

---

**UNIVERSITATEA DIN CRAIOVA**  
**Facultatea de Automatica, Calculatoare si Electronica**  
**Specializarea: Automatica si Informatica Aplicata**  
**Grupa: 2.2**

**PROIECT la disciplina BAZE DE DATE**

**“Moonlight Coffee Shops Chain”**



**Student,**  
**Guran Sebastian-Dieter**

**Coordonator,**  
**Prof. Dr. Ing. Viorel Stoian**

**Craiova, 2023**

---

---

**UNIVERSITATEA DIN CRAIOVA**  
**Facultatea de Automatica, Calculatoare si Electronica**  
**Specializarea: Automatica si Informatica Aplicata**  
**Grupa: 2.2**

**PROIECT la disciplina BAZE DE DATE**

**“Moonlight Coffee Shops Chain”**



**Student,**  
**Guran Sebastian-Dieter**

**Coordonator,**  
**Prof. Dr. Ing. Viorel Stoian**

**Craiova, 2023**

---

# **CUPRINS**

## ***CAPITOLUL I. INTRODUCERE***

## ***CAPITOLUL II. TEMA DE PROIECT***

## ***CAPITOLUL III. SCHEMA CONCEPTUALA***

### ***3.1. Notiuni teoretice***

### ***3.2. Schema conceptuala***

## ***CAPITOLUL IV. SCHEMA LOGICA***

### ***4.1. Notiuni teoretice***

### ***4.2. Schema logica***

## ***CAPITOLUL V. NORMALIZAREA BD***

### ***5.1. Notiuni teoretice***

### ***5.2. Normalizarea tabelelor bazei de date***

## ***CAPITOLUL VI. DENORMALIZAREA BD***

### ***6.1. Notiuni teoretice***

### ***6.2. Denormalizarea bazei de date***

## ***CAPITOLUL VII. SGBD utilizat***

### ***7.1. Notiuni teoretice***

### ***7.2. Exemple de interogari SQL pe baza de date creata***

## ***CAPITOLUL VIII. CONCLUZII***

## ***BIBLIOGRAFIE***

## CAPITOLUL I. INTRODUCERE

Datele reprezintă informații fixate pe un anumit suport fizic în vederea utilizării și prelucrării într-un anumit scop.

Baza de date (data base) este o colecție de date organizate care servește unui anumit scop (nu conține date care nu sunt relevante). Faptul că sunt organizate înseamnă că sunt stocate, reprezentate și accesate într-o manieră consistentă.

Dezvoltarea bazelor de date s-a bazat pe 2 cerințe:

- persistența datelor (datele trebuie să fie valide pentru mai multe rulări),
- simplitatea stocării și manipulării datelor.

Sistemul bazelor de date are 4 nivele:

Nivelul conceptual

Este nivelul fundamental ce descrie într-un mod natural și fără ambiguități sistemul ce urmează a fi modelat.

Nivelul extern

La acest nivel se realizează schema externă care este astfel realizată încât grupuri diferite de utilizatori să acceseze numai anumite subscheme ale schemei conceptuale globale.

Nivelul logic

Pentru o anumită aplicație dată schema conceptuală se convertește într-o structură de nivel inferior unde se alege un model logic adecvat de organizare a datelor .

Nivelul intern

După ce a fost realizată schema logică aceasta se concretizează într-o schemă internă care este specifică sistemului de gestiune a bazelor de date ales.

Un SGBD (Sistem de Gestiune a Bazelor de Date) sau DBMS (DataBase Management System) este un sistem software care gestionează toate procesele dintr-o bază de date și care permite utilizatorului să interacționeze cu aceasta.

Principalele funcțiuni ale unui SGBD sunt:

- stocarea datelor,
- definirea structurilor de date,
- manipularea datelor,
- interogarea (extragerea și prelucrarea) datelor,
- asigurarea securității datelor,
- asigurarea integrității datelor,
- accesul concurent la date cu păstrarea consistenței acestora,
- asigurarea unui mecanism de recuperare a datelor,
- asigurarea unui mecanism de indexare care să permită accesul rapid la date. [1]

## CAPITOLUL II. TEMA DE PROIECT

Recent, managerul unei organizatii a angajat firma ta pentru a proiecta aplicatia de care are nevoie, cu titlul: Moonlight Coffee Shops Chain

Moonlight Coffees este un lanț de cafenele de înaltă calitate în creștere rapidă, cu în prezent peste 500 de magazine în 12 țări ale lumii. Magazinele sunt situate în locații de primă clasă, cum ar fi principalele zone comerciale, de divertisment și de afaceri, aeroporturi, gări și muzee. Moonlight Coffees are aproximativ 9.000 de angajați.

Toate magazinele servesc cafea, ceaiuri, băuturi răcoritoare și diverse tipuri de produse de patiserie. Majoritatea magazinelor vând produse nealimentare, cum ar fi cărți poștale și uneori chiar bilete la teatru. Conducerea magazinului raportează zilnic cifrele de vânzări la sediul central, în moneda locală. Moonlight Coffees folosește o listă internă de cursuri de schimb care se modifică lunar. De la 1 ianuarie 1999, țările Comunității Europene trebuie să raporteze în euro.

## CAPITOLUL III. SCHEMA CONCEPTUALA

### 3.1. Notiuni teoretice

În prima fază, o echipă nominalizată colectează (achiziționează) datele corespunzătoare din sistem, apoi urmează faza de organizare a acestora utilizându-se modelul entitate-legătură. Principalele concepte folosite în acest model sunt: entitatea, relația (legătura) și atributul. Entitatea este un obiect de interes din sistem pentru care trebuie să existe date înregistrate. Observații:

- Fiecare entitate are o denumire unică în cadrul unui sistem.
- Entitățile sunt reprezentate prin substantive, dar nu orice substantiv folosit în descrierea sistemului este entitate, ci numai acelea care au o semnificație deosebită.
- Fiecare entitate trebuie să fie bine definită și precizată pentru a se evita confuziile.

Relația (legătura) este o asociere (raport) nedirecționată între 2 entități.

Cardinalitatea unei relații indică numărul de instanțe din fiecare entitate poate participa la relație. Există 3 tipuri de cardinalitate:

"mulți-la-unu" (many-to-one, M:1). Relația dintre entitățile A și B este de tipul "mulți-la-unu" dacă fiecărei instanțe din A i se poate asocia cel mult o singură instanță din B și fiecărei

instanțe din B i se pot asocia mai multe instanțe din A.

