

**UNIVERSITATEA TEHNICA “Gheorghe Asachi” din IAȘI**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**DOMENIUL: Calculatoare și Tehnologia Informației**

**SPECIALIZAREA: Tehnologia Informației**

**DISCIPLINA: EXTRAGEREA CUNOȘTIINȚELOR DIN BAZE DE DATE**

# **Gestiunea unui adăpost de animale În MongoDB**

**Coordonator,**

**Prof. Cătălin Mironeanu**

**Student,**

**Mertic Lucia-Maria,**

**Grupa 1409A**

## 1. Descrierea proiectului

Proiectul presupune analiza, proiectarea și implementarea unei baze de date pentru gestionarea unei organizații de adopție a animalelor, utilizând MongoDB. Baza de date trebuie să permită stocarea și gestionarea informațiilor legate de animale, persoanele care adoptă și detaliile adopțiilor în sine.

Acest sistem facilitează operațiunile de adăugare, actualizare și ștergere a înregistrărilor, precum și realizarea de interogări complexe pentru a sprijini activitățile administrative ale organizației.

Considerând că poate exista un număr mare de animale, persoane și adopții, sistemul ar trebui să includă funcționalități de căutare avansată care să faciliteze administrarea eficientă a organizației. Acestea ar putea include căutarea animalelor în funcție de specie, rasă sau statutul de disponibilitate, căutarea persoanelor în funcție de nume, oraș sau contact, sau căutarea adopțiilor în funcție de data acestora sau a statusului.

## 2. Structura colecțiilor

Schema detaliată a bazei de date include trei colecții principale:

### animale:

- `_id`: ObjectId - Un identificator unic generat de MongoDB pentru fiecare document (animal).
- `nume`: String - Numele animalului.
- `specie`: String - Specia animalului, de exemplu "Caine", "Pisica", etc.
- `rasa`: String - Rasa animalului.
- `varsta`: Integer - Vârsta animalului în ani.
- `gen`: String - Genul animalului, de exemplu "M", "F".
- `disponibil`: Boolean - Indică dacă animalul este disponibil pentru adopție sau nu.
- `greutate`: Double - Greutatea animalului în kilograme.
- `fotografie`: Binary - Fotografia animalului stocată în format binar.
- `vaccinari`: Array - Un array de obiecte care conține detalii despre vaccinările animalului.

Fiecare obiect conține:

- `_id`: ObjectId - Un identificator unic pentru fiecare înregistrare a vaccinului.
- `nume_vaccin`: String - Numele vaccinului.
- `data_vaccin`: Date - Data când vaccinul a fost administrat.

#### **persoane:**

- `_id`: ObjectId - Un identificator unic generat de MongoDB pentru fiecare document (persoana).
- `nume`: String - Numele persoanei.
- `prenume`: String - Prenumele persoanei.
- `date_contact`: Obiect - Un obiect care conține detalii de contact ale persoanei, care include:
  - `adresa`: String - Adresa persoanei.
  - `telefon`: String - Numărul de telefon al persoanei, validat cu o expresie regulată pentru a se asigura că este un număr de 10 cifre.
  - `email`: String - Adresa de e-mail a persoanei, validată cu o expresie regulată pentru a se asigura că este o adresă de e-mail validă.
  - `oras`: String - Orașul în care locuiește persoana.

#### **adoptii:**

- `_id`: ObjectId - Un identificator unic generat de MongoDB pentru fiecare document (adoptie).
- `id_animal`: ObjectId - Referință la ID-ul animalului adoptat.
- `id_persoana`: ObjectId - Referință la ID-ul persoanei care a adoptat animalul.
- `data`: Date - Data când a fost realizată adopția.
- `status`: String - Statusul adopției, de exemplu "rezervat", "adoptat", etc

### **3. Descrierea funcționalităților și a operațiilor corespunzătoare din fișierele de script**

#### **a. + b. Operații CRUD**

În continuare voi descrie cum am folosit operațiile CRUD pentru colecțiile de mai sus:

**Create:** operația `createCollection` este folosită pentru a crea o nouă colecție în baza de date

**Insert:** include `insertMany` și `insertOne`. Acestea sunt utilizate pentru a introduce noi înregistrări în colecții. `insertMany` introduce mai multe înregistrări în același timp, în timp ce `insertOne` introduce o singură înregistrare.

Am realizat:

- Înserare multiplă și singulară de elemente în colecția animale
- Înserare multiplă și singulară de elemente în colecția persoane
- Înserare multiplă de elemente în colecția adoptii

**Delete:** Include deleteOne și deleteMany. Acestea sunt utilizate pentru a șterge înregistrări din colecțiile MongoDB. deleteOne șterge o singură înregistrare, în timp ce deleteMany poate șterge mai multe înregistrări simultan.

Am realizat:

- Ștergerea primei înregistrări de adopție pentru un anumit animal
- Ștergerea tuturor înregistrărilor de adopție făcute de o persoană
- Ștergerea animalelor care sunt câini
- Ștergerea unei persoane după nume și prenume
- Ștergerea tuturor persoanelor din orașul Cluj

**Update:** Include updateOne și updateMany. Acestea sunt utilizate pentru a actualiza înregistrările existente în colecțiile MongoDB. updateOne actualizează o singură înregistrare, în timp ce updateMany poate actualiza mai multe înregistrări simultan.

