

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 UNEI LIBRĂRII

Coordonator,
Ș.l.dr.ing. Cristian Aflori

Student,
Modiga Camelia-Maria, 1410B

An universitar 2021-2022

1. Descrierea proiectului

Se propune analiza, proiectarea și implementarea unei baze de date care să modeleze gestiunea unei librării cu ajutorul MongoDB.

Proiectul trebuie să surprindă posibilitatea salvării unei comenzi cu toate produsele aferente, să ofere un total de plată pentru client dar și o afișare organizată a listei de cumpărături.

Administratorul magazinului trebuie să aibă posibilitatea de a adăuga noi cărți, de a modifica stocul cărților deja existente, de a adăuga noi clienți și de a modifica informațiile acestora, să creeze o nouă comandă și să modifice informațiile referitoare la o comandă deja existentă. De asemenea, dacă magazinul hotărăște să nu mai colaboreze cu o anumită editură vor fi șterse toate cărțile de la acea editură iar în cazul în care magazinul hotărăște să nu mai livreze într-un oraș este permisă ștergerea clienților din orașul respectiv.

În cazul în care există posibilitatea unui număr foarte mare de clienți, cărți și comenzi trebuie implementate căutări avansate care să conducă la o organizare benefică a magazinului: căutarea cărților după autor sau editura acestora, căutarea clienților după nume și prenume, după oraș sau căutarea comenzilor în funcție de suma pe care o are de plătit clientul.

2. Structura colecțiilor

Aplicația este organizată în trei colecții principale: cărți, clienți și comenzi și o colecție auxiliară: counter

1. Cărți

Reprezintă totalitatea cărților pe care le deține magazinul. Această colecție conține documente care oferă detalii despre titlul cărții prin câmpul “titlu” de tipul șir de caractere, genul literar prin câmpul “gen_literar” care este de tipul vector de valori scalare, autorul prin câmpul “autor” de tipul șir de caractere, “editura” de tipul șir de caractere, numărul de cărți disponibile prin câmpul “cantitate” de tip integer, prețul de tip double și câmpul “disponibil” de tip boolean care îi arată administratorului magazinului dacă o carte poate fi vândută. Câmpul _id este autodefinit de către MongoDB.

```
carti{
  _id: ObjectId,
  "titlu": string,
  "gen_literar": string,
  "autor": string
  "editura": string,
  "cantitate": integer,
  "pret": double,
  "disponibil": boolean
}
```

2. Clienți

În această colecție se vor reține informații referitoare la clienții magazinului. Un document de tip client are în vedere detaliile de identificare ale acestuia, câmpuri de tip șir de caractere, nume, prenume și un subdocument care conține datele de contact ale clientului (telefon, email, oraș). Câmpul “email” trebuie să fie de forma *@yahoo.com* pentru a putea fi inserat. Câmpul `_id` este autodefinit de către MongoDB.

```
clienti{
  _id: ObjectId,
  "nume": string,
  "prenume": string,
  "date_contact":
  {
    "telefon": string,
    "email": string,
    "oras": string
  }
}
```

3. Comenzi

În această colecție se vor reține informații referitoare la comenzile pe care le face un client. Un document de tip comandă conține un vector de cărți, în fiecare element are informații organizate ca un subdocument care conține un identificator al cărții de tipul `ObjectId` dar și cantitatea comandată. Importante pentru structura comenzii sunt și câmpurile: “`id_client`” prin care o comandă este asociată cu un client, “`data`” care reprezintă data calendaristică în care a fost generată comanda dar și totalul care reprezintă suma pe care o are de plătit clientul. Identificatorul fiecărei cărți reprezintă modul în care colecția comenzi este legată de cărți iar câmpul “`id_client`” modul în care colecția comenzi este legată de clienți. Atributul `_id` este definit de către utilizator.

```
comenzi{
  _id: integer,
  "id_client": ObjectId,
  "carti":[
    {
      "id_carte": ObjectId,
      "cantitate": integer
    }
  ],
  "data": date,
  "total": double
}
```

4. Counter

Reprezintă o colecție auxiliară folosită pentru a putea defini câmpul `_id` din colecția “comenzi” prin atribuirea unei valori de tip `integer`. Atributul “`_id`” este folosit în funcția JavaScript *getValueForNextSequence* pentru a putea identifica secvența a cărei valoare trebuie incrementată. Câmpul “`sequence_value`” este utilizat pentru a putea incrementa câmpul “`_id`” din colecția “comenzi”.

```
counter{
  "_id": string,
  "sequence_value": integer
}
```

3. Descrierea funcționalităților și a operațiilor corespunzătoare

Proiectul oferă posibilitatea adăugării de clienți, comenzi și cărți. Aplicația oferă posibilitatea inserării unui singur document, a unor documente multiple într-un mod secvențial dar și paralel prin `bulkWrite`. La inserarea comenzilor câmpul `total` va fi setat pe `null`, el urmând a fi calculat ulterior.