"unu-la-unu" (one-to-one, 1:1). Relația dintre entitățile A și B este de tipul "unu-la-unu" dacă fiecărei instanțe din A i se poate asocia cel mult o singură instanță din B și fiecărei instanțe din B i se poate asocia cel mult o singură instanță din A.

"mulți-la-mulți" (many-to-many, M:M) Relația dintre entitățile A și B este de tipul "mulți-la-unu" dacă fiecărei instanțe din A i se pot asocia mai multe instanțe din B și fiecărei instanțe

din B i se pot asocia mai multe instanțe din A.

Atributul Atributul este o caracteristică a unei entități sau a unei relații. Fiecare entitate are un anumit număr de atribute despre care sunt înregistrate date.

Observații:

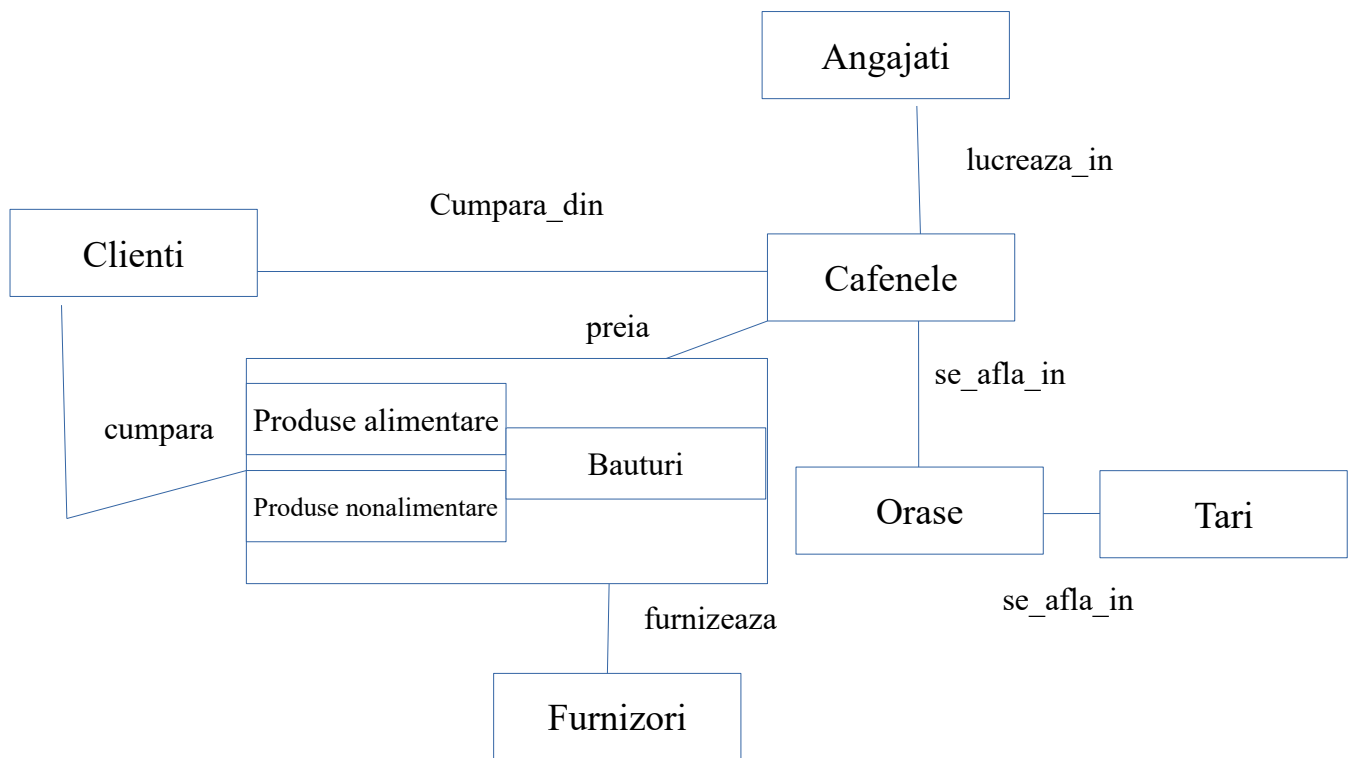
- Numele unui atribut este unic în cadrul unei entități sau al unei relații.
- Atributele sunt întotdeauna substantive, dar nu orice substantiv este atribut.
- Pentru fiecare atribut este necesară o descriere, împreună cu domeniul de valori
- Trebuie evitate atributele indirecte.

Chei primare, naturale, artificiale. Cheia unei entități este un atribut sau set de atribute care identifică în mod unic o instanță a acelei entități (face distincție între oricare 2 rânduri diferite ale tabelului asociat entității). Cheile sunt de 2 feluri: naturale (au semnificație reală pentru entitate) și artificiale (nu au semnificație reală pentru entitate)

