



UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" IAȘI  
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE  
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI  
DISCIPLINA BAZE DE DATE PROIECT

# Gestiunea produselor apicole într-un laborator

Coordonator  
Prof. Mironenu Cătălin

Student,  
Bulai Iustina-Bianca  
1307A

## **Titlul proiectului**

### **Gestiunea produselor apicole într-un laborator**

Analiza, proiectarea și implementarea unei baze de date care să modeleze gestiunea producției într-un laborator ce are la bază produse apicole.

### **Descrierea Proiectului**

În cadrul unui laborator, se dorește realizarea întregului proces de creare și împachetare a unor rețete rezultate în urma combinării cu materii ce nu provin din apicultură.

Așadar, informațiile de care avem nevoie sunt legate de materia primă, produsul finit rezultat, modul de ambalare și termenul de valabilitate al produselor.

Ne vom axa doar pe ideea de producție, ceea ce înseamnă că va trebui să știm cu ce materii prime putem lucra și în ce cantități, pentru a putea rezulta produsele finite. Mai apoi, acestea vor fi plasate în diferite tipuri de ambalaje și doar în momentul în care ne-am asigurat că avem atât cantitățile necesare cât și ambalaj disponibil, va putea fi atașată eticheta ce conține termenul de valabilitate.

### **Descrierea funcțională a aplicației**

Principalele funcții ale aplicației sunt:

- Evidențierea ingredientelor
- Evidențierea produselor rezultate
- Evidențierea ambalajelor
- Evidențierea etichetelor

### **Structura și relațiile dintre entități**

Entitățile din această aplicație sunt:

- Materia\_primă
- Produs\_finit
- Ambalaj
- Tipuri\_Ambalaje
- Etichete
- Termen

În ceea ce privește modelarea datelor, se folosesc relații 1:1(one-to-one), 1:n(one-to-many), m:n(many-to-many).

Între entitatea *Materia\_primă* și *Produs\_finit* există o relație m:n. Un ingredient poate fi prezent în mai multe produse finite, iar un produs finit poate conține mai multe ingrediente. Legătura dintre cele două este posibilă prin prezența unei noi entități formată pe relația m:n, denumită *Rețeta*. Legătura dintre *Materia\_primă* și *Rețeta* se realizează prin *Id\_mp*, iar legătura dintre *Materia\_primă* cu entitatea *Produs\_finit*, prin *Id\_produs*.

Între entitatea *Ambalaj* și *Produs\_finit*, avem o relație 1:n. Un tip de ambalaj poate conține mai multe produse finite. Un produs finit poate fi conținut de un singur ambalaj. Legătura dintre cele două entități se face prin câmpul *ID\_ambalaj*.

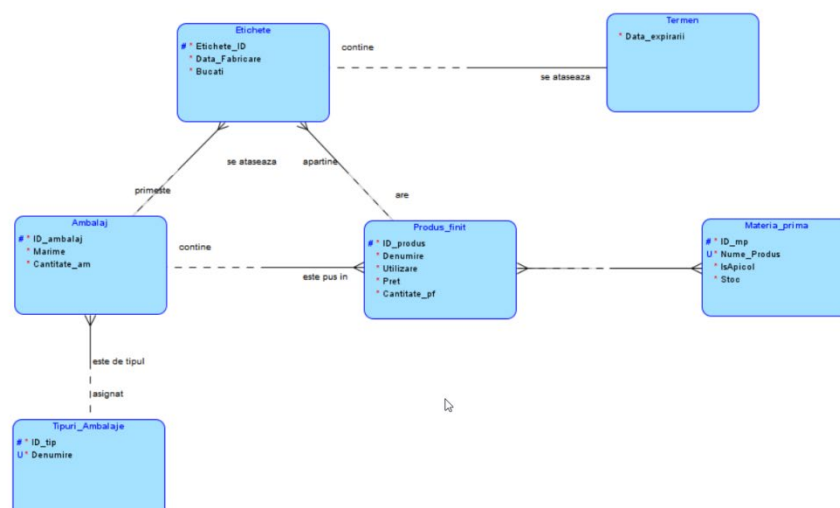
Între entitatea *Ambalaj* și *Etichete*, avem relația 1:n. Un ambalaj primește mai multe etichete, iar un tip de etichetă poate fi pe un singur ambalaj. Cu ajutorul câmpului *ID\_ambalaj*, se realizează legătura între entități.

Între entitatea *Produs\_finit* și *Etichete*, relația este 1:n. Un produs finit poate avea mai multe etichete, iar o etichetă aparține unui produs finit. Legătura între cele două se realizează cu ajutorul câmpului *ID\_produs*.