Operațiile de actualizare posibile în colecția “cărți” sunt cele de modificare a stocului unei cărți, modificarea cantității disponibile pentru o carte în funcție de editura pe care o are dar și adăugarea unui nou gen literar unui document deja existent. Colecția “clienți” oferă posibilitatea actualizării documentelor în felul următor: actualizarea numărului de telefon a unui client în funcție de numele și prenumele acestuia, actualizarea adresei în funcție de numele clientului dar și modificarea orașului din datele de contact. De asemenea, colecția “comenzi” permite adăugarea unui nou produs la o comandă existentă, modificarea cantității pe care un client dorește să o achiziționeze și modificarea totalului unei comenzi în funcție de data la care a fost plasată comanda.

În cazul în care un client dorește să nu i se mai prelucreze datele, este posibilă ștergerea acestuia pe baza numelui și a prenumelui, iar în cazul în care magazinul hotărăște să nu mai livreze într-un oraș este permisă ștergerea clienților din acel oraș. Colecția “comenzi” are implementată ștergerea unei comenzi atunci când totalul este nul sau ștergerea comenzilor dintr-un anumit interval de timp iar colecția “cărți” are implementată ștergerea cărților scrise de un autor specificat și ștergerea cărților de la o anumită editură.

Căutarea datelor este relevantă în fiecare colecție. Astfel, se pot afișa: toți clienții ordonați alfabetic, adresele clienților al căror nume de familie începe cu litera “M”, toți clienții care locuiesc în Iași, clienții care au lipsă câmpul “email” dar și afișarea clienților dintr-un oraș specificat. De asemenea, pentru colecția “cărți” sunt implementate următoarele căutări de documente: afișarea documentelor care au valoarea `null` pentru câmpul “cantitate” și au valoarea “Epica” pentru câmpul “editura”, afișarea titlurilor

cărților care sunt disponibile de la editura "Epica", afișarea cărților scrise de "Ali Hazelwood", căutarea cuvântului "Ali" în colecția cărți, afișarea cărților de la editura "Curtea Veche" care au prețul mai mare de 30 lei, afișarea ordonată descrescător a cărților după preț de la editura "Curtea Veche". Numărul mare de cărți conduce la implementarea unei căutări paginate. Astfel se folosește operatorul skip pentru a trece peste numărul de pagini înmulțit cu numărul de cărți pe pagina și apoi o limitare cu numărul de cărți pe pagină. Căutarea indexată este realizată la nivelul oraș pe colecția clienți. Pentru colecția "comenzi" au fost implementate afișarea comenzilor cu un total între 100 și 200 de lei și afișarea comenzilor cu 3 produse ordonate după total.

Cursorii sunt folosiți pentru implementarea operațiilor complexe. Cu ajutorul lor se poate face legătura între cele trei colecții. Astfel, au fost efectuate următoarele operații: calcularea totalului unei comenzi, afișarea comenzilor pentru fiecare client, afișarea cărților și a cantității vândute din fiecare carte.

Agregările sunt utilizate pentru: afișarea numărului de documente din colecția cărți, gruparea cărților în funcție de numele autorului, afișarea numărului de cărți disponibile în funcție de numele autorului, afișarea numărului de clienți din fiecare oraș, afișarea primelor 3 orașe cu cei mai mulți clienți, gruparea cărților în funcție de editura și afișarea celei mai scumpe cărți, celei mai ieftine carti și prețul total de la editura respectiva, afișarea zilelor în care s-au vândut cele mai multe cărți.

Cu ajutorul map-reduce au fost implementate afișarea numărului de cărți de la fiecare editură, afișarea clienților grupați pe orașe, gruparea cărților în funcție de editura și afișarea celei mai scumpe cărți, celei mai ieftine carti, gruparea comenzilor pe zile și calcularea totalului pe zile.