

Sumário

Conceitos.....	2
Tipos de Dados.....	3
Utilização.....	4
Importando dados	5
Comandos Gerais.....	5
Seleção – R (read)	6
Filtros.....	6
Operadores	7
AND	7
OR.....	8
IN	9
Exercícios	9
LIKE.....	11
Exercícios	12
\$gt e \$lt	13
Exercícios	14
Inserção – C (create).....	15
Seleção – EXTRA	15
Timestamp – ID.....	16
Exercícios	16
Atualização – U (update)	17
Exercícios	17
Deleção – D (delete)	19
Exercícios	20
Desafio.....	21
Conteúdo adicional.....	22
Instalação	22
Autenticação	26

Conceitos

O MongoDB é um document database (banco de dados de documentos) com informações no formato JSON. Um documento pode exibir um valor simples, como por exemplo, o nome da pessoa, um número, data ou uma lista de valores.

Esses documentos são agrupados em coleções. E um conjunto de coleções (vamos pensar em diferentes planilhas) forma um database (banco de dados).

Esquema flexível para persistência de dados.

Tipos de Datos

<https://docs.mongodb.com/manual/core/document/>

<https://data-flair.training/blogs/mongodb-data-types/>

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

Utilização

<https://stackoverflow.com/questions/4883045/mongodb-difference-between-running-mongo-and-mongod-databases>

Criar pasta `C:/data/db` – por padrão é a que o MongoDB utiliza para rodar o servidor

Ir até a pasta de instalação – `'C:\Program Files\MongoDB\Server\4.0\bin'`

Executar **mongod.exe**

mongod basicamente faz com o que processo inicie.

mongo faz com que você se conecte com uma instância específica do mongo.

Caso queira direcionar o caminho em que deseja armazenar as suas informações:

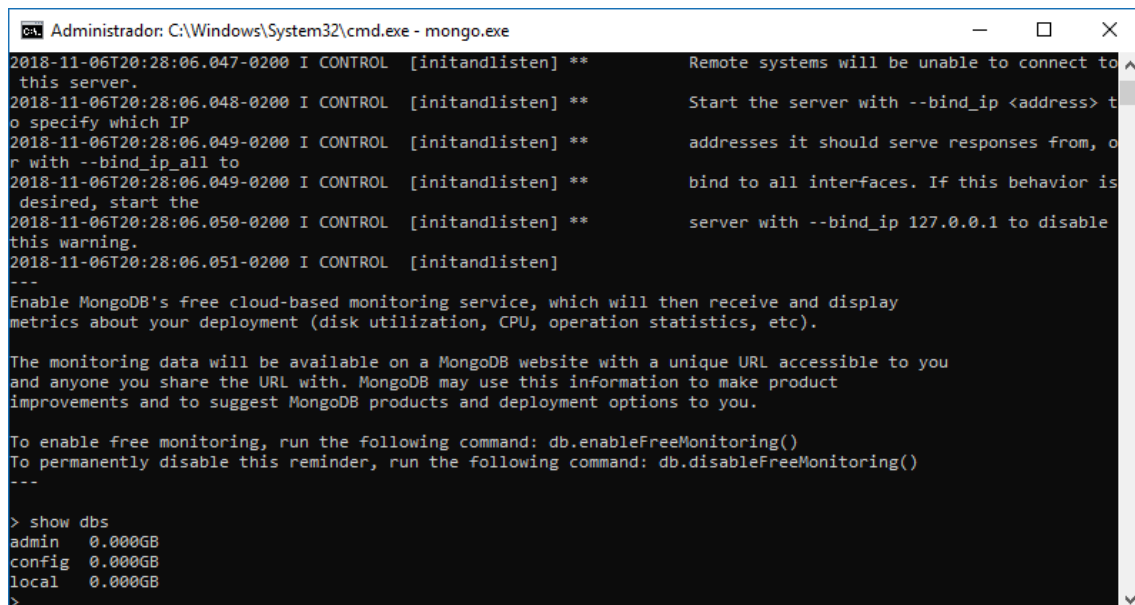
mongod.exe --dbpath="C:\Program Files\MongoDB\Server\4.0\data"

Abra outro CMD/Prompt de Comando e dentro da pasta: `'C:\Program Files\MongoDB\Server\4.0\bin'` execute:

mongo.exe

Podemos também utilizar: `./mongo localhost:27017/dbname`

Por padrão, estes são os bancos criados:



```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
2018-11-06T20:28:06.047-0200 I CONTROL [initandlisten] ** Remote systems will be unable to connect to
this server.
2018-11-06T20:28:06.048-0200 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> t
o specify which IP
2018-11-06T20:28:06.049-0200 I CONTROL [initandlisten] ** addresses it should serve responses from, o
r with --bind_ip_all to
2018-11-06T20:28:06.049-0200 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is
desired, start the
2018-11-06T20:28:06.050-0200 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable
this warning.
2018-11-06T20:28:06.051-0200 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

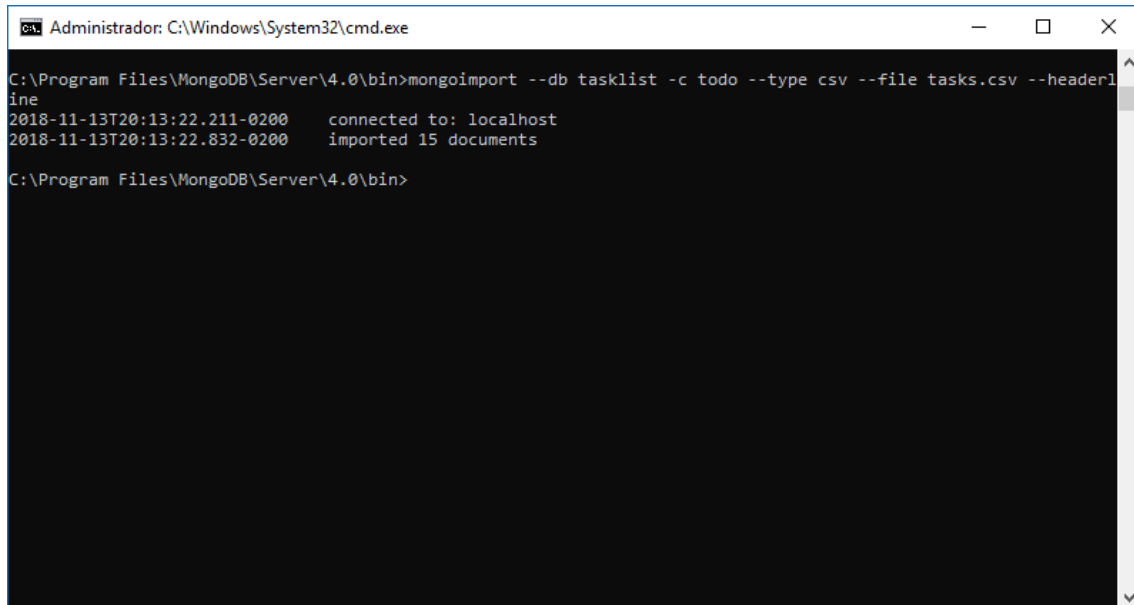
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
>
```

Importando dados

mongoimport serve para importar dados de um determinado documento (Excel, JSON, CSV) para o banco de dados do mongodb.

