

mongoDB

Banco de dados NoSQL - MongoDB
Manipulando Arrays

Prof. Gustavo Leitão

Busca em Array

Insira os seguinte dados

```
db.inventory.insertMany([
  { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
  { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
  { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
  { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
  { item: "postcard", qty: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }
]);
```

Busca documentos que possuem array exatamente como especificado (incluindo ordem)

```
db.inventory.find( { tags: [ "red", "blank" ] } )
```

Busca em Array

Busca documentos que contém ambos elementos (independente de ordem)

```
db.inventory.find({ tags: { $all: [ "red", "blank" ] } })
```

Busca por documentos que contém um elemento em específico

```
db.inventory.find({ tags: "red" })
```

Busca em Array

Busca todos documentos em que pelo menos um elemento do campo dim_cm é maior que 25

```
db.inventory.find( { dim_cm: { $gt: 25 } } )
```

Múltiplas condições: Busca documentos que possuem pelo menos um elemento no campo dim_cm maior que 15 e um elemento menor que 20.

```
db.inventory.find( { dim_cm: { $gt: 15, $lt: 20 } } )
```

Busca em Array

Múltiplas condições: Busca documentos que possui um elemento do array que satisfaça ambas as condições.

```
b.inventory.find( { dim_cm: { $elemMatch: { $gt: 22,  
$lt: 30 } } } )
```

Busca um elementos de uma dada posição do array e que satisfaça a condição.

```
db.inventory.find( { "dim_cm.1": { $gt: 25 } } )
```

Busca em Array

Busca por arrays que possuem determinado número de elementos:

```
db.inventory.find( { "tags": { $size: 3 } } )
```

Exercício

- Localize os restaurantes que possuem em suas coordenadas um elemento com valor maior que 3 e menor que 5.

Exercício

- Localize os restaurantes que possuem em suas coordenadas um elemento com valor maior que 3 e menor que 5.

```
db.restaurants
.find({ "address.coord": { $elemMatch: { $gt: 3,
$lt: 5 } } })
```


Busca em Array (Documentos internos)

Insira os seguinte dados

```
db.inventory.insertMany( [  
  { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },  
  { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },  
  { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 } ] },  
  { item: "planner", instock: [ { warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 } ] },  
  { item: "postcard", instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }  
]);
```

Busca documentos que possuem documento exatamente como especificado (incluindo ordem)

```
db.inventory.find( { "instock": { warehouse: "A", qty: 5 } } )
```

Busca em Array (Documentos internos)

Seleciona todos os documentos que possui pelo menos um elemento com quantidade menor ou igual a 20

```
db.inventory.find( { 'instock.qty': { $lte: 20 } } )
```

É possível buscar em um determinado elemento do array

```
db.inventory.find( { 'instock.0.qty': { $lte: 20 } } )
```

Busca em Array (Documentos internos)

Múltiplas condições: Busca documentos que no campo instock possui no elemento interno qty valor maior que 10 e menor igual a 20 (o valor de qty deve satisfazer ambas condições)

```
db.inventory.find( { "instock": { $elemMatch: { qty: { $gt: 10, $lte: 20 } } } } )
```

Já nesse caso, qualquer combinação de elemento satisfaz a condição. Ou seja, busca por documentos que algum elemento qty é maior que 10 e/ou menor que 20 (pode ser elementos diferentes do array)

```
db.inventory.find( { "instock.qty": { $gt: 10, $lte: 20 } } )
```

Exercício

- Localize os restaurantes que possuem pelo menos uma avaliação “B”.
- Localize os restaurantes que possuem exatamente 8 avaliações.
- Localize restaurantes que possuem avaliação A e score maior que 10 (na mesma avaliação)

Exercício

- Localize restaurantes que possuem avaliação A e score maior que 10 (na mesma avaliação)

```
db.restaurants
.find({ grades: { $elemMatch: { grade: "A", score:
{ $gte: 10 } } } })
```

Exercício

- Desafio: Localize os documentos que possuem 8 avaliações e todas com grade A!

Exercício

- Desafio:

```
db.restaurants
.find({
  $and: [
    {"grades": {
      $not: {$elemMatch: {"grade": {$nin: ['A']}} } } },
    {"grades": {"$size": 8}
  ]})
```

Removendo uma coleção

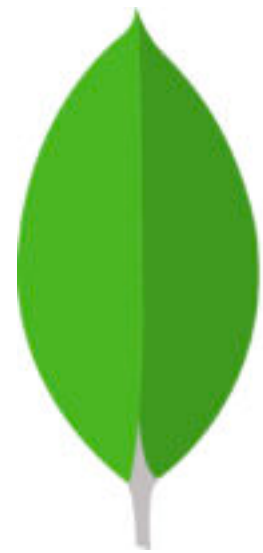
Drop a Collection

The remove all operation only removes the documents from the collection. The collection itself, as well as any indexes for the collection, remain. To remove all documents from a collection, it may be more efficient to drop the entire collection, including the indexes, and then recreate the collection and rebuild the indexes. Use the **drop()** method to drop a collection, including any indexes.

```
db.restaurants.drop()
```

Upon successful drop of the collection, the operation returns **true**.

```
true
```

mongoDB

Banco de dados NoSQL - MongoDB (Parte2)

Prof. Gustavo Leitão