UNIVERSITATEA “ȘTEFAN CEL MARE”, SUCEAVA

FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

PROGRAM DE STUDIU CALCULATOARE

PROIECT DE DIPLOMĂ

Site web ”Happy Bee” educativ, blog și manageriere stupină

Studentă Profesor coordonator

Istratoaie Ioana-Teodora conf. dr. ing. Schipor Ovidiu-Andrei

Suceava, 2021

Cuprins

[TEMA ȘI MOTIVAȚIA ALEGERII (2 pg) 3](#_Toc60583080)

[I. Capitolul I. Tehnologii utilizate (5 pagini) 4](#_Toc60583081)

[I.1. Tehnologii specifice bazelor de date 4](#_Toc60583082)

[I.1.1. SQL și MySQL 4](#_Toc60583083)

[I.1.2. Elemente specifice bazelor de date 4](#_Toc60583084)

[I.2. Tehnologii specifice web 5](#_Toc60583085)

[I.2.1. React 5](#_Toc60583086)

[I.2.2. Bootstrap 5](#_Toc60583087)

[I.3. Tehnologii pentru crearea de severe 6](#_Toc60583088)

[I.3.1. Node.js 6](#_Toc60583089)

[I.3.2. JavaScript 6](#_Toc60583090)

[I.3.3. Ngrok 6](#_Toc60583091)

[I.4. IDE-uri și platforme utilizate 6](#_Toc60583092)

[II. Capitolul II. Specificații de implementare (20 pg) 6](#_Toc60583093)

[II.1. Proiectarea aplicației 7](#_Toc60583094)

[II.1.1. Creare design web 7](#_Toc60583095)

[II.1.2. Proiectarea bazei de date (structura bazei de date, constrângeri asupra datelor) 7](#_Toc60583096)

[II.2. Proiectarea codului 10](#_Toc60583097)

[II.2.1. Diagrame UML(clase dezvoltate) 11](#_Toc60583098)

[II.2.2. Diagrame Workflow(logica principala a programului) 11](#_Toc60583099)

[II.2.3. Scenarii de testare 11](#_Toc60583100)

[II.3. Codarea aplicației 11](#_Toc60583101)

[II.3.1. Implementarea bazei de date 11](#_Toc60583102)

[II.3.2. Server baza de date 11](#_Toc60583103)

[II.3.3. Platformă web 11](#_Toc60583104)

[III. Capitolul III. Specificatii de utilizare (20 pagini) 11](#_Toc60583105)

[III.1. Pornirea aplicației 12](#_Toc60583106)

[III.2. Utilizarea aplicației 12](#_Toc60583107)

[IV. Concluzii (1 pagina) 12](#_Toc60583108)

[V. Bibliografie (1 pagina) 13](#_Toc60583109)

[VI. Anexe 14](#_Toc60583110)

# TEMA ȘI MOTIVAȚIA ALEGERII (2 pg)

(-prezentare\_tema  
-motivare\_alegere  
- solutii similare existente)

Lumea este un adevărat lanț misterios de lucruri, fenomene și elemente neprevăzute. Fiecare element, fie el chiar și o simplă albină, are un rol esențial in demersul natural al ecosistemului planetei.

# Capitolul I. Tehnologii utilizate (5 pagini)

Proiectul final este însumarea tuturor tehnologiilor care se completează una pe alta. Aplicația este constituită din trei categorii de tehnologii: tehnologii specifice bazelor de date, tehnologii specifice servering și respectiv tehnologii specifice web, Fig.I.1. În continuare vor fi prezentate succint aceste tehnologii.

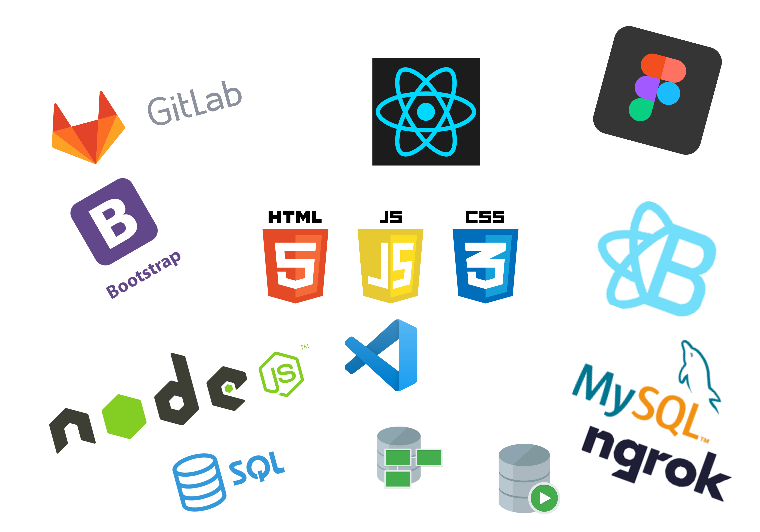


Fig. I.1 Tehnologii și IDE-uri utilizate în cadrul proiectului

## Tehnologii specifice bazelor de date

Pentru realizarea bazei de date s-au utilizat două limbaje specifice: SQL, pentru etapa de proiectare fiind limbajul de bază pentru MySQL iar cel din urma pentru gestionarea bazei de date la nivelul proiectului.