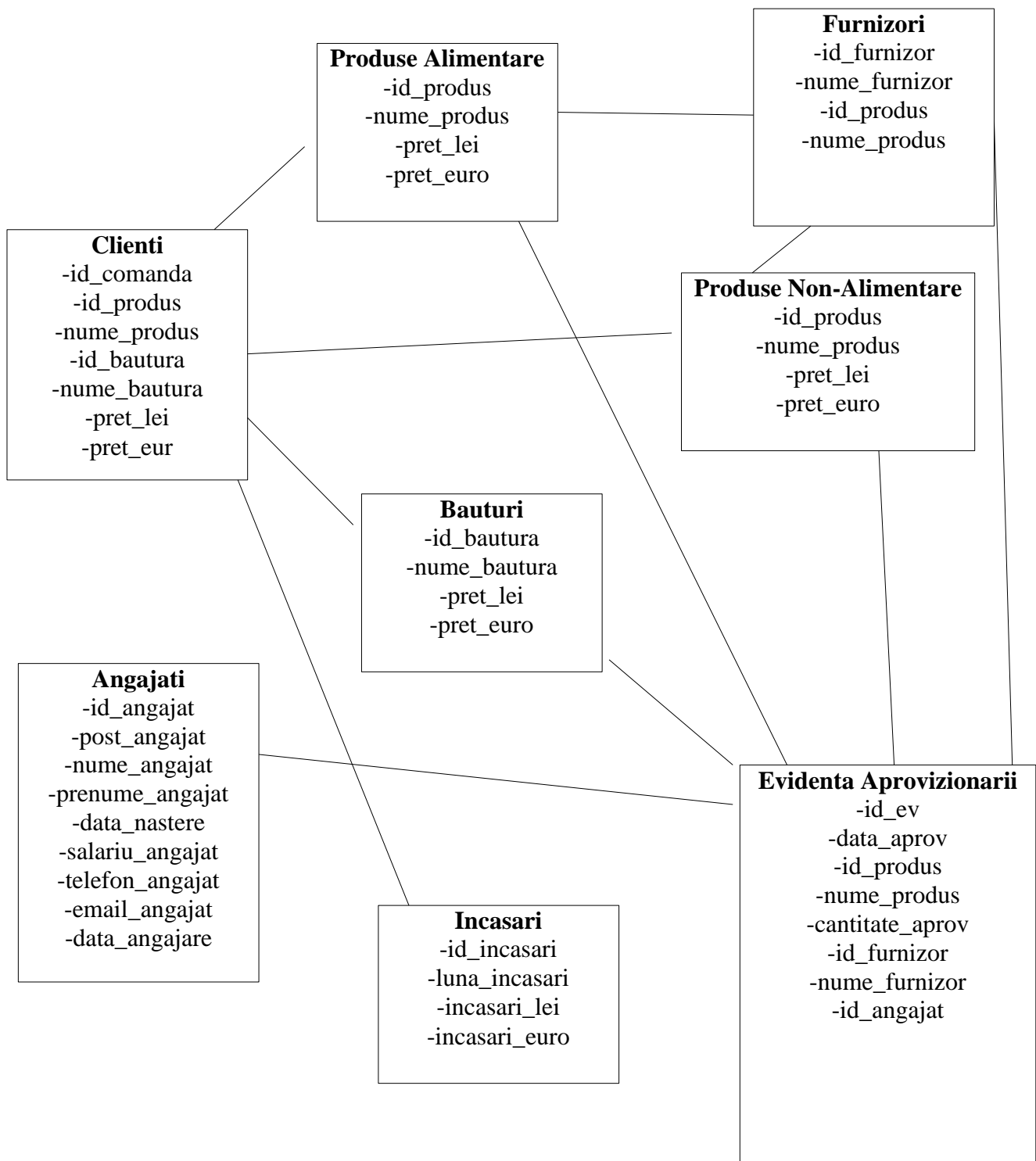
Avantajele cheilor artificiale primare:

- stabilitatea – Cheile primare artificiale se schimbă rar. Schimbarea unei chei primare presupune schimbarea cheilor străine care fac referire la ea.
- simplitatea – Au un număr de atribute și de caractere mai mic.
- Nu prezintă ambiguități în reprezentare
- Elimină apariția valorilor Null.[2]

- In mai multe tari se afla orase
- In orase se afla cafenele
- Cafenelele preiau produsele de la furnizori
- Clientii cumpara produsele din cafenele



### 3.2. Schema conceptuala





## CAPITOLUL IV. SCHEMA LOGICA

### 4.1. Noțiuni teoretice

Pentru realizarea schemei logice a unei baze de date se pornește de la scheme conceptuale (modelul entitate – legătură) urmărindu-se conversia entităților și a legăturilor în tabele relaționale. Regulile de conversie ale entităților, legăturilor și atributelor sunt următoarele:

**Transformarea entităților.** Regulă generală: entitățile se transformă în tabele. **Entitățile independente** devin tabele independente, adică tabele a căror cheie primară nu conține chei străine. **Entitățile dependente** devin tabele dependente (tabele detaliu) adică tabele a căror cheie primară conține cheia străină ce face referința la cheia primară a entității de care depinde entitatea în cauză. **Subentitățile** devin subtabele, adică tabele a căror cheie primară este cheia străină pentru tabelul superentitate.

**Transformarea relațiilor** (legăturilor) Regula generală: Relațiile (legăturile) se convertesc în chei străine.

Convenție: plasamentul cheii străine este simbolizat printr-o săgeată. Atunci când cheia străină este inclusă în cheia primară, vârful săgeții este plin și este gol în caz contrar.

**Relațiile 1:1** devin chei străine. Cheia străină este plasată în tabelul cu linii mai puține.

**Relațiile M:1** devin chei străine plasate în tabelul care se află în partea de “multi” a relației. Cheia străină nu poate avea valoarea Null, iar în cazul entităților dependente ea va face parte chiar din cheia primară a tabelului detaliu. Cheia străină poate avea valoarea Null și nu poate face parte din cheia primară.

