

# PROIECT: Gestiunea unui Petshop

**Apostol Alin-Constantin**

## Gestiunea Petshop-ului

Proiectul își propune realizarea unui sistem informatic pentru gestionarea activității unui magazin de tip Petshop. Scopul principal este eficientizarea procesului de vânzare și menținerea unei evidențe clare a stocurilor, clienților și performanței angajaților.

### Principalele funcționalități acoperite:

- **Gestiunea Produselor:** Organizarea produselor pe categorii (Hrană, Accesorii, Farmacie) și monitorizarea stocurilor.
- **Procesarea Comenzilor:** Înregistrarea vânzărilor, calcularea automată a totalurilor și asocierea cu angajatul care a procesat tranzacția.
- **Fidelizarea Clienților:** Sistem de carduri de fidelitate cu nivele (Bronze, Silver, Gold) pentru a încuraja revenirea clienților.
- **Monitorizarea Angajaților:** Evidența personalului și a rolurilor acestora (Manager, Casier, Farmacist).

Informațiile critice gestionate includ datele personale ale clienților (cu validări stricte), detaliile tranzacțiilor (ce s-a vândut și în ce cantitate) și istoricul prețurilor.

## Structura și relațiile dintre entități

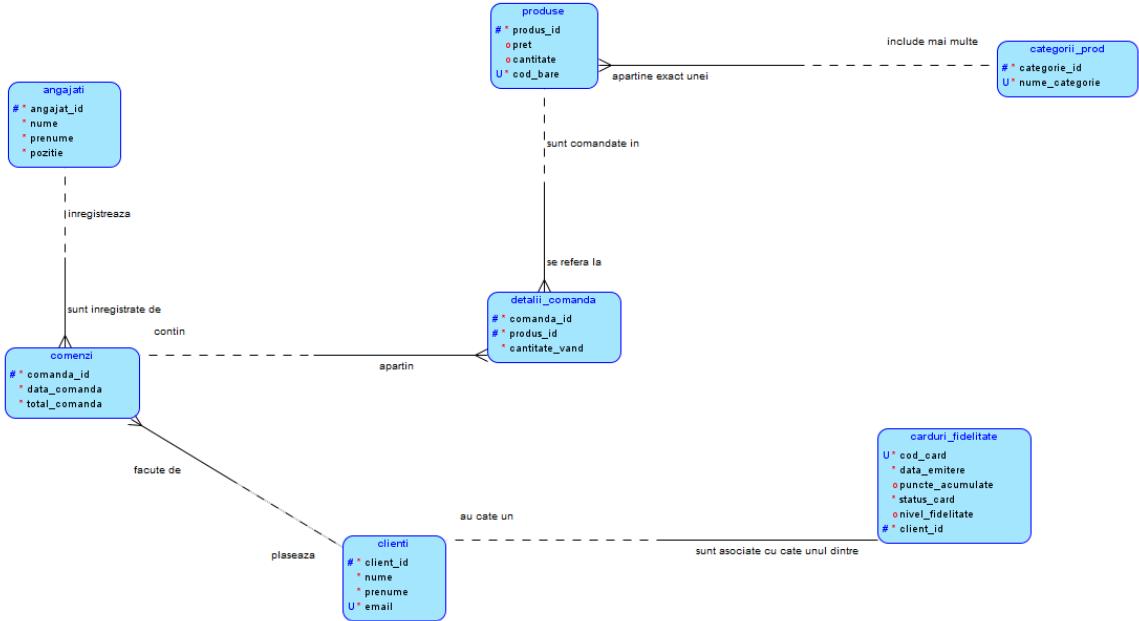
Modelul bazei de date a fost proiectat respectând principiile normalizării. S-au identificat următoarele entități principale:

1. **CLIENTI**: Stochează datele de contact.
2. **CARDURI\_FIDELITATE**: Extensie a entității clienți (Relație 1:1).
3. **ANGAJATI**: Personalul magazinului.
4. **COMENZI**: Tranzacțiile efectuate (Antetul bonului).
5. **DETALII\_COMANDA**: Conținutul tranzacției (Liniile bonului).
6. **PRODUSE**: Inventarul magazinului.
7. **CATEGORII\_PROD**: Clasificarea produselor.

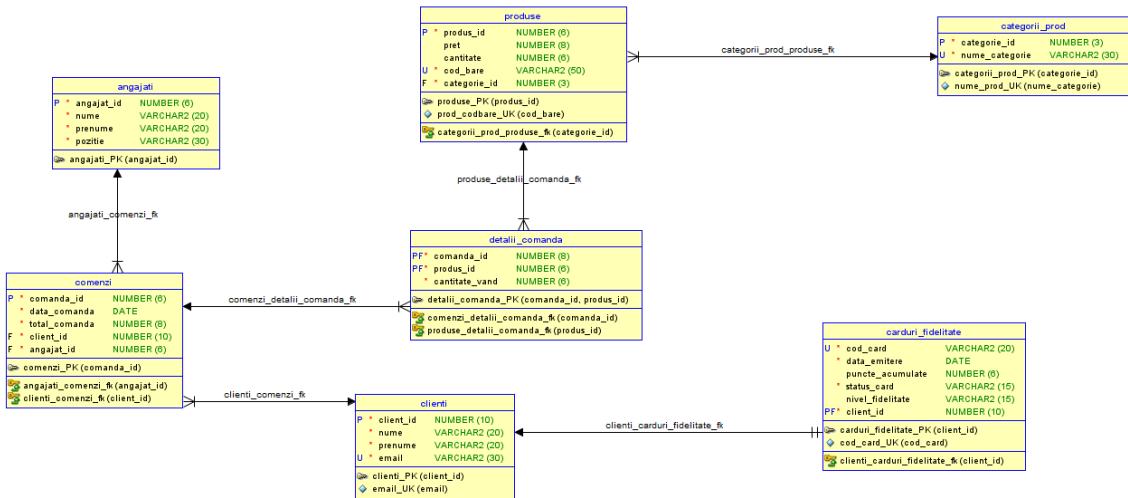
### Tipuri de relații implementate:

- **Relația 1:1 (One-to-One)**: Între CLIENTI și CARDURI\_FIDELITATE.
  - Un client poate avea cel mult un card de fidelitate, iar un card aparține unui singur client.
  - Implementare: Cheia primară din CARDURI\_FIDELITATE (client\_id) este simultan și cheie străină către CLIENTI.
- **Relația 1:N (One-to-Many)**:
  - **Categorii - Produse**: O categorie conține mai multe produse.
  - **Angajați - Comenzi**: Un angajat procesează mai multe comenzi.
  - **Clienți - Comenzi**: Un client poate plasa mai multe comenzi.
- **Relația M:N (Many-to-Many)**: Între COMENZI și PRODUSE.
  - O comandă conține mai multe produse, iar un produs apare în mai multe comenzi.
  - **Rezolvare**: Această relație a fost descompusă prin entitatea asociativă DETALII\_COMANDA, care conține cheile străine ale ambelor tabele părinte și atributul specific relației (cantitate\_vand).

# Modelul Logic



# Modelul relational



# Aspecte legate de Normalizare

Baza de date respectă formele normale pentru a elimina redundanța și anomaliiile:

- **Forma Normală 1 (1NF):** Toate atributele sunt atomice (ex: Nume și Prenume sunt separate). Nu există grupuri repetitive de date (produsele dintr-o comandă nu sunt stocate într-o listă într-un singur câmp, ci în tabela separată DETALII\_COMANDA).
- **Forma Normală 2 (2NF):** Toate atributele non-cheie depind de întreaga cheie primară. În tabela DETALII\_COMANDA (care are cheie compusă), cantitatea vândută depinde de perechea (Comandă + Produs).
- **Forma Normală 3 (3NF):** Nu există dependențe tranzitive. De exemplu, numele categoriei nu este stocat în tabela PRODUSE (unde s-ar repeta inutil), ci în tabela CATEGORII\_PROD, fiind referit doar prin ID.

## Descrierea Coloanelor (Selecție)

- **Tipuri de date:**
  - NUMBER(6): Folosit pentru ID-uri (permite până la 999.999 înregistrări) și stocuri.
  - VARCHAR2(n): Folosit pentru text (Nume, Email). Dimensiunea a fost aleasă în funcție de lungimea estimată (ex: 30 caractere pentru Email).
  - DATE: Pentru data comenzi și data emiterii cardului.
  - NUMBER(8): Pentru prețuri și totaluri, permitând valori monetare mari.

## Descrierea Constrângerilor (Validarea Datelor)

Pentru a asigura integritatea datelor, s-au implementat următoarele mecanisme:

#### A. Chei Primare (Primary Keys)

Generate automat prin **Secvențe și Triggere** (BEFORE INSERT). Acest mecanism de auto-increment asigură unicitatea fiecărei înregistrări (ex: angajat\_id, comanda\_id).

#### B. Chei Străine (Foreign Keys)

Asigură integritatea referențială.

- Nu se poate adăuga un produs într-o categorie inexistentă.
- Nu se poate crea o comandă pentru un client inexistent.

#### C. Constrângeri CHECK (Validare Logică)

- **Validare Email:** S-a folosit REGEX (regexp\_like) pentru a obliga formatul text@domeniu.extensie.
- **Validare Nume:** Numele și Prenumele trebuie să aibă minim 2 litere (length > 1).
- **Valori Pozitive:** Prețul, stocurile și totalul comenzi trebuie să fie > 0.
- **Liste de Valori:**
  - status\_card acceptă doar: 'Activ', 'Expirat', 'Suspendat'.
  - pozitie angajat acceptă doar: 'Casier', 'Manager', 'Farmacist veterinar'.
- **Limită Cantitate:** Un client nu poate cumpăra mai mult de 10 unități din același produs pe un bon (BETWEEN 1 AND 10).

#### D. Constrângeri UNIQUE

- **Email:** Nu pot exista doi clienți cu aceeași adresă de email.
- **Cod Card / Cod Bare:** Identifierii fizici trebuie să fie unici în sistem.

#### E. Triggere Complexe

- **Validare Dată:** S-au creat triggere (trg\_comenzi\_data, trg\_card\_data\_emitere) care împiedică introducerea unei date din trecut (mai mică decât data curentă). Acestea aruncă o eroare personalizată (RAISE\_APPLICATION\_ERROR).