INSTITUTO FEDERAL
Rondônia

Banco de Dados II

O que é um banco de dados de documentos?

Um banco de dados de documentos (também conhecido como banco de dados orientado a documentos ou armazenamento de documentos) é um banco de dados que armazena informações em documentos.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  }
  "hobbies": ["surfing", "coding"]
}
```

Em vez de armazenar dados em linhas e colunas fixas, os bancos de dados de documentos usam documentos flexíveis.

O que são documentos?

Os documentos armazenam dados em pares campo-valor. Os valores podem ser de vários tipos e estruturas, incluindo strings, números, datas, matrizes ou objetos. Os documentos podem ser armazenados em formatos como JSON, BSON e XML.

BSON Types

Type	Number	Alias	Notes
Double	1	"double"	
String	2	"string"	
Object	3	"object"	
Array	4	"array"	
Binary data	5	"binData"	
Undefined	6	"undefined"	Deprecated.
ObjectId	7	"objectId"	
Boolean	8	"bool"	
Date	9	"date"	
Null	10	"null"	
Regular Expression	11	"regex"	
DBPointer	12	"dbPointer"	Deprecated.
JavaScript	13	"javascript"	
Symbol	14	"symbol"	Deprecated.
JavaScript code with scope	15	"javascriptWithScope"	Deprecated in MongoDB 4.4.
32-bit integer	16	"int"	
Timestamp	17	"timestamp"	
64-bit integer	18	"long"	
Decimal128	19	"decimal"	
Min key	-1	"minKey"	
Max key	127	"maxKey"	

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

BSON (Binary JSON)

BSON é uma representação binária do JSON, utilizada pelo MongoDB para armazenar seus documentos. É utilizado devido a:

- Rápida escaneabilidade (fast scanability), ou seja, torna possível varrer um documento procurando um valor rapidamente.
- Novos tipos de dados (Date, ObjectId, Binary Data)

Todo documento tem um campo chamado _id obrigatoriamente.

Por padrão, é um campo do tipo ObjectId ,mas pode ser definido como qualquer outro tipo de dado.

https://bsonspec.org/

Características do MongoDB