mongoimport --db tasklist -c todo --type csv --file tasks.csv --headerline



```
Administrador: C:\Windows\System32\cmd.exe
C:\Program Files\MongoDB\Server\4.0\bin>mongoimport --db tasklist -c todo --type csv --file tasks.csv --headerline
2018-11-13T20:13:22.211-0200    connected to: localhost
2018-11-13T20:13:22.832-0200    imported 15 documents
C:\Program Files\MongoDB\Server\4.0\bin>
```

use tasklist

Comandos Gerais

show dbs – mostrar todos os bancos de dados

show collections – mostrar todas as coleções

use tasklist – colocar o banco de dados em uso

db.dropDatabase() – deletar todo o banco de dados

db.todo.count() – contar a quantidade de registros inserida

Seleção – R (read)

`db.todo.findOne()` – seleciona o primeiro registro

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.count()
15
> db.todo.findOne()
{
  "_id" : ObjectId("5beb4c82cfc209629fbacbbba"),
  "ordem" : 1,
  "titulo" : "T@tulo A",
  "descricao" : "mongodb",
  "concluida" : "false",
  "dataInclusao" : "01/01/2018"
}
```

`db.todo.find()` – seleciona todos os registros da coleção selecionada

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find()
{ "_id" : ObjectId("5beb4c82cfc209629fbacbbba"), "ordem" : 1, "titulo" : "T@tulo A", "descricao" : "mongodb", "concluida" : "false", "dataInclusao" : "01/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbbb"), "ordem" : 2, "titulo" : "T@tulo B", "descricao" : "nodejs", "concluida" : "false", "dataInclusao" : "02/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbbc"), "ordem" : 3, "titulo" : "T@tulo B", "descricao" : "native script", "concluida" : "true", "dataInclusao" : "03/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbbd"), "ordem" : 4, "titulo" : "T@tulo C", "descricao" : "angularjs", "concluida" : "true", "dataInclusao" : "04/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbbe"), "ordem" : 5, "titulo" : "T@tulo D", "descricao" : "react", "concluida" : "false", "dataInclusao" : "05/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbbf"), "ordem" : 6, "titulo" : "T@tulo D", "descricao" : "react native", "concluida" : "false", "dataInclusao" : "06/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbc0"), "ordem" : 7, "titulo" : "T@tulo D", "descricao" : "javascript", "concluida" : "true", "dataInclusao" : "07/01/2018" }
{ "_id" : ObjectId("5beb4c82cfc209629fbacbc1"), "ordem" : 8, "titulo" : "T@tulo E", "descricao" : "chocolate", "concluida" : "true", "dataInclusao" : "08/01/2018" }
```

`db.todo.find().pretty()` – busca todos os registros da coleção com uma melhor apresentação

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find().pretty()
{
  "_id" : ObjectId("5beb4c82cfc209629fbacbbba"),
  "ordem" : 1,
  "titulo" : "T@tulo A",
  "descricao" : "mongodb",
  "concluida" : "false",
  "dataInclusao" : "01/01/2018"
}
{
  "_id" : ObjectId("5beb4c82cfc209629fbacbbb"),
  "ordem" : 2,
  "titulo" : "T@tulo B",
  "descricao" : "nodejs",
  "concluida" : "false",
  "dataInclusao" : "02/01/2018"
}
```

Filtros

`db.todo.find({ "descricao" : "es6" }).pretty()` – Realizando filtro. Busca apenas a descrição aonde o valor será igual a es6.

Ele é case sensitive. Isso significa que, caso você digite Descricao no valor da chave, o mesmo não irá ser encontrado.

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({"descricao" : "es6"}).pretty()
{
  "_id" : ObjectId("5beb4c82cfc209629fbacbc8"),
  "ordem" : 15,
  "titulo" : "Título H",
  "descricao" : "es6",
  "concluida" : "false",
  "dataInclusao" : "15/01/2018"
}
> db.todo.find({"Descricao" : "es6"}).pretty()
>
```

`db.todo.find({ "titulo" : "Título B" }).pretty()`

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
{
  "titulo" : "Título A",
  "descricao" : "mongodb",
  "concluida" : "false",
  "dataInclusao" : "01/01/2018"
}
> db.todo.find({ "titulo" : "Título B" }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff884c"),
  "ordem" : 2,
  "titulo" : "Título B",
  "descricao" : "nodejs",
  "concluida" : "false",
  "dataInclusao" : "02/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8851"),
  "ordem" : 3,
  "titulo" : "Título B",
  "descricao" : "native script",
  "concluida" : "true",
  "dataInclusao" : "03/01/2018"
}
>
```

Operadores

`db.todo.find({ "titulo" : "Título B" }, { "titulo" : "Título C" }).pretty()` – Quando são colocadas duas condições simultaneamente, ele irá considerar apenas o primeiro registro da lista.

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({ "titulo" : "Título D" }, { "titulo" : "Título C" })
{ "_id" : ObjectId("5bed52684a295f9527ff884a"), "titulo" : "Título D" }
{ "_id" : ObjectId("5bed52684a295f9527ff884b"), "titulo" : "Título D" }
{ "_id" : ObjectId("5bed52684a295f9527ff884d"), "titulo" : "Título D" }
> db.todo.find({ "titulo" : "Título B" }, { "titulo" : "Título C" }).pretty()
{ "_id" : ObjectId("5bed52684a295f9527ff884c"), "titulo" : "Título B" }
{ "_id" : ObjectId("5bed52684a295f9527ff8851"), "titulo" : "Título B" }
>
```

AND

Trazer todos os registros cujo título é igual a 'Título H' e descrição é igual a 'es6'.

`db.todo.find({ "titulo" : "Título H", "descricao" : "es6" }).pretty()`

```
C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({ "titulo" : "Titulo H", "descricao" : "es6" }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8855"),
  "ordem" : 15,
  "titulo" : "Titulo H",
  "descricao" : "es6",
  "concluida" : "false",
  "dataInclusao" : "15/01/2018"
}
```

OR

Porém, caso quiséssemos trabalhar com a mesma chave correspondente, aonde o título fosse A *ou* B, por exemplo, se utilizássemos a mesma condição, nos traria a lista incorreta.

```
db.todo.find({ "titulo" : "Titulo B", "titulo" : "Titulo D" }).pretty()
```

```
C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({ "titulo" : "Titulo B", "titulo" : "Titulo D" }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff884a"),
  "ordem" : 5,
  "titulo" : "Titulo D",
  "descricao" : "react",
  "concluida" : "false",
  "dataInclusao" : "05/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884b"),
  "ordem" : 6,
  "titulo" : "Titulo D",
  "descricao" : "react native",
  "concluida" : "false",
  "dataInclusao" : "06/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884d"),
  "ordem" : 7,
  "titulo" : "Titulo D",
  "descricao" : "javascript",
  "concluida" : "true",
  "dataInclusao" : "07/01/2018"
}
```

Ele considera apenas o último registro. Para realizar essa condição, devemos incluir o operador \$or. Para nos dizer que queremos um valor *ou* outro.

```
db.todo.find({ $or: [{ "titulo" : "Titulo H"}, {"titulo" : "Titulo A" }] }).pretty()
```

