



Semantix

MongoDB - Básico

Aula 2

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





Comandos Básicos

Banco de dados, Collections e Documentos

Criação Banco de dados

- Criar banco de dados

- Sintaxe:

use <nomeBD>

- Visualizar o banco de dados conectado

- Comando:

db

```
> use test
switched to db test
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> db
test
```

- Para visualizar o banco de dados é necessário incluir um documento

Criação Collection

○ Criar Collection

- Sintaxe:

`db.createCollection('<nomeCollection>')`

```
> use test
switched to db test
> db.createCollection('users')
{ "ok" : 1 }
```

```
> db.createCollection('users')
{
  "ok" : 0,
  "errmsg" : "Collection already exists. NS: test.users",
  "code" : 48,
  "codeName" : "NamespaceExists"
}
```

○ Visualizar Collections

- Comando:

`show collections`

```
> show collections
users
> show dbs
admin    0.000GB
config  0.000GB
local   0.000GB
test     0.000GB
```

Exclusão Collection e Banco de dados

- Excluir Collection

- Sintaxe:

db.<nomeCollection>.drop()

- Excluir Banco de dados

- Comando:

db.dropDatabase()

```
> use test
switched to db test
> db.users.drop()
true
> db.dropDatabase()
{ "dropped" : "test", "ok" : 1 }
```

Renomear Collection

- Renomear Collection

- Sintaxe:

db. <nomeCollection>.renameCollection('<nomeNovaCollection>')

```
> show collections
users
> db.users.renameCollection('cliente')
{ "ok" : 1 }
> show collections
cliente
```

Criação Documentos

- Adicionar documentos em uma collection

- Inserir um documento

- Sintaxe:

- ```
db.<nomeCollection>.insertOne({<documento>})
```

- Inserir N documentos (Array)

- Sintaxe:

- ```
db. <nomeCollection>.insertMany(  
  [{<documento1>}, {<documento2>},  
  {<documentoN>}])
```

- Exemplo de um documento na collection cliente

```
db.cliente.insertOne(  
  
  {  
  
    nome : "Lucas",  
    idade : 20,  
    conhecimento : "Windows, Linux, Hadoop"  
  
  }  
  
)
```

```
> db.cliente.insertOne({nome: "Lucas", idade: 20, conhecimento: "Windows, Linux, Hadoop"})  
{  
  "acknowledged" : true,  
  "insertedId" : ObjectId("5f322bae1da23003d673c7d6")  
}
```


Criação Documentos

- Exemplo de N documentos na collection cliente

```
db.cliente.insertMany(  
  [  
    {  
      nome : "Ana",  
      idade : 25,  
      conhecimento : "Windows, Linux, NoSQL"  
    },  
    {  
      nome : "João"  
    }  
  ]  
)
```

Visualizar Documentos

- Listar documentos da collection

- Sintaxe:

`db.<nomeCollection>.find().pretty()`

```
> db.cliente.find().pretty()
{
  "_id" : ObjectId("5f322c7f1da23003d673c7d8"),
  "nome" : "Lucas",
  "idade" : 20,
  "conhecimento" : "Windows, Linux, Hadoop"
}
{
  "_id" : ObjectId("5f322e121da23003d673c7d9"),
  "nome" : "Ana",
  "idade" : 25,
  "conhecimento" : "Windows, Linux, NoSQL"
}
{ "_id" : ObjectId("5f322e121da23003d673c7da"), "nome" : "João" }
```

Data Type

- ObjectId
- String
- Número
 - Long
 - Int
 - Decimal
 - Double
- Boolean
- Array
- Date
- Null
- Regex
- Timestamp

<https://docs.mongodb.com/manual/reference/bson-types/>

Exemplo Documento com Array

```
db.cliente.insertOne(  
  {  
    nome : "Lucas",  
    idade : 20,  
    conhecimento : ["Windows", "Linux", "Hadoop"]  
  }  
)
```

Exemplo Documento Alinhado

```
db.cliente.insertOne(  
  {  
    nome : "Lucas",  
    idade : 20,  
    conhecimento : ["Windows", "Linux", "Hadoop"],  
    endereco: {  
      rua: "Eusébio Matoso",  
      numero: 1375  
      bairro: "Pinheiros",  
      cidade: "São Paulo"  
    }  
  }  
)
```

Exercício – Comandos Básicos

1. Criar o banco de dados com seu nome
2. Listar os banco de dados
3. Criar a collection produto no bd com seu nome
4. Listar os banco de dados
5. Listar as collections
6. Inserir os seguintes documentos na collection produto:
 - `_id: 1, "nome": "cpu i5", "qtd": "15"`
 - `_id: 2, nome: "memória ram", qtd: 10, descricao: {armazenamento: "8GB", tipo:"DDR4"}`
 - `_id: 3, nome: "mouse", qtd: 50, descricao: {conexao: "USB", so: ["Windows", "Mac", "Linux"]}`
 - `_id: 4, nome: "hd externo", "qtd": 20, descricao: {conexao: "USB", armazenamento: "500GB", so: ["Windows 10", "Windows 8", "Windows 7"]}`
7. Mostrar todos os documentos

Consultas Básicas em Documentos

Consulta Documentos

- Forma básica de consultar documentos
 - Sintaxe:
 - `db.<nomeCollection>.find({<critérioConsulta>},{<projeção>})`
 - Critério de Consulta
 - Operadores aplicados nos atributos
 - `{ <atributo> : { <operador> : <valor> }`
 - Projeção
 - Atributos que são visualizados pela consulta
 - `{ <atributo> : 0 }`
 - Ocultar o atributo
 - `{ <atributo> : 1 }`
 - Mostrar o atributo

