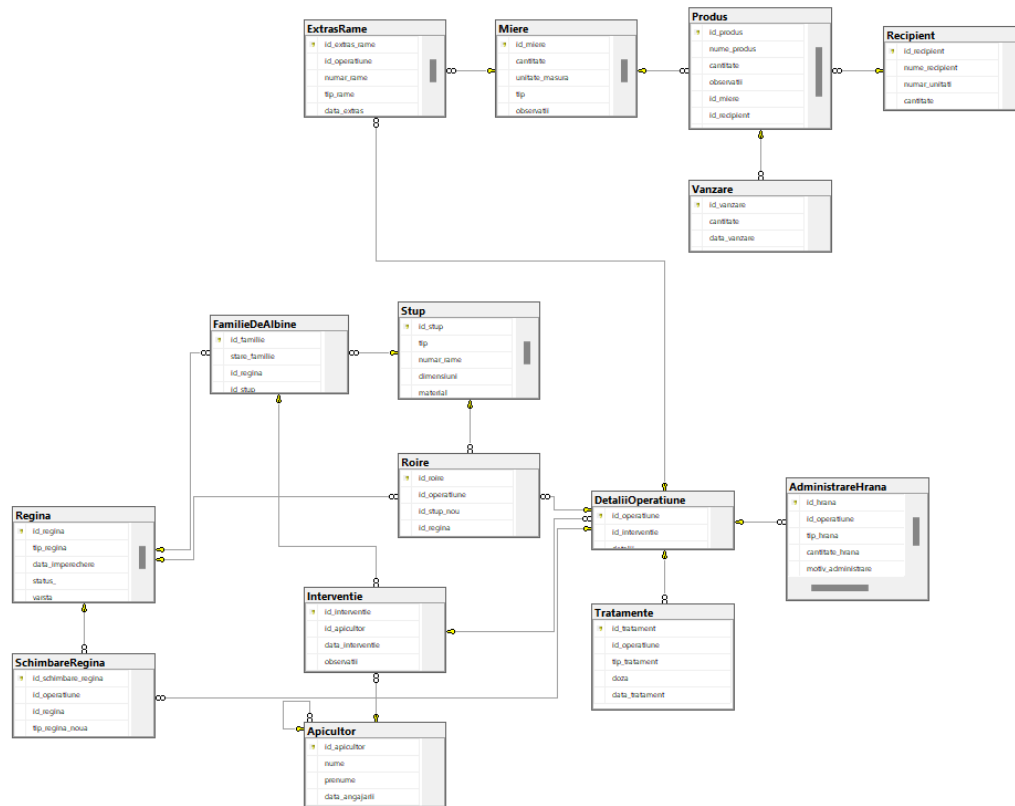


Bază de date – Fermă Apicolă  
Student: Olivian Ghiță Mărgescu

**Ferma Apicola** este un proiect bazat pe gestionarea activităților specifice unei ferme apicole, inclusiv gestionarea stupilor, raselor de albine, tratamentele aplicate, hrănirea albinelor și altele. Proiectul include o aplicație care interacționează cu o bază de date relațională, oferind rapoarte utile despre starea și activitățile fermei.

## 1. Descrierea bazei de date



Baza de date cuprinde 15 tabele, împărțite în 2 zone, una pentru operațiunile legate de îngrijirea și exploatarea albinelor, iar cea de a doua se ocupă de înbutelierea și vânzarea mierii.

Prima parte cuprinde tabele care rețin apicultorii, date despre familii de albine, stupi, regine și mai ales despre intervenții (pentru simplificarea aplicației am ales să proiectez doar cinci): Administrare Hrană, Tratamente, Roire, Schimbare Regină, Extras Rame. Cea din urmă tabelă face legătura dintre prima și a doua zonă.

Cea de a doua zonă cuprinde produsele finale (tabela Podus), care sunt obținute din înbutelierea mierii (tabele Miere) în recipientele din stoc (tabela Recipient), și vândute.

