



UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” DIN IAȘI

FACULTATEA DE AUTOMATICP ȘI CALCULATOARE

SPECIALIZAREA: TI

EXTRAGEREA CUNOȘTIȚELOR DIN BAZA DE DATE

# Organizarea unei școli de balet

Student: Iorga Elena, 1411B

Cadru didactic coordinator: Ș.l.dr.ing. Cristian Aflori

## 1. Descrierea proiectului (scopul aplicației)

Baza de date „Organizarea unei școli de balet” are ca scop oferirea de suport pentru împărțirea studenților școlii de balet pe grupe, asignarea instructorilor și a pianistilor grupei, organizarea spectacolelor și a asignării sălilor pentru repetiții. Este necesară o astfel de structură întrucât este greu să se țină o astfel de evidență fără un ajutor structurat. A fost aleasă o structură de tip NoSql pentru ușurința modelării datelor având în vedere că aceasta permite adăugarea de subdocumente (ex studenții unei grupe), astfel nefiind nevoie să existe o tabelă separată pentru studenți, reducând complexitatea problemei.

O grupă are diverși studenți și poate participa la spectacole. Totodată, aceasta trebuie să aibă un instructor și un pianist, fie certificat (profesionist) sau nu. Fiecare grupă are ca id numele grupei pentru ușurință în folosirea acestora în celelalte tabele pentru join-uri. Fiecărei grupe îi sunt asignate anumite săli în care poate să facă repetiții, acestea având ca proprietăți capacitatea (câte persoane încap în sală) și dimensiunea acestora astfel încât să fie facilitată efectuarea anumitor verificări (de ex: încap toți elevii dintr-o grupă într-o anumită sală?).

Studenții școlii de balet au opțiunea de a participa la spectacole care sunt planificate pe o anumită dată. Tot ce trebuie să facă este să aibă grupa înscrisă la respectiva serbare. Totodată, ei sunt obligați să își completeze anumite date personale dar nu pe toate luând în considerare eventualitatea în care studentul respectiv nu poate oferi acele informații.

Această bază de date poate fi folosită de administratorii școlii pentru a ține evidența proceselor instituției și a introduce date noi atunci când este nevoie.

## 2. Structura colecțiilor

### 1. „Grupe” :

- `_id` : String
- `instructor` : (object)
  - o `nume`: String
  - o `prenume`: String
  - o `an_incepere`: Integer
  - o `certificat`: Boolean
- `pianist` : (object)
  - o `nume`: String
  - o `prenume`: String
  - o `an_incepere`: Integer
  - o `certificat`: Boolean
- `students` : Array(object)
  - o Array(
    - `nume`: String

- prenume: String
- detalii\_personale:
  - numar\_telefon : String (Regex “^0[0-9]{9}\$”) \*\*\*
  - data\_nastere: Date
  - cnp: String

)

Colecția „grupe” este folosită pentru a stoca studenții care va fi în cazul acesta un vector de subdocumente unde fiecare subdocument corespunde unui student. Despre acesta se vor cere: numele, prenumele, detaliile personale (numărul de telefon sub formă de String care este inițializat ca un Regex astfel încât să înceapă cu cifra 0 și să aibă alte 9 cifre în rest și cnp-ul). Totodată, fiecărei grupe îi este asignat câte un pianist și un instructor, fiecare dintre acestea fiind câte un document.

Totodată, id-ul va fi reprezentat de numele grupei și nu de ObjectId predefinit de mongo pentru ușurința de a adăuga grupe atât la săli cât și la spectacole.

Numărul de telefon este marcat cu trei steluțe mai sus deoarece acesta nu este un câmp obligatoriu. Pentru pianiști și instructori, valorile pentru „certificat” pot fi și nule.

## 2. Sali

- \_id: ObjectId
- Dimensiune : (object)
  - latime: Integer
  - lungime: Integer
- Capacitate : Integer
- Grupe: (Array(String))
- Numar\_sală : String

Tabela „săli” are rolul de a stoca camerele folosite pentru repetițiile grupelor. Pentru a asigna grupele într-o anumită sală, se adaugă la vectorul „grupe” id-ul grupelor în cauză, acesta fiind chiar numele grupei. Bineînțeles, fiecare grupă pe lângă ID are un nume pentru a fi recunoscută ex: 4A. Fiecare sală are o anumită dimensiune (caracterizată prin lățime și lungime) și o anumită capacitate pentru a verifica cât de mare e grupa care are loc în acel spațiu.