### SQL și MySQL

*„Datele din cele mai obișnuite tipuri de baze de date sunt distribuite de regulă pe linii și coloane, în diferite tabele, pentru eficientizarea procesării și interogării datelor. Datele pot fi accesate, gestionate, modificate, actualizate, controlate și organizate cu ușurință. Majoritatea bazelor de date utilizează un limbaj structurat de interogare (SQL) pentru scrierea și interogarea datelor.*”([Conform Oracle](https://www.oracle.com/ro/database/what-is-database/))

**SQL** este un limbaj de programare neprocedural, acesta este utilizat de aproape toate bazele de date relaționare, pentru definirea, interogarea și gestionarea datelor dar și pentru organizarea datelor în tabele între care se pot definii anumite relații specifice. **MySQL**  este un SGBD open-source care se bazează pe SQL pentru interogarea datelor din cadrul tabelelor. Acest limbaj permite implementarea unei baze de date pe un calculator personal astfel încât gestionarea manipularea și întreținerea acesteia să se poată realiza independent de alte servere externe.

### Elemente specifice bazelor de date

Pentru a înțelege modul de creare al bazei de date vom pune in evidență câțiva termeni specifici:

* Un **SGBD( SISTEME DE GESTIUNE A BAZELOR DE DATE)** reprezintă un sistem de programe care permit utilizatorului definirea, crearea, întreținerea bazei de date și accesul controlat la aceasta.
* Intre tabelele bazei de date se pot construi mai multe tipuri de legături:
* **1:1 - unei entități din mulțimea 1 îi corespunde o singură entitate din mulțimea 2 si reciproc;**
* **1:N- unei entități din mulțimea 1 îi corespund una sau mai multe entității din mulțimea 2, dar fiecărei entități din mulțimea 2 îi corespunde o singură entitate din mulțimea 1;**
* **M:N- unei entități din mulțimea 1 îi corespund una sau mai multe entități din mulțimea 2 și reciproc;**
* Pentru crearea **SGBD-ului** final s-au avut în vedere următorii pași:

1. Analiza proiectului;

2. Proiectare conceptuală a bazei de date(Diagrama ERD);

3. Proiectarea logică a bazei de date;

4. Proiectarea fizică a bazei de date;

5. Implementarea bazei de date.

* Diagrama entitate relație (ERD) are rolul de a reda cât mai precis legăturile ce se stabilesc între entități (tabele), descrierea legăturilor prin simboluri specifice și descrierea entităților prin atributele acestora.

Lista simbolurilor din cadrul **diagramei ERD** și semnificațiile acestora sunt prezentate în tabelul următor:

|  |  |
| --- | --- |
| Simbol | Semnificație |
| #/ P | Cheie primară |
| \* | Câmp obligatoriu |
| o | Câmp opțional |
|  | Entitatea **poate** conține o legătură către o altă entitate (**nu este obligatoriu**) |
|  | Entitatea **trebuie să** conțină o legătură către o altă entitate (**este obligatoriu**) |
| A B | Legătură 1:N în care entitatea B preia id-ul din entitatea A, lui A îi corespund mai multe entități B |
| A B | Legătură M:N în care entitatea in care B poate avea mai multe entități A și invers |
| A B | Legătura 1:1 în care unei singure entități A îi corespunde o singură entitate B |

Tabelul I.1 Simboluri specifice ERD

## Tehnologii specifice web

### React

### Bootstrap

Sasa

D

## Tehnologii pentru crearea de severe

### Node.js

### JavaScript

### Ngrok

## IDE-uri și platforme utilizate

Pentru realizarea proiectului s-au utilizat o serie de platforme pentru a venii atât în ușurarea muncii cât și pentru eficientizarea lucrului în echipă. Pentru realizarea designului web dar și mobile sa utilizat editorul grafic vectorial **Figma**. Acesta este util pentru colaborarea în timp real cu membrii echipei. O altă platformă esențială în realizarea proiectului este **GitLab**, care are principalele două caracteristici: facilitează lucrul în echipă, controlează versiunile aplicației.