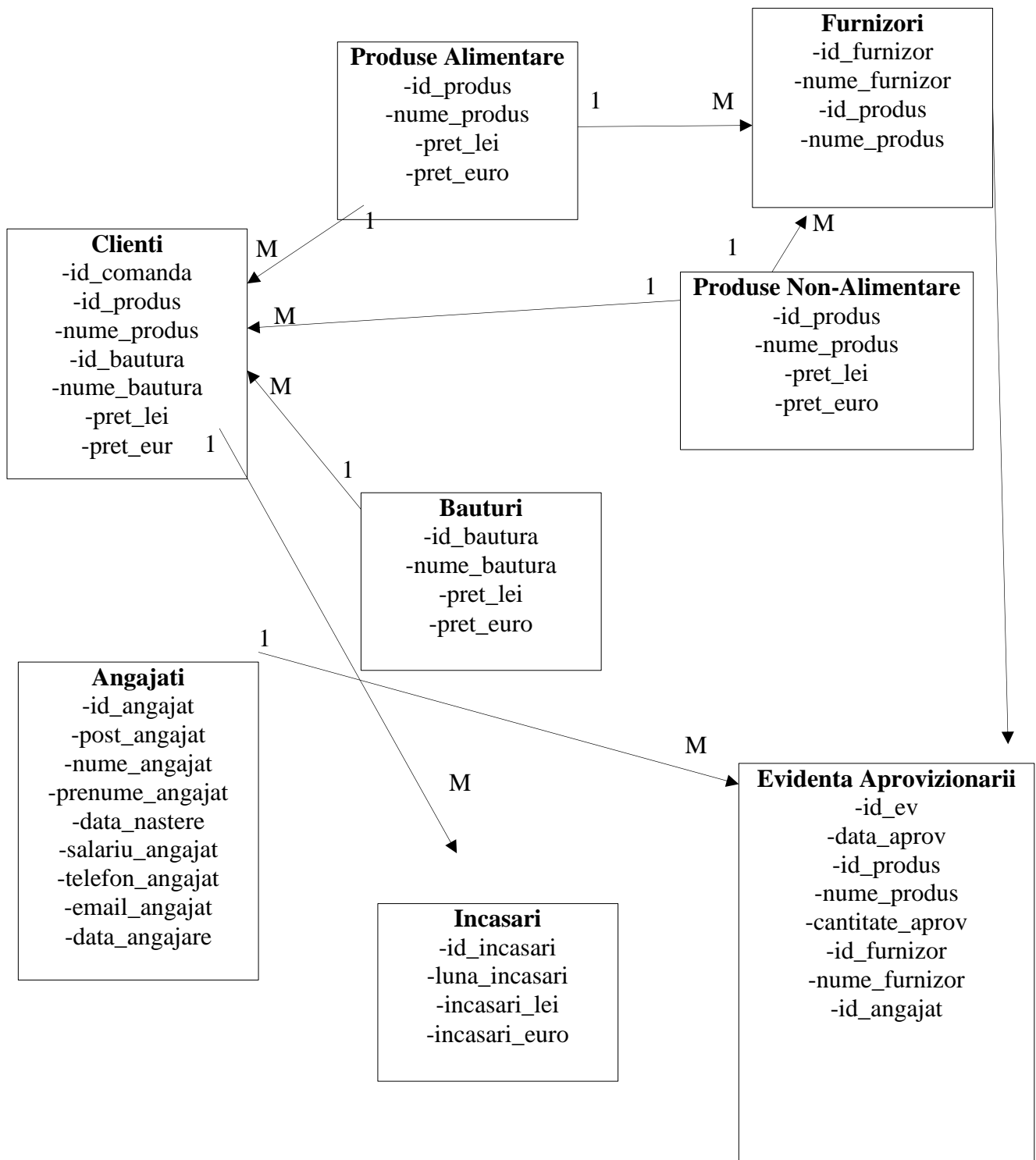
**Relațiile M:M** se transformă în 2 relații M:1. În acest caz se construiește un tabel special numit tabel asociativ care are 2 chei străine care fac referința la cheile primare ale celor 2 tabele aflate în relația M:M. Cheia sa primară este formată din cele 2 chei străine plus (eventual) alte atribute suplimentare.

**O relație de tip 3** se transformă într-un număr de relații de tip 2, egal cu numărul de tabele asociate. Aceste relații (legături) se stabilesc între un tabel asociativ și tabelele asociate. Tabelul asociativ are câte o cheie străină pentru fiecare tabel asociat, iar cheia sa primară este formată din toate aceste chei străine plus (eventual) alte atribute suplimentare.

**Transformarea atributelor** Regula generală: Atributele se convertesc în coloane ale tabelelor provenite din entități sau chiar în tabele. **Atributele simple** ale unei entități devin coloane în tabelul provenit din acea entitate. Toate componentele unui atribut compus devin coloane. **Atributele repetitive** (multivaloare) ale unei entități devin tabele dependente ce conțin fiecare o cheie străină (care face referința la cheia primară a entității) și atributul multivaloare. Cheia primară a unui astfel de nou tabel este formată din cheia străină plus alte coloane suplimentare. **Atributele simple** ale unei relații 1:1 sau M:1 devin coloane ale tabelului care conține cheia străină. Atributele simple ale unei relații M:M vor deveni coloane ale tabelului asociativ. Atributele simple ale unei relații M:M vor deveni coloane ale tabelului asociativ.

**Diagrama logică a BD și tabelele asociat.** Prezentăm diagrama logică a bazei de date pentru sistemul Magazin de închirieri video (ANGAJATI, FURNIZORI, FILME, CLIENTI) care a rezultat în urma transformărilor prezentate mai înainte și efectuate asupra diagramei entitate-legătură.[2]

## 4.2. Schema logica



**furnizori:** (id\_furnizor,nume\_furnizor)

**clienti:**(id\_comanda,id\_produș,nușe\_produș,id\_bautura,nușe\_bautura,pret\_lei, pret\_euro)  
**produse\_alimentare:**(id\_produș,nușe\_produș,pret\_lei,pret\_euro)  
**produse\_nonalimentare:**(id\_produș,nușe\_produș,pret\_lei,pret\_euro)  
**bauturi:**(id\_bautura,nușe\_bautura,pret\_lei,pret\_euro)  
**angajati:**(id\_angajat,post\_angajat,nușe\_angajat,preșume\_angajat,data\_nastere, salariu\_angajat,telefon\_angajat,email\_angajat,data\_angajare)  
**incasari:**(id\_incasari,luna\_incasari,incasari\_lei,incasari\_euro)  
**evidenta\_ąprovizionarii:**(id\_ev,data\_ąprov,id\_produș,nușe\_produș,cantitate\_ąprov,id\_furnizor,nușe\_furnizor,id\_angajat)

## CAPITOLUL V. NORMALIZAREA BD

### 5.1. Notiuni teoretice

Normalizarea reprezinta procesul de descompunere a unui tabel relational in mai multe tabele care satisfac anumite reguli si care stocheaza aceleasi date ca si tabelul initial astfel incat sa fie eliminate redundanta in date si anomalii la actualizare.

**Caracterul reversibil al normalizarii.** Prin caracter reversibil al normalizarii se intelege faptul ca descompunerea se face fara pierdere de informatie, adica tabelul initial poate fi reconstituit prin compunerea naturala atributelor comune, a tabelelor rezultate.

**Regula Casey-Delobel** (caz particular de descompunere fara pierdere de informatie) Fie un tabel  $R(X, Y, Z)$  care se descompune prin proiectie in tabelele  $R_1(X, Y)$  si  $R_2(X, Z)$  unde prin  $X$  notam setul de coloane comune ale tabelelor  $R_1$  si  $R_2$ , iar prin  $Y$  si  $Z$  coloanele specifice lui  $R_1$ , respectiv  $R_2$ . Conditia de descompunere fara pierdere de informatie presupune ca tabelul  $R$  sa fie obtinut prin compunerea naturala a tabelelor  $R_1$  si  $R_2$ .

**Dependenta functionala.** Fie  $R$  un tabel relational si  $X$  si  $Y$  doua submultimi de coloane ale lui  $R$ . Spunem ca  $X$  determina functional pe  $Y$  sau ca  $Y$  depinde functional de  $X$  daca nu exista doua randuri in tabelul  $R$  care sa aiba aceleasi valori pentru coloanele din  $X$  si sa aiba valori diferite pentru cel putin o coloana din  $Y$ .

