

Bancos de Dados Não-Relacionais

CET 0620 - Módulo 03f

Professor Franklin Amorim

MongoDB

Arrays

Arrays

Find com Arrays

```
db.collection.find({<array>: { <operator1>: <value1>, ... } })
```

Arrays também podem ser usados em consultas, mas tem algumas particularidades

Arrays

```
db.inventory.drop()
```

```
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 ] }  
]);
```

Vamos criar alguns documentos com arrays para nos ajudar a ilustrar este tópico

Arrays

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

Com uma igualdade retornamos todos os documentos onde o array possui este elemento

Arrays

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

Para buscar um array inteiro, devemos listar todos os seus elementos na ordem correta.

Arrays

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

Se alterarmos a ordem dos elementos do array diferentes documentos são retornados

Arrays

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

Se quisermos retornar documentos com todos os elementos de um array ignorando a ordem precisamos usar o operador "\$all".

Arrays

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

Podemos usar outros operadores, como "\$gt" com arrays.
Documentos onde ao menos um elemento do array corresponda ao operador é retornado

Arrays

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

Podemos usar o operador posicional "\$" para retornar o primeiro elemento do Array que corresponda ao filtro.

Arrays

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

Podemos usar várias condições no filtro. Neste caso qualquer array que possuem uma combinação de elementos que corresponda às condições são retornados

Arrays

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

Use o operador `$elemMatch` para especificar vários critérios nos elementos de array, de modo que pelo menos um elemento do array satisfaça todos os critérios especificados.

Arrays

```
db.inventory.find({"dim_cm.0": 10})
```

Usando a notação de ponto, você pode especificar condições de consulta para um elemento em um determinado índice ou posição do array.

Arrays

Para a query anterior:

```
db.inventory.find({"dim_cm.0": 10})
```

Como projetar apenas o array "dim_cm" nos resultados?

Como projetar apenas o segundo elemento do array?

Mini Lab: Answer

Arrays

```
db.inventory.find({"dim_cm.0": 10}, {_id: 0, dim_cm: 1})
```

Podemos usar project para retornar apenas o array.

Arrays

```
db.inventory.find({"dim_cm.0": 10}, {"dim_cm.1": 1})
```

Mas não podemos usar dot notation para retornar um elemento do array

Mini Lab: Answer

Arrays

```
db.inventory.find({"dim_cm.0": 10}, {"dim_cm.1": 1})
```

Wrong

Mas não podemos usar dot notation para retornar um elemento do array

Mini Lab: Answer

Arrays

```
db.inventory.find({"dim_cm.0": 30}, {"dim_cm.1": 1})
```

E como estamos usando a dot notation para filtrar, não podemos usar o operador posicional "\$"

Mini Lab: Answer

Arrays

```
db.inventory.find({"dim_cm.0": 30}, {"dim_cm.1": 1})
```

Wrong

E como estamos usando a dot notation para filtrar, não podemos usar o operador posicional "\$"

Arrays

```
db.inventory.find({"dim_cm.0": 10}, {_id: 0, "Second": {$arrayElemAt: ["$dim_cm", 1]}})
```

Neste caso podemos usar "\$arrayElemAt" para criar um campo com a sintaxe { \$arrayElemAt: [<array>, <idx>] }

Arrays

```
db.inventory.find({"dim_cm.0": 10}, {"Second": {$slice: ["$dim_cm", 1, 1]}})
```

Outra opção é usar "\$slice", que pode ser usado para retornar um subconjunto do array usando a sintaxe: { \$slice: [<array>, <position>, <n>] }

Arrays

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

Com "\$size" podemos encontrar documentos com um tamanho específico de arrays



Arrays

Como encontrar documentos com atributo “tags” de tamanho maior que 2?