O MongoDB foi projetado para ser escalável.

Por isso, algumas funcionalidades NÃO foram incorporadas ao seu sistema, como:

- Junções (JOINs) entre coleções
- Transações ACID [Elmasri and Navathe 2014]

Características do MongoDB

E como viver em um mundo sem junções e transações?

- Aninhando documentos (embedded documents)
- Criando links artificiais (textual, não é uma chave estrangeira)
- Lembrando que as operações de escritas são atômicas no nível de um único documento, mesmo quando uma operação modificar vários documentos.

Problemas?

O que pode acontecer com os nossos dados?

- A consistência pode ser eventual
- Pode haver duplicação e inconsistência entre campos usados como links de documentos

Em compensação, ganhamos desempenho e escalabilidade.

A aplicação torna-se responsável pela integridade dos dados duplicados

MongoDB x SQL

MongoDB	SQL
Documentos	Linhas
Campos/chaves/atributos	Colunas
Coleções	Tabelas

Operações básicas

CRUD	SQL	MongoDB
Create	Insert	Insert
Read	Select	Find
Update	Update	Update
Delete	Delete	Remove

Não existe uma linguagem separada para descrever as operações de CRUD no MongoDB.
As operaçõs existem como métodos/funções dentro da API.

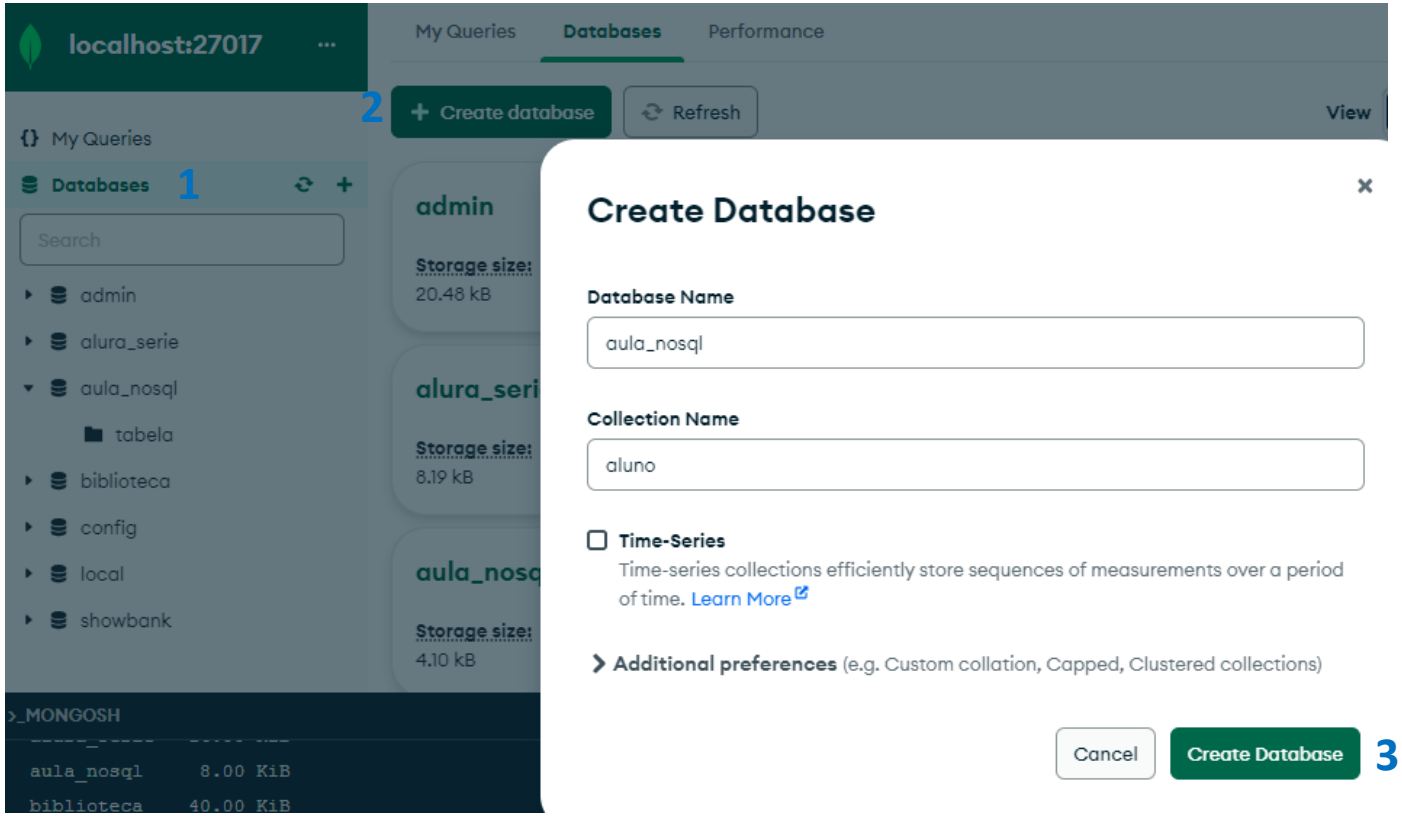
https://www.mongodb.com/docs/manual/crud/

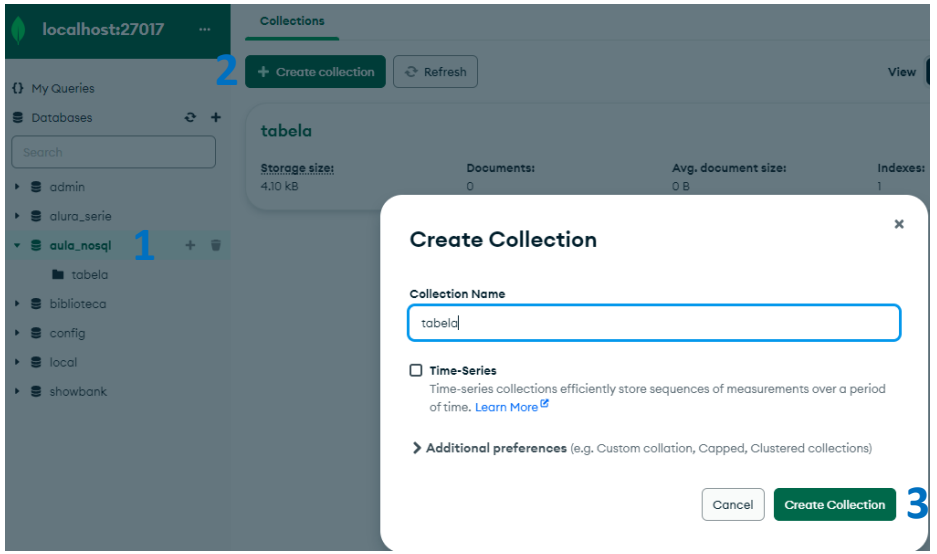
Criando uma base de dados no Mongodb Compass

use <“basededados”> // no ambiente de shell

Case-sensitive

db.createCollection(“tabela”) //Para a base existir é necessário uma collection





Para saber mais: restrições de criação - Coleção

Assim como para criar um banco de dados, também existem restrições para se criar uma coleção aqui no MongoDB, que são:

- Os nomes das coleções devem começar com um sublinhado ou um caractere de letra.
- Não podem:
  - ✓Conter o $.
  - ✓Ser uma string vazia (por exemplo "",).
  - ✓Conter o caractere nulo.
  - ✓Começar com o system.prefixo. (Reservado para uso interno).