**Dependenta functionala tranzitiva.** Fie  $R$  un tabel relational,  $X$  o submultime de coloane a lui  $R$  si  $A$  o coloana a lui  $R$ . Spunem ca  $A$  este dependenta tranzitiv de  $X$  daca exista o submultime de coloane  $Y$  care nu include coloana  $A$ .

**Prima forma normala (1NF - First Normal Form).** Un tabel relational este in 1NF daca fiecarei coloane ii corespunde o valoare indivizibila (atomica). Deci orice valoare nu poate fi compusa dintr-o multime de valori (atributele compuse) si nu sunt admise attributele repetitive.

#### **Algoritmul 1NF:**

1. Se inlocuiesc in tabel coloanele corespunzatoare atributelor compuse cu coloane ce contin componentele acestora.
2. Fiecare grup de attribute repetitive se plaseaza in cate un nou tabel.
3. Se introduce in fiecare tabel nou creat la pasul 2 cheia primara a tabelului din care a fost extras atributul respectiv, care devine cheie straina in noul tabel.
4. Se stabileste cheia primara a fiecarui nou tabel creat la pasul 2. Aceasta va fi compusa din cheia straina introdusa la pasul 3 plus una sau mai multe coloane suplimentare.

**A doua forma normala (2NF - Second Normal Form).** Un tabel relational  $R$  este in a doua forma normala (2NF) daca si numai daca:  $R$  este in 1NF, orice coloana care depinde partial de o cheie a lui  $R$  este inclusa in acea cheie. Deci, 2NF nu permite dependentele functionale partiale fata de cheile tabelului, cu exceptia dependentelor triviale, de incluziune.

#### **A treia forma normala (3NF - Third Normal Form)**

Definitie\_1: Un tabel relational  $R$  este in a treia forma normala (3NF) daca si numai daca:  
-  $R$  este in 2NF

- Pentru orice coloana  $A$  necontinuta in nici o cheie a lui  $R$ , daca exista un set de coloane  $X$

astfel încât  $X \in A$ , atunci fie  $X$  conține o cheie a lui  $R$ , fie  $A$  este inclusă în  $X$ .

Obs.: A doua condiție din definiție interzice dependențele funcționale totale față de alte coloane în afara celor care constituie chei ale tabelului. Prin urmare, un tabel este în 3NF dacă orice coloană care nu este conținută într-o cheie, depinde de cheie, de întreaga cheie și numai de cheie. Dacă există astfel de dependențe funcționale trebuie efectuate noi descompuneri ale tabelelor. Ținând cont de definirea noțiunii de dependență funcțională tranzitivă (4.1.) se poate formula:

#### **A 4-a formă normală (4NF – Fourth Normal Form)**

Definiții: Fie  $R$  un tabel relațional,  $X$  și  $Y$  două submulțimi de coloane ale lui  $R$  și  $Z = R - X - Y$  mulțimea coloanelor din  $R$  care nu sunt nici în  $X$  nici în  $Y$ . Spunem că există o dependență multivaloare  $Y$  de  $X$  sau că  $X$  determină multivaloare pe  $Y$  (notația folosită:  $X \twoheadrightarrow Y$ ), dacă, pentru orice valoare a coloanelor lui  $X$ , sunt asociate valori pentru coloanele din  $Y$  care nu sunt corelate în nici un fel cu valorile coloanelor lui  $Z$ .

#### **A 5-a formă normală (5NF – Fifth Normal Form)**

Se întâlnește destul de rar în practică și are rolul de a elimina relațiile M:M dependente care pot introduce redundanțe în date.

Regulile de descompunere stabilite pentru formele normale 1NF-4NF permitteau descompunerea prin proiecție a unui tabel relațional  $R$  în 2 tabele relaționale  $R_1$  și  $R_2$ . Există însă tabele relaționale care nu pot fi descompuse în 2 tabele, ci în 3 sau mai multe, fără

pierdere de informație. Astfel de situații sunt tratate de 5NF. S-a introdus conceptul de join-dependență sau dependență la compunere sau dependență de uniune.

**Forma normală Boyce-Codd (BCNF – Boyce-Codd Normal Form).** Un tabel relațional  $R$  este în forma normală Boyce-Codd (BCNF) doar dacă pentru orice dependență funcțională totală  $X \twoheadrightarrow A$ , unde  $X$  este un subset de coloane ale lui  $R$ , iar  $A$  este o coloană neconținută în  $X$ ,  $X$  este o cheie a lui  $R$ . [4]

## **5.2. Normalizarea tabelelor bazei de date**

Tabelele rezultate în urma proiectării bazei de date respectă toate cele 3 forme (FN1, FN2, FN3), astfel nu este necesar procesul de normalizare.

**Tabelul Angajati**

ID_ANGAJAT	POST_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	DATA_NASTERE	SALARIU_ANGAJAT	TELEFON_ANGAJAT	EMAIL_ANGAJAT	DATA_ANGAJARE
3	Store Manager	Guran	Sebastian	26-JAN-02	4000	0764450016	s.dieter	03-APR-22
4	Asistent Manager	Greere	Ionut	15-APR-02	4000	0721345089	q.cristi	12-MAY-22
5	Barista	Enache	Cristina	14-JAN-02	2500	0720981744	scriti.enache	20-OCT-21
6	Barista	Voinea	Alexandru	09-JUN-02	2500	0744732911	vo.alex	21-AUG-22
7	Contabil	Martin	Byrde	09-MAY-89	6000	0251498597	m.byrde	10-SEP-16
8	Vanzator	Popescu	Marcel	19-MAR-96	2300	0784323798	marcel.popi	01-FEB-18
9	Vanzator	Matei	Carmen	13-AUG-00	2300	0721698724	carmelita.matei	15-MAY-19

**Tabelul Furnizori**

ID_FURNIZOR	NUME_FURNIZOR	ID_PRODUS	NUME_PRODUS
1	Organical Coffee	1	Cafea Arabica
2	Organical Coffee	2	Cafea Etiopia
3	Laptarescu	21	Lapte degresat
4	Laptarescu	22	Lapte de migdale
5	Laptarescu	23	Lapte de soia
6	Laptarescu	24	Lapte de cocos
7	Papetaria.ro	10	Stilou
8	Papetaria.ro	13	Agenda
9	Ceramic Vibe	33	Cana emailata