Mini Lab: Answer

Arrays

```
db.inventory.find( { tags: { $gt: { $size: 2 } } } )
```

Infelizmente \$size não pode ser usado aqui e este comando não retorna nenhum documento

Mini Lab: Answer

Arrays

```
db.inventory.find( { tags: { $gt: { $size: 2 } } } )
```

Wrong

Infelizmente \$size não pode ser usado aqui e este comando não retorna nenhum documento

Arrays

```
db.inventory.find( { "tags.2" : { $exists : true } } )
```

A alternativa é user dot notation.

Neste caso, se o array possui um elemento na posição 2 isso significa que ele tem ao menos 3 elementos

Arrays de Documentos

Arrays de Documentos

```
db.inventory.drop()
```

```
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 } ] }  
]);
```

Vamos criar alguns documentos com
array de documentos para ilustrar este tópico

Arrays de Documentos

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

Podemos buscar documentos com igualdade passado o subdocumento completo

Arrays de Documentos

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

Mas só vamos retornar o documento
se não passarmos o subdocumento completo, na order certa.

Arrays de Documentos

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

Podemos usar operadores para retornar todos os documentos onde pelo menos um subdocumento corresponde ao filtro

Arrays de Documentos

Como encontrar documentos onde **todos** os subdocumentos possuem "instock.qty" menor que 20?

Arrays de Documentos

```
db.inventory.find( { "instock.qty": { $not: { $gt: 20 } } } )
```

Podemos utilizar o operador “\$not” para inverter a questão e retornar todos os documentos que não possuem nenhum subdocumento com “instock.qty” maior que 20

Arrays de Documentos

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

Podemos usar dot notation para especificar um subdocumento no array.

Neste exemplo retornamos documentos onde o primeiro subdocumento do array "instock" possui "qty" igual ou menor que 20

Arrays de Documentos

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

Podemos usar o operador `$elemMatch` para especificar vários critérios, de modo que pelo menos um subdocumento satisfaça todos os critérios especificados.

Arrays de Documentos

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

Se não usarmos o operador "\$elemMatch", a consulta selecionará os documentos onde o array contém qualquer combinação de elementos que satisfaça as condições.

Update Arrays

Update Arrays

```
db.students.drop();
```

```
db.students.insertMany( [  
  { "_id" : 1, "grades" : [ 85, 80, 80 ] },  
  { "_id" : 2, "grades" : [ 88, 90, 92 ] },  
  { "_id" : 3, "grades" : [ 85, 100, 90 ] }  
] );
```

Vamos criar alguns documentos com arrays para nos ajudar a ilustrar este tópico

Update Arrays

```
db.students.updateMany(  
  { _id: 3 },  
  { $push: { grades : 200 } }  
);  
  
db.students.find();
```

Use o operador "\$push" para adicionar um item ao array.

Update Arrays

```
db.students.updateMany( {}, { $pop: { grades : -1 } } )
```

O operador \$pop remove o primeiro ou o último elemento de um array.

Passe "-1" para remover o primeiro elemento e "1" para remover o último elemento.

Update Arrays

```
db.students.updateMany(  
  { _id: 1, grades: 80 },  
  { $set: { "grades.$" : 82 } }  
)
```

Para atualizar o primeiro item do array que satisfaça a consulta, usamos o operador "\$".

Neste caso, o documento correspondente possui dois itens de array com valor 80, mas apenas um é atualizado para 82.

Update Arrays

```
db.students.updateMany(  
  {},  
  { $inc: { "grades.$[]": 10 } }  
)
```

Para atualizar todos os itens do array do documento que satisfazem o filtro usamos o operador "\$[]"

Update Arrays

```
db.students.updateMany(  
  { },  
  { $set: { "grades.$[element]" : 100 } },  
  { arrayFilters: [ { "element": { $gte: 100 } } ] }  
)
```

Para atualizar apenas alguns elementos do array podemos usar o "\$[<identifier>]" com "arrayFilters".

Neste caso, apenas os itens do array maiores que 100 serão definidos como 100.