



**UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” IAȘI**  
**FACULTATEA AUTOMATICĂ ȘI CALCULATOARE**  
**SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI**

**DISCIPLINA BAZE DE DATE - PROIECT**

## ***Gestiunea unui magazin cu articole pentru animale de companie***

**Coordonator,  
Mironeanu Cătălin**

**Student,  
Pandeleanu Cosmin-Constantin**

**Iași, 2021**

## Titlu proiect : Gestiunea unui magazin cu articole pentru animale de companie

Analiza, proiectarea și implementarea unei baze de date care modelează activitatea unui magazin cu privire la gestiunea produselor și a clienților (un caz particular îl constituie clienții care nu au card de fidelitate).

### Descrierea cerințelor și modul de organizare al proiectului

Volumul mare de informații existente în cazul unui magazin cu o gamă foarte variată de produse și numeroși clienți determină necesitatea fluidizării fluxurilor de date ce traversează sistemele informaționale ale magazinului, gestiunea acestora fiind destul de grea.

Informațiile de care avem nevoie sunt cele legate de:

- **clienți:** datele referitoare la clienți se află în tabelele *clienți* și respectiv în *detalii\_clienți*, datele care ne interesează cel mai mult sunt numele clientului, numărul cardului de fidelitate care se află în tabela *clienți*, iar datele referitoare la adresa de e-mail, data nașterii, genul, orașul și adresa la care locuiește se află în tabelă *detalii\_clienți*.

Constrângerile folosite sunt:

- de tip primary key pe *nr\_card* pentru a avea un număr de card unic și pentru a putea face legătura între cele două tabele, *clienți* și *detalii\_clienți*.
- de tip check pe *nume\_client* pentru a verifica că lungimea numelui clientului este mai mare de 3 litere și că numele introdus nu conține cifre.
- de tip unique pe *nume\_client* pentru ca numele să fie unic, să nu existe două persoane cu același nume.
- de tip check pe *e-mail* pentru a verifica că o adresă de e-mail introdusă este validă.
- de tip unique pe *e-mail* pentru a nu exista două persoane cu adrese de e-mail identice.
- de tip foreign key pentru *nr\_card* pentru a face legăturile dintre tabelele *clienți* și *detalii\_clienți*.

- **produse:** datele referitoare la produse se află în tabela *produse*, informațiile asociate sunt id-ul produsului, denumirea produsului, stocul produsului respectiv, prețul de vânzare, unitatea de măsură asociată produsului, tipul produsului și furnizorul.

Constrângerile folosite sunt:

- de tip primary key pe *id\_produs*
- de tip check pentru *stoc* și *preț* pentru a nu avem valori negative.
- de tip check pe *denumire\_produs* pentru a verifica că denumirea are mai mult de două caractere.
- de tip check pentru a verifica că unitatea de măsură *um* se află în lista de valori impusă.
- de tip foreign key pentru *id\_furnizor* și *id\_tip* pentru a face legăturile cu tabele *furnizori* și *tipuri\_produse*.

- **furnizori:** datele referitoare la furnizorii magazinului se află în tabela *furnizori*, informațiile asociate sunt id-ul furnizorului și denumirea furnizorului.

Constrângerile folosite sunt:

- de tip primary key pe *id\_furnizor*.
- de tip unique pe *denumire\_furnizor*, pentru a nu avea mai mulți furnizori cu aceeași denumire.
- de tip check pe *denumire\_furnizor* pentru a verifica că lungimea denumirii furnizorului este mai mare de 3 litere și că nu conține cifre.

- **tipuri de produse:** datele referitoare la tipurile de produse disponibile în magazin se afla în tabela *tipuri\_produce*, informațiile asociate sunt id-ul tipului și denumirea tipului de produs.

Constrângerile folosite sunt:

- de tip primary key pe *id\_tip*.
- de tip unique pe *denumire\_tip*, pentru a nu avea mai multe tipuri cu aceeași denumire.
- de tip check pe *denumire\_tip* pentru a verifica că lungimea denumirii tipului este mai mare de 3 litere și că nu conține cifre.

- **categorii de animale:** datele referitoare la categoriile de animale pentru care sunt disponibile produsele din magazin se află în tabela *categorii\_animale* și respectiv în tabela *categorie*, informațiile asociate tabelii *categorii\_animale* sunt id-ul categoriei și denumirea categoriei de animale, iar în tabela *categorie* informațiile asociate sunt id-ul categoriei și id-ul produsului.

Constrângerile folosite sunt:

- de tip primary key pe *id\_categorie*.
- de tip unique pe *denumire\_categorie*, pentru a nu avea mai multe categorii cu aceeași denumire.
- de tip check pe *denumire\_categorie* pentru a verifica că lungimea categoriei este mai mare de 3 litere și că nu conține cifre.
- de tip foreign key pentru *id\_categorie* și *id\_produs* pentru a face legăturile cu tabele *categorii\_animale* și *produse*.