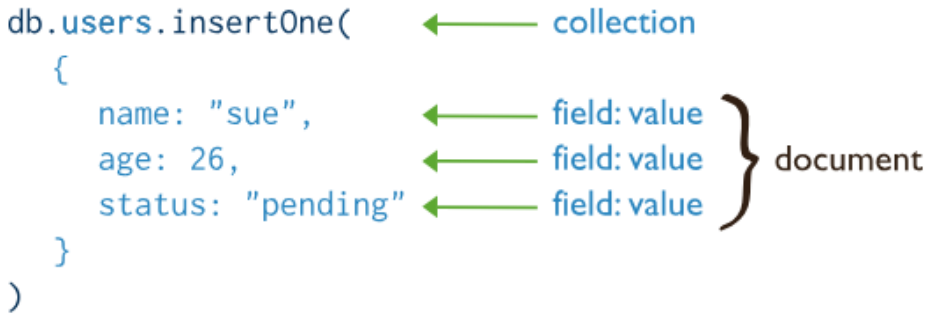
Excluindo uma base de dados no Mongodb Compass

- db.collection.drop()
- db.dropDatabase()

```
>_MONGOSH

>
> db.tabela.drop()
< true
> db.dropDatabase()
< { ok: 1, dropped: 'aula_nosql' }
aula_nosql>
```

Inserir documentos



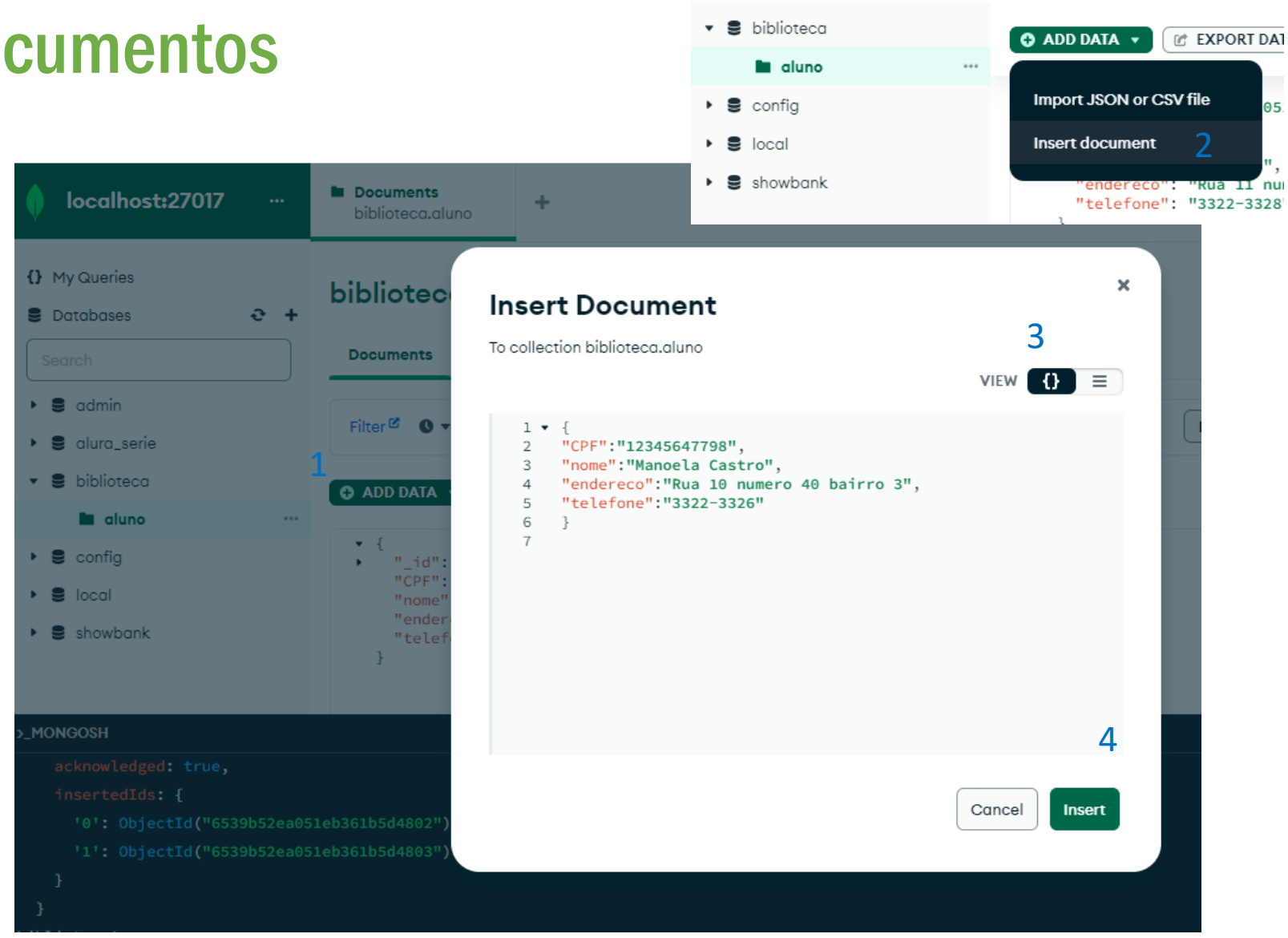
db.collection.insertOne()

```
db.aluno.insertOne({
"CPF":"12345647798",
"nome":"Manoela Castro",
"endereco":"Rua 10 numero 40 bairro 3",