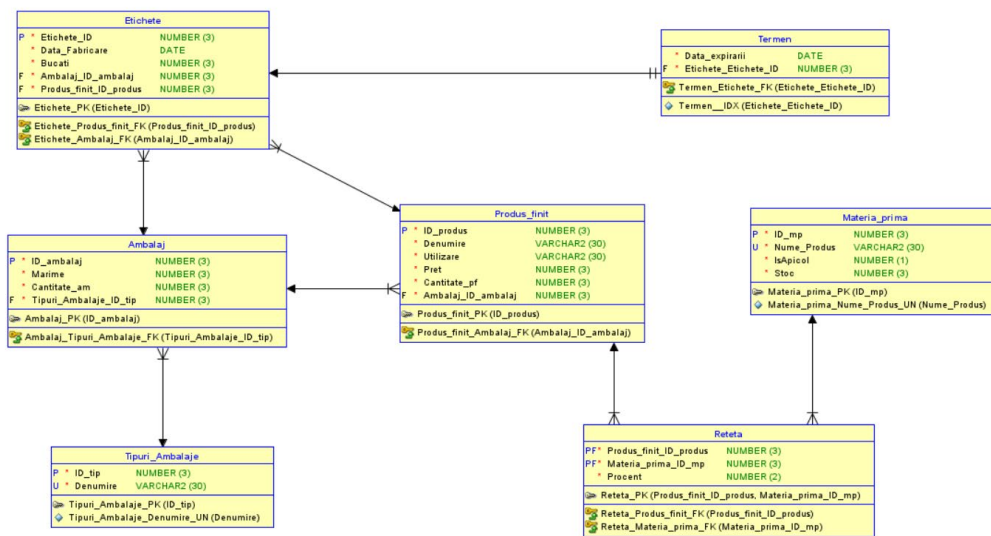
Între *Tipuri\_Ambalaje* și *Ambalaj*, relația 1:n. Un ambalaj poate fi de un singur tip, iar tipul poate fi asignat mai multor ambalaje. Legătura dintre cele două entități este dată de *ID\_tip*.

Între *Etichete* și *Termenul expirării*, relația 1:1. O etichetă conține un termen de valabilitate, iar termenul este atașat pe o singură etichetă. Legătura e realizată cu ajutorul câmpului *Etichete\_ID*.

## Modelul Logic:



## Modelul relațional:



## Descrierea constrângerilor

Constrângerile de tip primary key sunt prezente aproximativ în toate entitățile. Acestea sunt generate de baza de date prin intermediul unui mecanism de tip autoincrement (avem câte un id pentru a identifica unic elementele necesare în cadrul producției: Id\_mp, Id\_produs, Id\_ambalaj, Id\_tip, Etichete\_ID).

Constângerile de tip check au fost utilizate în cadrul atributelor legate de denumire sau stoc/preț. Pentru denumiri, a fost utilizată, ca exemplu, funcția: REGEXP\_LIKE(Denumire, '^[A-Za-z]+((s)?([a-z])+)\*)\$'), deoarece nu putem avea cifre sau alte caractere într-un nume. Pentru attributele stoc/preț/cantitate, s-a impus ca valoarea acestora să fie  $\geq 0$ , deoarece nu putem introduce valori negative.

Constrângerea de tip unique am acordat-o atributelor nume\_produs și denumire(din entitatea tipuri\_ambalaje), deoarece nu putem introduce de mai multe ori același ingredient/tip de ambalaj.

Constrângerile de tip Not Null sunt prezente la toate attributele entităților.

În cadrul entității Rețeta, formată în urma legăturii m:n între Materia\_primă și Produs\_finit, vom avea un atribut procent, căruia i s-a setat precizia 2 deoarece un produs finit trebuie să aibă în compoziția sa suma de 100% a procentelor ingredientelor utilizate.

## Normalizarea datelor

1. Entitatea Materia\_primă este în FN1 (are attribute atomice, grupurile nu se repetă), FN2 (atributele depind în totalitate de cheia primară), nu este în FN3 pentru că avem mai multe chei candidat (ID\_mp, Nume\_Produs).
2. Entitatea Produs\_finit este în FN1 (are valori atomice, grupurile nu se repetă). Este în FN2 (atributele depind în totalitate de cheia primară). Este în FN3 deoarece avem o singură cheie candidat (ID\_produs). În FN3.5 pentru că nu avem dependențe funcționale. În FN4 deoarece nu avem dependențe multi-valoare. Nu este în FN5 deoarece join-urile se fac și pe baza Ambalaj\_ID\_ambalaj.
3. Entitatea Ambalaj este în FN1 (are valori atomice, grupurile sunt unice). Este în FN2 pentru că atributele depind în totalitate de cheia primară (ID\_ambalaj). Este în FN3 deoarece toate atributele non-cheie depind doar de cheia candidat (ID\_ambalaj). Nu avem dependențe funcționale, deci este în FN3.5. Este și în FN4 pentru că nu avem dependențe multi-valoare. Nu este în FN5 pentru că join-urile se fac și pe baza Tipuri\_Ambalaje\_ID\_tip.
4. Entitatea Etichete este în FN1 (are valori atomice, grupurile nu se repetă), FN2 (atributele depind în totalitate de cheia primară), în FN3 (avem o singură cheie candidat - Etichete\_ID). Nu avem dependență funcțională (FN3.5) și nici multi-valoare (FN4). Join-urile pot fi făcute și după Ambalaj\_ID\_ambalaj și Produs\_finit\_ID\_produs, deci entitatea nu e în FN5.
5. Entitatea Tipuri\_ambalaje este în FN1 deoarece are attribute atomice, grupurile nu se repetă. Nu avem attribute non-cheie, prin urmare entitatea va fi și în FN2, FN3, FN3.5. Nu avem dependențe multi-valoare, deci este în FN4. Join-urile se fac doar pe baza cheii primare (ID\_tip), deci entitatea este și în FN5.
6. Entitatea Termen este în FN1 deoarece are attribute atomice, grupurile nu se repetă.

Autoincrementul se folosește deoarece ne dorim să nu inserăm mereu valori în cheile primare numerice. Prin utilizarea acestuia, se poate pune în câmpul cheii primare valoarea null, prima înregistrare pornind de la valoarea minimă setată.