- **vânzări:** datele referitoare la produsele vândute se află în tabela *vânzări*, informațiile asociate sunt numărul bonului, numărul cardului de fidelitate, id-ul produsului, cantitate de produs cumpărată și data achiziției.

Constrângerile folosite sunt:

- de tip primary key pe *nr\_bon*.
- de tip foreign key pentru *nr\_card* și *id\_produs* pentru a face legăturile cu tabele *clienți* și *produse*.
- constrângere implementată cu un trigger pentru verificarea datei calendaristice *data\_achizitiei*, pentru care se emite bonul. Bonul nu se poate emite înainte de data înființării magazinului.

Toate primary key-urile sunt generate de baza de date printr-un mecanism de tip autoincrement. Majoritatea pornesc de la valoarea 1, cu excepția primary key-ului pentru *id\_produs* care pornește de la 1000.

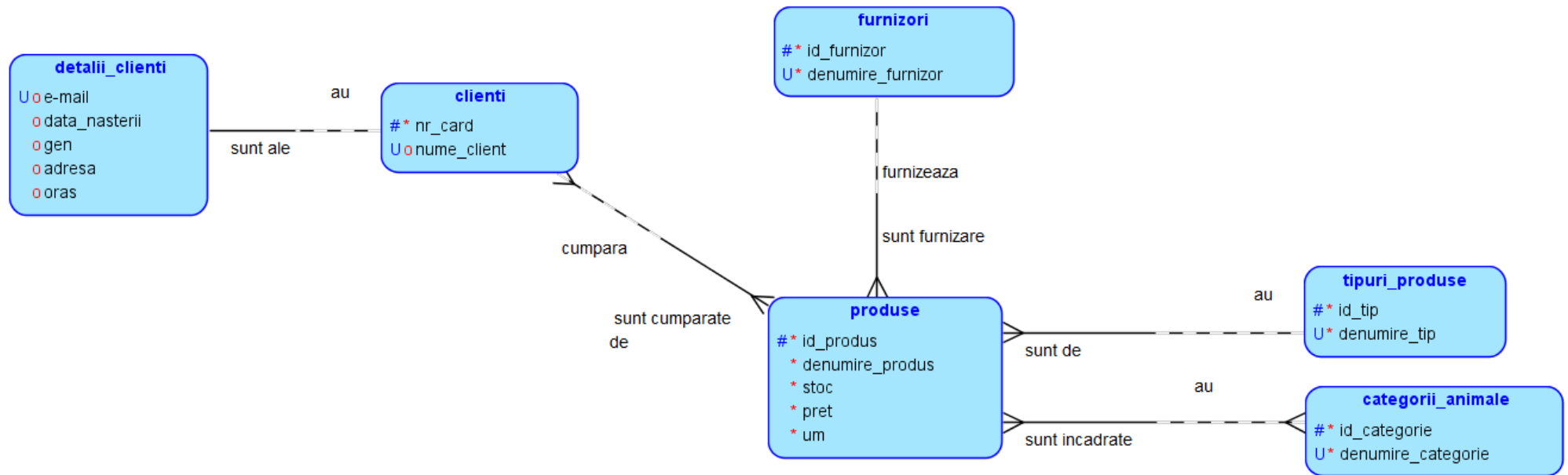
## Descrierea funcțională a aplicației

Principalele funcții care se pot întâlni într-un magazin sunt:

- ✓ Evidența produselor din magazin
- ✓ Evidența clienților
- ✓ Evidența tranzacțiilor efectuate

## Diagrama Entitate Relatie

### Modelul Logic



Entitățile din această aplicație sunt:

- ✓ clienți ;
- ✓ categorii\_animale ;
- ✓ detalii\_clienți ;
- ✓ furnizori ;
- ✓ produse ;
- ✓ tipuri\_produce ;

## Descrierea detaliată a entităților și a relațiilor dintre ele

În proiectarea acestei baze de date s-au identificat tipurile de **relații** 1:1, 1: n și n : n.

Între entitățile **clienți** și **detalii\_clienți** se întânește o relație de tip **one-to-one** deoarece unui client din entitatea *clienți* îi corespunde un singur set de detalii din entitatea *detalii\_clienți*. Legătura dintre cele două entități este realizată prin câmpul *nr\_card*.