**Tabelul Produse Alimentare**

ID_PRODUS	NUME_PRODUS	PRET_LEI	PRET_EURO
1	Croissant	10	2
2	Senvis bacon	20	4
3	Senvis cu ou	15	3
4	Bagel	10	2

**Tabelul Produse Non-Alimentare**

ID_PRODUS	NUME_PRODUS	PRET_LEI	PRET_EURO
1	Agenda	30	6
2	Stilou	100	20
3	Cana emailata	60	12

**Tabelul Bauturi**

ID_BAUTURA	NUME_BAUTURA	PRET_LEI	PRET_EURO
1	Cappucino	15	3
2	Espresso	12	2
3	Americano	10	2
4	Cafe Latte	13	3
5	Ciocolata calda	10	2
6	Ceai fierbinte	10	2

**Tabelul Clienti**

ID_COMANDA	ID_PRODUS	NUME_PRODUS	ID_BAUTURA	NUME_BAUTURA	PRET_LEI	PRET_EURO
2	1	agenda	1	Cappucino	45	9
3	1	croissant	1	Cappucino	25	5
7	(null)	(null)	5	Ciocolata calda	10	2

**Tabelul Incasari**

ID_INCASARI	LUNA_INCASARI	INCASARI_LEI	INCASARI_EURO
1	Ianuarie	1500000	300000
2	Februarie	1200000	240000
3	Martie	1350000	270000
4	Aprilie	1320000	264000
5	Mai	1455000	291000

**Tabelul Evidenta Aprovizionarii**

ID_EV	DATA_APROV	ID_PRODUS	NUME_PRODUS	CANTITATE_APROV	ID_FURNIZOR	NUME_FURNIZOR	ID_ANGAJAT
5	21-OCT-22	1	Cafea Arabica	100	1	Organical Coffee	3
6	21-OCT-22	2	Cafea Etiopia	100	2	Organical Coffee	3

## CAPITOLUL VI. DENORMALIZAREA BD

**6.1. Notiuni teoretice.** Denormalizarea unei BD reprezinta procesul invers operatiei de normalizare si duce la cresterea redundantei datelor. Prin aceasta se doreste, in principal, cresterea performantei si simplificarea programelor de interogare a datelor.

Obs.: - Denormalizarea se face numai dupa o normalizare corecta.

- Denormalizarea se face printr-o selectare strategica a structurilor care aduc avantaje semnificative

- Denormalizarea trebuie insotita de masuri suplimentare de asigurare a integritatii datelor.

**Cresterea performantei.** Un caz frecvent intalnit in interogarea BD este cazul unor operatii sau calcule foarte des utilizate.

**Cresterea performantei.** Un caz frecvent intalnit in interogarea BD este cazul unor operatii sau calcule foarte des utilizate.

Denormalizarea este utilizată pe scară largă, în special în contexte precum depozitul de date. În acest context, în care viteza de răspuns a bazei de date la o interogare are o importanță mai mare decât organizarea datelor, este util să existe o denormalizare a tabelelor pentru a evita realizarea de imbinari suplimentare. Denormalizarea este utilizată în general prin transformarea relațiilor ierarhice unu-la-multe într-o singură relație și apoi într-un singur tabel.

De exemplu, în cazul seriilor temporale, datele după ce au fost normalizate (adică fixate într-un interval care se extinde de la 1 la infinit, în funcție de perioada luată în considerare), sunt returnate la forma lor originală.[4]

### **6.2. Denormalizarea bazei de date**

Deoarece beneficiarul bazei de date nu solicita cerinte sau specificatii, nu este necesar procesul de denormalizare a tabelelor.

#### **Tabelul Angajati**

```
CREATE TABLE Angajati_Moonlight(  
id_angajat number(3) primary key,  
post_angajat varchar2(30),  
nume_angajat varchar2(30),  
prenume_angajat varchar2(30),  
data_nastere date,  
salariu_angajat number(4),  
telefon_angajat varchar2(30),  
email_angajat varchar2(30),  
data_angajare date);  
create sequence PK_angajat start with 1 increment by 1;  
insert into Angajati_Moonlight values(PK_angajat.nextval, 'Store Manager', 'Guran', 'Sebastian',  
'26.January.2002', '4000', '0764450016', 's.dieter','03.April.2022');  
insert into Angajati_Moonlight values(PK_angajat.nextval, 'Asistent Manager', 'Greere', 'Ionut',  
'15.April.2002', '4000', '0721345089', 'g.cristi','12.May.2022');  
insert into Angajati_Moonlight values(PK_angajat.nextval, 'Barista', 'Enache', 'Cristina',  
'14.January.2002', '2500', '0720981744', 'scriti.enache','20.October.2021');
```



```

insert into Angajati_Moonlight values(PK_angajat.nextval, 'Barista', 'Voinea', 'Alexandru',
'09.June.2002', '2500', '0744732911', 'vo.alex','21.August.2022');
insert into Angajati_Moonlight values(PK_angajat.nextval, 'Contabil', 'Martin', 'Byrde',
'09.May.1989', '6000', '0251498597', 'm.byrde','10.September.2016');
insert into Angajati_Moonlight values(PK_angajat.nextval, 'Vanzator', 'Popescu', 'Marcel',
'19.March.1996', '2300', '0784323798', 'marcel.popi','01.February.2018');
insert into Angajati_Moonlight values(PK_angajat.nextval, 'Vanzator', 'Matei', 'Carmen',
'13.August.2000', '2300', '0721698724', 'carmelita.matei','15.May.2019');

```

### **Tabelul Furnizori**

```

CREATE TABLE Furnizori(
id_furnizor number(3) primary key,
nume_furnizor varchar2(30),
id_produs number(3),
nume_produs varchar2(30));
create sequence PK_furnizori start with 1 increment by 1;
insert into furnizori values(PK_furnizori.nextval, 'Organical Coffee', '1','Cafea Arabica');
insert into furnizori values(PK_furnizori.nextval, 'Organical Coffee', '2','Cafea Etiopia');
insert into furnizori values(PK_furnizori.nextval, 'Laptarescu', '21','Lapte degresat');
insert into furnizori values(PK_furnizori.nextval, 'Laptarescu', '22','Lapte de migdale');
insert into furnizori values(PK_furnizori.nextval, 'Laptarescu', '23','Lapte de soia');
insert into furnizori values(PK_furnizori.nextval, 'Laptarescu', '24','Lapte de cocos');
insert into furnizori values(PK_furnizori.nextval, 'Papetaria.ro', '10','Stilou');
insert into furnizori values(PK_furnizori.nextval, 'Papetaria.ro', '13','Agenda');
insert into furnizori values(PK_furnizori.nextval, 'Ceramic Vibe', '33','Cana emailata');