"telefone":"3322-3326"
});
```

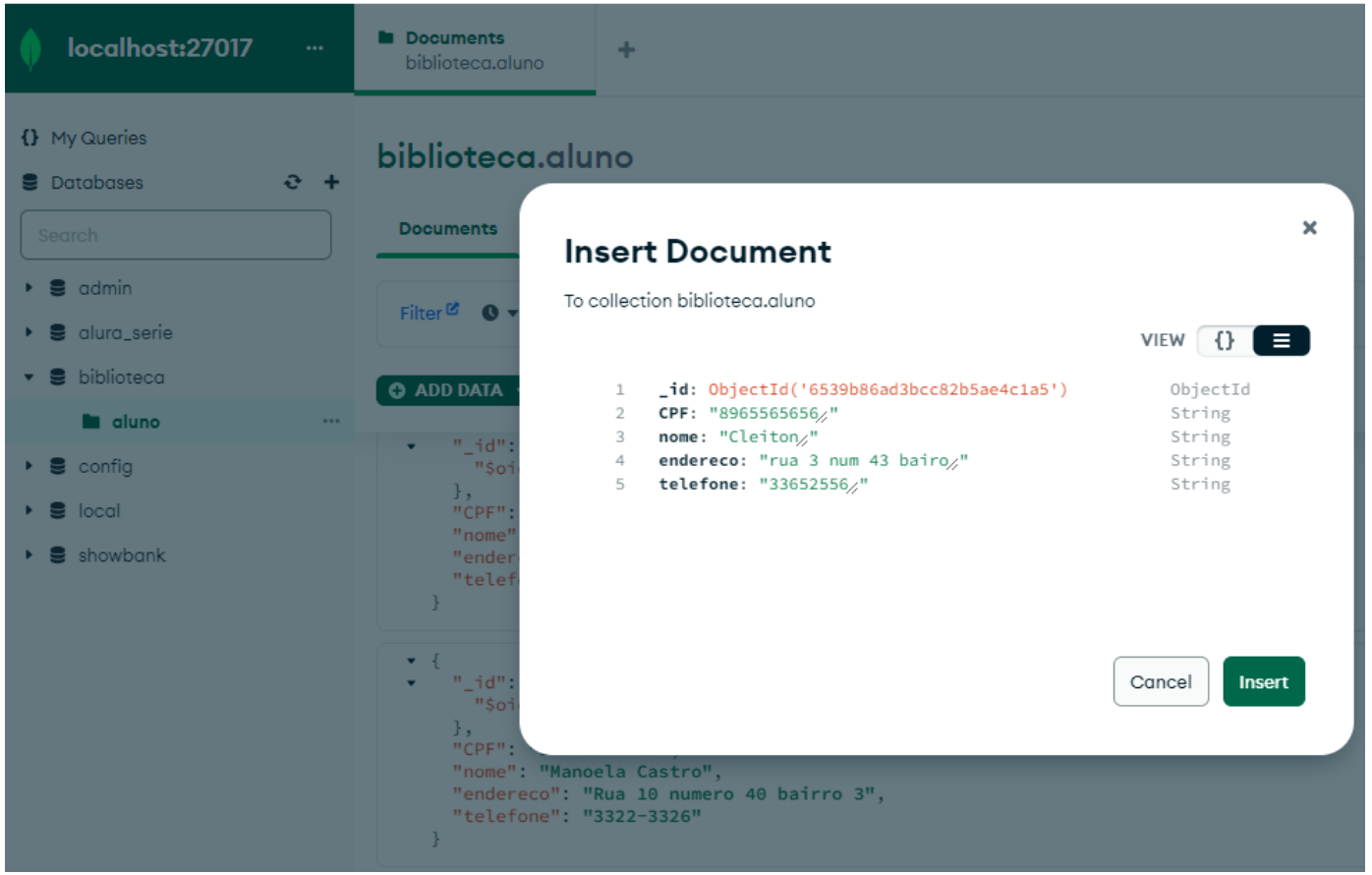
db.collection.insertMany()

```
db.aluno.insertMany([{
"CPF":"12345647798",
"nome":"Carlos Pereira",
"endereco":"Rua 17 numero 42 bairro 3",
"telefone":"3322-3328"
},{
"CPF":"5656565655",
"nome":"Maria Ribas",
"endereco":"Rua 11 numero 41 bairro 5",
"telefone":"3322-3328"
}])
```

Inserir documentos



Inserir documentos



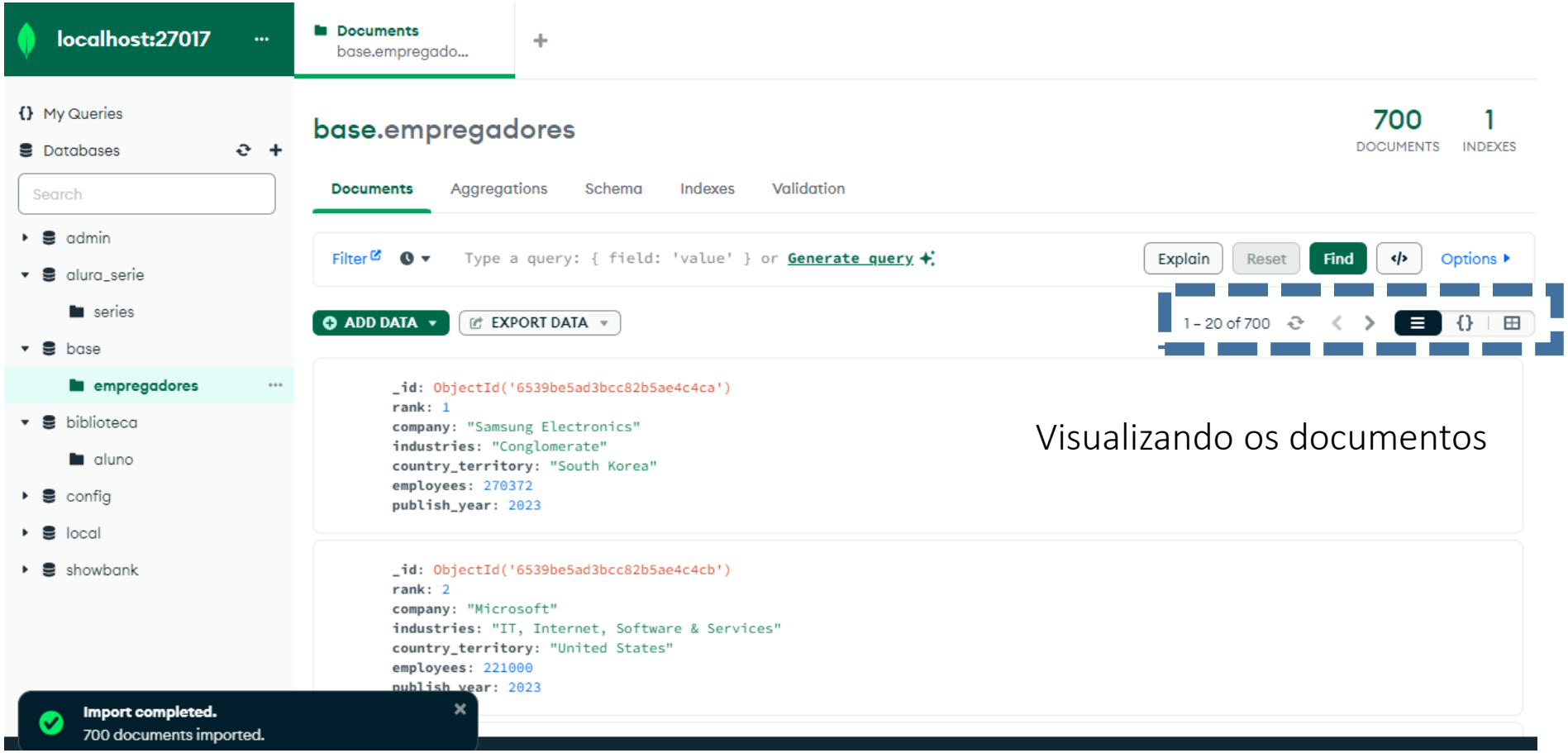
Restrição em coleções

Assim, como para criar um banco de dados e coleções, também existem restrições ao se criar um documento:

- O nome do campo _id é reservado para uso como chave primária. Seu valor deve ser único na coleção, é imutável e pode ser de qualquer tipo que não seja um array.
- Os nomes dos campos não podem conter o caractere NULL.
- Os documentos BSON, também possuem restrições de tamanho:
- O tamanho máximo de um documento BSON é 16 megabytes.

