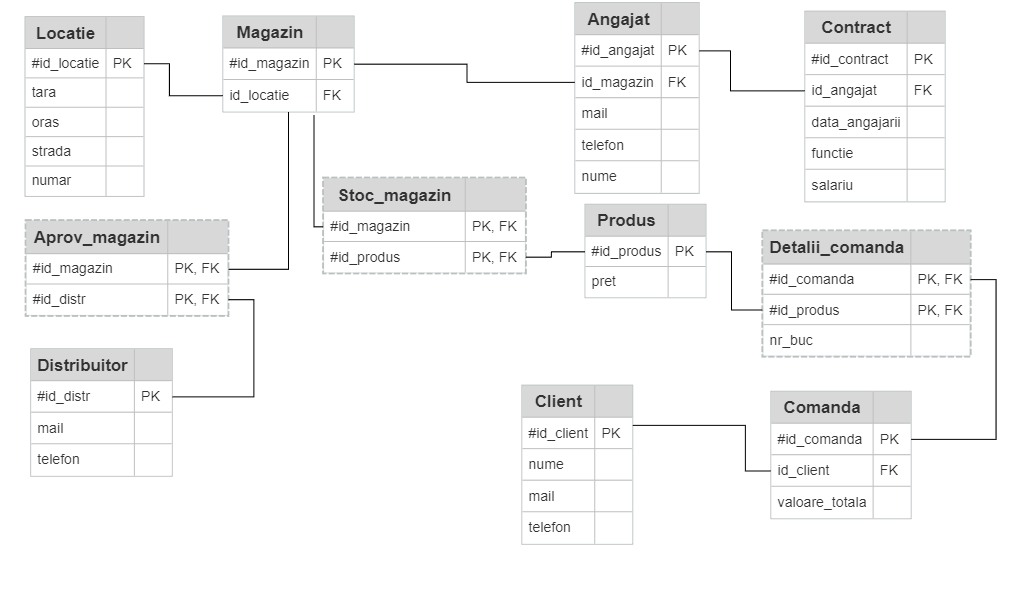


Figura 2 - Diagrama conceptuală

**3.1. Tabelul „LOCAȚIE”**

Acesta a fost creat folosind codul:

*CREATE TABLE locatie (*

*id\_locatie INT,*

*tara VARCHAR(15) NOT NULL,*

*oras VARCHAR(20) NOT NULL,*

*strada VARCHAR(50) NOT NULL,*

*numar INT*

*);*

*ALTER TABLE locatie ADD PRIMARY KEY (id\_locatie);*

Acest tabel stochează informații despre locul unde se află un magazin, folosind 3 atribute de tip varchar, care nu pot lipsi, prin constrângerea „NOT NULL” asigurându-se că nu sunt nule și două atribute de tip integer. Atributul „id\_locatie” primeste rolul de cheie primară, stocând numărul de ordine al fiecărei locații. Mai multe date au fost inserate dupa urmatorul model:

*INSERT INTO locatie VALUES*

*(1,'Romania', 'Bucuresti','Calea Vitan',59);*

**3.2. Tabelul „MAGAZIN”**

*CREATE TABLE Magazin (*

*id\_magazin VARCHAR(15) PRIMARY KEY,*

*id\_locatie INT*

*);*

*ALTER TABLE magazin ADD FOREIGN KEY (id\_locatie) REFERENCES locatie (id\_locatie);*

„MAGAZIN” este un tabel simplu, care stochează numărul de ordine al unui magazin, în atributul de tip integer „id\_magazin” și face legătura cu tabelul „LOCAȚIE” prin cheia străină care este adăugată printr-o constrângere la atributul „id\_locatie”. Datele au fost inserate în felul următor, magazinul „NewFashion” aflându-se, deci, în locația „1”:

INSERT INTO magazin VALUES

('NewFashion',1);

**3.3. Tabelul „ANGAJAT”**

CREATE TABLE angajat (

id\_angajat INT PRIMARY KEY,

id\_magazin VARCHAR(15),

mail VARCHAR(50),

telefon VARCHAR(15),

nume VARCHAR(25) NOT NULL

);