Am realizat:

- Modificarea statusului după id-ul unui animal
- Schimbarea datei adopției
- Schimbarea persoanei care a adoptat animalul
- Modificarea disponibilității unui animal
- Adăugarea unui vaccin pentru un animal
- Modificarea numelui unui vaccin (al doilea) pentru un animal
- Actualizarea numărului de telefon a unei persoane
- Actualizarea adresei unei persoane

**Bulk:** MongoDB oferă operațiunea bulkWrite pentru a efectua operațiuni CRUD în lot. Aceasta poate include o combinație de operațiuni insertOne, updateOne, updateMany, deleteOne și deleteMany.

Am realizat:

**Adoptii bulk:** Am realizat următoarele operațiuni în colecția adoptii:

- InsertOne: Am introdus două noi înregistrări de adopție cu specificarea detaliilor fiecărei adopții, inclusiv ID-urile persoanei și animalului, data și statusul.
- UpdateMany: Am actualizat toate înregistrările care aveau statusul "rezervat", schimbându-le în "adoptat".
- DeleteMany: Am șters toate înregistrările care aveau statusul "adoptat".

**Animale bulk:** Operațiunile realizate în colecția animale sunt:

- InsertOne: Am adăugat două noi înregistrări de animale, furnizând detaliile fiecărui animal, inclusiv numele, specia, rasa, vârsta, genul, disponibilitatea, greutatea și lista de vaccinări.
- UpdateMany: Am actualizat toate înregistrările cu specia "Caine", setându-le ca indisponibile.
- DeleteMany: Am șters toate înregistrările care erau marcate ca indisponibile.

**Persoane bulk:** Operațiunile în colecția persoane sunt:

- InsertOne: Am adăugat două noi înregistrări de persoane, oferind detaliile fiecărei persoane, inclusiv numele, prenumele și informațiile de contact.
- UpdateMany: Am actualizat orașul tuturor persoanelor care locuiau în "Iasi", schimbându-l în "Bucuresti".
- DeleteMany: Am șters toate înregistrările persoanelor care locuiau în "Bucuresti".

### c. Căutări simple

În interogările pe care le-am realizat, am utilizat operația "read" din CRUD, care în MongoDB este realizată prin metoda find(). Aceasta permite căutarea înregistrărilor din colecția specificată. Metoda find() poate primi două argumente - un obiect de interogare și un obiect de proiecție.

Am realizat următoarele interogări:

**Căutarea documentelor/subdocumentelor/elementelor din vector folosind find() cu diverse opțiuni**

- interogare care caută toate animalele cu numele "Bella"
- interogare care caută toate persoanele care au ca oraș "Iasi" în subdocumentul date\_contact.
- interogare care caută persoanele al căror nume începe cu 'P' sau al căror email se termină cu '@student.tuiasi.ro'

**Proiecția atributelor, valori null sau attribute inexistente**

- interogare care caută toate animalele care nu au un atribut fotografie
- interogare care caută toate adopțiile cu statusul "rezervat" și returnează doar anumite attribute precum id-ul animalului și al persoanei pentru fiecare înregistrare găsită

**Sortări**

- interogare ce returnează toate persoanele, sortate după nume și apoi după prenume în ordine alfabetică

**Căutare paginată pentru unele operații de căutare**

- Interogare ce returnează adopțiile care au avut loc înainte de 2024, sortate descrescător după dată. De asemenea, aplică paginarea, returnând un număr specific de înregistrări de pe o anumită pagină
- Paginarea după cainii care au vaccinul pentru Rabia
- Paginarea în ordine descrescătoare după numele persoanelor din Bucuresti al căror număr de telefon începe cu '07'

## d. Căutări complexe

Am prezentat mai multe tehnici avansate de lucru cu MongoDB, inclusiv folosirea cursorilor, frameworkul de agregare și conceptul de pipeline.

### **Folosirea cursorilor pentru find() pentru procesări suplimentare**

Cursorii MongoDB permit procesarea fiecărui document returnat de o interogare. Am folosit un cursor pentru a itera prin fiecare document din colecția "persoane" și a afișa numele și prenumele.

De asemenea, am folosit cursori pentru a asocia adopțiile cu persoanele și animalele corespunzătoare, demonstrând astfel posibilitatea de a combina date din mai multe colecții.

### **Folosirea frameworkului aggregate (operatori de tip group, push/addFields, project, unwind, sort, limit) si conceptul de pipeline pentru înlănțuirea operatorilor**

Am folosit frameworkul de agregare MongoDB pentru a executa interogări complexe care implică manipularea și transformarea datelor.

Un exemplu este interogarea în care am folosit operatorul \$lookup pentru a combina datele din colecțiile "adoptii" și "persoane", și operatorul \$unwind pentru a diviza fiecare document în funcție de câmpul "persoana\_info". Am folosit, de asemenea, \$project pentru a alege ce câmpuri să afișez și \$addFields pentru a adăuga un nou câmp care calculează numărul de adopții pentru fiecare persoană.

Un alt exemplu de utilizare a frameworkului de agregare este interogarea în care am grupat animalele după specie, am colectat numele tuturor animalelor pentru fiecare specie și am sortat rezultatele în funcție de numărul de animale pentru fiecare specie.