```

### **Tabelul Produse\_Alimentare**

```

id_produs number(3) primary key,
nume_produs varchar2(30),
pret_lei number(3),
pret_euro number(3));
create sequence PK_produsealimentare start with 1 increment by 1;
insert into produse_alimentare values(PK_produsealimentare.nextval, 'Croissant',
'10','2');
insert into produse_alimentare values(PK_produsealimentare.nextval, 'Senvis bacon',
'20','4');
insert into produse_alimentare values(PK_produsealimentare.nextval, 'Senvis cu ou',
'15','3');
insert into produse_alimentare values(PK_produsealimentare.nextval, 'Bagel', '10','2');

```

### **Tabelul Produse\_NonAlimentare**

```

id_produs number(3) primary key,
nume_produs varchar2(30),
pret_lei number(3),
pret_euro number(3));
create sequence PK_produsenonalimentare start with 1 increment by 1;
insert into produse_nonalimentare values(PK_produsenonalimentare.nextval, 'Agenda',
'30','6');

```

```
insert into produse_nonalimentare values(PK_produsenonalimentare.nextval, 'Stilou',
'100','20');
insert into produse_nonalimentare values(PK_produsenonalimentare.nextval, 'Cana
emailata', '60','12');
```

### **Tabelul Bauturi**

```
id_bautura number(3) primary key,
nume_bautura varchar2(30),
pret_lei number(3),
pret_euro number(3));
create sequence PK_bauturi start with 1 increment by 1;
insert into bauturi values(PK_bauturi.nextval, 'Cappucino', '15','3');
insert into bauturi values(PK_bauturi.nextval, 'Espresso', '12','2');
insert into bauturi values(PK_bauturi.nextval, 'Americano', '10','2');
insert into bauturi values(PK_bauturi.nextval, 'Cafe Latte', '13','3');
insert into bauturi values(PK_bauturi.nextval, 'Ciocolata calda', '10','2');
insert into bauturi values(PK_bauturi.nextval, 'Cei fierbinte', '10','2');
```

### **Tabelul Clienti**

```
CREATE TABLE clienti_moonlight(
id_comanda number(3) primary key,
id_produș number(3),
nume_produș varchar2(30),
id_bautura number(3),
nume_bautura varchar2(30),
pret_lei number(3),
pret_euro number(3),
FOREIGN KEY (id_produș) REFERENCES produse_alimentare (id_produș) ON
DELETE CASCADE,
FOREIGN KEY (id_produș) REFERENCES produse_nonalimentare(id_produș) ON
DELETE CASCADE,
FOREIGN KEY (id_bautura) REFERENCES bauturi (id_bautura) ON DELETE
CASCADE);
create sequence PK_clienti start with 1 increment by 1;
insert into clienti_moonlight values(PK_clienti.nextval, '1', 'agenda','1',
'Cappucino','45','9');
insert into clienti_moonlight values(PK_clienti.nextval, '1', 'croissant','1',
'Cappucino','25','5');
insert into clienti_moonlight values(PK_clienti.nextval, NULL, NULL,'5','Ciocolata
calda','10','2');
```

### **Tabelul Incasari**

```
create table incasari_moonlight(
id_incasari number(5) primary key,
luna_incasari varchar2(30),
incasari_lei number(10),
incasari_euro number(10));
create sequence PK_incasari start with 1 increment by 1;
```

```

insert into incasari_moonlight values(PK_incasari.nextval, 'Ianuarie','1500000','300000');
insert into incasari_moonlight values(PK_incasari.nextval,
'Februarie','1200000','240000');
insert into incasari_moonlight values(PK_incasari.nextval, 'Martie','1350000','270000');
insert into incasari_moonlight values(PK_incasari.nextval, 'Aprilie','1320000','264000');
insert into incasari_moonlight values(PK_incasari.nextval, 'Mai','1455000','291000');

```

### **Tabelul Evidenta Aprovizionarii**

```

id_ev number(3) primary key,
data_aprov date,
id_produs number(3) not null,
nume_produs varchar2(30),
cantitate_aprov number(5),
id_furnizor number(3) not null,
nume_furnizor varchar2(30),
id_angajat number(3) not null,
FOREIGN KEY (id_furnizor) REFERENCES furnizori (id_furnizor) ON DELETE
CASCADE,
FOREIGN KEY (id_angajat) REFERENCES angajati_moonlight (id_angajat) ON
DELETE CASCADE,
FOREIGN KEY (id_produs) REFERENCES produse_nonalimentare (id_produs) ON
DELETE CASCADE);
create sequence PK_evidenta_aprovizionarii start with 1 increment by 1;
insert into evidenta_aprovizionarii values(PK_evidenta_aprovizionarii.nextval,
'21.October.2022', '1', 'Cafea Arabica', '100','1','Organical Coffee','3');
insert into evidenta_aprovizionarii values(PK_evidenta_aprovizionarii.nextval,
'21.October.2022', '2', 'Cafea Etiopia', '100','2','Organical Coffee','3');

```

## CAPITOLUL VII. SGBD utilizat

### 7.1. Noțiuni teoretice

Un SGBD (Sistem de Gestionare a Bazelor de Date) sau DBMS (DataBase Management System) este un sistem software care gestionează toate procesele dintr-o bază de date și care permite utilizatorului să interacționeze cu aceasta.

Principalele funcțiuni ale unui SGBD sunt:

- stocarea datelor,
- definirea structurilor de date,
- manipularea datelor,
- interogarea (extragerea și prelucrarea) datelor,
- accesul concurrent la date cu păstrarea consistenței acestora,
- asigurarea unui mecanism de recuperare a datelor,
- asigurarea unui mecanism de indexare care să permită accesul rapid la date.

După modelul folosit există mai multe tipuri de SGBD:

SGBD ierarhic - Modelul ierarhic stochează datele în structuri de tip arbore.

SGBD rețea - Datele sunt stocate sub formă de înregistrări cu legături multiple și complexe între ele.

Caracteristicile principale ale SGBD rețea sunt:

- reprezentare date complexe
- extrem de puțin flexibil
- design extrem de complicat

SGBD relational - Reprezintă cea mai simplă structură pe care o poate avea o bază de date. Datele sunt organizate în tabele formate din înregistrări și câmpuri. În acest caz bazele de date relaționale sunt foarte flexibile și ușor de mânuit.