## 3. Spectacole

- \_id: ObjectId
- Nume: String
- Grupe : Array(String)
- Data : Date

Tabela „spectacole” are rolul de a ține evidența evenimentelor ce vor avea loc la o anumită dată. Fiecare spectacol are o temă care e reprezentată prin numele spectacolului, iar la acesta pot participa oricâte grupe vor, dar cu condiția de a se înscrie.

Se poate observa că atât între tabelele săli și grupe cât și între grupe și spectacole, pentru a accesa de exemplu ce studenți merg la ce spectacole sau în ce săli sunt repartizați anumiți elevi, este nevoie de join-uri între tabele.

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

#### a. Structura colecțiilor

Pentru acest subpunct au fost realizate schemele folosind \$jsonSchema colecțiilor pentru a fixa parametrii obligatorii și cei care nu sunt necesari. Totodată au fost hotărâte tipurile fiecărui obiect al documentului (ex: bsonType: "int") și au fost fixați parametrii de tip Regex (ex: pattern: "^0[0-9]{9}\$").

- Valorile de id **autodefinite**: cele corespunzătoare colecțiilor săli și spectacole
- Valorile de id **definite de utilizator**: colecția grupe
- Atributele **scalare** folosite:
  - **String** : \_id (grupe), nume(grupe), prenume(grupe), nume(grupe.pianist), prenume(grupe.pianist), nume(grupe.studenti), prenume(grupe.studenti), numar\_telefon(grupe.studenti.detalii\_personale), cnp(grupe.studenti.detalii\_personale), numar\_sala (Sali)
  - **Integer** : an\_incepere(grupe.instructor), an\_incepere(grupe.pianist), latime(Sali.dimensiune), lungime(Sali.dimensiune), capacitate(Sali)
  - **Boolean**: certificat (grupe.instructor), certificat (grupe.pianist)
  - **Date**: data(spectacole), data\_nastere (grupe.studenti.detalii\_personale)
  - **Regular expression**: numar\_telefon (grupe.studenti.detalii\_personale)
- Atribute de tip **vector de valori scalare**: grupe (Sali) : Array(String), grupe (spectacole) : Array(String)
- Atribute de tip **subdocument** : pianist (grupe), instructor (grupe), dimensiune (Sali)
- Atribute de tip **vector de subdocumente** : studenti (grupe)

#### b. Operații CRUD

- **insertGrupe()** – funcție ce are ca rol golirea inițială a colecției grupe pentru a ne asigura că nu am inserat deja datele, iar apoi folosind funcția insertMany inserează 3 grupe diferite cu id-urile „Mare”, „Mică”, „Mijlocie” și le afișează după folosind cursori
- **insertSali()** - funcție ce are ca rol golirea inițială a colecției săli pentru a ne asigura că nu am inserat deja datele, iar apoi folosind funcția insertMany inserează 3 săli și le afișează folosind cursori
- **insertSpectacole()** - funcție ce are ca rol golirea inițială a colecției spectacole pentru a ne asigura că nu am inserat deja datele, iar apoi folosind funcția insertMany inserează 5 spectacole cu numele „Dansând printre stele”, „Primăvara”, „Ghiociei”, „Vara” și le afișează folosind cursori
- **updateDataSpectacole(numaSpectacol, noua\_data)** – funcție ce modifică data pe care este planificat un anumit spectacol în caz că este nevoie de o amânare sau de mutarea spectacolului mai devreme folosind \$updateOne și \$set și cursori pentru afișare
- **updateNumeInstructor(numaNou, idGrupa)** – funcție ce modifică numele de familie al unui instructor în caz că acesta s-a căsătorit sau din alte motive și-a schimbat numele folosind \$updateOne și \$set și cursori pentru afișare
- **changeInstructor(newNume, newPrenume, newAn, certificare, idGrupa)** – funcție ce schimbă un instructor al unei anumite grupe date ca id cu un altul folosind \$updateOne și \$set și cursori pentru afișare