Pentru proiectarea bazei de date sa utilizat **SQL Developer Data Modeler**. Principalul avantaj al acestei aplicații este faptul că modul de utilizare al acesteia este intuitiv iar dupa proiectarea fizică a bazei de date se poate genera automat codul pentru baza de date.

Codarea server-ului dar și a site-ului web au fost realizate cu ajutorul IDE-ului **Visual Studio Code.** Principalele avantaje ale acestuia sunt: se pot instala extensii pentru autocompletarea instrucțiunilor în funcție de limbajul de programare, se pot cerea snippet-uri proprii pentru generare de cod, este intuitiv și ușor de utilizat, este compatibil cu proiectele de git.

# Capitolul II. Specificații de implementare (20 pg)

(- clase dezvoltate  
- structura bazei de date  
- constrangeri asupra datelor  
- cod sursa pentru logica principala a programului  
- fisiere de test (daca s-a utilizat testarea automata))

Implementarea aplicației “Happy Bee” s-a realizat trecând prin următorii pași:

1. Definire obiective și structura proiectului;
2. Alegerea tehnologiilor pentru realizarea proiectului;
3. Proiectarea aplicației;
4. Implementarea și testarea aplicației.

Primii doi pași au fost descriși în capitolele anterioare. Proiectarea și implementarea vor fi descrise în continuare.

## Proiectarea aplicației

Pentru orice aplicație partea de proiectare este esențială ea oferă o vedere de ansamblu asupra proiectului pentru a eficientiza etapele ulterioare ale proiectului.

### Creare design web

Design-ul web a fost realizat cu ajutorul tool-ului online [Figma](https://www.figma.com/). O vedere de ansamblu a design-ului site-ului web se poate observa în Fig. II.1. Aceasta este o vedere de ansamblu pentru un utilizator normal. Atunci când se larghează un administrator va avea acces al alte meniuri specifice administrării paginii web.

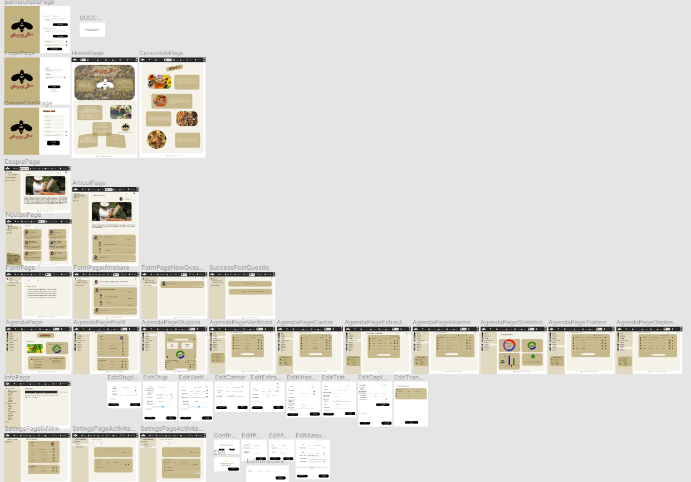


Fig. II.1 Design site web cont normal utilizator

### Proiectarea bazei de date (structura bazei de date, constrângeri asupra datelor)

* + - 1. **Analiza site-ului web ”Happy Bee” și scopul acestuia**

După etapa de creare a design-ului site-ului web s-au analizat necesitățile pentru crearea bazei de date. Din analiza structurii site-ului web “Happy Bee” s-a observat faptul că pentru baza de date vor fi necesare următoarele elemente:

* entitate **Utilizatori** care va reține date despre utilizatori. Această entitate va avea două categorii de utilizatori: administratori și utilizatori normali;
* Pentru postarea informațiilor despre apicultură din cadrul site-ului aceste se vor reține in cadrul unui tabel;
* Utilizatorii au posibilitatea să: posteze anunțuri în secțiunea de noutăți, să comunice cu alți utilizatori prin intermediul blog-ului, să posteze anumite comentarii la anunțuri, să transmită anumite sesizări în cazul în care observă o neregulă, să-și vizualizeze statisticile din propria stupină[[1]](#footnote-1), dacă acesta o deține;
* Vor exista entități separate care vor reține date despre Anunțuri, Sesizări și Blog.
* Actualizarea datelor se realizează relativ frecvent.