SGBD orientat pe obiect - Este tipul cel mai nou de SGBD fiind introdus conceptul de obiect. Integrează principiile programării orientate pe obiect (C++, Actor etc.) cu cele ale bazelor de date.

În cazul existenței mai multor utilizatori, un SGBD trebuie să gestioneze accesul curent al acestora la date, menținând în același timp integritatea bazei de date.

a) Când două sau mai multe persoane vor să vizualizeze aceleași date fără a le modifica însă, totul este în ordine și nu trebuie luate măsuri suplimentare.

b) Când cel puțin o persoană dorește să facă modificări asupra unor date care, în același timp, sunt vizualizate de alte persoane, atunci SGBD trebuie să realizeze și să stocheze mai multe copii ale datelor și să transfere toate modificările copiilor într-o singură versiune atunci când întreaga operațiune este terminată.

c) În cazul când mai multe persoane încearcă să modifice aceleași date în același timp SGBD utilizează metoda blocării (locking). Utilizatorul care a efectuat primul modificarea datelor le blochează, ceilalți utilizatori neputându-le modifica până ce operația efectuată de acesta nu este încheiată. În Oracle blocarea se poate face la nivel de tabel sau la nivel de rând. Cu cât unitatea de date este mai mică cu atât concurența este mai eficientă și utilizatorii așteaptă mai puțin. Oracle blochează în mod implicit orice rând asupra căruia se execută o operație de modificare.

#### **Tranzacții**

Întregul mecanism care gestionează concurența într-o bază de date are la bază conceptul de tranzacție. Tranzacția este cea mai mică unitate de lucru. Nu poate exista o operație mai mică decât o tranzacție. Toate tranzacțiile trebuie să fie atomice adică fiecare tranzacție, fie este complet executată, fie nu este executată deloc. Când o tranzacție este încheiată se spune că

este permanentizată (committed); când nu poate fi încheiată și trebuie anulată se spune că este derulată înapoi (rolled back). Derularea înapoi poate fi făcută explicit, printr-o comandă a utilizatorului sau poate fi executată în mod automat, în cazul unei probleme în funcționarea sistemului. În Oracle, o tranzacție poate cuprinde una sau mai multe operații de interogare sau manipulare a datelor și doar cel mult o operație de definire a datelor. Un SGBD dispune de un administrator de tranzacții (transaction manager).

### Baze de date distribuite

Baza de date distribuită reprezintă o mulțime de date corelate logic, dar distribuite pe mai multe mașini interconectate printr-o rețea de comunicație. Din punct de vedere al utilizatorului o bază de date distribuită reprezintă o singură bază de date. Acesta are acces la niște date fără să cunoască localizarea lor. Avantajele unei baze de date distribuite sunt:

- administrare distribuită, descentralizată a bazei de date,
- ușurință în administrare,
- eficiență în exploatare,
- localizarea mai rapidă a defectiunilor. [1]

### 7.2. Exemple de interogari SQL pe baza de date creata

1. Afișați persoanele care au caracterul ,a' pe a treia poziție în Nume;

select \* from angajati\_moonlight where nume\_angajat like '\_\_a%';

ID_ANGAJAT	POST_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	DATA_NASTERE	SALARIU_ANGAJAT	TELEFON_ANGAJAT	EMAIL_ANGAJAT	DATA_ANGAJARE
5	Barista	Enache	Cristina	14-JAN-02	2500	0720981744	scriti.enache	20-OCT-21

2. Afișați toți angajații care au fost angajați după 15-06-2022;

select \*from angajati\_moonlight where data\_angajare > '15.June.2022';

ID_ANGAJAT	POST_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	DATA_NASTERE	SALARIU_ANGAJAT	TELEFON_ANGAJAT	EMAIL_ANGAJAT	DATA_ANGAJARE
6	Barista	Voinea	Alexandru	09-JUN-02	2500	0744732911	vo.alex	21-AUG-22

3.Ordoneza numele angajatilor dupa nume si prenume;

select \*from angajati\_moonlight order by nume\_angajat, prenume\_angajat;

ID_ANGAJAT	POST_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	DATA_NASTERE	SALARIU_ANGAJAT	TELEFON_ANGAJAT	EMAIL_ANGAJAT	DATA_ANGAJARE
5	Barista	Enache	Cristina	14-JAN-02	2500	0720981744	scriti.enache	20-OCT-21
4	Asistent Manager	Greere	Ionut	15-APR-02	4000	0721345089	g.cristi	12-MAY-22
3	Store Manager	Guran	Sebastian	26-JAN-02	4000	0764450016	s.dieter	03-APR-22
7	Contabil	Martin	Byrde	09-MAY-89	6000	0251498597	m.byrde	10-SEP-16
9	Vanzator	Matei	Carmen	13-AUG-00	2300	0721698724	carmelita.matei	15-MAY-19
8	Vanzator	Popescu	Marcel	19-MAR-96	2300	0784323798	marcel.popi	01-FEB-18
6	Barista	Voinea	Alexandru	09-JUN-02	2500	0744732911	vo.alex	21-AUG-22

## **CAPITOLUL VIII. CONCLUZII**

Baza de date creata corespunde criteriilor impuse de catre beneficiar, asigurand un mod optim de stocare si regasire a datelor aferente inchirierii de filme in cadrul magazinului de inchirieri video.

In urma analizelor si discutiilor am reusit sa cream o baza de data care sa gestioneze inchirierea de filme. Baza de date a fost normalizata si denormalizata pentru a optimiza timpul de executie al interogarilor, dimensiunea bazei de date si implicit consumul de resurse sistem.

Sistemul de gestiune a bazelor de date propus a fi folosit, sql, asigura o exploatare optima a bazelor de date, facand posibila consultarea datelor introduse, chiar si in cazul util unei rate mici de transfer (conexiune internet).

## **BIBLIOGRAFIE**

1. Viorel Stoian, "BD-Notițe de curs" \_Capitolul 1 \_Introducere.pdf
2. Viorel Stoian, "BD-Notițe de curs" \_Capitolul 2 \_Proiectarea bazelor de date relationale.pdf
3. Viorel Stoian, "BD-Notițe de curs" \_Capitolul 3 \_Modelul relational.pdf
4. Viorel Stoian, "BD-Notițe de curs" \_Capitolul 4 \_Normalizarea bazei de date.pdf
5. [Moonlight Coffee Shops Chain](#)

=