

IMPLEMENTARE BAZA DE DATE ÎN MONGODB

-SCHI, PATINAJ, CICLISM-

Şanta Valentina-Larisa

Proiectul își propune crearea unei baze de date pentru compania SC. BRAND SRL, o companie care are ca și domeniu de activitate închirierea sau vânzarea de articole sportive pentru activitățile de iarnă precum: patinaj, sky sau ciclism. Aflându-ne în plin sezon de activitate, numărul mare de cereri a dus la necesitatea unei aplicații care să faciliteze și să ușureze munca angajaților. Baza de date implementată reprezintă scheletul unei aplicații de tip web, desktop sau mobile care are ca și obiectiv principal schimbarea modului în care serviciul de închiriere era făcut până în acest moment. Pentru a evita situația în care clienții sunt nevoiți să stea la coadă, ei vor avea posibilitatea de a vedea în timp real produsele disponibile spre închiriere dar și ora la care alte produse dorite se vor elibera. De asemenea, tot prin intermediul viitoarei aplicații clienții vor putea să-și rezerve dacă doresc anumite produse destinate vânzării, pentru a putea fi ridicate ulterior de la sediul companiei, economisindu-se astfel timp prin faptul că fiecare persoană poate vedea în aplicație ce produse sunt disponibile și ce produse l-ar interesa.

Baza de date va fi proiectată în MongoDB. MongoDB este o bază de date NoSQL având ca și caracteristică faptul că stocarea datelor nu se va face folosind tabele ca și într-o bază de date relațională, ci vor fi salvate sub formă de colecții care conțin documente. Pentru a face o legătură cu bazele de date relaționale, colecțiile sunt analogia pentru tabele iar documentele echivalentul înregistrărilor. Modul în care vom crea baza de date este în felul următor: vom crea patru colecții în interiorul cărora vom avea mai multe documente. Prima colecție va fi reprezentată de clienți unde vom stoca cele mai importante informații despre aceștia cum ar fi: numele, prenumele, orașul din care aceștia provin precum și numărul de telefon al fiecăruia. Produsele vor avea și ele o colecție, în care vom avea numele acestora dar și cărui tip de serviciu îi aparțin, astfel unele produse vor fi destinate vânzărilor iar altele vor fi destinate închirierii. Vom mai avea o colecție pentru închirierile de produse și una pentru rezervarea produselor având o evidență clară a fiecărui serviciu pus la dispoziția clienților. O reprezentare a colecțiilor poate fi văzută în felul următor:

Clienți:

id
nume
prenume
oras
telefon

Inchirieri:

id
data
produs { id, ora_start
ora_finaal }
client.id

Produse:

id
nume
serviciu
preț

Rezervari:

id
data
produs.id
client.id

Primul pas va fi reprezentat de crearea bazei de date în care vom salva toate colecțiile enumerate anterior. Pentru a face acest lucru vom folosi comanda *use branddb*:

```
> use branddb
switched to db branddb
```

Odată creată baza de date vom putea trece la crearea colecțiilor. Pentru a face acest lucru ne folosim de comanda *db.createCollection*. Pentru a verifica dacă colecțiile noastre au fost create folosim comanda *show collections*:

```
> show collections
Clienti
Produse
Rezervari
închirieri
```

Popularea colecțiilor cu date o vom face folosindu-ne de comanda *db.numecolectie.insertMany()*. În MongoDB fiecare colecție are în mod obligatoriu un id, care va fi generat în mod automat. Un exemplu de inserare al datelor pentru colecția Produse:

```
db.Produse.insertMany([
{
  nume: "schiuri",
  serviciu: "închiriere",
  preț: "30 "
},
{
  nume: "casti patinaj",
  serviciu: "închiriere",
  preț: "25 "
},
{
  nume: "manusi",
  serviciu: "vanzare",
  preț: "60 "
},
{
  nume: "clapari",
  serviciu: "închiriere",
  preț: "45 "
},
{
  nume: "bicicleta down hill",
  serviciu: "vanzare",
  preț: "250"
```

```
},
{
  nume: "costum sky",
  serviciu: "vanzare",
  preț: "340"
},
{
  nume: "genunchiere",
  serviciu: "închiriere",
  preț: "15 "
},
{
  nume: "cotiere",
  serviciu: "închiriere",
  preț: "20 "
},
{
  nume: "patine",
  serviciu: "vanzare",
  preț: "450"
},
{
  nume: "placa",
  serviciu: "vanzare",
  preț: "600"
},
])
```