Lembrando: não esquecer de incluir as chaves e colchetes correspondentes.


```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({ $or: [{ "titulo" : "Titulo H"}, {"titulo" : "Titulo A" }] }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8850"),
  "ordem" : 1,
  "titulo" : "Titulo A",
  "descricao" : "mongodb",
  "concluida" : "false",
  "dataInclusao" : "01/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8855"),
  "ordem" : 15,
  "titulo" : "Titulo H",
  "descricao" : "es6",
  "concluida" : "false",
  "dataInclusao" : "15/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8856"),
  "ordem" : 11,
  "titulo" : "Titulo H",
  "descricao" : "bootstrap",
  "concluida" : "true",
  "dataInclusao" : "11/01/2018"
}
>
```

IN

Trabalhando com o operador IN, nós podemos colocar dois valores para uma única chave, por exemplo:

```
db.todo.find({ "titulo" : { $in : ["Titulo A", "Titulo B"] } }).pretty()
```

```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({ "titulo" : { $in : ["Titulo A", "Titulo B"] } }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff884c"),
  "ordem" : 2,
  "titulo" : "Titulo B",
  "descricao" : "nodejs",
  "concluida" : "false",
  "dataInclusao" : "02/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8850"),
  "ordem" : 1,
  "titulo" : "Titulo A",
  "descricao" : "mongodb",
  "concluida" : "false",
  "dataInclusao" : "01/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8851"),
  "ordem" : 3,
  "titulo" : "Titulo B",
  "descricao" : "native script",
  "concluida" : "true",
  "dataInclusao" : "03/01/2018"
}
>
```

Exercícios

1. Selecione todos os registros onde o título seja igual a 'Titulo B' e a descrição 'nodejs'. Após, o Título seja 'Titulo K' e a descrição seja 'html'.

```
db.todo.find({"titulo" : "Titulo B" , "descricao" : "nodejs"}).pretty()
```

```
db.todo.find({ "titulo" : "Titulo K", "descricao" : "html" }).pretty()
```

2. Selecione todos os registros em que o título seja igual a 'Titulo B' OU 'Titulo C'.

```
db.todo.find({ $or: [{ "titulo" : "Titulo B"}, { "titulo" : "Titulo C" }] }).pretty()
```

3. Pegue o exercício acima e inclua também aonde a descrição seja igual 'javascript'.

```
db.todo.find({ $or: [{ "titulo" : "Titulo B"}, { "titulo" : "Titulo C" }, { "descricao": "javascript" }] }).pretty()
```

4. Trabalhe com um operador diferente de 'OU' para trazer todos os registros aonde as descrições sejam iguais a 'mongodb', 'css' ou 'jquery'.

```
db.todo.find({ $or: [{ "descricao" : "mongodb" }, { "descricao" : "css" }, { "descricao" : "jquery" }] }).pretty()
```

```
db.todo.find( { "descricao" : { $in : [ "mongodb", "css", "jquery" ] } }).pretty()
```

Utilizando os dois operadores juntos. \$or e \$in.

5. Procure todas as tarefas cujo título seja igual a A ou a descrição seja igual a es6 ou bootstrap.

```
db.todo.find({
```

```
  $or : [
```

```
    { "titulo" : "Titulo A" },
```

```
    { "descricao" : { $in : [ "es6", "bootstrap" ] } }
```

```
  ]
```

```
}).pretty()
```

```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
... ]
... }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8850"),
  "ordem" : 1,
  "titulo" : "Título A",
  "descricao" : "mongodb",
  "concluida" : "false",
  "dataInclusao" : "01/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8855"),
  "ordem" : 15,
  "titulo" : "Título H",
  "descricao" : "es6",
  "concluida" : "false",
  "dataInclusao" : "15/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8856"),
  "ordem" : 11,
  "titulo" : "Título H",
  "descricao" : "bootstrap",
  "concluida" : "true",
  "dataInclusao" : "11/01/2018"
}
>
```

LIKE

`db.todo.find({ "descricao" : /e/ }).pretty()` – uma descrição que contenha um e em qualquer parte.

```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe localhost:27017/tasklist
> db.todo.find({ "descricao" : /e/ }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff884a"),
  "ordem" : 5,
  "titulo" : "Título D",
  "descricao" : "react",
  "concluida" : "false",
  "dataInclusao" : "05/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884b"),
  "ordem" : 6,
  "titulo" : "Título D",
  "descricao" : "react native",
  "concluida" : "false",
  "dataInclusao" : "06/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884c"),
  "ordem" : 2,
  "titulo" : "Título B",
  "descricao" : "nodejs",
  "concluida" : "false",
  "dataInclusao" : "02/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884e"),
```

`db.todo.find({ "descricao" : /e$/ }).pretty()` – uma descrição que termine com a letra e

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe localhost:27017/tasklist

    "dataInclusao" : "12/01/2018"
  }
}
> db.todo.find( { "descricao" : /e$/ }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff884b"),
  "ordem" : 6,
  "titulo" : "Titulo D",
  "descricao" : "react native",
  "concluida" : "false",
  "dataInclusao" : "06/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884f"),
  "ordem" : 8,
  "titulo" : "Titulo E",
  "descricao" : "chocolate",
  "concluida" : "true",
  "dataInclusao" : "08/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8853"),
  "ordem" : 12,
  "titulo" : "Titulo J",
  "descricao" : "backbone",
  "concluida" : "true",
  "dataInclusao" : "12/01/2018"
}
}
```

`db.todo.find({ "descricao" : /^r/ }).pretty()` – uma descrição que comece com a letra r

```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe localhost:27017/tasklist

}
> db.todo.find( { "descricao" : /^r/ }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff884a"),
  "ordem" : 5,
  "titulo" : "Titulo D",
  "descricao" : "react",
  "concluida" : "false",
  "dataInclusao" : "05/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884b"),
  "ordem" : 6,
  "titulo" : "Titulo D",
  "descricao" : "react native",
  "concluida" : "false",
  "dataInclusao" : "06/01/2018"
}
}
```

Exercícios

1. Selecione todos os registros em que as descrições contenham em qualquer parte da palavra, 'js'.

`db.todo.find({ "descricao" : /js/ }).pretty()`

2. Selecione todos os registros em que as descrições terminem com a letra 't'.

`db.todo.find({ "descricao" : /t$/ }).pretty()`

3. Selecione todos os registros em que as descrições terminem com a letra 'l'.

`db.todo.find({ "descricao" : /l$/ }).pretty()`

4. Selecione todos os registros em que as descrições terminem com a letra 't' ou 'l'.

```
db.todo.find({ "descricao" : {$in: [/l$/, /t$/] }}).pretty()
```

```
db.todo.find({ $or : [ { "descricao" : /l$/ }, { "descricao" : /t$/ } ] }).pretty()
```

5. Selecione todos os registros em que as descrições comecem com a letra 'b'.

