UNIVERSITATEA TEHNICĂ “GHEORGHE ASACHI” DIN IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

DISCIPLINA EXTRAGEREA CUNOȘTIINȚELOR DIN BAZE DE DATE

PROIECAREA UNEI BAZE DE DATE MONGODB PENTRU MONITORIZAREA STARII PACIENTILOR

Coordonator,

Prof. Catalin Mironeanu

Student,

Pasa Larisa, grupa 1410A

Cuprins

1. **Descrierea proiectului**

Scopul proiectului este de a implementa o soluție de gestionare a fiselor de observație ale pacienților dintr-un anumit sector medical prin intermediul unei baze de date NoSQL MongoDB. Această platformă are ca obiectiv centralizarea și organizarea informațiilor legate de pacienți, doctori și fise medicale într-un format structurat și ușor accesibil.

Prin intermediul celor trei colecții distincte - pacienti, doctori și fise medicale - proiectul asigură o metodă coerentă și sistematică de stocare a datelor. Astfel, informațiile clinice și observațiile medicale notate de către doctori sunt documentate în mod detaliat și stocate în baza de date, facilitând astfel monitorizarea, analiza și gestionarea acestora.

De asemenea, structura proiectului permite o integrare eficientă între diferitele entități implicate. De exemplu, informațiile notate în fisa medicală a unui pacient sunt actualizate și în istoricul medical al aceluiași pacient, asigurând astfel coerența și continuitatea informației medicale.

2. **Structura colectiilor**

2.1 Doctors:

Colectia "doctors" reprezintă un set organizat de date care păstrează informații despre cadrele medicale angajate în sistemul de sănătate. Aceste date sunt potrivite pentru identificarea și contactarea rapidă a medicilor în situații medicale urgente sau pentru referințe medicale specifice. Prin colectarea și organizarea acestor date, colecția "doctors" facilitează accesul la resursele medicale și imbunatateste coordonarea îngrijirii pacienților.

Structura colectiei cuprinde:

- _id: String – cnp-ul doctorului
- full_name: { last_name:String, first_name : String}
- specialization: String
- personal_details:{
  - address: String
  - phone: String
  - email: String

}

2.2 Patients:

Colectia "patients" constituie un ansamblu de informații ce reflectă profilurile medicale ale pacienților. Colectia conține istoricul medical al fiecărui pacient, care cuprinde detalii despre diagnosticuri anterioare, tratamente urmate și datele vizitelor

medicale. Prin centralizarea și gestionarea corecta a acestor date, colecția "patients" contribuie la asigurarea unui tratament medical acordat pe baza istoricului propriu.

Structura colectiei cuprinde:

- _id: String
- full_name: { last_name:String, first_name : String}
- personal_details:{
  - age: Integer
  - gender: String
  - address: String
  - phone: String
  - email: String

}

- medical_history: Array(
  {
    - visit_date: Date
    - diagnostic: String
    - treatment: String

  }

)

Id-ul colectiei este reprezentat de C.N.P.-ul pacientului, fiind un identificator sugestiv pentru fiecare document. Vectorul cu istoric medical cuprinde toate vizitele pacientului la spitalul care utilizeaza aplicatia . El este actualizat automat cu o noua vizita de fiecare data cand se creeaza o fisa de observatie .

2.3 Patient Observation Sheet:

Colectia "patient_obs_sheet" este un set organizat de date care conține observații medicale detaliate asociate fiecărui pacient înregistrat. Aceasta reunește informații despre diagnostic, tratament și evoluția stării de sănătate a pacienților, incluzând parametri vitali și simptome observate. Colecția este gestionată de cadrele medicale responsabile cu îngrijirea pacienților. Datele conținute în colecție oferă informații pentru adaptarea planurilor de tratament si pentru mentinerea unei evidente a starii pacientului.