- **changePianist(newNume, newPrenume, newAn, certificare, idGrupa)** - funcție ce schimbă un pianist al unei anumite grupe date ca id cu un altul folosind \$updateOne și \$set și cursori pentru afișare folosind \$updateOne și \$set și cursori pentru afișare
- **addStudent(umeStudent, prenume, dataNastere, numarTelefon, idGrupa, cnps)** – funcție ce înrolează un nou student într-o anumită grupă folosind \$updateOne și \$push și cursori pentru afișare
- **schimbaNumarTelefonStudent(umeStudent, idGrupa, newTelefon)** – funcție ce modifică numărul de telefon al unui student în caz că este nevoie folosind \$updateOne și \$set și cursori pentru afișare
- **asigneazăSalaUneiGrupe(idGrupa,numarSala)** – funcție responsabilă de asignarea unei grupe unei săli folosind \$updateOne și \$push și cursori pentru afișare
- **schimbăNumeleSpectacolului(oldName, newName)** – funcție responsabilă de modificarea numelui unui spectacol în cazul în care tema acestuia este modificată folosind \$updateOne, \$set și cursori pentru afișare
- **schimbăDataSpectacolului(umeSpectacol, newDate)** – funcție responsabilă de modificarea datei unui spectacol folosind \$updateOne, \$set și cursori pentru afișare
- **adăugareGrupaLaSpectacol(idGrupa, umeSpectacol)** – funcție care înscrie o grupă la un anumit spectacol folosind \$updateOne, \$push și cursori pentru afișare
- **scoateGrupaDinSpectacol(idGrupa, umeSpectacol)** – funcție responsabilă de a șterge o grupă din lista celor care participă la spectacol în caz că vrea să se retraga folosind \$updateOne, \$pull și cursori pentru afișare
- **scoateGrupaDinSala(idGrupa, umeSala)** – funcție responsabilă de a șterge o grupă din lista celor asiguate unei anumite săli în cazul în care nevoile acesteia se schimbă folosind \$updateOne, \$pull și cursori pentru afișare
- **scoateStudent(cnps, idGrupa)** – funcție care va scoate un student dintr-o anumită grupă folosind cnps-ul acestuia în cazul în care renunță la școala de balet, sau în care este exmatriculat sau mutat în altă grupă folosind \$updateOne, \$pull și cursori pentru afișare
- **bulkActualizareCertificareȘiAdăugareGrupe()** – funcție care va folosi \$updateMany pentru a atesta că toți instructorii și pianiștii sunt certificați și pentru a adăuga încă două grupe folosind \$insertOne, toate acestea cu \$bulkWrite
- **bulkActualizareSpectacoleAdaugareȘiStergereGrupe(umeSpectacolVechi,umeSpectacolNou, umeSpectacolDeSters, umeSpectacolDeInlocuit)** – funcție ce va modifica numele unui spectacol folosind \$updateOne, filter și update, apoi va insera un nou spectacol folosind insertOne, va șterge un spectacol existent cu deleteOne iar apoi va înlocui un spectacol existent folosind replaceOne, toate acestea folosind bulkWrite

### c. Căutări simple

- **găseșteGrupa(idGrupa)** – funcția care identifică grupa cu un anumit id din colecția grupe folosind find după \_id și afișează documentul folosind cursori – **căutarea documentelor**
- **găseșteSalaDupăNumăr(numărSala)** – funcția care identifică sala cu un anumit nume folosind find după numele grupe. Dacă nu este nicio grupă găsită cu acel nume, folosind count se va verifica dacă există elemente în cursor și dacă nu, se va afișa mesajul „Sala nu a fost găsită” – **căutarea documentelor**
- **găseșteSpectacolDupăNume(umeSpectacol)** - funcția care identifică spectacolul cu un anumit nume folosind find după numele spectacolului. Dacă nu este niciun spectacol găsit cu acel nume, folosind count se va verifica dacă există elemente în cursor și dacă nu, se va afișa mesajul „Spectacolul nu există” – **căutarea documentelor**