```
db.todo.find( { "descricao" : /^b/ }).pretty()
```

6. Selecione todos os registros em que as descrições comecem com a letra 'j' ou 'h'.

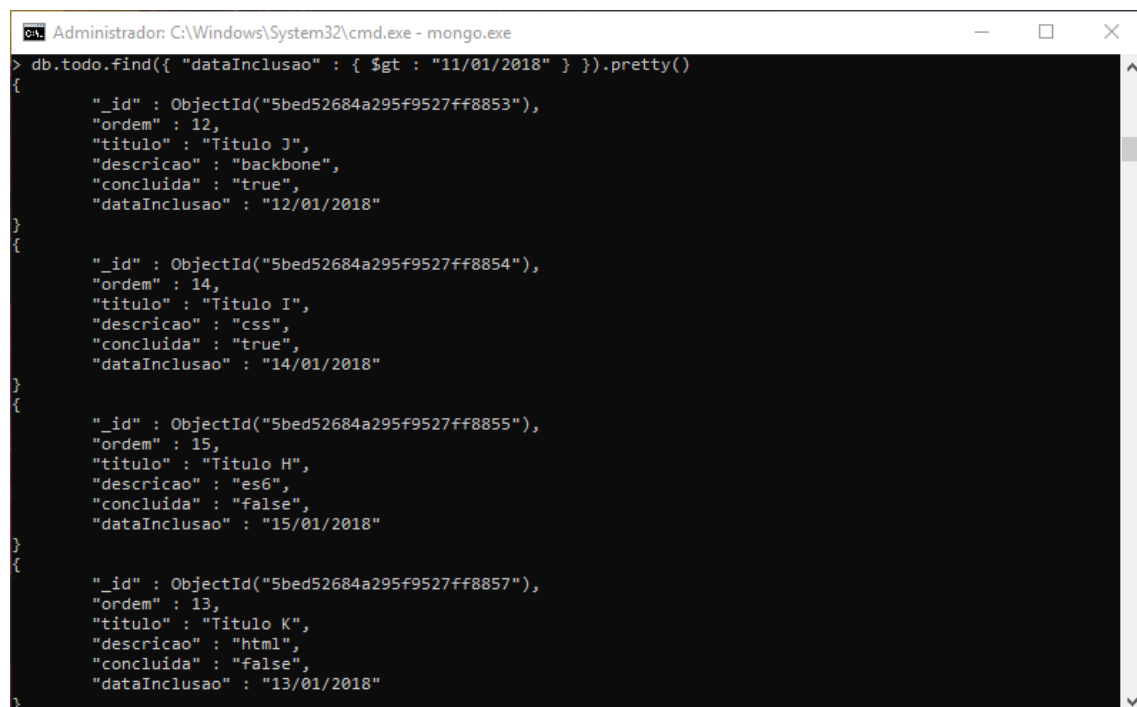
```
db.todo.find({ $or : [ { "descricao" : /^j/ }, { "descricao" : /^h/ } ] }).pretty()
```

```
db.todo.find( { "descricao" : { $in: [ /^j/, /^h/ ] } }).pretty()
```

\$gt e \$lt

Todas as tarefas cuja data seja superior a 11/01/2018.

```
db.todo.find({ "dataInclusao" : { $gt : "11/01/2018" } }).pretty()
```



```
C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find( { "dataInclusao" : { $gt : "11/01/2018" } }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8853"),
  "ordem" : 12,
  "titulo" : "Título J",
  "descricao" : "backbone",
  "concluida" : "true",
  "dataInclusao" : "12/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8854"),
  "ordem" : 14,
  "titulo" : "Título I",
  "descricao" : "css",
  "concluida" : "true",
  "dataInclusao" : "14/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8855"),
  "ordem" : 15,
  "titulo" : "Título H",
  "descricao" : "es6",
  "concluida" : "false",
  "dataInclusao" : "15/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8857"),
  "ordem" : 13,
  "titulo" : "Título K",
  "descricao" : "html",
  "concluida" : "false",
  "dataInclusao" : "13/01/2018"
}
```

Todas as tarefas cuja data seja inferior a 11/01/2018.

```
db.todo.find({ "dataInclusao" : { $lt : "11/01/2018" } }).pretty()
```

```
CA: Administrador: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.find({ "dataInclusao" : { $lt : "11/01/2018" } }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8849"),
  "ordem" : 4,
  "titulo" : "Titulo C",
  "descricao" : "angularjs",
  "concluida" : "true",
  "dataInclusao" : "04/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884a"),
  "ordem" : 5,
  "titulo" : "Titulo D",
  "descricao" : "react",
  "concluida" : "false",
  "dataInclusao" : "05/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884b"),
  "ordem" : 6,
  "titulo" : "Titulo D",
  "descricao" : "react native",
  "concluida" : "false",
  "dataInclusao" : "06/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884c"),
  "ordem" : 2,
  "titulo" : "Titulo B",
  "descricao" : "nodejs",
  "concluida" : "false",
  "dataInclusao" : "02/01/2018"
}
```

Exercícios

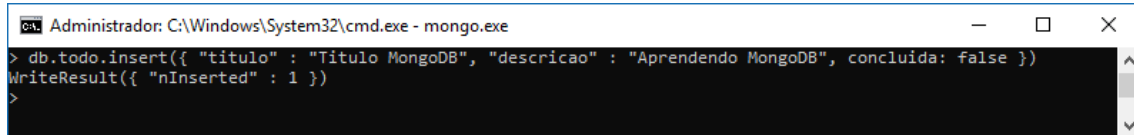
Traga somente os registros aonde o valor da data seja superior a 11/01/2018 e inferior a 13/01/2018.

```
db.todo.find({ "dataInclusao" : { $gt : "11/01/2018", $lt : "13/01/2018" } }).pretty()
```

```
CA: Administrador: C:\Windows\System32\cmd.exe - mongo.exe
"dataInclusao" : "01/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8851"),
  "ordem" : 3,
  "titulo" : "Titulo B",
  "descricao" : "native script",
  "concluida" : "true",
  "dataInclusao" : "03/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8852"),
  "ordem" : 10,
  "titulo" : "Titulo G",
  "descricao" : "jquery",
  "concluida" : "false",
  "dataInclusao" : "10/01/2018"
}
> db.todo.find({ "dataInclusao" : { $gt : "11/01/2018", $lt : "13/01/2018" } }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8853"),
  "ordem" : 12,
  "titulo" : "Titulo J",
  "descricao" : "backbone",
  "concluida" : "true",
  "dataInclusao" : "12/01/2018"
}
>
```

Inserção – C (create)

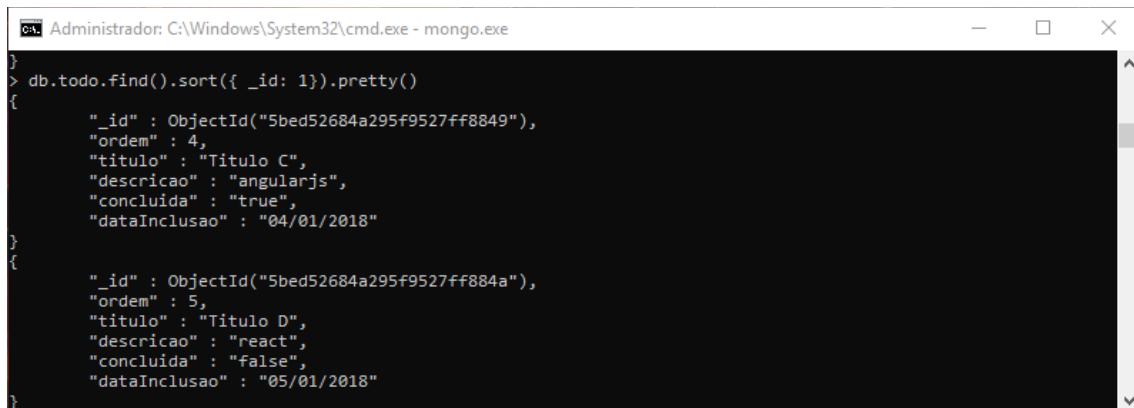
```
db.todo.insert({ "titulo" : "Titulo MongoDB", "descricao" : "Aprendendo MongoDB", concluida: false })
```



