



Semantix

MongoDB - Básico

Aula 3

Quem sou eu?

Eu sou Rodrigo Augusto Rebouças.

Engenheiro de dados da Semantix
Instrutor do Semantix Academy

Você pode me encontrar em:
rodrigo.augusto@semantix.com.br





Atualizar Documentos

Atualizar Documentos

- Principais métodos
 - `db.<nomeCollection>.updateOne(<filtro>, <atualização>)`
 - `db.<nomeCollection>.updateMany(<filtro>, <atualização>)`
- Filtro é o Critério de Consulta
 - Operadores aplicados nos atributos
 - `{ <atributo> : { <operador> : <valor> } }`
- Atualização
 - Operadores de atualização aplicados nos atributos
 - <https://docs.mongodb.com/manual/reference/operator/update/>
 - `{<operadorAtualização>: { <atributo>: <valor>, <atributo>: <valor>, ...}}`

Atualizar 1 Documento

○ Operador set

- Atualizar o atributo
- Caso o atributo não exista, é criado
- Ex

```
db.cliente.updateOne(  
  {_id: 1},  
  {$set: {idade: 25, estado: "SP"}}  
)
```

```
> db.cliente.updateOne({_id: 1}, {$set: {idade: 25}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

○ Operador unset

- Remover o atributo
- O valor do atributo deve ser ""
- Ex

```
db.cliente.updateOne(  
  {_id: 1},  
  {$unset: {idade: ""}}  
)
```

Atualizar N Documentos

○ Operador set

- Atualizar o atributo
- Caso o atributo não exista, é criado
- Ex

```
db.cliente. updateMany(  
    {idade: {$gt: 27}},  
    {$set: {seguro_carro: "baixo"}}  
)
```

Atualizar nome do Atributo

○ Operador rename

- Atualizar os nomes dos atributos
- {\$rename: { <nomeAtributo>: <novoNome>, <nomeAtributo2>: <novoNome2>, ... } }
- Ex

```
db.cliente.updateMany(  
    {},  
    {$rename: {"nome": "nome_completo"}}  
)
```

Documentos com data

○ Tipos de Data e tempo

- Date() – Retorna a data atual como string
- new Date() - Retorna a data atual como objeto de data UTC (ISODate)
- new Date("<YYYY-mm-ddTHH:MM:ssZ>")
- New Timestamp() – Retorna o timestamp atual

○ Ex.

```
db.test.insertOne( {  
  ts: new Timestamp(),  
  date: new Date(),  
  data_string: Date(),  
  config_date: new Date("2020-08")  
});
```

```
> db.test.find().pretty()  
{  
  "_id" : ObjectId("5f33f723098fa6f852019977"),  
  "ts" : Timestamp(1597241123, 1),  
  "date" : ISODate("2020-08-12T14:05:23.953Z"),  
  "data_string" : "Wed Aug 12 2020 14:05:23 GMT+0000 (UTC)",  
  "config_date" : ISODate("2020-08-01T00:00:00Z")  
}
```


Atualizar data de Documentos

- Operador currentDate

- Setar um valor de data atual
- { \$currentDate: { <atributo>: \$type: <tipo> } }

- Tipo

- timestamp
 - date

- Ex

```
db.cliente.updateMany(  
  {idade: {$gt: 27}},  
  {  
    $set: {seguro_carro: "baixo"},  
    $currentDate: { atualizado: { $type: "timestamp" } }  
  }  
)
```

Atualizar Array do Documento

○ Operador set

- Atualizar um elemento do Array
- O elemento que deseja atualizar deve estar no Filtro
- Operador \$ - atualiza o primeiro elemento da pesquisa em relação ao filtro
- Ex

```
db.cliente.updateOne(  
    {_id: 2, "conhecimento":"Mongo"},  
    {$set: {"conhecimento.$": "MongoDB"}}  
)
```