Însumând toate elementele enumerate mai sus, este ușor de observat faptul că gestionarea acestora se poate realiza eficient printr-un **SGBD.**

Principalul scop al acestei baze de date este pentru gestionarea cât mai eficienta a site-ului „Happy Bee”, atât a informațiilor cât și a postărilor utilizatorilor. Principala caracteristică pentru a realiza o bază de date cât mai eficientă este organizarea precisă a tuturor relațiilor specifice dintre entități.

* + - 1. **Proiectarea și crearea structurii bazei**

Pentru a determina care sunt entitățile și legăturile dintre acestea am construit o listă de întrebări la care trebuie să răspundă baza de date urmate în urma analizei de la punctul anterior:

1. Care sunt utilizatorii care au postat anunțuri, întrebări în blog sau au realizat reclamații?
2. Există conturi de utilizatori blocate? Cine este administratorul care a blocat contul?
3. Câte articole de informații conține site-ul?
4. La câte postări de blog s-au anunțuri s-a dat cel puțin un răspuns?
5. Câți utilizatori dețin stupine?
6. Câți utilizatori au postat anunțuri de tipul vânzare?

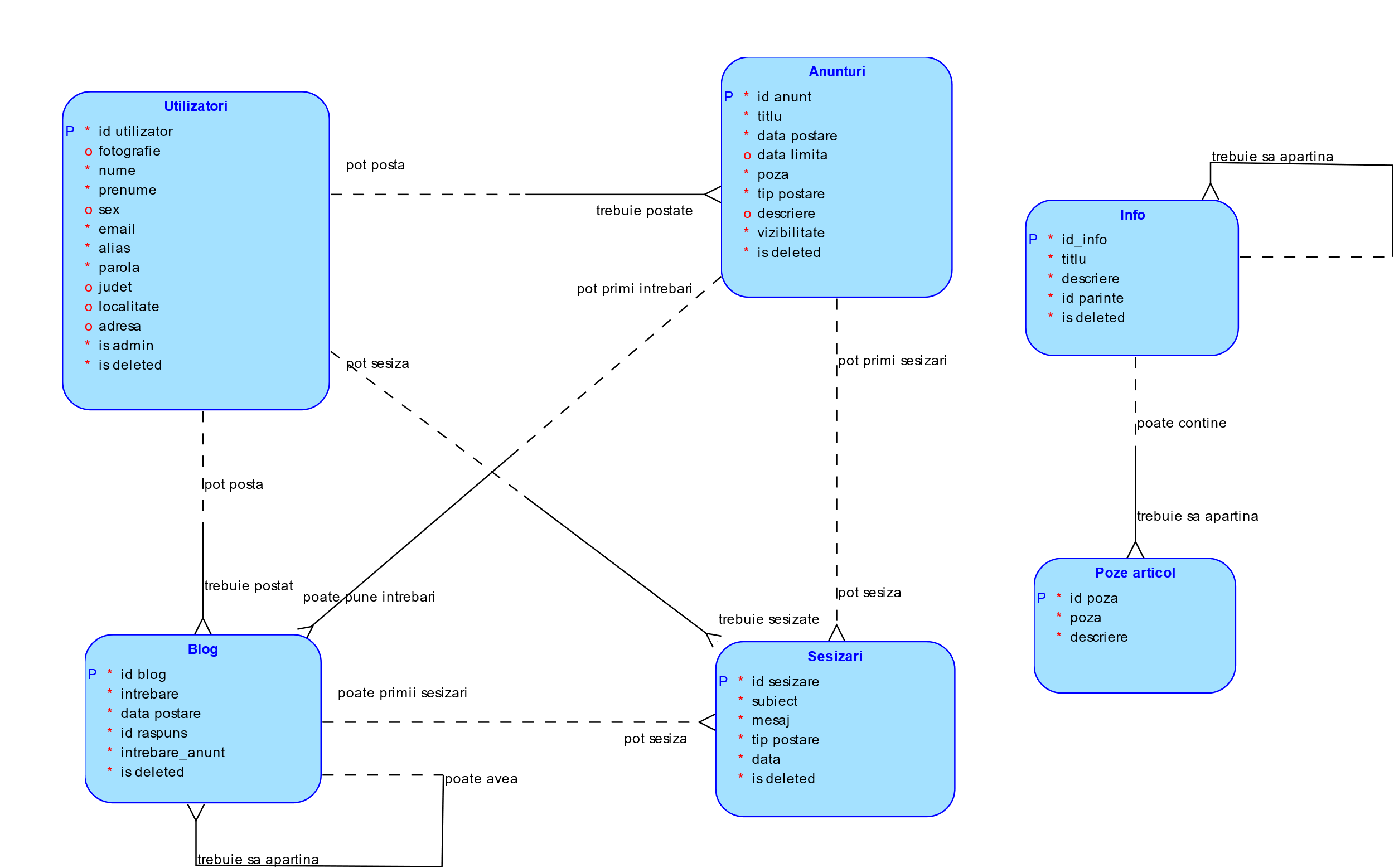
Lista întrebărilor de mai sus și analiza de la punctul anterior au fost principalul reper pentru construirea diagramei Logice, diagramă ERD, Fig. II.1.