În mod similar vom proceda și cu celelalte colecții având însă date diferite. Pentru a afișa toate documentele inserate în colecții ne putem folosi de comanda *db.numecolectie.find().pretty()*:

```

> db.Inchirieri.find(<>).pretty(<>)
<
  "_id" : ObjectId<"5dfba8992df2ff645145336c">,
  "data" : "12.12.2019",
  "produs" : <
    "produs_id" : ObjectId<"5dfb97282df2ff6451453352">,
    "ora_start" : "12:00",
    "ora_final" : "13:00"
  >,
  "client_id" : ObjectId<"5dfb9e2a2df2ff645145335d">
<
  "_id" : ObjectId<"5dfba9252df2ff645145336d">,
  "data" : "02.12.2019",
  "produs" : <
    "produs_id" : ObjectId<"5dfb97282df2ff6451453353">,
    "ora_start" : "9:00",
    "ora_final" : "10:00"
  >,
  "client_id" : ObjectId<"5dfb9ea92df2ff645145335f">
<
  "_id" : ObjectId<"5dfba97b2df2ff645145336e">,
  "data" : "30.11.2019",
  "produs" : <
    "produs_id" : ObjectId<"5dfb97792df2ff6451453354">,
    "ora_start" : "14:00",
    "ora_final" : "15:00"
  >,
  "client_id" : ObjectId<"5dfb9f372df2ff6451453362">
<
  "_id" : ObjectId<"5dfba9e32df2ff645145336f">,
  "data" : "28.09.2019",
  "produs" : <
    "produs_id" : ObjectId<"5dfb97f52df2ff6451453358">,
    "ora_start" : "10:00",
    "ora_final" : "11:00"
  >,
  "client_id" : ObjectId<"5dfba0472df2ff6451453363">
<
  "_id" : ObjectId<"5dfbaa182df2ff6451453370">,
  "data" : "18.09.2019",
  "produs" : <
    "produs_id" : ObjectId<"5dfb97f52df2ff6451453359">,
    "ora_start" : "18:00",
    "ora_final" : "19:00"
  >,
  "client_id" : ObjectId<"5dfba0cd2df2ff6451453365">
<
  _

```

```

> db.Rezervari.find(<>).pretty(<>)
<
  "_id" : ObjectId<"5dfba52b2df2ff6451453367">,
  "data" : "13.10.2019",
  "produs_id" : ObjectId<"5dfb97792df2ff6451453354">,
  "client_id" : ObjectId<"5dfb9eda2df2ff6451453360">
<
  "_id" : ObjectId<"5dfba5dc2df2ff6451453368">,
  "data" : "03.02.2019",
  "produs_id" : ObjectId<"5dfb97b12df2ff6451453356">,
  "client_id" : ObjectId<"5dfb9e662df2ff645145335e">
<
  "_id" : ObjectId<"5dfba62a2df2ff6451453369">,
  "data" : "21.11.2019",
  "produs_id" : ObjectId<"5dfb97b12df2ff6451453357">,
  "client_id" : ObjectId<"5dfba07d2df2ff6451453364">
<
  "_id" : ObjectId<"5dfba6b52df2ff645145336a">,
  "data" : "07.09.2019",
  "produs_id" : ObjectId<"5dfb98272df2ff645145335a">,
  "client_id" : ObjectId<"5dfb9d912df2ff645145335c">
<
  "_id" : ObjectId<"5dfba7012df2ff645145336b">,
  "data" : "18.12.2019",
  "produs_id" : ObjectId<"5dfb98272df2ff645145335b">,
  "client_id" : ObjectId<"5dfb9f0f2df2ff6451453361">
<
  _