## 2. Constrângeri de Integritate (Chei Străine)

### 1. AdministrareHrana

- [id\_operatiune] → DetaliiOperatiune(id\_operatiune)

### 2. DetaliiOperatiune

- [id\_interventie] → Interventie(id\_interventie)

### 3. ExtrasRame

- [id\_operatiune] → DetaliiOperatiune(id\_operatiune)
- [id\_miere] → Miere(id\_miere)

### 4. FamilieDeAlbine

- [id\_regina] → Regina(id\_regina)
- [id\_stup] → Stup(id\_stup)

### 5. Interventie

- [id\_apicultor] → Apicultor(id\_apicultor)
- [id\_familie] → FamilieDeAlbine(id\_familie).

### 6. Produs

- [id\_miere] → Miere(id\_miere)
- [id\_recipient] → Recipient(id\_recipient)

### 7. Roire

- [id\_operatiune] → DetaliiOperatiune(id\_operatiune)
- [id\_regina] → Regina(id\_regina)
- [id\_stup] → Stup(id\_stup)

### 8. SchimbareRegina

- [id\_operatiune] → DetaliiOperatiune(id\_operatiune)
- [id\_regina] → Regina(id\_regina)

### 9. Tratamente

- [id\_operatiune] → DetaliiOperatiune(id\_operatiune)

### 10. Vanzare

- [id\_produs] → Produs(id\_produs)

## 3. Descrierea procedurilor

Procedurile stocate sunt apelate din backend pentru a gestiona diferite operațiuni ale bazei de date, folosind endpoint-uri RESTful implementate cu Flask. Fiecare procedură are un scop bine definit, fie pentru inserarea, actualizarea, ștergerea, sau extragerea de date. Aceste proceduri oferă integrare între aplicație și baza de date SQL Server, asigurând integritatea și eficiența operațiunilor.

1. **GetFamiliiDeAlbine**
  - Extrage toate familiile de albine din baza de date pentru a permite afișarea lor în aplicație.
2. **AddFamilieDeAlbine**
  - Adaugă o nouă familie de albine în baza de date, asigurând legătura cu un stup și o regină specifică.
3. **GetRegine**
  - Returnează lista reginelor pentru gestionarea și afișarea detaliilor despre ele.
4. **AddRegina**
  - Adaugă o regină în baza de date, inclusiv detalii precum tipul, vârsta și proveniența.
5. **GetStupi**
  - Oferă lista tuturor stupilor, utilă pentru monitorizarea resurselor.
6. **AddStup**
  - Permite adăugarea unui stup cu detalii despre tip, dimensiuni și materiale.
7. **GetAllApicultori**
  - Extrage informațiile despre apicultori pentru a oferi o evidență a personalului.
8. **AddApicultor**
  - Adaugă un nou apicultor în baza de date, incluzând legătura cu un maestru apicultor, dacă este cazul.
9. **PackHoney**
  - Permite ambalarea mierii într-un recipient specific, cu gestionarea stocului și prețurilor.
10. **VanzareProcedura**
  - Asigură înregistrarea unei vânzări, inclusiv actualizarea cantității disponibile.
11. **GetInterventii**
  - Extrage toate intervențiile înregistrate pentru familiile de albine, utile pentru raportare.
12. **AddInterventieFromJSON**
  - Înregistrează o intervenție bazată pe un JSON detaliat, pentru a simplifica gestionarea operațiunilor complexe.
13. **GetDataFamAlbine**
  - Oferă statistici și detalii specifice despre familiile de albine.
14. **GetTotalVanzariProdus**
  - Returnează totalul vânzărilor pentru un produs, util pentru monitorizarea performanței vânzărilor.
15. **GetStupRasa**
  - Oferă o asociere între stupi și rasa albinelor pentru a monitoriza diversitatea.
16. **GetTratamentHrana**
  - Returnează informații despre tratamentele și hrana utilizată pentru familiile de albine.
17. **GetStatisticiStupi**
  - Generează statistici legate de operațiuni efectuate per stup și rasă.
18. **GetDataApicultori**
  - Furnizează statistici despre experiența apicultorilor și relația cu maeștrii lor.