Fig. II.2 Diagrama ERD a bazei de date

Pe baza diagramei sa construit matricea legăturilor, Fig. II.2, în care se pot observa care sunt legăturile dintre entități. Pentru a construii această matrice s-au utilizat etichetele de pe legăturile dintre tabele.



Fig. II.3 Matricea legăturilor

În Tabelul II.1 Se poate observa descrierea fiecărei entități utilizate în SGBD.

|  |  |  |
| --- | --- | --- |
| **Nr. crt.** | **Entități** | **Descriere** |
| **1** | **Utilizatori** | Conține toți utilizatorii aplicației.pot fi doua categorii de utilizatori: administratori sau utilizatori normali. |
| **2** | **Anunțuri** | Conține informații despre anunțurile pe care le postează utilizatorii. |
| **3** | **Blog** | Conține informații despre întrebările postate de utilizatori. Acestea pot fi  postate la secțiunea de Blog sau Anunțuri. |
| **4** | **Sesizări** | Conține informații despre sesizările realizate de utilizatori. |
| **5** | **Info** | Conține informații despre apicultură. |
| **6** | **Poze articol** | Conține poze specifice entității Info. |

Tabelul II.1 Descrierea entităților utilizate

Pentru a pune în evidență atât cardinalitatea cât și obligativitatea legăturilor s-a construit un tabel pentru a le descrie, Tabelul II.2. La rândul 7 și respectiv 9 se pot observa două legături recursive care au rolul de a integra subcategorii de aceiași categorie.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.crt.** | **Relație** | **Tip relație** | **Descriere** |
| **1** | **Utilizatori - Anunțuri** | 1:N | Un utilizator poate posta unul sau mai multe anunțuri. |
| **2** | **Utilizatori - Blog** | 1:N | Un utilizator poate posta una sau mai multe întrebări într-un blog. |
| **3** | **Utilizatori - Sesizări** | 1:N | Un utilizator poate realiza una sau mai multe sesizări. |
| **4** | **Anunțuri - Blog** | 1:N | La un anunț se pot adresa una sau mai multe întrebări. |
| **5** | **Anunțuri - Sesizări** | 1:N | La un anunț se poate răspunde cu una sau mai multe sesizări. |
| **6** | **Blog - Sesizări** | 1:N | La un blog se poate răspunde cu una sau mai multe sesizări. |
| **7** | **Blog - Blog** | 1:N | La o întrebare din blog se poate răspunde cu unul sau mai multe răspunsuri. |
| **8** | **Info - Poze articol** | 1:N | O informație poate conține una sau mai multe poze specifice. |
| **9** | **Info - Info** | 1:N | O informație poate conține una sau mai multe subcapitole. |

Tabelul II.2 Descrierea relațiilor din baza de date

Cu ajutorul diagramei Fizice se pot observa detaliile despre atributele și constrângerile aplicate acestora, Fig. II.3.