```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.insert({ "titulo" : "Titulo MongoDB", "descricao" : "Aprendendo MongoDB", concluida: false })
WriteResult({ "nInserted" : 1 })
>
```

Seleção – EXTRA

```
db.todo.find().sort({ _id: 1 }).pretty()
```



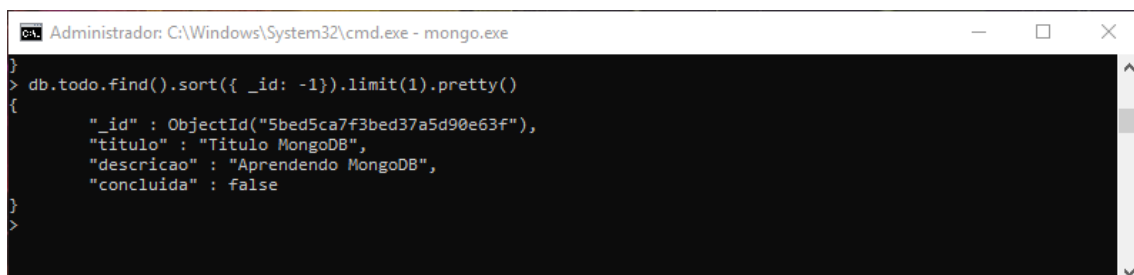
```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
}
> db.todo.find().sort({ _id: 1 }).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8849"),
  "ordem" : 4,
  "titulo" : "Titulo C",
  "descricao" : "angularjs",
  "concluida" : "true",
  "dataInclusao" : "04/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884a"),
  "ordem" : 5,
  "titulo" : "Titulo D",
  "descricao" : "react",
  "concluida" : "false",
  "dataInclusao" : "05/01/2018"
}
```

```
db.todo.find().sort({ _id: -1 }).pretty()
```



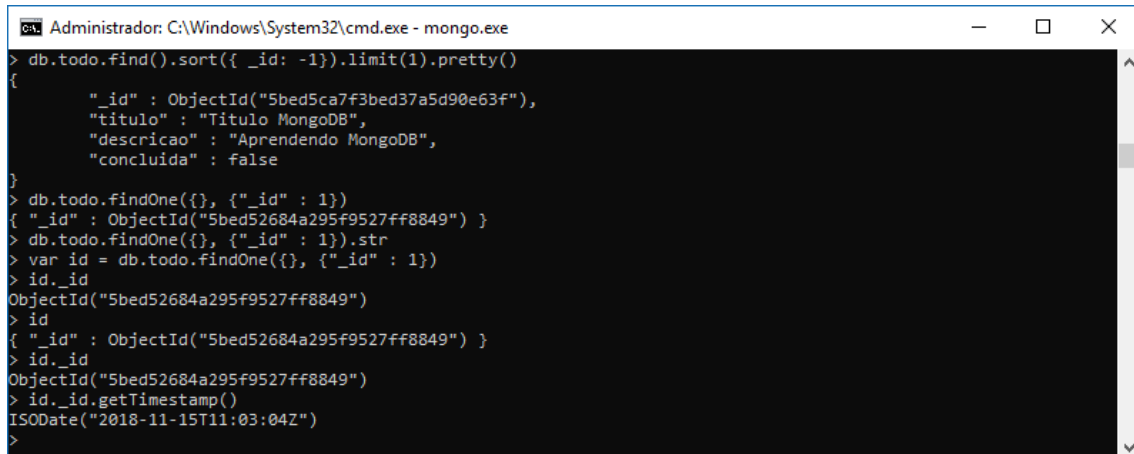
```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
}
> db.todo.find().sort({ _id: -1 }).pretty()
{
  "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f"),
  "titulo" : "Titulo MongoDB",
  "descricao" : "Aprendendo MongoDB",
  "concluida" : false
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8857"),
  "ordem" : 13,
  "titulo" : "Titulo K",
  "descricao" : "html",
  "concluida" : "false",
  "dataInclusao" : "13/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8856"),
```

```
db.todo.find().sort({ _id: -1 }).limit(1).pretty()
```



```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
}
> db.todo.find().sort({ _id: -1 }).limit(1).pretty()
{
  "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f"),
  "titulo" : "Titulo MongoDB",
  "descricao" : "Aprendendo MongoDB",
  "concluida" : false
}
>
```

Timestamp – ID



```
> db.todo.find().sort({ _id: -1}).limit(1).pretty()
{
  "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f"),
  "titulo" : "Titulo MongoDB",
  "descricao" : "Aprendendo MongoDB",
  "concluida" : false
}
> db.todo.findOne({}, {"_id" : 1})
{ "_id" : ObjectId("5bed52684a295f9527ff8849") }
> db.todo.findOne({}, {"_id" : 1}).str
> var id = db.todo.findOne({}, {"_id" : 1})
> id._id
ObjectId("5bed52684a295f9527ff8849")
> id
{ "_id" : ObjectId("5bed52684a295f9527ff8849") }
> id._id
ObjectId("5bed52684a295f9527ff8849")
> id._id.getTimestamp()
ISODate("2018-11-15T11:03:04Z")
>
```

Exercícios

1. Insira 5 novos registros conforme descrições abaixo:

Título = Título U

Descrição = Java

Concluída = verdadeiro

Título = Título O

Descrição = CSharp

Concluída = falso

Título = Título P

Descrição = Firebase

Título = Título Y

Descrição = Microsoft

Título = Título R

Descrição = Oracle

Data de Inclusão = data de hoje

Atualização – U (update)

Caso eu faça isso, sem setar os valores que eu quero no campo, o documento inteiro será atualizado com apenas o campo que foi passado.

```
db.todo.update(  
  { "titulo": "Titulo MongoDB" },  
  { "descricao": "Alterando a descricao" }  
)
```

```
db.todo.update(  
  { "titulo": "Titulo MongoDB" },  
  {  
    $set : { "descricao": "Alterando a descricao" }  
  }  
)
```



```
Administrador: C:\Windows\System32\cmd.exe - mongo.exe  
> db.todo.find().sort({ _id: -1}).limit(1).pretty()  
{  
  "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f"),  
  "titulo" : "Titulo MongoDB",  
  "descricao" : "Aprendendo MongoDB",  
  "concluida" : false  
}  
> db.todo.update(  
... { "titulo": "Titulo MongoDB" },  
... {  
... $set : { "descricao": "Alterando a descricao" }  
... }  
... )  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.todo.find().sort({ _id: -1}).limit(1).pretty()  
{  
  "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f"),  
  "titulo" : "Titulo MongoDB",  
  "descricao" : "Alterando a descricao",  
  "concluida" : false  
}  
>
```

Exercícios

1. Atualize os registros criados no exercício anterior, com os filtros abaixo (não se esqueça de buscar pelo _id):

Titulo = Titulo Uava

Descrição = Java

Concluída = verdadeiro

Titulo = Titulo OSharp

Descrição = CSharp

Concluída = falso

Titulo = Titulo Pirebase

Descrição = Firebase

Titulo = Titulo YMicrosoft

Descrição = Microsoft

Titulo = Titulo Roracle

Descrição = Oracle

Data de Inclusão = data de hoje