## 4. Descrierea aplicației

Aplicația este un sistem de management al unei ferme apicole, construit pentru a monitoriza și gestiona diverse aspecte ale operațiunilor, resurselor și producției. Aceasta interacționează cu o bază de date SQL Server folosind un backend RESTful bazat pe Flask.

Aplicația se conectează la baza de date folosind driverul ODBC pentru SQL Server și librăria `pyodbc`. Conexiunea se stabilește cu detalii despre server, bază de date și autentificare. După stabilirea conexiunii, se execută proceduri stocate sau interogări SQL pentru a manipula și returna datele.

Principalele funcționalități includ:

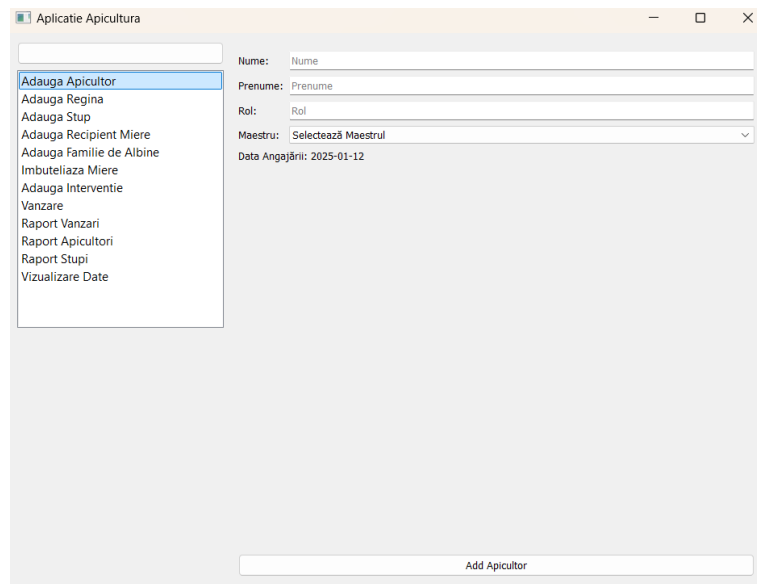
- Gestionarea familiilor de albine, stupilor, apicultorilor și reginelor.
- Monitorizarea intervențiilor și tratamentelor aplicate.
- Evidența producției, ambalării și vânzării de miere și alte produse apicole.
- Generarea de statistici și rapoarte despre activitățile din fermă.

---

## Workflow-ul Aplicației

### 1. Adăugare Date (Resurse și Operațiuni):

Apicultorii introduc date despre resurse (regine, stupi, apicultori, recipiente) sau operațiuni (intervenții, tratamente, producție). Datele sunt validate la nivel de backend și stocate în baza de date prin proceduri stocate.



The screenshot shows a web application window titled "Aplicatie Apicultura". On the left, there is a vertical menu with the following items: "Adauga Apicultor" (highlighted), "Adauga Regina", "Adauga Stup", "Adauga Recipient Miere", "Adauga Familie de Albine", "Imbuteliază Miere", "Adauga Interventie", "Vanzare", "Raport Vanzari", "Raport Apicultori", "Raport Stupi", and "Vizualizare Date". The main area on the right contains a form with the following fields: "Nume:" (text input), "Prenume:" (text input), "Rol:" (text input), "Maestru:" (dropdown menu with "Selectează Maestrul" as the selected option), and "Data Angajării:" (text input with the value "2025-01-12"). At the bottom right of the form area, there is a button labeled "Add Apicultor".

2.

### 3. Vizualizare Date:

Utilizatorii pot accesa datele stocate prin endpoint-uri de tip GET, care apelează proceduri pentru extragerea și formatarea datelor.