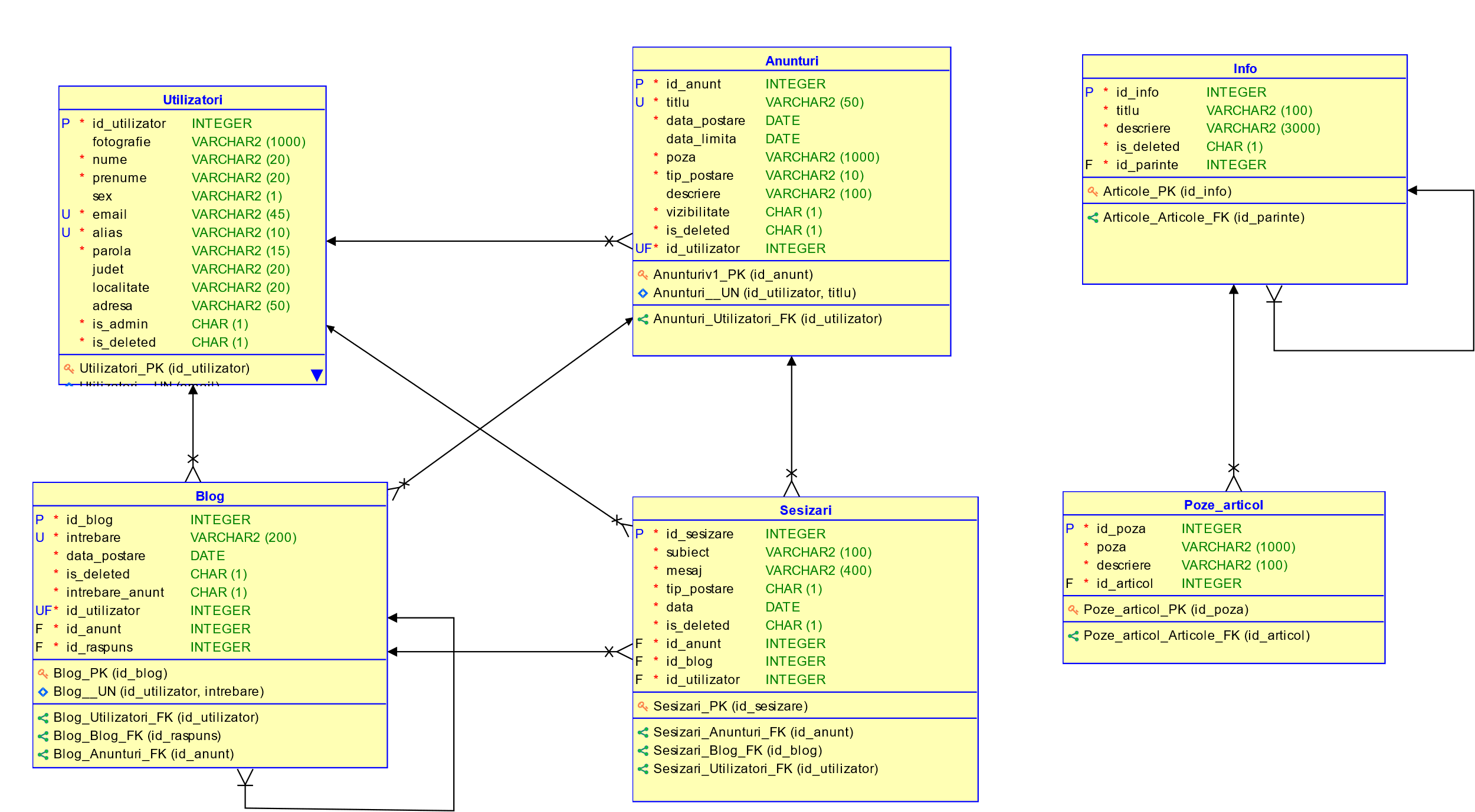


Fig. II.4 Diagrama fizică a bazei de date

* + - 1. **Constrângeri asupra bazei de date**

În cadrul bazei de date există două categorii de constrângeri: constrângeri de cheie și constrângeri de dimensiune și validare. Astfel asupra datelor s-au pus următoarele constrângeri:

* Câmpuri obligatorii, marcate cu steluță(\*);
* Constrângeri de cheie primară, străină sau câmpuri unice, marcate cu literele P,F respectiv U;
* Constrângeri de tip(VARCHAR, CHAR, INTEGER etc.);
* Constrângeri de dimensiune a datelor.

Constrângerile și validările au rolul esențial de a prevenii introducerea eronată a datelor, acestea trebuie efectuate la fiecare nivel de programare, astfel ne asiguram de faptul că eliminăm pe cat posibil incontinența datelor la nivelul bazei de date.

## Proiectarea codului

dsafd

### Diagrame UML(clase dezvoltate)

### Diagrame Workflow(logica principala a programului)

### Scenarii de testare

fsfds

## Codarea aplicației

sad

### Implementarea bazei de date

Pentru implementarea bazei de date s-a [instalat](https://www.youtube.com/watch?v=WuBcTJnIuzo&ab_channel=ProgrammingKnowledge) mai întâi MySQL. După instalare și configurare s-a convertit codul SQL generat la etapa de proiectare în cod MySQL. Codul final a fost rulat în clientul MySQL de pe calculatorul local. În figura II.5 se pot observa tabele din cadrul bazei de date.

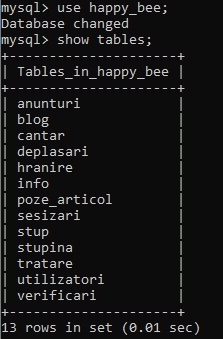


Fig. II.5 Tabelele din baza de date Happy Bee

O dată deținută baza de date următorul pas a fost crearea server-ului pentru comunicarea între baza de date și site-ul web.

