



UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI
FACULTATEA DE AUTOMATICĂ ȘI
CALCULATOARE



SPECIALIZAREA: TEHNOLOGIA INFORMAȚIEI
DISCIPLINA: EXTRAGEREA CUNOSTINTELOR DIN
BAZE DE DATE

Gestionarea vanzarilor de masini intr-o reprezentanta auto

Coordonator,

Asist.dr.ing. Cătălin Mironeanu

Student,

Butnaru Silviu, 1409B

Iași, 2022

1. Descrierea proiectului;

Titlul proiectului: Gestionarea vanzarilor de masini intr-o reprezentanta auto.

In ceea ce priveste proiectul dezvoltat, se prezinta o abordare facila care usureaza procesul de vanzare a automobilelor intr-o reprezentanta auto.

Utilizatorul unei astfel de aplicatii poate sa adauge obiecte (documente): de la automobile (masini), care prezinta attributele specifice acestora pana la reprezentante auto, unde se pot realiza vanzari auto, care inglobeaza mai multi angajati si un numar mare de masini.

De asemenea este expusa functionalitatea de achizitie a unei masini dintr-o reprezentanta.

2. Structura colectiilor;

Sunt expuse 3 colectii principale care faciliteaza acest proces intr-un mod eficient:

- Masini
- Reprezentante
- Achizitii

In ceea ce priveste colectia „Masini”, acesta contine documente cu datele principale ale unei masini:

- `_id`: este VIN-ul masinii, Vehicle Identification Number sau seria de sasiu;
- `serie_motor`: codul unic al motorului masinii, este supus unui anumit tipar validat printr-un regex (Regular expression);
- `marca`: denumirea marcii auto;
- `mode`: modelul autovehiculului specific marcii;
- `an_fabricatie`: anul in care s-a fabricat masina;
- `combustibil`: tipul de carburant utilizat;
- `motor`: subdocument cu mai multe campuri:
 - `capacitate_cilindrica`: capacitatea cilindrica a motorului;
 - `putere`: puterea autovehiculului in KWh (Kilowatt-ora) sau hp (Horsepower);

- cuplu: cuplu motorului in Nm (Newton-metru);
- hybrid: daca masina este sau nu capabila de incarcare electrica;

Masini:

```
{
  "_id": <VIN> (String),
  "serie_motor": (String)),
  "marca": (String),
  "model": (String),
  "an_fabricatie": (Date),
  "combustibil": (String),
  "motor": {
    "capacitate_cilindrica": (Integer),
    "putere": (Integer),
    "cuplu": (Integer)
  },
  "hybrid": (Boolean)
}
```

Din punct de vedere al colectiei „Reprezentante”, acesta contine documente cu datele principale ale unei reprezentante auto:

- nume: numele reprezentantei;
- localitate: localitatea in care este amplasata reprezentanta;
- angajati: vector de subdocumente cu angajatii reprezentantei;
 - nume_angajat: numele angajatului;
 - calificare: calificarea acestuia;
 - marca_specializata: marca de masina pe care este specializat;
- masini: un vector ce contine VIN-urile masinilor aflate in reprezentanta;

Reprezentante:

```
{
  "nume": (String),
  "localitate": (String),
  "angajati": [
    {
      "nume_angajat": (String),
      "calificare": (String),
      "marca_specializata": (String)
    },
    {
      "nume_angajat": (String),
      "calificare": (String),
      "marca_specializata": (String)
    },
    ...
  ],
  "masini": ["_id": <VIN> (String), "_id": <VIN> (String), ...]
}
```

Colectia „Achizitii” contine documente cu datele principale ale unui formular de achizitie auto:

- vin: VIN-ul masinii vandute;
- nume_reprezentanta: numele reprezentantei de la care s-a vandut masina;
- nume_angajat_reprezentanta: numele angajatului din reprezentanta care a vandut masina respectiva;
- data_achizitie: data in care s-a efectuat achizitia;

- pret: pretul achizitiei;

Achizitii:

```
{
    "vin": (String),
    "nume_reprezentanta": (String),
    "nume_angajat_reprezentanta": (String),
    "data_achizitie": (Date),
    "pret": (Double)
}
```

3. Descrierea functionalitatilor si a operatiilor corespunzatoare din fisierele de script;

a) Script structura colectii:

Acest fisier prezinta structura colectiilor si crearea acestora. Pentru fiecare colectie in parte, acestea ii este validata structura prin intermediul functiei de validare cu o structura de tip JSON, prin \$jsonSchema.

b) Script operatii crud:

Fisierul ce contine scriptul cu operatii CRUD contine logica de inserare, update si stergere a documentelor din colectiile prezentate mai sus. Pentru fiecare dintre colectii au fost facute operatiile de inserare a cate unui document dar si a mai multor documente printr-o singura comanda, insertMany, update la un singur document dupa _id dar si a mai multor documente prin updateMany.

Pe langa acestea, stergerea este implementata si ea, de la eliminarea unui singur document dupa un _id, a mai multor documente in functie de un anumit cap pana la stergerea tuturor documentelor.

c) Script cautari simple:

Cautarile simple vor fi cele mai folosite operatii in cadrul unei aplicatii de acest fel iar in fisierul cu scriptul aferent sunt prezentate mai multe tipuri de interogari.

Este prezentata cautarea documentelor folosind find() cu diverse optiuni pentru cautari in vectori/subdocumente, proiectia atributelor, valori null sau attribute inexistente si sortari.

Colectia Masini:

- Verificare serie_motor prin regex
- Primele 2 cele mai noi masini
- A 2 a cea mai veche masina

Colectia Reprezentante:

- Reprezentantele care au masina specifica
- Afisare achizitii mai mici de 40000.00 si care sunt de la reprezentanta Volkswagen SRL descrescator

Alte comenzi regasite in acest fisier de script sunt cele de creare a indecsilor necesari si afisarea utilizarii acestora pentru toate colectiile (masini, reprezentante si achizitii); functii de cautare paginata cu parametri pentru numarul de documente pe pagina si numarul paginii pentru afisare.

d) Script cautari complexe:

Operatiile de cautare descrise mai sus nu sunt singurele functionalitati pe care aplicatia le va folosi. In acest capitol sunt prezentate 2 modalitati de a cauta si procesa date din colectie.

Prima metoda este folosirea cursorilor. In colectia masini sunt afisate automobilele care nu sunt hybrid si au capacitatea cilindrica $< 2000 \text{ cm}^3$, pentru reprezentante, afisarea fiecarui angajat cu ce calificare are si marca pentru care este specializat iar pentru achizitii, calculul total de vanzari.

A doua metoda ilustreaza functionalitatea frameworkului aggregate si conceptului de pipeline pentru inlantuirea operatorilor fiind implementate 2 exemple: media achizitiilor in functie de reprezentanta si afisare fiecare angajat in parte de la fiecare reprezentanta.

e) Script folosirea Map-Reduce:

In ilustrarea functionalitatii de Map-Reduce a fost facut un exemplu in care se poate observa cum fiecare dintre etapele map si reduce se executa la rand: media achizitiilor in functie de reprezentanta, dezvoltat si in capitolul precedent cu frameworkul aggregate.