```

```

> db.Produse.find().pretty()
{
  "_id" : ObjectId("5dfb97282df2ff6451453352"),
  "nume" : "schiuri",
  "serviciu" : "inchiriere",
  "pret" : 30
}
{
  "_id" : ObjectId("5dfb97282df2ff6451453353"),
  "nume" : "casti patinaj",
  "serviciu" : "inchiriere",
  "pret" : 30
}
{
  "_id" : ObjectId("5dfb97792df2ff6451453354"),
  "nume" : "manusi",
  "serviciu" : "vanzare",
  "pret" : 60
}
{
  "_id" : ObjectId("5dfb97792df2ff6451453355"),
  "nume" : "clapari",
  "serviciu" : "inchiriere",
  "pret" : 45
}
{
  "_id" : ObjectId("5dfb97b12df2ff6451453356"),
  "nume" : "bicicleta down hill",
  "serviciu" : "vanzare",
  "pret" : 250
}
{
  "_id" : ObjectId("5dfb97b12df2ff6451453357"),
  "nume" : "costum sky",
  "serviciu" : "vanzare",
  "pret" : 340
}
{
  "_id" : ObjectId("5dfb97f52df2ff6451453358"),
  "nume" : "geunuchiere",
  "serviciu" : "inchiriere",
  "pret" : 15
}
{
  "_id" : ObjectId("5dfb97f52df2ff6451453359"),
  "nume" : "cotiere",
  "serviciu" : "inchiriere",
  "pret" : 20
}
{
  "_id" : ObjectId("5dfb98272df2ff645145335a"),
  "nume" : "patine",
  "serviciu" : "vanzare",
  "pret" : 450
}

```

```

> db.Clienti.find().pretty()
{
  "_id" : ObjectId("5dfb9d912df2ff645145335c"),
  "nume" : "Popa",
  "prenume" : "Anamaria",
  "oras" : "Baia Mare",
  "telefon" : "0732647185"
}
{
  "_id" : ObjectId("5dfb9e2a2df2ff645145335d"),
  "nume" : "Cosma",
  "prenume" : "Adina",
  "oras" : "Satu Mare",
  "telefon" : "0784747585"
}
{
  "_id" : ObjectId("5dfb9e662df2ff645145335e"),
  "nume" : "Giurgiu",
  "prenume" : "Ioan",
  "oras" : "Targu Lapus",
  "telefon" : "0784352671"
}
{
  "_id" : ObjectId("5dfb9ea92df2ff645145335f"),
  "nume" : "Man",
  "prenume" : "Andrei",
  "oras" : "Cluj-Napoca",
  "telefon" : "0745319504"
}
{
  "_id" : ObjectId("5dfb9eda2df2ff6451453360"),
  "nume" : "Mutiu",
  "prenume" : "Alexandra",
  "oras" : "Brasov",
  "telefon" : "0775319367"
}
{
  "_id" : ObjectId("5dfb9f0f2df2ff6451453361"),
  "nume" : "Salajan",
  "prenume" : "Mircea",
  "oras" : "Baia",
  "telefon" : "0784219364"
}
{
  "_id" : ObjectId("5dfb9f372df2ff6451453362"),
  "nume" : "Popa",
  "prenume" : "Cornel",
  "oras" : "Baia",
  "telefon" : "0784219364"
}
{
  "_id" : ObjectId("5dfba0472df2ff6451453363"),
  "nume" : "Santa",
  "prenume" : "Ioana",
  "oras" : "Gherla",
  "telefon" : "0784452906"
}

```

Pentru a efectua o ștergere și o actualizare vom adăuga un nou produs:

```
> db.Produse.insert({nume:"casca", serviciu: "vanzare", pret:"400 lei"})
WriteResult({ "nInserted" : 1 })
```

Pentru operația de actualizare dorim ca serviciul căruia îi este destinat produsul să devină închiriere, iar prețul să fie mai mare cu 50 lei. Realizarea acestei operații se va face folosind funcția *db.numecolectie.update()*. În colecția produse se va căuta produsul cu id-ul specificat de noi și îl va actualiza cu noile date.

```
> db.Produse.update({_id: ObjectId("5dff606c938f95eb1575eb76")},
... {serviciu:"inchiriere", pret:"450 lei", })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Cu ajutorul funcției *db.Produse.find().pretty()* putem verifica dacă produsul a fost actualizat.

```
{
  "_id" : ObjectId("5dff606c938f95eb1575eb76"),
  "serviciu" : "inchiriere",
  "pret" : "450 lei"
}
```

Pentru a efectua operația de ștergere ne ajutăm de funcția *db.numecolectie.remove()*.

```
> db.Produse.remove({_id: ObjectId("5dff606c938f95eb1575eb76")})
WriteResult({ "nRemoved" : 1 })
```

În urma executării acestei comenzi produsul va fi șters din colecția noastră.

Pentru o dezvoltare ulterioară a aplicației, în tabela Produse am putea avea și o scurtă descriere a produselor precum și numărul de bucăți disponibile, aceste informații suplimentare ajutând clienții în procesul decizional. Astfel, spre exemplificare, vom adăuga două produse cu o structură diferită.