```
Administrator: C:\Windows\System32\cmd.exe - mongo
> db.todo.insert({ "titulo" : "Titulo R", "descricao" : "Oracle", "dataInclusao" : new Date() })
WriteResult({ "nInserted" : 1 })
> db.todo.find().sort({_id : -1}).limit(1).pretty()
{
  "_id" : ObjectId("5bfaf989796edb16931958e8"),
  "titulo" : "Titulo R",
  "descricao" : "Oracle",
  "dataInclusao" : ISODate("2018-11-25T19:35:37.156Z")
}
> db.todo.update( {_id:ObjectId("5bfaf6b4796edb16931958e7")}, { $set: { "descricao" : "Oracle Database" } } )
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.todo.update( {_id: ObjectId("5bfaf989796edb16931958e8") }, { $set: { "descricao" : "Oracle Database" } } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.todo.find().sort({_id : -1}).limit(1).pretty()
{
  "_id" : ObjectId("5bfaf989796edb16931958e8"),
  "titulo" : "Titulo R",
  "descricao" : "Oracle Database",
  "dataInclusao" : ISODate("2018-11-25T19:35:37.156Z")
}
>
```

Deleção – D (delete)

`db.todo.remove({ "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f") })`

```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.remove({ "_id" : ObjectId("5bed5ca7f3bed37a5d90e63f") })
WriteResult({ "nRemoved" : 1 })
>
```

Deletando com filtros

`db.todo.remove({ "dataInclusao" : { $gt : "13/01/2018" } })`

```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
> db.todo.remove({ "dataInclusao" : { $gt : "13/01/2018" } })
WriteResult({ "nRemoved" : 2 })
>
```

Ordenando por data de inclusão

`db.todo.find().sort({ dataInclusao: -1}).pretty()`