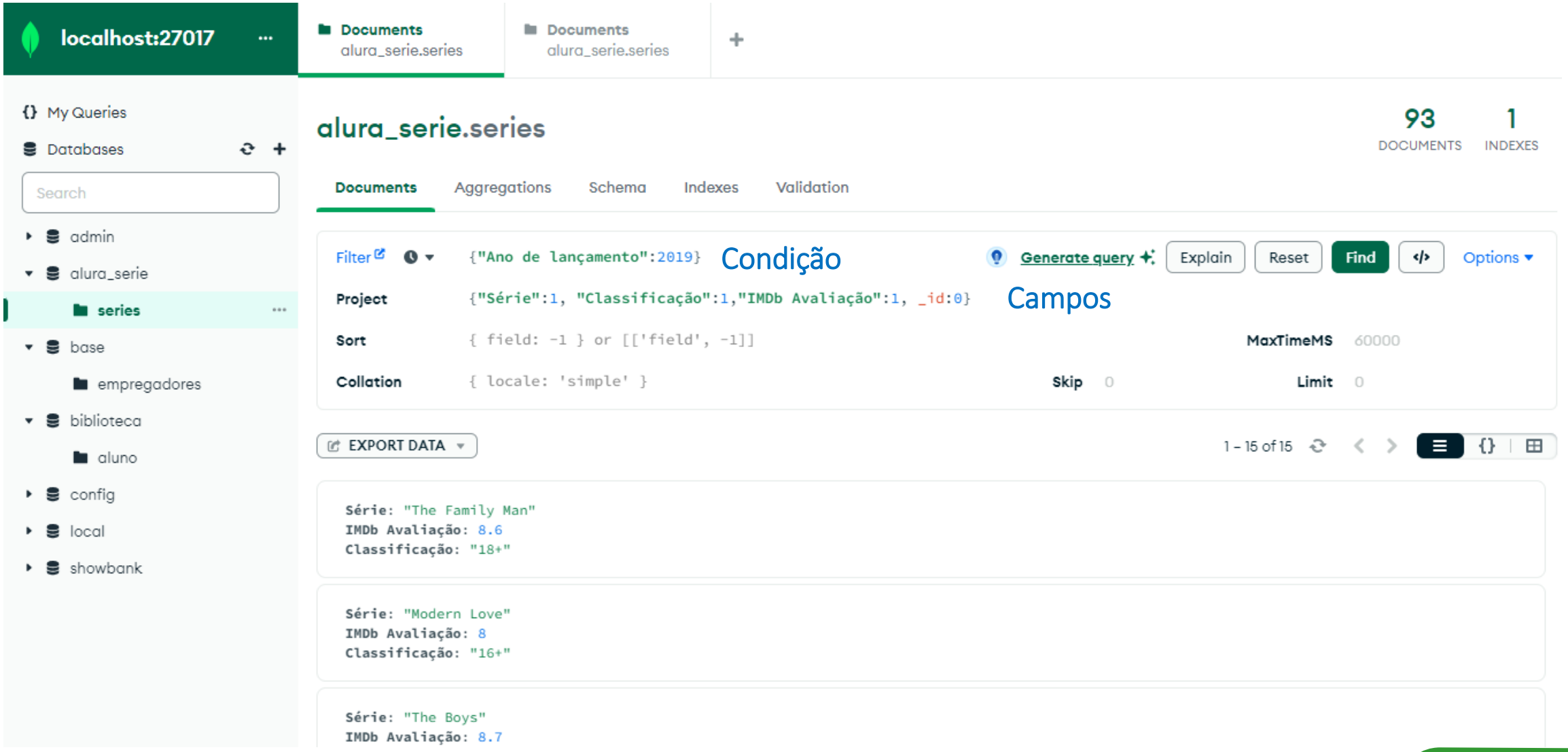
https://www.kaggle.com/datasets/

Consultando os documentos



Visualizando os documentos

Consultando os documentos



- FILTER: utilizado para especificar qual será a condição que os documentos devem atender para serem retornados na busca.
- PROJECT: utilizado para especificar quais campos serão ou não retornados na consulta.

  Ao Informar o nome do campo e informar 0, todos os campos, exceto os campos especificados no campo project, são retornados.

