1. Selecione todos os registros.

db.tasks.find()

1. Selecione todos os registros e deixe a com apresentação melhor.

db.tasks.find().pretty()

1. Selecione todos os registros em que o título seja igual a ‘Titulo A’.

db.tasks.find({"titulo" : "Titulo A"}).pretty()

1. Selecione todos os registros em que o título seja igual a ‘Titulo C’.

db.tasks.find({"titulo" : "Titulo C"}).pretty()

1. Selecione todos os registros em que a descrição seja “mongodb”.

db.tasks.find({"descricao" : "mongodb"}).pretty()

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

db.tasks.find({“titulo”: “Titulo B”},{"descricao" : "nodejs"}).pretty()

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

db.tasks.find($ or: [{“titulo”: “Titulo B”},{"titulo" : "titulo C"}]).pretty()

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

db.tasks.find($ or: [{“titulo”: “Titulo B”},{"titulo" : "titulo C"}, {“descrição”: “javascript”}]).pretty()

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

db.tasks.find({descrição: {$in: [“mongodb”, “css”, “jquery”] }} ).pretty()

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

1. Procure todas as tarefas cujo título seja igual a A ou a descrição seja igual a es6 ou bootstrap.
2. db.tasks.find({$or: [{“titulo”: “Titulo A”}, {descricao: {$in: [“es6”, “bootstrap”]}} ]})pretty()

Para procurar uma letra:

db.tasks.find({descricao: /e/}).pretty() (contem e)

db.tasks.find({descricao: /e$/}).pretty() (finaliza com e)

db.tasks.find({descricao: /^e/}).pretty() (inicia com e)

db.tasks.find({descricao: /E/i}).pretty() (independente de maiúscula ou minúscula) i=ignorecase

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

db.tasks.find({descricao: /js/}).pretty()

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

db.tasks.find({descricao: /t$/}).pretty()

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

> db.tasks.find({descricao: /l$/}).pretty()

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

> db.tasks.find( {$or: [{descricao: /l$/}, {descricao: /t$/}]}).pretty()

> db.tasks.find({descricao: {$in: [ /t$/, /l$/ ]}}).pretty()

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

> db.tasks.find({descricao: /^b/}).pretty()

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

> db.tasks.find( {$or: [{descricao: /^j/}, {descricao: /^h/}]}).pretty()

> db.tasks.find({descricao: {$in: [ /^j/, /^h/ ]}}).pretty()

> db.tasks.find().sort({ \_id : 1}).limit(1).pretty() – crescente

> db.tasks.find().sort({ \_id : -1}).limit(1).pretty() – descrescente

Ordene pela data de inclusão.

> db.tasks.find().sort({ \_id : 1}).pretty()

Ordene pelas descrições.

db.tasks.find().sort({ \_descricao : 1}).pretty()

Ordene pela quantidade de caracteres das descrições.

Procure somente registros aonde o dia da data de inclusão seja igual a 01.

> db.tasks.find({dataInclusao: /^01/}).sort({ \_id : 1}).pretty()

03/12/2019

1. Insira 5 novos registros conforme descrições abaixo:

Titulo = Titulo U

Descrição = Java

Concluída = verdadeiro

db.tasks.insert({"titulo": "Titulo U", "descricao": "Java", "concluida": "verdadeiro"})

Titulo = Titulo O

Descrição = CSharp

Concluída = falso

db.tasks.insert({"titulo": "Titulo O", "descricao": "CSharp", "concluida": "falso"})

Titulo = Titulo P

Descrição = Firebase

db.tasks.insert({"titulo": "Titulo P", "descricao": "Firebase"})

Titulo = Titulo Y

Descrição = Microsoft

db.tasks.insert({"titulo": "Titulo Y", "descricao": "Microsoft"})

Titulo = Titulo R

Descrição = Oracle

Data de Inclusão = data de hoje

> db.tasks.insert({"titulo": "Titulo R", "descricao": "Oracle", "dataInclusao":"03/12/2019"})

Db.tasks.insertMany([{“titulo”: “Titulo O”, “descricao”: “”}, {“titulo”: “ndjefkhkjw”, “descricao”: “nfjkdhswk”}])

“dataInclusao”: new Date() – mostra a data

Let teste = new Date ()

Teste – mostra a data

Para atualizar:

db.tasks.update({ “titulo”: “Titulo Q”, “descrição” : “Descricao A”})

db.tasks.update({\_id: fjioewhjori}, {$ set “titulo”: “Titulo Q”, “descrição” : “Descricao A”})

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

Titulo = Titulo Uava

Descrição = Java Best

Concluída = verdadeiro

db.tasks.update({ “titulo”: “Titulo U”})

db.tasks.update({\_id: ObjectId("5de6419a89ef1177d42fef21")}, {$set: {“titulo”: “Titulo Uava”, “descricao” : “Java Best”, “concluida”: “true”}})

Titulo = Titulo OSharp

Descrição = CShark

Concluída = falso

Titulo = Titulo Pirebase

Descrição = Fireb4se

Titulo = Titulo YMicrosoft

Descrição = MicroSOFT

Titulo = Titulo Roracle

Descrição = Oracle Inc.

Data de Inclusão = data de hoje

> db.tasks.deleteOne({"\_id": ObjectId("5de642fa89ef1177d42fef25")})