```
> db.cliente.find({_id:2})  
{ "_id" : 2, "nome" : "João", "conhecimento" : [ "Hadoop", "Mongo" ] }  
> db.cliente.updateOne({_id: 2, "conhecimento":"Mongo"},{$set: {"conhecimento.$":"MongoDB"}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }  
> db.cliente.find({_id:2})  
{ "_id" : 2, "nome" : "João", "conhecimento" : [ "Hadoop", "MongoDB" ] }
```

Atualizar Array do Documento

○ Operador pull e push

- Adicionar elemento no Array - push
- Remover elemento no Array - pull
- Ex

```
db.cliente.updateOne({_id: 2},{ $push: {conhecimento:"Redis"}})
```

```
db.cliente.updateOne({_id: 2},{ $pull: {conhecimento:"Redis"}})
```

```
> db.cliente.find({_id:2})
{ "_id" : 2, "nome" : "João", "conhecimento" : [ "Hadoop", "Mongo" ] }
> db.cliente.updateOne({_id: 2},{ $push: {conhecimento:"Redis"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.cliente.find({_id:2})
{ "_id" : 2, "nome" : "João", "conhecimento" : [ "Hadoop", "Mongo", "Redis" ] }
> db.cliente.updateOne({_id: 2},{ $pull: {conhecimento:"Redis"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.cliente.find({_id:2})
{ "_id" : 2, "nome" : "João", "conhecimento" : [ "Hadoop", "Mongo" ] }
```

Exercício – Atualização de Documentos

1. Mostrar todos os documentos da collection produto do banco de dados seu nome
2. Atualizar o valor do campo nome para “cpu i7” do id 1
3. Atualizar o atributo quantidade para o tipo inteiro do id: 1
4. Atualizar o valor do campo qtd para 30 de todos os documentos, com o campo qtd \geq 30
5. Atualizar o nome do campo “descricao.so” para “descricao.sistema” de todos os documentos
6. Atualizar o valor do campo descricao.conexao para “USB 2.0” de todos os documentos, com o campo descricao.conexão = “USB”
7. Atualizar o valor do campo descricao.sistema para “USB 3.0” de todos os documentos, com o campo descricao.sistema = “USB 2.0” e adicionar o campo “data_modificacao”, com a data da atualização dos documentos
8. Atualizar um dos elementos do array descricao.so de “Windows” para “Windows 10” do id 3
9. Acrescentar o valor “Linux” no array descricao.so do id 4
10. Remover o valor “Mac” no array descricao.sistema do id 3 e adicionar o campo “ts_modificacao”, com o timestamp da atualização dos documentos



Deletar Documentos



Deletar Documentos

- Principais métodos
 - `db.<nomeCollection>.deleteOne(<filtro>)`
 - `db.<nomeCollection>.deleteMany(<filtro>)`
- Filtro é o Critério de Consulta
 - Operadores aplicados nos atributos
 - `{ <atributo> : { <operador> : <valor> }`
- Deletar todos os documentos
 - `db.cliente.deleteMany({})`

Deletar Documentos

○ Exemplos

- Deletar N documentos

- `db.cliente.deleteMany({status: "Inativo"})`

- Deletar um documento

- Se a busca encontrar mais de um documento, irá deletar apenas o primeiro

- `db.cliente.deleteOne({_id: 2})`

```
> db.cliente.deleteOne({_id: ObjectId("5f322e121da23003d673c7da")})  
{ "acknowledged" : true, "deletedCount" : 1 }
```

- Deletar collection

- `db.cliente.drop()`

Exercício – CRUD de Documentos

1. Criar a collection teste no banco de dados seu nome
2. Inserir o seguinte documento:
 - Campo: usuario, valor: Semantix
 - Campo: data_acesso, valor: data atual no formato ISODate
3. Buscar todos os documentos do ano ≥ 2020
4. Atualizar o campo “data_acesso” para timestamp com o valor da atualização do usuario Semantix
5. Buscar todos os documentos
6. Deletar o documento criado pelo id
7. Deletar a collection



Semantix

Obrigado!

Alguma pergunta?



Você pode me encontrar em:
rodrigo.augusto@semantix.com.br

GET SMARTER