```
Administrator: C:\Windows\System32\cmd.exe - mongo.exe
}
> db.todo.find().sort({ dataInclusao: -1}).pretty()
{
  "_id" : ObjectId("5bed52684a295f9527ff8857"),
  "ordem" : 13,
  "titulo" : "Título K",
  "descricao" : "html",
  "concluida" : "false",
  "dataInclusao" : "13/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8853"),
  "ordem" : 12,
  "titulo" : "Título J",
  "descricao" : "backbone",
  "concluida" : "true",
  "dataInclusao" : "12/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8856"),
  "ordem" : 11,
  "titulo" : "Título H",
  "descricao" : "bootstrap",
  "concluida" : "true",
  "dataInclusao" : "11/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff8852"),
  "ordem" : 10,
  "titulo" : "Título G",
  "descricao" : "jquery",
  "concluida" : "false",
  "dataInclusao" : "10/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884e"),
  "ordem" : 9,
  "titulo" : "Título F",
  "descricao" : "vuejs",
  "concluida" : "false",
  "dataInclusao" : "09/01/2018"
}
{
  "_id" : ObjectId("5bed52684a295f9527ff884f"),
  "ordem" : 8,
  "titulo" : "Título E",
  "descricao" : "chocolate",
  "concluida" : "false",
  "dataInclusao" : "08/01/2018"
}
```

Exercícios

1. Exclua os últimos 5 itens criados e atualizados nos itens anteriores com as seguintes condições:

Os dois primeiros pelo _id.

Titulo = Titulo Uava

Descrição = Java

Concluída = verdadeiro

Titulo = Titulo OSharp

Descrição = CSharp

Concluída = falso

Faça o filtro pelo título no item abaixo.

Titulo = Titulo Pirebase

Descrição = Firebase

Faça o filtro pela descrição, nos itens que seguem:

Titulo = Titulo Ycrosoft

Descrição = Microsoft

Titulo = Titulo Roracle

Descrição = Oracle

Data de Inclusão = data de hoje

Desafio

Você receberá um arquivo com o formato JSON contendo alguns personagens de quadrinhos e/ou desenhos animados e você deverá realizar as alterações que foram solicitadas abaixo pelo cliente.

1. Selecione todos os registros.
 - a. `db.personagens.find().pretty()`
2. Selecione apenas o primeiro registro.
 - a. `db.personagens.findOne()`
 - b. `db.personagens.find().sort({_id:1}).limit(1).pretty()`
3. Selecione todos os registros em que o nome seja igual a 'Carla' ou a quantidade de gatos seja igual a 2.
 - a. `db.personagens.find({ $or : [{"Nome" : "Carla" }, {"Qty Gatos" : 2}] }).pretty()`
4. Selecione todos os registros em que o local de nascimento seja igual a Roma e necessariamente que a quantidade de cachorros seja igual a 3.
 - a. `db.personagens.find({"Local" : "Roma", "Qty Cachorros" : 3 }).pretty()`
5. Selecione todos os registros em que a quantidade de gatos seja igual a 3 ou a quantidade de cachorros seja igual a 6.
 - a. `db.personagens.find({ $or: [{"Qty Gatos" : 3}, {"Qty Cachorros" : 6 }] }).pretty()`
6. Selecione todos os registros em que o nome comece com C.
 - a. `db.personagens.find({"Nome" : /^C/ }).pretty()`
7. Selecione todos os registros em que o nome comece com H.
 - a. `db.personagens.find({"Nome" : /^H/ }).pretty()`
8. Selecione todos os registros em que o nome termine com a.
 - a. `db.personagens.find({"Nome" : /a$/ }).pretty()`
9. Selecione todos os registros em que o nome contenha s.
 - a. `db.personagens.find({"Nome" : /s/ }).pretty()`
10. Selecione todos os registros em que o nome contenha 'e' ou 'o'.
 - a. `db.personagens.find({ $or : [{"Nome" : /e/}, {"Nome": /o/ }] }).pretty()`