ALTER TABLE angajat ADD FOREIGN KEY (id\_magazin) REFERENCES magazin(id\_magazin);

ALTER TABLE angajat ADD CONSTRAINT mail\_ang UNIQUE (mail);

ALTER TABLE angajat ADD CONSTRAINT telefon\_ang UNIQUE (telefon);

Tabelul „ANGAJAT” reține în atributele de tip varchar „mail”, „telefon” și „nume” datele de contact ale unui angajat. Acestora li se aplică o constrângere care să le asigure unicitatea, acestea fiind specifice fiecărui angajat, iar numele este constrâns să nu fie null. În atributele „id\_angajat” de tip integer si „id\_magazin” sunt memorate numerele de ordine ale fiecărui angajat și cel al magazinului la care lucrează. Cheia primară este „id\_anagajat”, iar „id\_magazin” este cheia străină ce face legătura între angajat și magazin. Un exemplu de date ale unui angajat, care lucrează la magazinul „NewFashion”, este:

INSERT INTO angajat VALUES

(1,'NewFashion','bogdanoctavian@email.com','0742750707', 'Bogdan Octavian');

**3.4. Tabelul „CONTRACT”**

CREATE TABLE contract (

id\_contract INT PRIMARY KEY,

id\_angajat INT,

data\_angajarii DATE NOT NULL,

functie VARCHAR(255),

salariu NUMBER(8,2),

CONSTRAINT ang\_salariu\_min CHECK (salariu > 0)

);

ALTER TABLE contract

ADD FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat);

Tabelul „CONTRACT” stochează datele contractuale în atributele „data\_angajării” de tip date, căreia i se aplică constrângerea „NOT NULL”, „functie” de tip varchar și „salariu” care este un număr real și pozitiv și diferit de zero prin constrângerea aplicată. Cheia primară este reprezentată de „id\_contract”, iar cheia secundară „id\_angajat” face legătura cu angajatul care deține un contract.

INSERT INTO contract VALUES

(1,1,'01-Apr-2022','Manager',4000);

**3.5. Tabelul „DISTRIBUITOR”**

CREATE TABLE distribuitor (

id\_distr VARCHAR(15) PRIMARY KEY,

mail VARCHAR(50),

telefon VARCHAR(15)

);

ALTER TABLE distribuitor ADD CONSTRAINT mail\_distr UNIQUE (mail);

ALTER TABLE distribuitor ADD CONSTRAINT telefon\_distr UNIQUE (telefon);

Tabelul conține cate un număr de ordine stocat în „id\_distr”, ce are rol de cheie primară și datele distribuitorului prin atributele „mail” și „telefon”, care sunt constrânse să fie unice. Tipul de date utilizat este VARCHAR cu o anumită precizie în funcție de tipul de dată memorat.

INSERT INTO distribuitor VALUES

('Fashion Place','fshplc@email.com','+5492954684076');

**3.6. Tabelul „APROV\_MAGAZIN”**

CREATE TABLE aprov\_magazin (

id\_magazin VARCHAR(15),

id\_distr VARCHAR(15),

constraint id\_aprov primary key(id\_magazin, id\_distr),

constraint fk\_aprov\_mag

foreign key(id\_magazin) references magazin(id\_magazin)

on delete cascade,

constraint fk\_aprov\_distr

foreign key(id\_distr) references distribuitor(id\_distr)

on delete cascade

);

„APROV\_MAGAZIN” este o tabelă asociativă, care stabilește legătura între magazin și distribuitor. Atributul „id\_magazin” face legătura cu tabelul „MAGAZIN”, având rol de cheie străină, iar pe același model se stabilește și relatia cu distribuitorul. Cheia primară este formată prin compunere celor două chei străine. Astfel se creează două relații one-to-many, care compuse asigură relația many-to-many. Constrângerea „on delete cascade” asigură că stergerea unui element din tabelele „MAGAZIN” si „DISTRIBUITOR” va genera și stergerea în tabelul „APROV\_MAGAZIN” pentru a se păstra logica și integritatea datelor și a relațiilor. În urmatorul exemplu, se face legătura între primul magazin și doi distribuitori:

INSERT INTO aprov\_magazin VALUES

('NewFashion','Fashion Place');

INSERT INTO aprov\_magazin VALUES

('TrendyVibes','Fashion Place');

**3.7. Tabelul „PRODUS”**

CREATE TABLE produs (

id\_produs VARCHAR(15) PRIMARY KEY,

pret NUMBER(6,2) CHECK (pret > 0)

);

Acest tabel reține numele produsului memorat ca un id în atributul „id\_produs”, care are rol si de cheie primară. Pretul este reținut în atributul cu același nume, căruia i se aplică o constrângere pentru a fi pozitiv și diferit de zero.

INSERT INTO produs VALUES

('Bluza',54.99);

**3.8. Tabelul „STOC\_MAGAZIN”**

CREATE TABLE stoc\_magazin (

id\_magazin VARCHAR(15),

id\_produs VARCHAR(15),

constraint id\_stoc primary key(id\_magazin, id\_produs),

constraint fk\_stoc\_magazin

foreign key(id\_magazin) references magazin(id\_magazin)

on delete cascade,

constraint fk\_stoc\_produs

foreign key(id\_produs) references produs(id\_produs)

on delete cascade

);

Tabelul asociativ „STOC\_MAGAZIN” face legătura între produs și magazinul la care se află. Cheile străine sunt „id\_magazin” și „id\_produs”, iar cheia primară se generează prin compunerea lor și aceeași constrângere „on detele cascade” asigură stergerea datelor în cazul pierderii de date într-unul dintre tabelele „părinte”. În următorul exemplu se introduc în stocul magazinului „NewFashion” două produse:

INSERT INTO stoc\_magazin VALUES

('NewFashion','Bluza');

INSERT INTO stoc\_magazin VALUES

('NewFashion','Rochie');

**3.9. Tabelul „COMANDĂ”**

CREATE TABLE comanda (

id\_comanda INT PRIMARY KEY,

id\_client INT,

valoare\_totala NUMBER(6,2) CHECK (valoare\_totala >= 0)

);

În acest tabel, atributele „id\_comanda” și „id\_client” rețin două numere de ordine, cel din urmă reprezentând o cheie străină ce face legătura cu clienții. Cheia primară este „id\_comandă”, iar atributul „valoare\_totală”, număr real pozitiv, este calculat printr-o formulă, care este adăugată dupa introducerea datelor în toate tabelele:

UPDATE comanda

SET valoare\_totala = (SELECT SUM(nr\_buc\*pret)

FROM detalii\_comanda

JOIN produs ON detalii\_comanda.id\_produs = produs.id\_produs

WHERE id\_comanda = comanda.id\_comanda);

Codul de mai sus preia valoarea prețului din tabelul „PRODUS” și numărul de bucăți din tabelul „DETALII\_COMANDA”, folosind funcția „SUM” pentru a calcula suma, pentru produsele care se află în aceeași comandă și, deci, au același id. „Join”-ul ne permite accesul la tabelul „PRODUS” prin intermediul tabelului asociativ „DETALII\_COMANDA”. Datele au fost introduse prin următorul cod:

INSERT INTO comanda VALUES

(1,1,0);

Astfel, comanda „1”, aparținând clientului „1” are valoarea inițială 0.

**3.10. Tabelul „DETALII\_COMANDA”**

CREATE TABLE detalii\_comanda (

id\_comanda INT,

id\_produs VARCHAR(15),

nr\_buc INT,

constraint id\_detalii primary key(id\_comanda, id\_produs),

constraint fk\_det\_com

foreign key(id\_comanda) references comanda(id\_comanda)

on delete cascade,

constraint fk\_det\_prod

foreign key(id\_produs) references produs(id\_produs)

on delete cascade

);

Acest tabel este creat pe modelul tabelelor asociative anterioare, făcând legătura între produse și comenzi și asigurând relația many-to-many. Cheile strâine sunt „id\_comanda” și „id\_produs”, cheia primară este reprezentată de compunerea lor, iar atributul „nr\_buc” reprezintă numărul de bucăți dintr-un anume produs pe care l-a cumpărat un client, fiind folosit la calculul valorii comenzii. În comanda „1” sunt adăugate două bluze astfel:

INSERT INTO detalii\_comanda VALUES

(1,'Bluza',2);

**3.11. Tabelul „CLIENT”**

CREATE TABLE client (

id\_client INT PRIMARY KEY,

nume VARCHAR(25),

mail VARCHAR(50),

telefon VARCHAR(15)

);

ALTER TABLE client ADD CONSTRAINT mail\_client UNIQUE (mail);

ALTER TABLE client ADD CONSTRAINT telefon\_client UNIQUE (telefon);

Detaliile despre clienți sunt, în final, memorate în tabelul „CLIENT”, în atributele „nume”, „mail și „telefon”, aplicându-se constrângerea pentru ca adresa de mail și numărul de telefon să fie unice. Cheia primară „id\_client” reține numărul de ordine al clientului.

INSERT INTO client VALUES

(1,'Popescu Andrei','popescuandrei@email.com','0702078753');

**4. Codul sursă**

CREATE TABLE locatie (

id\_locatie INT,

tara VARCHAR(15) NOT NULL,

oras VARCHAR(20) NOT NULL,

strada VARCHAR(50) NOT NULL,

numar INT

);

ALTER TABLE locatie ADD PRIMARY KEY (id\_locatie);

CREATE TABLE Magazin (

id\_magazin VARCHAR(15) PRIMARY KEY,

id\_locatie INT

);

ALTER TABLE magazin ADD FOREIGN KEY (id\_locatie) REFERENCES locatie (id\_locatie);

CREATE TABLE angajat (

id\_angajat INT PRIMARY KEY,

id\_magazin VARCHAR(15),

mail VARCHAR(50),

telefon VARCHAR(15),

nume VARCHAR(25) NOT NULL

);

ALTER TABLE angajat ADD FOREIGN KEY (id\_magazin) REFERENCES magazin(id\_magazin);

ALTER TABLE angajat ADD CONSTRAINT mail\_ang UNIQUE (mail);

ALTER TABLE angajat ADD CONSTRAINT telefon\_ang UNIQUE (telefon);

CREATE TABLE contract (

id\_contract INT PRIMARY KEY,

id\_angajat INT,

data\_angajarii DATE NOT NULL,

functie VARCHAR(255),

salariu NUMBER(8,2),

CONSTRAINT ang\_salariu\_min CHECK (salariu > 0)

);

ALTER TABLE contract

ADD FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat);

CREATE TABLE distribuitor (

id\_distr VARCHAR(15) PRIMARY KEY,

mail VARCHAR(50),

telefon VARCHAR(15)

);

ALTER TABLE distribuitor ADD CONSTRAINT mail\_distr UNIQUE (mail);

ALTER TABLE distribuitor ADD CONSTRAINT telefon\_distr UNIQUE (telefon);

CREATE TABLE aprov\_magazin (

id\_magazin VARCHAR(15),

id\_distr VARCHAR(15),

constraint id\_aprov primary key(id\_magazin, id\_distr),

constraint fk\_aprov\_mag

foreign key(id\_magazin) references magazin(id\_magazin)

on delete cascade,

constraint fk\_aprov\_distr

foreign key(id\_distr) references distribuitor(id\_distr)

on delete cascade

);

CREATE TABLE produs (

id\_produs VARCHAR(15) PRIMARY KEY,

pret NUMBER(6,2) CHECK (pret > 0)

);

CREATE TABLE stoc\_magazin (

id\_magazin VARCHAR(15),

id\_produs VARCHAR(15),

constraint id\_stoc primary key(id\_magazin, id\_produs),

constraint fk\_stoc\_magazin

foreign key(id\_magazin) references magazin(id\_magazin)

on delete cascade,

constraint fk\_stoc\_produs

foreign key(id\_produs) references produs(id\_produs)

on delete cascade

);

CREATE TABLE comanda (

id\_comanda INT PRIMARY KEY,

id\_client INT,

valoare\_totala NUMBER(6,2) CHECK (valoare\_totala >= 0)