- **găseșteGrupeleDinAnumitaSala(numarSala)** – funcție ce va afișa grupele care sunt repartizate unei anumite săli folosind find pentru numar\_sala și făcând o **proiecție** pentru a afișa doar elementul grupe – **căutarea elementelor**
- **găseștePianistulȘiInstructorulUneiGrupe(idGrupa)** – funcție ce va afișa pianistul și instructorul unei anumite grupe folosind find și **proiecția** pentru a afișa doar pianistul și instructorul – **căutarea subdocumentelor**
- **găseșteSalaCuOAnumităCapacitateȘiDimensiune(capacitateSala, lungimeSala, latimeSala)** – afișează sala care are o anumită capacitate, lungime și lățime – **căutarea documentelor**
- **găseștePianiștiiCertificați()** – funcția care returnează câți pianiști certificați sunt în total folosind funcția count a cursorilor – **căutarea subdocumentelor**
- **găseșteInstructoriiCertificați(page, items\_on\_page)** - funcția care returnează câți instructori certificați sunt în total folosind funcția count a cursorilor și folosind a – **căutarea subdocumentelor + afișare paginată**
- **găseșteLaCeSpectacoleParticipăOGrupa(idGrupa)** – afișează spectacolele la care participă o anumită grupă – **căutarea elementelor**
- **găsireaStudentilorNăscuțiÎnainteDeUnAnumitAnSauCareÎncepCuOAnumităLiteră(nrAn)** : afișează studenții născuți înainte de nrAn folosind \$lt sau studenții a căror nume încep cu litera N folosind /^N/ – **căutarea elementelor din vector**
- **afișareaNumerelorDeTelefonFixe()** – afișează toți studenți care au numere de telefon fix (care încep cu 02) folosind \$elemMatch
- **selectareaSpectacolelorLaCareParticipăMaiMultDeNGrupe(nrGrupe)** – afișează spectacolele la care participă mai mult de nrGrupe grupe folosind \$expr, \$gte și făcând proiecție pentru a afișa doar numele grupelor. Dacă nu există spectacole la care participă mai mult de numărul dat de grupe se va afișa un mesaj. Spectacolele vor fi afișate în ordine alfabetică după nume. - **sortare**
- **selectareaStudentilorCareNuAuIntrodusNumarulDeTelefon()** – va returna studenții care au ca numar de telefon null – **valori null**
- **grupeAfișarePaginată(page, items\_on\_page)** – afișare paginată a grupelor
- **countSpectacoleForGrupa(groupName)** – afișează și returnează la câte spectacole participă o anumită grupă folosind findOne și countDocuments

#### d. Căutări complexe

- **afișeazăVârsteleFolosindCursori()** – funcție ce afișează vârstele studenților folosind cursori
- **getStudentsWithCertainAgesUsingCursor(min, max)** – funcție ce afișează doar numele, prenumele și vârstele studenții care au vârstele cuprinse între min și max folosind cursori
- **getStudentsWithCertainAges(min,max)** - funcție ce afișează doar numele, prenumele și vârstele studenții care au vârstele cuprinse între min și max folosind \$aggregate cu: \$unwind, \$addFields, \$floor, \$divide, \$subtract, \$match, \$project
- **sortareSpectacoleDupaData()** – funcție ce afișează spectacolele sortate în ordine cronologică după dată folosind \$aggregate și \$sort
- **sortareSpectacoleData()** – afișează câți ani mai sunt până la spectacole, numele lor, grupele și le ordonează în ordinea numărului de ani obținut folosind \$aggregate, \$project, \$subtract, \$sort
- **cautăÎnCeSăliSuntRepartizateGrupele()** – pentru fiecare grupă sunt afișate numărul și capacitatea sălii în care este repartizată respectiva grupă folosind \$lookup pentru join, \$unwind pentru valorile obținute din join, \$group pentru a grupa sălile în vectorul săli\_repartizate și \$project pentru a afișa doar numărul sălii și capacitatea

- **afişarePaginatăTotiStudentii(page, items\_on\_page)** – afişarea paginată a tuturor studenţilor folosind aggregate, \$unwind, \$skip, \$limit, \$project
- **outputSpectacolGrupe()** – afişarea grupelor care participă la un anumit spectacol folosind aggregate, \$lookup şi \$project
- **checksGroupFitIntoSala(groupName)** – funcţie care afişează dacă studenţii încap în sala asignată şi afişează mesajul cu dacă încap şi cu capacitatea sălii folosind cursori, aggregate, \$match, \$project, \$size, \$ifNull (pentru a verifica), \$filter