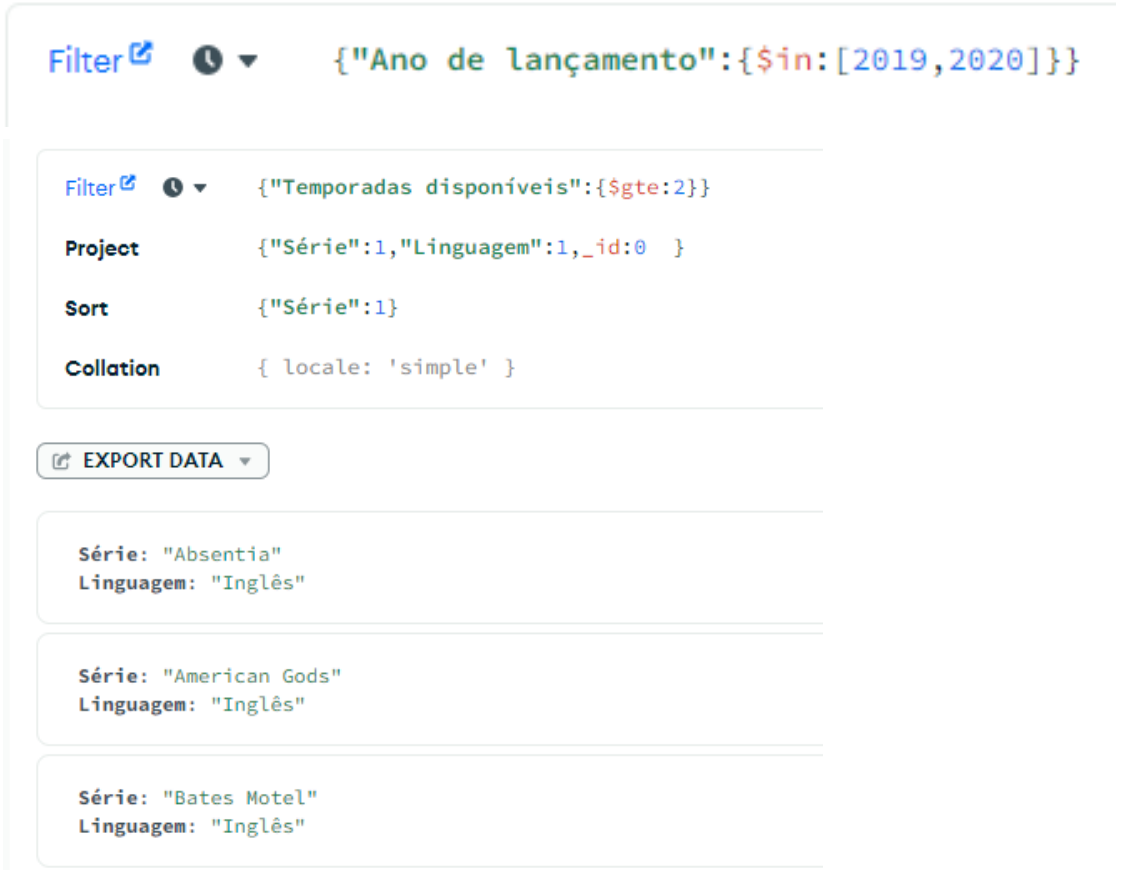
  Se o campo receber o valor de 1, ele será retornado na consulta. O campo _id é retornado por padrão, a menos que este seja especificado no campo project e definido como 0.
- SORT: especifica a ordem de classificação dos documentos retornados.

  Para especificar a ordem crescente de um campo, defina o campo como 1.

  Para especificar a ordem decrescente de um campo, defina o campo como -1.
- MAX TIME MS: define o limite de tempo cumulativo em milissegundos para processar as operações da barra de consulta. Se o limite de tempo for atingido antes da conclusão da operação, o Compass interrompe a operação.
- COLLATION: utilizado para especificar regras específicas do idioma para comparação de textos, como regras para letras maiúsculas ou minúsculas, acentos, entre outros.
- SKIP: especifica quantos documentos devem ser ignorados antes de retornar o conjunto de resultados.
- LIMIT: especifica o número máximo de documentos a serem retornados.

Consultando os documentos – Operadores de Comparação

Nome	Descrição
`$eq`	Corresponde a valores iguais a um valor especificado.
`$gt`	Corresponde a valores maiores que um valor especificado.
`$gte`	Corresponde a valores maiores ou iguais a um valor especificado.
`$in`	Corresponde a qualquer um dos valores especificados em uma matriz.
`$lt`	Corresponde a valores menores que um valor especificado.
`$lte`	Corresponde a valores menores ou iguais a um valor especificado.
`$ne`	Corresponde a todos os valores que não são iguais a um valor especificado.
`$nin`	Não corresponde a nenhum dos valores especificados em uma matriz.



https://www.mongodb.com/docs/manual/reference/operator/query/

Consultando os documentos – Operadores Lógico

Nome	Descrição
`$and`	Unir cláusulas de consulta com uma lógica `AND` retorna todos os documentos que correspondem às condições de ambas as cláusulas.
`$not`	Inverte o efeito de uma expressão de consulta e retorna documentos que não *correspondem* à expressão de consulta.
`$nor`	Unir cláusulas de consulta com uma lógica `NOR` retorna todos os documentos que não correspondem a ambas as cláusulas.
`$or`	Une cláusulas de consulta com uma lógica `OR` que retorna todos os documentos que correspondem às condições de qualquer uma das cláusulas.

```
Filter   {$and:[{"Ano de lançamento":2018},{"Classificação":"18+"}]}
Project  { field: 0 }
Sort     { field: -1 } or [['field', -1]]
Collation { locale: 'simple' }
```

```
Filter   {$or:[{"Ano de lançamento":2018},{"Classificação":"18+"}]}
Project  { field: 0 }
Sort     { field: -1 } or [['field', -1]]
Collation { locale: 'simple' }
```

https://www.mongodb.com/docs/manual/reference/operator/query/

Consultando os documentos

- As operações de leitura recuperam documentos de uma coleção ; ou seja, consulte uma coleção de documentos. O MongoDB fornece os seguintes métodos para ler documentos de uma coleção:
- db.collection.find()
- Você pode especificar filtros de consulta ou critérios que identifiquem os documentos a serem retornados.