);

CREATE TABLE detalii\_comanda (

id\_comanda INT,

id\_produs VARCHAR(15),

nr\_buc INT,

constraint id\_detalii primary key(id\_comanda, id\_produs),

constraint fk\_det\_com

foreign key(id\_comanda) references comanda(id\_comanda)

on delete cascade,

constraint fk\_det\_prod

foreign key(id\_produs) references produs(id\_produs)

on delete cascade

);

CREATE TABLE client (

id\_client INT PRIMARY KEY,

nume VARCHAR(25),

mail VARCHAR(50),

telefon VARCHAR(15)

);

ALTER TABLE client ADD CONSTRAINT mail\_client UNIQUE (mail);

ALTER TABLE client ADD CONSTRAINT telefon\_client UNIQUE (telefon);

INSERT INTO locatie VALUES

(1,'Romania', 'Bucuresti','Calea Vitan',59);

INSERT INTO locatie VALUES

(2,'Romania', 'Oradea','Str. Nufarului',30);

INSERT INTO locatie VALUES

(3,'Romania', 'Cluj-Napoca','Str. Avram Iancu',492);

INSERT INTO locatie VALUES

(4,'Italia', 'Roma','Della Spiga',103);

INSERT INTO locatie VALUES

(5,'Germania', 'Berlin','Augsburg',51);

INSERT INTO magazin VALUES

('NewFashion',1);

INSERT INTO magazin VALUES

('TrendyVibes',2);

INSERT INTO magazin VALUES

('RevolutionAge',3);

INSERT INTO magazin VALUES

('Memoir',4);

INSERT INTO magazin VALUES

('HouseOfGucci',5);

INSERT INTO angajat VALUES

(1,'NewFashion','bogdanoctavian@email.com','0742750707', 'Bogdan Octavian');

INSERT INTO angajat VALUES

(2,'NewFashion','bicuaida@email.com','0722480283', 'Bicu Aida');

INSERT INTO angajat VALUES

(3,'TrendyVibes','lascudan@email.com','0249415901', 'Lascu Dan');

INSERT INTO angajat VALUES

(4,'TrendyVibes','mihaigeorge@email.com','0769658750', 'Mihai George');

INSERT INTO angajat VALUES

(5,'RevolutionAge','papaandreea@email.com','0727944931', 'Popa Andreea');

INSERT INTO angajat VALUES

(6,'RevolutionAge','teodoroemil@email.com','0261840554 ', 'Teodoro Emil');

INSERT INTO angajat VALUES

(7,'Memoir','francescobasso@email.com','+390600004917', 'Francesco Basso');

INSERT INTO angajat VALUES

(8,'Memoir','giovannicontadino@email.com','+390649159995', 'Giovanni Contadino');

INSERT INTO angajat VALUES

(9,'HouseOfGucci','hahndagmar@email.com','05445544137 ','Hahn Dagmar');

INSERT INTO contract VALUES

(1,1,'01-Apr-2022','Manager',4000);

INSERT INTO contract VALUES

(2,2,'06-Apr-2022','Agent de vanzari',3000);

INSERT INTO contract VALUES

(3,3,'10-May-2022','Manager',3500);

INSERT INTO contract VALUES

(4,4,'16-May-2022','Agent de vanzari',2500);

INSERT INTO contract VALUES

(5,5,'11-Jan-2021','Manager',3800);

INSERT INTO contract VALUES

(6,6,'15-Jan-2021','Agent de vanzari',2850);

INSERT INTO contract VALUES

(7,7,'15-Aug-2021','Manager',7000);

INSERT INTO contract VALUES

(8,8,'15-Aug-2021','Agent de vanzari',4000);

INSERT INTO contract VALUES

(9,9,'20-Aug-2021','Manager',7000);

INSERT INTO contract VALUES

(10,9,'24-Aug-2021','Agent de vanzari',4000);

INSERT INTO distribuitor VALUES

('Fashion Place','fshplc@email.com','+5492954684076');