### Server baza de date

### Platformă web

dsada

# Capitolul III. Specificatii de utilizare (20 pagini)

(- scenarii de utilizare  
- capturi din aplicatie)

## Pornirea aplicației

sadfs

## Utilizarea aplicației

dsafa



Fig. III.1

dsad

adsad

# Concluzii (1 pagina)

Dasd

# Bibliografie (1 pagina)

<https://www.figma.com/>

<http://www.eed.usv.ro/~mdanub/cursuri/>

<https://www.oracle.com/ro/database/what-is-database/>

<https://ro.m.wikipedia.org/wiki/MySQL>

<https://en.wikipedia.org/wiki/Figma_(software)>

<https://www.youtube.com/watch?v=WuBcTJnIuzo&ab_channel=ProgrammingKnowledge>

# Anexe

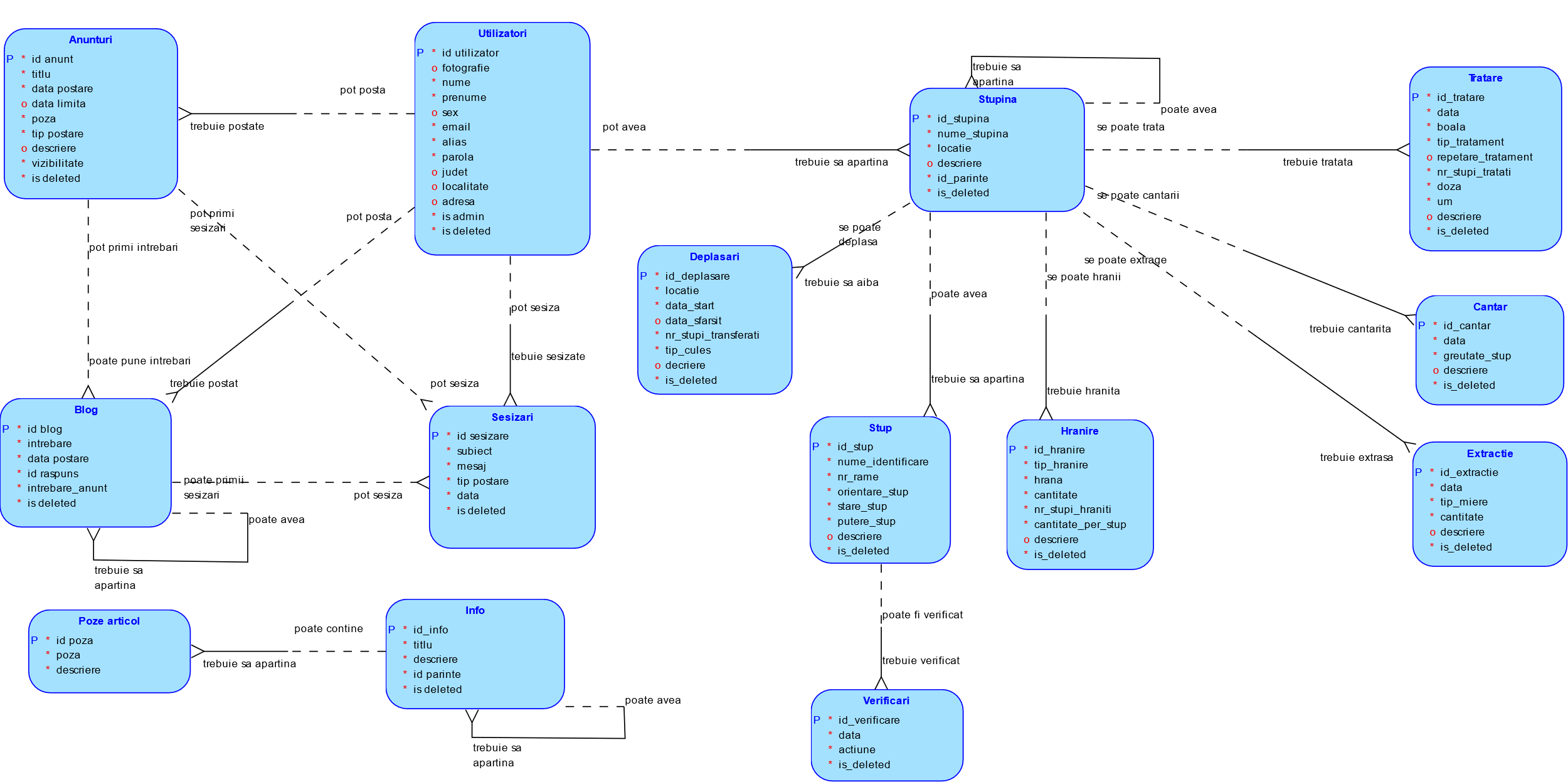


Fig. VI.1 Diagrama logică completă a bazei de date

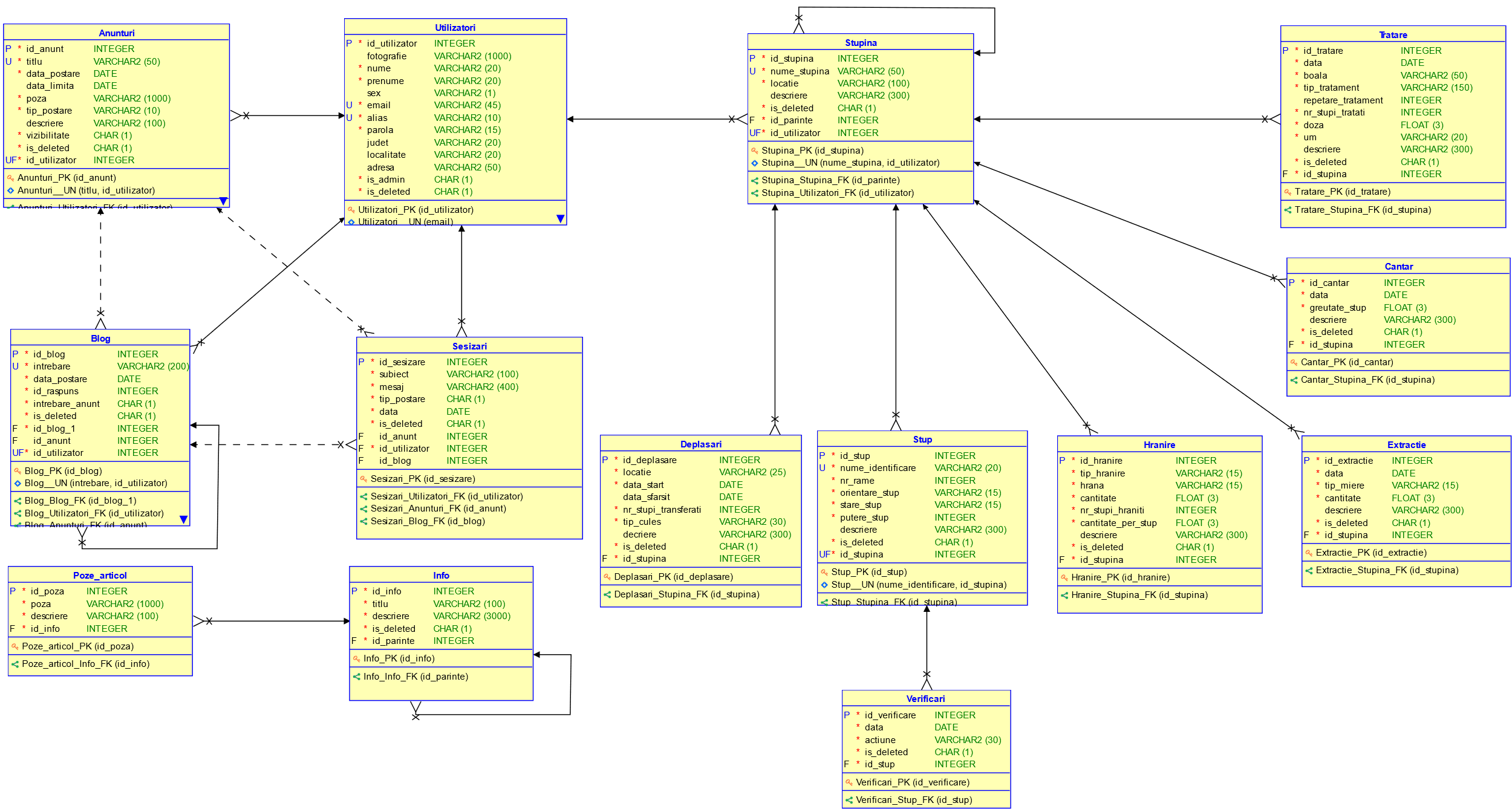


Fig. VI.2 Diagrama fizică completă a bazei de date

1. Lucrarea de față este în completarea cu lucrarea “Agendă apicolă” de Andrei Marius-Vasile, în care se vor găsii detaliile despre baza de date referitoare la stupinele deținute de utilizatori. La anexe se poate regăsii diagrama logică și fizică a bazei de date complete. [↑](#footnote-ref-1)