```
db.users.find(                          ← collection
   { age: { $gt: 18 } },                ← query criteria
   { name: 1, address: 1 }              ← projection
).limit(5)                              ← cursor modifier
```

Consultando os documentos

/lembre-se de selecionar o database para realizar as consultas
//Consulta todos os documentos da celeção series
db.series.find()
//Consulta todos os registros que tem ano de lançamento 2018
db.series.find({"Ano de lançamento":2018})
//Consulta todos os registros e retorna a projeção (campos) série ano de lançamento e exclui do resultado o id
db.series.find({},{"Série":1,"Ano de lançamento":1,"_id":0})
//Consulta todos os registros que em ano de lançamento tem 2019 e 2020
db.series.find({"Ano de lançamento": {$in:[2019,2020]}})
//Consulta todos os registros limitando a resposta a apenas 5
db.series.find().limit(5)

https://devopedia.org/mongodb-query-language

Atualizando os documentos

ADD DATA | EXPORT DATA | 1 – 20 of 93

series

	_id ObjectId	Série String	Ano de lançamento Int32	Temporadas disponíveis Int32	Linguagem String	Genero Array
1	ObjectId('6539bc0cd3bcc82b5ae…	Mirzapur	String	1	"Hindi"	[] 1 elements
	Document modified.				CANCEL	UPDATE
2	ObjectId('6539bc0cd3bcc82b5ae…	"The Family Man"	2019	1	"Hindi"	[] 1 elements
3	ObjectId('6539bc0cd3bcc82b5ae…	"Modern Love"	2019	1	"Inglês"	[] 3 elements
4	ObjectId('6539bc0cd3bcc82b5ae…	"Comicstaan"	2018	2	"Hindi"	[] 3 elements

Atualizando os documentos

As operações de atualização modificam documentos existentes em uma coleção . MongoDB fornece os seguintes métodos para atualizar documentos de uma coleção:

- db.collection.updateOne() Novo na versão 3.2
- db.collection.updateMany() Novo na versão 3.2
- db.collection.replaceOne() Novo na versão 3.2

No MongoDB, as operações de atualização têm como alvo uma única coleção. Todas as operações de gravação no MongoDB são atômicas no nível de um único documento.

```
db.users.updateMany(                    <-- collection
   { age: { $lt: 18 } },                <-- update filter
   { $set: { status: "reject" } }       <-- update action
)
```

Atualizando os documentos

- Alura_Series> db.series.find({"Série": "Grimm"})

- db.series.updateOne({"Série": "Grimm"},{$set: {"Temporadas disponíveis": 6}})

//Atualizando o documento que corresponde a série Grimm adicionado um campo que não existia

- db.series.updateOne({"Série": "Grimm"},{$set: {"Classificação": "16+"}})

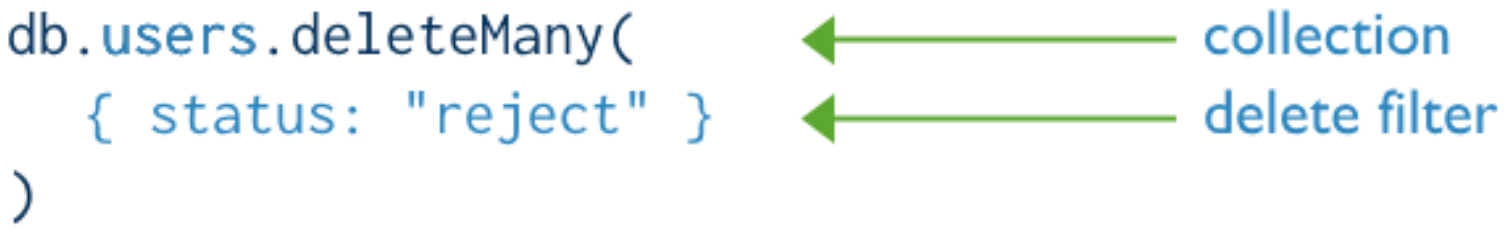
- db.series.find({"Série": {$in: ["Four More Shots Please", "Fleabag"]}})

- db.series.updateMany({"Série":{$in:["Four More Shots Please", "Fleabag"]}},{$set: {"Classificação": "18+"}})

Excluindo documentos

As operações de exclusão removem documentos de uma coleção. O MongoDB fornece os seguintes métodos para excluir documentos de uma coleção:

- db.collection.deleteOne() Novo na versão 3.2
- db.collection.deleteMany() Novo na versão 3.2
- No MongoDB, as operações de exclusão têm como alvo uma única coleção . Todas as operações de gravação no MongoDB são atômicas no nível de um único documento.



Excluindo documentos

- Alura_Series> db.series.find({"Série": "The Boys"})
- db.series.deleteOne({"Série": "The Boys"})
- db.series.deleteMany({"Temporadas disponíveis": 1})
- db.series.deleteMany({})