11. Selecione todos os registros em que a ordem seja maior do que 10.
 - a. `db.personagens.find({"Ordem" : { $gt : 10 } }).pretty()`
12. Selecione todos os registros em que a quantidade de gatos seja maior do que 2 ou menor do que 6.
 - a. `db.personagens.find({"Qty Gatos" : { $gt : 2, $lt : 6 } }).pretty()`
13. Insira 3 novos super-heróis.
14. Delete 1 super-herói em que o nome seja igual a Thor.
15. Atualize a quantidade de gatos em que o usuário com o registro de nome 'Carla' possui. Atualize a quantidade de gatos de '0' para '7'.
16. Selecione todos os registros que foram trabalhados e exporte para um arquivo JSON.

Conteúdo adicional

Instalação

MongoDB Atlas

<https://www.mongodb.com/download-center?jmp=nav#atlas>

Atlas	Community Server	Enterprise Server	Ops Manager	Compass	Connector for BI	Charts
-------	------------------	-------------------	-------------	---------	------------------	--------

MongoDB Atlas

Database as a Service

Deploy, operate, and scale a MongoDB database in the cloud with just a few clicks. Fully elastic and highly available by default, MongoDB Atlas is the easiest way to try out the latest version of the database, **MongoDB 4.0**.

- Available on AWS, GCP, Azure
- Secure from the start
- Fully managed backups
- Comprehensive monitoring and customizable alerts
- Easily migrate existing deployments with minimal downtime

[Click here](#) to learn more about MongoDB Atlas.

No download necessary

Deploy a free cluster now

Email Address
h.strada@hotmail.com

First Name
Helena

Last Name
Strada

Password
••••••••

✓ 8 characters minimum
✓ One number
✓ One letter
✓ One special character

Company
SENAI

Phone Number
+5511999659429

Other
▼

Brazil
▼

☒ I agree to the [terms of service](#).

Get started free

MongoDB Community Server

https://www.mongodb.com/dr/fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.1-signed.msi/download

[Atlas](#) [Community Server](#) [Enterprise Server](#) [Ops Manager](#) [Compass](#) [Connector for BI](#) [Charts](#)

[Current Release](#) | [Previous Releases](#) | [Development Releases](#)

Current Stable Release (4.0.1)
08/03/2018: [Release Notes](#) | [Changelog](#)
Download Source: [tgz](#) | [zip](#)

Windows

Linux

OSX

Version:

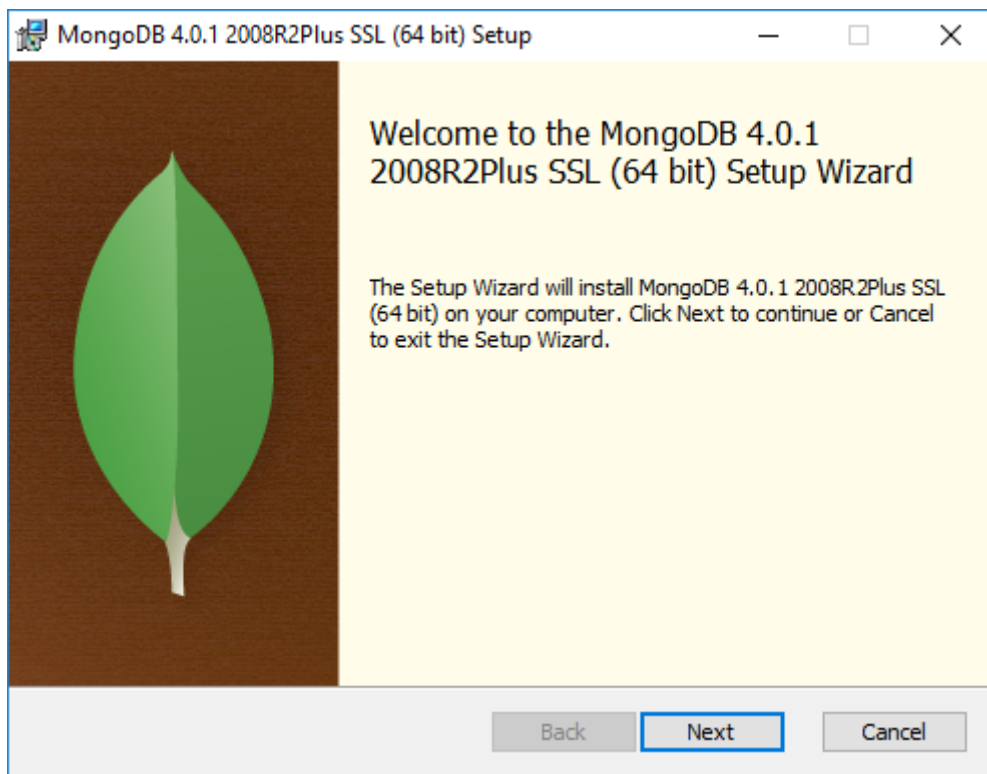
Windows 64-bit x64

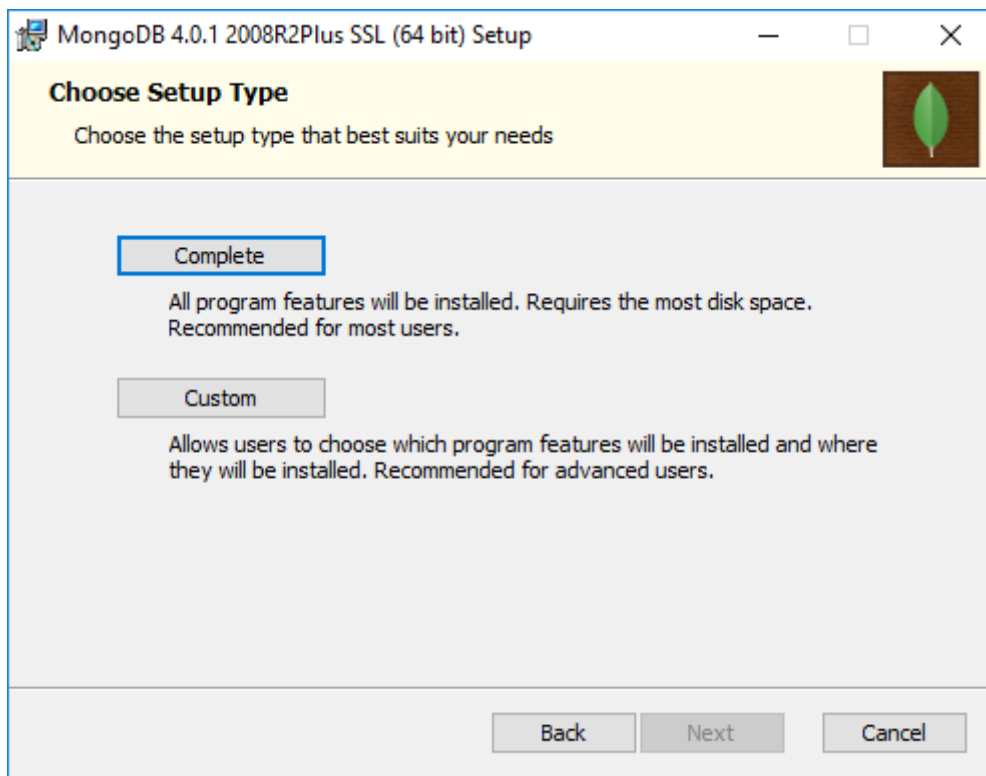
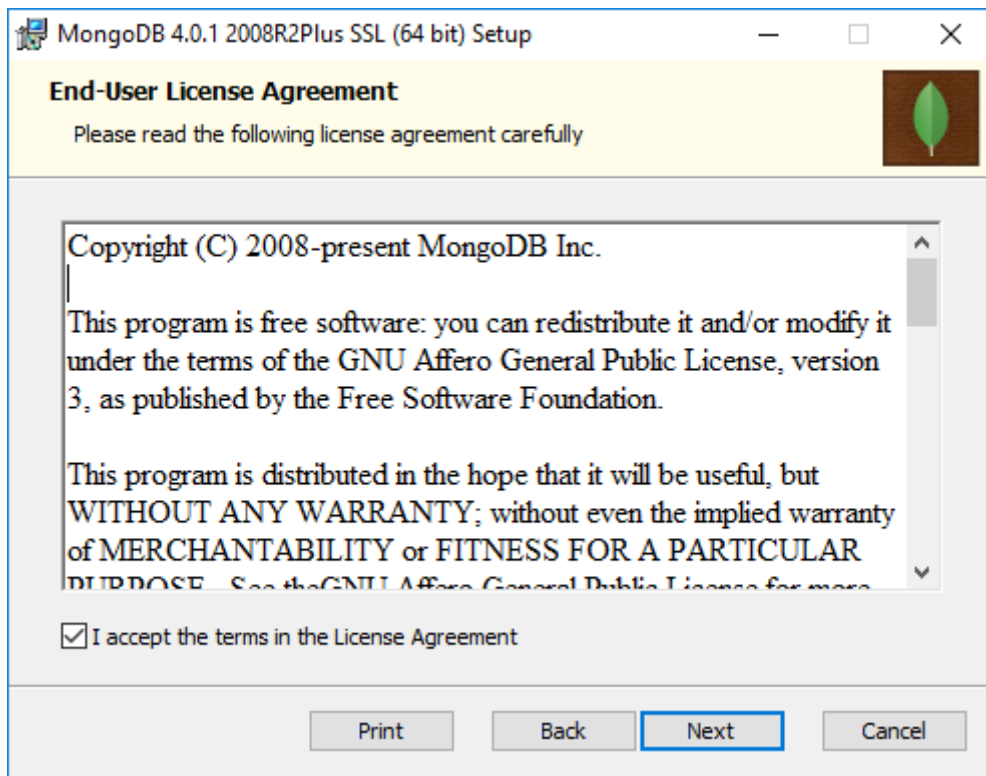
Installation Package:

DOWNLOAD (msi)

Binary: [Installation Instructions](#) | [All Version Binaries](#)

Deploy a free cluster in the cloud with MongoDB Atlas, our database service that gets you up and running in minutes.





MongoDB 4.0.1 2008R2Plus SSL (64 bit) Service Customization

Service Configuration

Specify optional settings to configure MongoDB as a service.

☒ Install MongoDB as a Service

☒ Run service as Network Service user

☐ Run service as a local or domain user:

Account Domain:

Account Name:

Account Password:

Service Name:

Data Directory:

Log Directory:

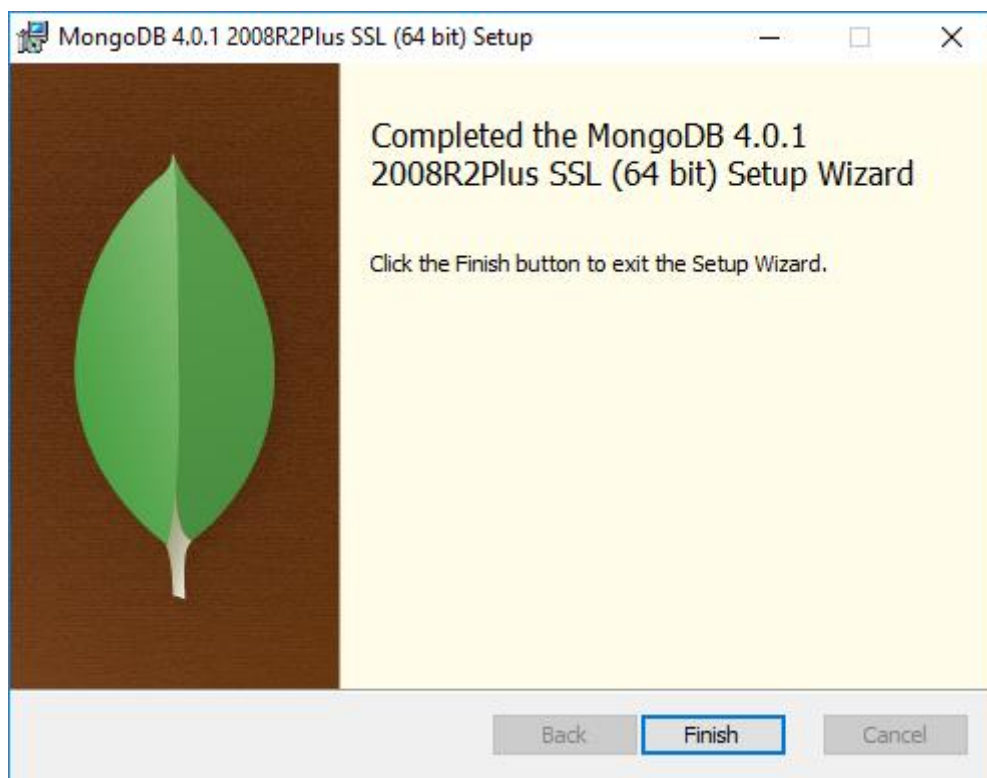
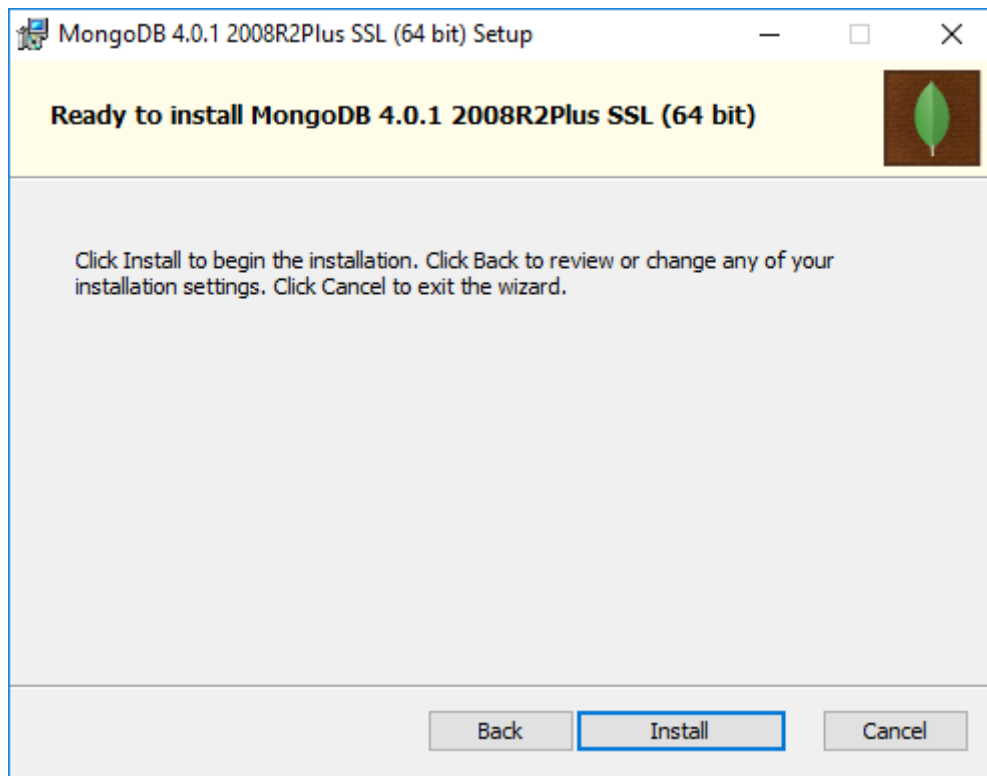
MongoDB Compass

Install MongoDB Compass

MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: <https://www.mongodb.com/products/compass>

☒ Install MongoDB Compass



Autenticação

<https://docs.mongodb.com/manual/tutorial/enable-authentication/>