INSERT INTO distribuitor VALUES

('Nelmarc','nelmarc@email.com','+244959150383');

INSERT INTO distribuitor VALUES

('JA Shop','jashop@email.com','+40702085192');

INSERT INTO aprov\_magazin VALUES

('NewFashion','Fashion Place');

INSERT INTO aprov\_magazin VALUES

('TrendyVibes','Fashion Place');

INSERT INTO aprov\_magazin VALUES

('RevolutionAge','Fashion Place');

INSERT INTO aprov\_magazin VALUES

('Memoir','Nelmarc');

INSERT INTO aprov\_magazin VALUES

('HouseOfGucci','JA Shop');

INSERT INTO produs VALUES

('Bluza',54.99);

INSERT INTO produs VALUES

('Rochie',159.99);

INSERT INTO produs VALUES

('Pantaloni',84.99);

INSERT INTO produs VALUES

('Geaca',254.99);

INSERT INTO produs VALUES

('Fusta',64.99);

INSERT INTO stoc\_magazin VALUES

('NewFashion','Bluza');

INSERT INTO stoc\_magazin VALUES

('NewFashion','Rochie');

INSERT INTO stoc\_magazin VALUES

('NewFashion','Pantaloni');

INSERT INTO stoc\_magazin VALUES

('NewFashion','Geaca');

INSERT INTO stoc\_magazin VALUES

('NewFashion','Fusta');

INSERT INTO stoc\_magazin VALUES

('TrendyVibes','Bluza');

INSERT INTO stoc\_magazin VALUES

('TrendyVibes','Rochie');

INSERT INTO stoc\_magazin VALUES

('TrendyVibes','Pantaloni');

INSERT INTO stoc\_magazin VALUES

('TrendyVibes','Fusta');

INSERT INTO stoc\_magazin VALUES

('RevolutionAge','Rochie');

INSERT INTO stoc\_magazin VALUES

('RevolutionAge','Fusta');

INSERT INTO stoc\_magazin VALUES

('Memoir','Rochie');

INSERT INTO stoc\_magazin VALUES

('HouseOfGucci','Bluza');

INSERT INTO stoc\_magazin VALUES

('HouseOfGucci','Rochie');

INSERT INTO stoc\_magazin VALUES

('HouseOfGucci','Pantaloni');

INSERT INTO stoc\_magazin VALUES

('HouseOfGucci','Geaca');

INSERT INTO stoc\_magazin VALUES

('HouseOfGucci','Fusta');

INSERT INTO client VALUES

(1,'Popescu Andrei','popescuandrei@email.com','0702078753');

INSERT INTO client VALUES

(2,'Radu Otilia','raduotilia@email.com','0702070553');

INSERT INTO client VALUES

(3,'Georgia Vasilica','georgianavasilica@email.com','0703497033');

INSERT INTO client VALUES

(4,'Rusu Dragos','rusudragos@email.com','0702015005');

INSERT INTO comanda VALUES

(1,1,0);

INSERT INTO comanda VALUES

(2,2,0);

INSERT INTO comanda VALUES

(3,3,0);

INSERT INTO comanda VALUES

(4,3,0);

INSERT INTO comanda VALUES

(5,4,0);

INSERT INTO detalii\_comanda VALUES

(1,'Bluza',2);

INSERT INTO detalii\_comanda VALUES

(1,'Pantaloni',1);

INSERT INTO detalii\_comanda VALUES

(2,'Geaca',1);

INSERT INTO detalii\_comanda VALUES

(2,'Rochie',1);

INSERT INTO detalii\_comanda VALUES

(3,'Fusta',3);

INSERT INTO detalii\_comanda VALUES

(3,'Bluza',2);

INSERT INTO detalii\_comanda VALUES

(4,'Pantaloni',2);

INSERT INTO detalii\_comanda VALUES

(5,'Rochie',2);

UPDATE comanda

SET valoare\_totala = (SELECT SUM(nr\_buc\*pret)

FROM detalii\_comanda

JOIN produs ON detalii\_comanda.id\_produs = produs.id\_produs

WHERE id\_comanda = comanda.id\_comanda);