Consulta Documentos

- Critérios de consulta

- Todos os documentos

- Sintaxe: `db.<nomeCollection>.find()`
- Exemplo: `db.cliente.find({ })`

- Por um atributo

- Sintaxe: `db.<nomeCollection>.find({ <atributo> : <valor> })`
- Exemplo:

`db.cliente.find({ nome: "Jõao" })`

`db.cliente.find({ "endereco.cidade": "São Paulo" })`

- Por um atributo com operador

- Sintaxe: `db.<nomeCollection>.find({ <atributo> : { <operador> : <valor> } })`
- Exemplo: `db.cliente.find({ idade: { $lt : 18 } })`

Operadores Pesquisa

Operador	Descrição	Sintaxe
\$eq	Igual a	{ <field> : { \$eq: 10 } } ou { <field> : 10 }
\$ne	Diferente de	{ <field> : { \$ne: 10 } }
\$gt	Maior que	{ <field> : { \$gt: 10 } }
\$gte	Maior ou igual	{ <field> : { \$gte: 10 } }
\$lt	Menor que	{ <field> : { \$lt: 10 } }
\$lte	Menor ou igual	{ <field> : { \$lte: 10 } }
\$in	Na lista	{ <field> : { \$in: [10,20] } }
\$nin	Fora da lista	{ <field> : { \$nin: [10,20] } }
\$not	Não	{ <field> : { \$not: { \$eq: 10 } } } ou { <field> : { \$ne: 10 } }

Consulta Documentos

○ Projeção na consulta

- Exemplo: Consultar o nome Lucas e não apresentar o atributo conhecimento

```
> db.cliente.find( { nome: "Lucas" }, {conhecimento: 0} )  
{ "_id" : ObjectId("5f322c7f1da23003d673c7d8"), "nome" : "Lucas", "idade" : 20 }
```

- Exemplo: Consultar o nome Lucas e apresentar apenas o atributo conhecimento

```
> db.cliente.find( { nome: "Lucas" }, {conhecimento: 1} )  
{ "_id" : ObjectId("5f322c7f1da23003d673c7d8"), "conhecimento" : "Windows, Linux, Hadoop" }
```

- Exemplo: Consultar o nome Lucas e não apresentar o atributo _id e conhecimento

```
> db.cliente.find( { nome: "Lucas" }, { _id: 0, conhecimento: 0 } )  
{ "nome" : "Lucas", "idade" : 20 }
```

Exercício – Consulta básica em documentos

1. Mostrar todos os documentos da collection produto
2. Pesquisar na collection produto, os documentos com os seguintes atributos:
 - a) Nome = mouse
 - b) Quantidade = 20 e apresentar apenas o campo nome
 - c) Quantidade \leq 20 e apresentar apenas os campos nome e qtd
 - d) Quantidade entre 10 e 20
 - e) Conexão = USB e não apresentar o campo _id e qtd
 - f) SO que contenha “Windows” ou “Windows 10”



Outras Opções com Consultas

Operador And e Or

- Pesquisar múltiplos critérios

- And – Default

- `db.cliente.find({ "endereco.cidade": "São Paulo", idade: {$gte: 18} })`

- Or

- `db.cliente.find({ $or: [`
`{"endereco.cidade": "São Paulo"},`
`{idade: {$gte: 18}}] })`

- And e Or

- `db.cliente.find({`
`conhecimento: "Windows",`
`{ $or: [{"endereco.cidade": "São Paulo"}, {idade: {$gte: 18}}]`
`})`

Ordenando Pesquisas

- Função sort do método find
 - Sintaxe:
 - `db.<nomeCollection>.find({...}).sort({<atributo>: <valor>})`
 - Atributo
 - `.sort({<atributo>: 1})`
 - Ordenação ascendente
 - `.sort({<atributo>: -1})`
 - Ordenação descendente
 - `.sort({<atributo1>: 1, <atributo2>: 1})`
 - Ordenação por N atributos
 - Exemplo
 - `db.cliente.find({}).sort({"endereco.cidade": 1, nome: 1})`

Limitar Pesquisas

- Função limit do método find
 - Sintaxe:
 - `db.<nomeCollection>.find({...}).limit(<valor>)`
 - Exemplo
 - Mostrar os 5 primeiros documentos
 - `db.cliente.find({}).limit(5)`

Buscar 1 documento

○ FindOne

- Mais performático que o Find
- Garantir na saída sempre 1 documento
- Sintaxe:
 - `db.<nomeCollection>.findOne({...})`
- Exemplo
 - `db.cliente.findOne({nome: "lucas"},{_idade: 0})`
 - `db.cliente.find({nome: "lucas"},{_idade: 0}).limit(1)`

Exercício – Outras Opções com Consultas

1. Mostrar todos os documentos da collection produto
2. Realizar as seguintes pesquisas na collection produto:
 - a) Mostrar os documentos ordenados pelo nome em ordem alfabética
 - b) Mostrar os 3 primeiros documentos ordenados por nome e quantidade
 - c) Mostrar apenas 1 documento que tenha o atributo Conexão = USB
 - d) Mostrar os documentos de tenham o atributo conexão = USB e quantidade menor que 25
 - e) Mostrar os documentos de tenham o atributo conexão = USB ou quantidade menor que 25
 - f) Mostrar apenas os id dos documentos de tenham o atributo conexão = USB ou quantidade menor que 25



Semantix

Obrigado!

Alguma pergunta?



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

GET SMARTER