#### 4. Fluxul de Tranzacții (Operațiuni):

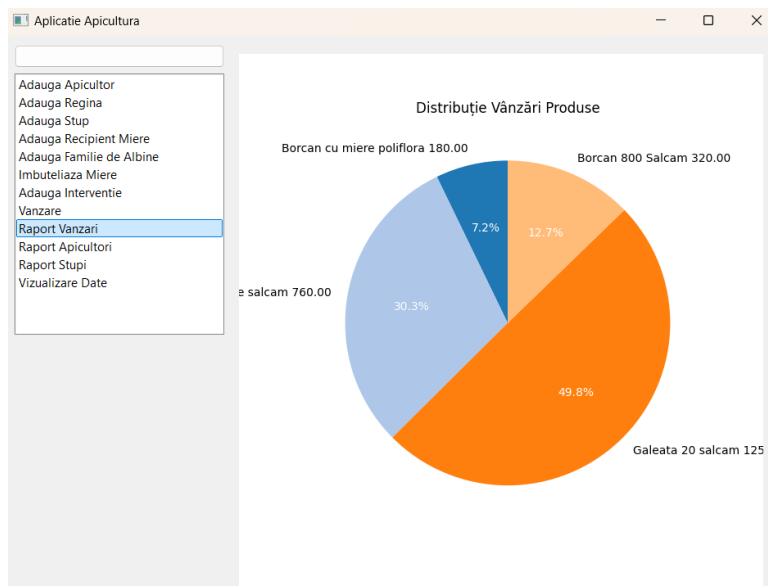
Operațiunile, precum tratamentele, intervențiile, sau vânzările, sunt gestionate prin endpoint-uri specifice, care asigură integritatea datelor și aplică logica de afaceri prin proceduri stocate.

The screenshot shows the 'Aplicatie Apicultura' application window. On the left is a sidebar menu with options: Adauga Apicultor, Adauga Regina, Adauga Stup, Adauga Recipient Miere, Adauga Familie de Albine, Imbuteliaza Miere, Adauga Interventie (highlighted), Vanzare, Raport Vanzari, Raport Apicultori, Raport Stupi, and Vizualizare Date. The main area displays the 'Adauga Interventie' form. Fields include: ID Apicultor (ID Apicultor), ID Familie (ID Familie), Data Interventiei (2025-01-12), checkboxes for Tratament, Roire, Schimbare Regina (checked), Administrare Hrana, and Extras Rame, Observatii (Observatii), Detalii (Schimbare Regina), Tip Tratament (Tip Tratament), Doza Tratament (Doza Tratament), Provenienta Regina (Provenienta Regina), Tip Stup (Tip Stup), ID Regina (ID Regina), Tip Hrana (Tip Hrana), Cantitate Hrana (Cantitate Hrana), Motiv Administrare (Motiv Administrare), Numar Rame Extrase (Numar Rame Extrase), Cantitate Miere (Cantitate Miere), and Observatii (Observatii). A button labeled 'Adauga Interventie' is at the bottom.

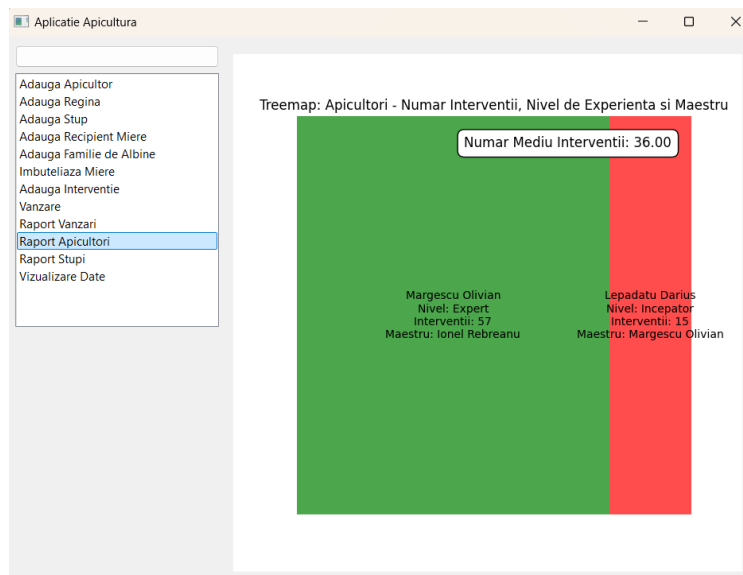
#### Statistici și Rapoarte:

Aplicația include trei rapoarte principale:

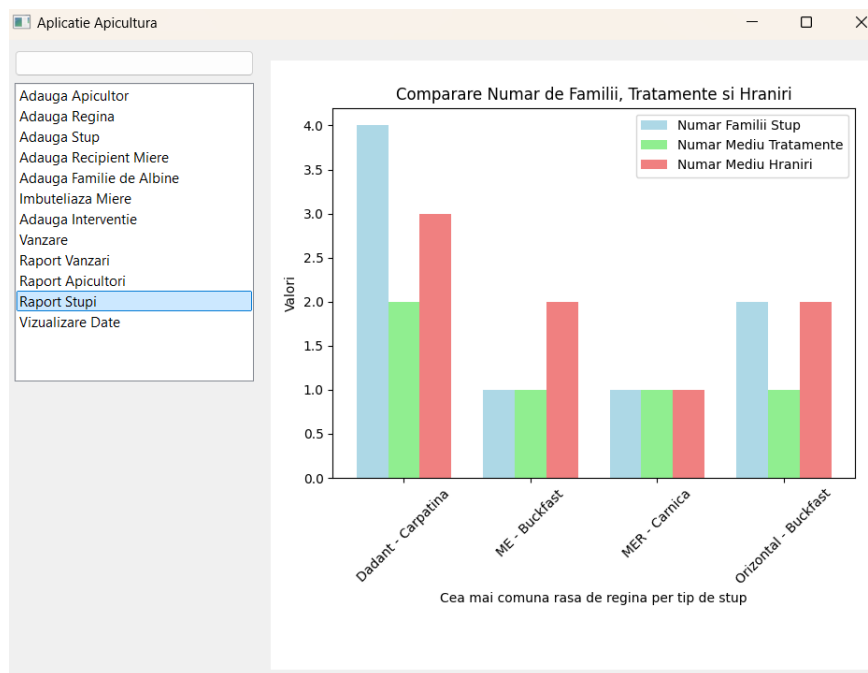
1. **Total Vânzări Produse:** Monitorizează cantitatea și venitul generat din vânzarea produselor, oferind o imagine clară a performanței financiare a fermei.



2. **Statistici Stupi și Rase:** Analizează operațiunile și producția stupilor, pe rase de albine, pentru a optimiza resursele și a îmbunătăți performanța.



3. **Experiența Apicultorilor:** Urmărește activitatea și experiența apicultorilor, oferind date despre intervenții și performanța acestora în ferme.



## 5. Concluzie:

Aplicația dezvoltată are scopul de a gestiona și analiza datele referitoare la activitățile apicole, precum gestionarea familiilor de albine, intervențiile efectuate, produsele obținute, vânzările și alte aspecte ale fermei apicole. Prin utilizarea unui backend robust bazat pe Flask și integrarea cu o bază de date SQL Server, aplicația oferă o interfață eficientă și ușor de utilizat pentru interacțiunea cu datele. De asemenea, folosirea procedurilor stocate pentru manipularea datelor asigură o performanță îmbunătățită și o gestionare centralizată a operațiunilor de backend. Cu ajutorul acestui sistem, utilizatorii pot adăuga, modifica și interoga datele într-un mod structurat, având acces rapid la rapoarte esențiale pentru administrarea fermei apicole.

## Bibliografie:

1. Flask Documentation. (2024). Retrieved from <https://flask.palletsprojects.com/>
2. PyODBC Documentation. (2024). Retrieved from <https://github.com/mkleehammer/pyodbc>
3. Microsoft SQL Server Documentation. (2024). Retrieved from <https://learn.microsoft.com/en-us/sql/>