Structura colectiei cuprinde:

- _id: String
- doctor: String
- patient: String
- active_status: Boolean
- diagnostic: String
- treatment: String
- observations: Array(

{

- observation_date: Date
- serious_condition: Boolean
- heart_rate: Integer
- oxygen_saturation: Array(Integer)
- temperature : Double
- symptoms: Array(String)
- radiography: BinaryData(00)

}

)

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

3.1 **Structura colectiilor:**

Crearea structurii default a colectiilor se realizeaza prin 3 functii care cuprind operatii de tip deleteMany() si insertMany() pentru a curata si popula colectia la fiecare rulare a functiei:

- initialPopulateDoctors – insereaza doctori cu valori pentru toate atributele
- initialPopulatePatients – insereaza pacienti cu cate un element in vectorul de istoric medical
- initialPopulatePatientObsSheet – insereaza fise de observatie pentru cativa dintre pacientii inserati default, insa fara a adauga radiografii (camp null default)

3.2 Operatii CRUD:

Fiecare funcție asociată operațiilor CRUD este definită pentru a efectua prelucrări de bază, însă într-un mod astfel structurat încât anumite operații să aibă ca efect modificarea celorlalte colecții, menținând un context corect și coerent în cadrul unui sistem medical integrat

3.2.1 Colectia „doctors”:

- **checkDoctorEmployment** - verifică dacă un medic cu CNP-ul specificat este deja angajat în sistem.
- **createDoctor** - Funcția creează un nou medic în sistem cu detaliile furnizate. Înainte de inserare, se verifică dacă medicul este deja angajat folosind funcția **checkDoctorEmployment**
- **updateDoctorPersonalDetails** - actualizează detaliile personale ale unui medic existent identificat prin CNP-ul său. Verifică mai întâi dacă medicul este angajat în sistem.

- **deleteDoctor** - șterge un medic din sistem. Dacă medicul are fișe de observație active asociate, se alege un alt medic aleatoriu pentru acele fișe.

3.2.2 Colectia „patients”:

- **checkPatientExistence** - verifică dacă un pacient cu CNP-ul specificat este înregistrat în sistem
- **createPatient** - creează un nou pacient în sistem cu detaliile furnizate. Înainte de inserare, se verifică dacă pacientul este deja înregistrat folosind funcția **checkPatientExistence**.
- **addPatientMedicalHistory** - adaugă o nouă înregistrare în istoricul medical al unui pacient existent identificat prin CNP-ul său. Este folosita in functia **createPatientSheet** pentru a inregistra in istoric orice vizita medicala.
- **deletePatient** - șterge un pacient din sistem și invalideaza toate fișele medicale asociate pacientului.

3.2.3 Colectia „patient_obs_sheet”:

- **checkObservationSheetExistence** - verifică dacă o fișă de observație activă există între un medic și un pacient specificați.
- **createPatientSheet** - creează o nouă fișă de observație pentru un pacient și un medic specificați. Verifică dacă pacientul și medicul sunt înregistrati și dacă o fișă activă nu există deja între ei.
- **deactivateSheet** - dezactivează o fișă de observație existentă între un medic și un pacient specificați. Functie folosita in cazul in care un pacient este sters din baza de date.
- **addObservationsToSheet** - adaugă noi observații la o fișă de observație existentă între un medic și un pacient specificați. Radiografia poate fi un camp populat sau poate fi default null.
- **addRadiographyToObservation** – cum radiografiile si alte materiale medicale in format digital necesita timp de prelucrare, ele pot fi disponibile dupa ce a fost creata observatia corespunzatoare unei anumite zile de internare a pacientului.
- **updateOxygenSaturation** - actualizează nivelul de saturație cu oxigen într-o observație specifică dintr-o fișă de observație între un medic și un pacient specificați.

3.3 Cautari simple:

Fiecare functie din acest subpunct contine cautari, filtrari simple, unele avand parametri, altele nu.

- **findCardiologists** - identifică toți cardiologii din sistem, specializați în domeniul cardiologiei.
- **findPatientsAboveAge** - selectează pacienții cu vârsta peste 40 de ani din baza de date medicală.
- **findObservationSheetsWithFeverSymptoms** - localizează fișele de observație care conțin informații despre simptome de febră.
- **findPatientsSortedByAge** - afișează pacienții ordonați descrescător în funcție de vârstă.
- **findDoctorsNamesAndSpecializations** – aplicand proiectia, listează numele și specializările tuturor medicilor din sistem.
- **findObservationSheetsWithoutRadiography** - identifică fișele de observație care nu conțin informații despre radiografii.
- **findPaginates** – afisare paginata in functie de parametrii skip si limita elementelor in pagina

3.4 Cautari complexe:

Functiile implementate în acest subpunct sunt structurate în trei categorii distincte: prima categorie se referă la funcții care se concentrează pe utilizarea cursorilor, a doua categorie include funcții care utilizează framework-ul aggregate pentru prelucrarea datelor, iar cea de-a treia categorie este reprezentată de funcții care utilizează operatorul $lookup pentru interogarea datelor din mai multe colecții.