```
> db.Produse.insert({nume:"bete sky", serviciu: "vanzare", pret:"250 lei", descriere:"Betele Atomic sunt confectionate din material dur, avand o greutate redusa. Acestea sunt echipate cu un sistem de siguranta care presupune eliberarea instantă a chingii in cazul unui impact. Gripul betelor este ergonomic si ofera spatiu pentru o prindere ferma.", numar_bucati:4})
WriteResult({ "nInserted" : 1 })
> db.Produse.insert({nume:"ochelari sky", serviciu: "inchiriere", pret:"50 lei", descriere:"Ochelarii TSAR sunt unicii ochelari de forma universala. Oferă o caracteristica anti-ceata ultra performanta. Lentilele lor au adaptare rapida de la intuneric la lumina la orice temperatura. Dispun de un scut protector impotriva zgarieturilor si ofera o imagine clara", numar_bucati:1})
WriteResult({ "nInserted" : 1 })
```

Indexarea în MongoDB este un proces care eficientizează execuția unei instrucțiuni, evitându-se astfel să se scaneze întregile documente din colecție. MongoDB crează un index în mod automat, *_id* care împiedică introducerea a două documente care să aibă același id. Vom crea un index simplu pentru colecția produse folosind funcția *db.numecolectie.createIndex*.

```
> db.Clients.createIndex ({nume : 1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Un index compus vom crea pentru colecția Produse:

```
> db.Produse.createIndex (<{nume : 1, pret:-1}>)
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Interogări:

- ✓ Să se găsească toți clienții al căror nume este sau conține numele Ioan pentru oferirea unui discount cu ocazia onomasticii:

```
> db.Cienti.find(<{prenume: {$regex: /^Ioan/}}>).pretty()
{
  "_id" : ObjectId("5dfb9e662df2ff645145335e"),
  "nume" : "Giurgiu",
  "prenume" : "Ioan",
  "oras" : "Targu Lapus",
  "telefon" : "0784352671"
}
{
  "_id" : ObjectId("5dfba0472df2ff6451453363"),
  "nume" : "Santa",
  "prenume" : "Ioana",
  "oras" : "Gherla",
  "telefon" : "0784452906"
}
```

- ✓ Să se găsească toate produsele destinate vânzării:

```
> db.Produse.find(<{serviciu: {$regex: /^vanzare/}}>).pretty()
{
  "_id" : ObjectId("5dfb97792df2ff6451453354"),
  "nume" : "manusi",
  "serviciu" : "vanzare",
  "pret" : 60
}
{
  "_id" : ObjectId("5dfb97b12df2ff6451453356"),
  "nume" : "bicicleta down hill",
  "serviciu" : "vanzare",
  "pret" : 250
}
{
  "_id" : ObjectId("5dfb97b12df2ff6451453357"),
  "nume" : "costum sky",
  "serviciu" : "vanzare",
  "pret" : 340
}
{
  "_id" : ObjectId("5dfb98272df2ff645145335a"),
  "nume" : "patine",
  "serviciu" : "vanzare",
  "pret" : 450
}
{
  "_id" : ObjectId("5dfb98272df2ff645145335b"),
  "nume" : "placa",
  "serviciu" : "vanzare",
  "pret" : 600
}
{
  "_id" : ObjectId("5dff69b7938f95eb1575eb77"),
  "nume" : "bete sky",
  "serviciu" : "vanzare",
  "pret" : 250,
  "descriere" : "Betele Atomic sunt confectionate din material dur, avand o greutate redusa. Acestea sunt echipate cu un sistem de siguranta care presupune eliberarea instantă a chingii in cazul unui impact. Gripul betelor este ergonomic si ofera spatiu pentru o prindere ferma.",
  "numar_bucati" : 4
}
```


- ✓ Să se găsească toate produsele al căror preț este de 60 lei:

```
> db.Produse.find(<<"pret": < $eq: 60>>>).pretty(<>
{
  "_id" : ObjectId("5dfb97792df2ff6451453354"),
  "nume" : "manusi",
  "serviciu" : "vanzare",
  "pret" : 60
}
```

- ✓ Să se afișeze rezervările făcute în data de 18.12.2019:

```
> db.Rezervari.find(<<data: <$regex: /^18.12.2019/>>>>).pretty(<>
{
  "_id" : ObjectId("5dfba7012df2ff645145336b"),
  "data" : "18.12.2019",
  "produs_id" : ObjectId("5dfb98272df2ff645145335b"),
  "client_id" : ObjectId("5dfb9f0f2df2ff6451453361")
}
```

- ✓ Să se găsească produsele cu pretul >200 lei:

```
> db.Produse.find(<<"pret": < $gt: 200>>>).pretty(<>
{
  "_id" : ObjectId("5dfb97b12df2ff6451453356"),
  "nume" : "bicicleta down hill",
  "serviciu" : "vanzare",
  "pret" : 250
}
{
  "_id" : ObjectId("5dfb97b12df2ff6451453357"),
  "nume" : "costum sky",
  "serviciu" : "vanzare",
  "pret" : 340
}
{
  "_id" : ObjectId("5dfb98272df2ff645145335a"),
  "nume" : "patine",
  "serviciu" : "vanzare",
  "pret" : 450
}
{
  "_id" : ObjectId("5dfb98272df2ff645145335b"),
  "nume" : "placa",
  "serviciu" : "vanzare",
  "pret" : 600
}
{
  "_id" : ObjectId("5dff69b7938f95eb1575eb77"),
  "nume" : "bete sky",
  "serviciu" : "vanzare",
  "pret" : 250,
  "descriere" : "Betele Atomic sunt confectionate din material dur, avand o greutate redusa. Acestea sunt echipate cu un sistem de siguranta care presupune eliberarea instanta a chingii in cazul unui impact. Gripul betelor este ergonomic si ofera spatiu pentru o prindere ferma.",
  "numar_bucati" : 4
}
```

- ✓ Să se afișeze informațiile despre clientul cu _id= ObjectId("5dfb9d912df2ff645145335c"):

```
> db.Cienti.find(<<_id:ObjectId("5dfb9d912df2ff645145335c")>>>).pretty(<>
{
  "_id" : ObjectId("5dfb9d912df2ff645145335c"),
  "nume" : "Popa",
  "prenume" : "Anamaria",
  "oras" : "Baia Mare",
  "telefon" : "0732647185"
}
```

- ✓ Să se găsească produsele al căror nume conține litera p și care au prețul mai mare de 200 lei:

```
> db.Produse.find(<<$and: [{pret: { $gt: 200}}, {nume:{$regex:/^p/}}]>>).pretty()
{
  "_id" : ObjectId("5dfb98272df2ff645145335a"),
  "nume" : "patine",
  "serviciu" : "vanzare",
  "pret" : 450
}
{
  "_id" : ObjectId("5dfb98272df2ff645145335b"),
  "nume" : "placa",
  "serviciu" : "vanzare",
  "pret" : 600
}
```

- ✓ Să se găsească închirierile făcute în data de 30.11.2019:

```
> db.Inchirieri.find(<<data:"30.11.2019">>).pretty()
{
  "_id" : ObjectId("5dfba97b2df2ff645145336e"),
  "data" : "30.11.2019",
  "produs" : {
    "produs_id" : ObjectId("5dfb97792df2ff6451453354"),
    "ora_start" : "14:00",
    "ora_final" : "15:00"
  },
  "client_id" : ObjectId("5dfb9f372df2ff6451453362")
}
```

- ✓ Să se afișeze toți clienții din Cluj-Napoca:

```
> db.Cienti.find(<<oras:"Cluj-Napoca">>).pretty()
{
  "_id" : ObjectId("5dfb9ea92df2ff645145335f"),
  "nume" : "Man",
  "prenume" : "Andrei",
  "oras" : "Cluj-Napoca",
  "telefon" : "0745319504"
}
```

- ✓ Să se identifice dacă clientul Brașovean Cristi a efectuat sau nu rezervări:

```
> db.Rezervari.find(<<client_id:ObjectId("5dfba07d2df2ff6451453364")>>).pretty()
{
  "_id" : ObjectId("5dfba62a2df2ff6451453369"),
  "data" : "21.11.2019",
  "produs_id" : ObjectId("5dfb97b12df2ff6451453357"),
  "client_id" : ObjectId("5dfba07d2df2ff6451453364")
}
```