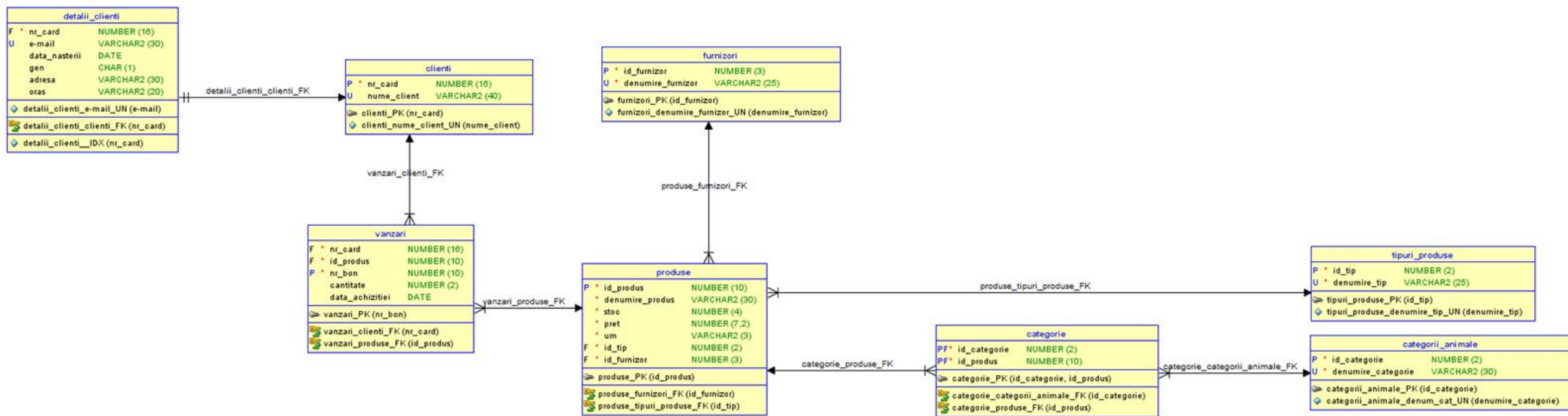
Între entitățile **furnizori** și **produse** se întâlnește o relație de tip **one-to-many** deoarece un furnizor poate furniza mai multe produse, iar un produs este furnizat de către un singur furnizor. Legătura dintre cele două entități este realizată prin câmpul *id\_furnizor*.

Între entitățile **tipuri\_produse** și **produse** se întâlnește o relație de tip **one-to-many** deoarece un produs poate fi asociat unui singur tip, iar un tip de produs are mai multe produse asociate. Legătura dintre cele două entități este realizată prin câmpul *id\_tip*.

Între entitățile **clienți** și **produse** se întâlnește o relație de tip **many-to-many** deoarece un client poate cumpăra mai multe produse, iar un produs poate fi cumpărat de mai mulți clienți. Legătura dintre cele două entități este realizată prin câmpurile *nr\_card* și *id\_produs*.

Între entitățile **categorii\_animale** și **produse** se întâlnește o relație de tip **many-to-many** deoarece un produs poate fi destinat mai multor categorii de animale, iar o categorie de animale are mai multe produse asociate. Legătura dintre cele două entități este realizată prin câmpurile *id\_cateogrie* și *id\_produs*.

## Modelul Relațional



**Tabelele** din aceasta aplicatie sunt:

- ✓ clienți ;
- ✓ categorii ;
- ✓ categorii\_animale ;
- ✓ detalii\_clienti ;
- ✓ furnizori ;
- ✓ produse ;
- ✓ tipuri\_produce ;
- ✓ vânzări ;

### Descrierea detaliată a tabelelor și a relațiilor dintre ele

Față de modelul logic, în modelul relațional mai apar încă două tabele necesare în realizarea relațiilor de tipul **n:n**.

Între tabelele *clienți* și *produse*, mai apare tabela *vânzări*, prin urmare relația de **many-to-many** se transformă în relație de **one-to-many** între tabela *clienți* și tabela *vânzări* și în relație de **many-to-one** între tabela *vânzări* și tabela *produse*.

Între tabelele *produse* și *categorii\_animale*, mai apare tabela *categorie*, prin urmare relația de **many-to-many** se transformă în relație de **one-to-many** între tabela *produse* și tabela *categorie* și în relație de **many-to-one** între tabela *categorie* și tabela *categorii\_animale*.

### Pentru simplificarea proiectului s-au luat în considerare următoarele aspecte

- Un furnizor aprovizionează maxim o dată pe zi;
- Nu este obligatoriu ca vânzarea să se poată face doar pe baza de card de fidelitate ;
- Nu se poate vinde mai mult decât este în stoc;
- Nu se gestionează termenul de garanție al produselor;
- Vânzarea produselor se face doar în cantități întregi;
- Nu sunt promoții;
- Prețul unui produs nu variază în timp;