- Cursori:
  - **iterateCursor** - funcție care utilizează un cursor pentru a intera prin colecția patients și a afișa anumite câmpuri ale fiecărui document.
  - **findMean_HeartRate** - funcție care calculează media ratei cardiace pentru toți pacienții utilizând un cursor pentru a intera prin colecțiile patients și patient_obs_sheet.
  - **findMean_Temperature** - funcție care calculează media temperaturii pentru pacienții cu simptomul specificat, utilizând un cursor pentru a intera prin colecția patient_obs_sheet.
- Aggregate:
  - **countObservationSheetsForDoctor** - Această funcție utilizează agregarea pentru a determina numărul de fișe de observație asociate fiecărui doctor. Operatorul **$group** este folosit pentru a grupa înregistrările după **doctor** și a calcula numărul total de fișe pentru fiecare doctor folosind **$sum**. Rezultatul este apoi sortat în ordine descrescătoare pentru a identifica cei mai activi doctori în ceea ce privește numărul de fișe de observație.
  - mainSymptomsForPatient - Această funcție folosește agregarea pentru a identifica simptomele principale raportate de fiecare pacient. Prin utilizarea operatorilor **$unwind**, se descompun înregistrările și simptomele într-o structură liniară, permițând apoi

gruparea lor utilizând **$group**. Folosind **$addToSet**, se asigură că simptomele sunt unice pentru fiecare pacient, fără duplicări. Rezultatul este formatat pentru a afișa simptomele pentru fiecare pacient.

- classifyPatientsByHeartRate - Această funcție utilizează agregarea pentru a identifica primii trei pacienți cu cele mai ridicate rate ale pulsului. Prin utilizarea **$unwind**, fiecare înregistrare este descompusă într-o singură observație, permițând calculul sumei tuturor ratelor de puls folosind **$sum** în cadrul operatorului **$group**. Rezultatele sunt sortate în ordine descrescătoare după suma ratelor de puls și limitate la primele trei înregistrări pentru a identifica pacienții cu cele mai ridicate rate ale pulsului.

- Lookup:
  - **getPatientsWithSympton** - Această funcție utilizează agregarea împreună cu operatorul **$lookup** pentru a obține detalii complete despre pacienții care prezintă un simptom specific, cum ar fi febra. Inițial, se folosește **$match** pentru a filtra înregistrările care conțin simptomul specificat. Apoi, cu ajutorul **$unwind**, se descompun înregistrările și simptomele pentru a lucra cu ele individual. Utilizarea **$lookup** combină colecția **patient_obs_sheet** cu colecția **patients**, permițând extragerea detaliilor complete ale pacienților. Operatorul **$project** este folosit pentru a selecta și proiecta doar informațiile relevante în rezultatul final, cum ar fi ID-ul și numele pacientului, detaliile personale, temperatura și simptomele raportate.
  - **findMean_OxygenLevel** - Această funcție calculează nivelul mediu de oxigen pentru fiecare pacient. Prin utilizarea operatorului **$unwind**, înregistrările sunt descompuse în observații individuale și niveluri de saturare a oxigenului. Apoi, cu ajutorul **$group**, se grupează înregistrările după ID-ul pacientului și se calculează media nivelurilor de saturare a oxigenului folosind **$avg**. Operatorul **$lookup** este utilizat pentru a combina colecția **patient_obs_sheet** cu colecția **patients**, permițând adăugarea detaliilor complete ale pacienților în rezultat. În final, operatorul **$project** este folosit pentru a proiecta doar numele pacientului și nivelul mediu de oxigen în rezultatul final.