**mongoimport --db m\_alunos -c alunos --jsonArray --file alunos.json**

#### Exercícios

1. Selecione todos os registros.

**db.alunos.find()**

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

**db.alunos.find().pretty()**

1. Selecione todos os registros em que o nome seja igual a ‘Aya’.

**db.alunos.find({ "nome" : "Aya" }).pretty()**

1. Selecione todos os registros em que o aluno nasceu em São Paulo.

**db.alunos.find({ "nasceuEmSp" : true }).pretty()**

1. Selecione todos os registros em que contenha a título do livro igual a “Java - Web Developers”.

**db.alunos.find({ "livros.titulo" : "Java - Web Developers"}).pretty()**

1. Selecione todos os registros em que a aluna tenha o livro “Clean Code A” e o nome “Helena”.

**db.alunos.find({"livros.titulo" : "Clean Code A" , "nome" : “Helena” }).pretty()**

1. Selecione todos os registros em que a aluna não tenha nascido em sp e tenha o livro Cangaceiro JavaScript.  
   **db.alunos.find({"nasceuEmSp" : false , "livros.titulo" : "Cangaceiro JavaScript" }).pretty()**
2. Selecione todos os registros em que contenham os alunos que tenham o livro Java – Web Developers ou Clean Code B. Primeiro: realize o exercício com o operador ou e posteriormente realize o exercício com o operador in.  
   **db.alunos.find({ $or: [{ "livros.titulo" : "Clean Code B"} , { "livros.titulo" : "Java - Web Developers" }] }).pretty()**

**db.alunos.find({ "livros.titulo" :{ $in: ["Clean Code B", "Java - Web Developers"] }}).pretty()**

1. Pegue o exercício acima e inclua também aonde o aluno pode ter nascido em sp.  
   **db.alunos.find({ $or: [{ "livros.titulo" : "Clean Code B"} , { "livros.titulo" : "Java - Web Developers" }, {“nasceuEmSp” : true}] }).pretty()**
2. Procure todos os alunos que tenham nascido em são paulo ou os livros ‘MySQL’ e ‘MongoDB’.

**db.alunos.find({**

**$or : [**

**{ "nasceuEmSp" : true },**

**{ "livros.titulo" : { $in : [ "MySQL", "MongoDB" ] } }**

**]**

**}).pretty()**

#### Exercícios

1. Selecione todos os registros em que os livros contenham em qualquer parte da palavra, 'Clean'.

**db.alunos.find({ "livros.titulo" : /clean/i }).pretty()**

1. Selecione todos os registros em que os títulos dos livros terminem com a letra 'S'.

**db.alunos.find({ "livros.titulo" : /s$/ }).pretty() – sem o ignore case**

**db.alunos.find({ "livros.titulo" : /s$/i }).pretty() – com o ignore case**

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

**db.alunos.find({ "livros.titulo" : /t$/ }).pretty()**

1. Selecione todos os nomes das alunos terminem com a letra 'k' ou 'o'.

**db.alunos.find({ "nome" : {$in: [/k$/, /o$/] }}).pretty()**

**db.alunos.find({ $or : [ { "nome" : /k$/ }, { "nome" : /o$/ } ] }).pretty()**

1. Selecione todos os registros em que os nomes comecem com a letra 'a'.

**db.alunos.find( { "nome" : /^a/i } ).pretty()**

1. Selecione todos os registros em que nomes comecem com a letra 'a' ou 'h'.

**db.alunos.find({ $or : [ { "nome" : /^a/i}, {"nome" : /^h/i } ] }).pretty()**

**db.alunos.find( { "nome" : { $in : [ /^h/i, /^a/i ] } } ).pretty()**

#### Ordenação

**db.alunos.find().sort({ \_id: 1}).pretty()** - ordena pelo campo **\_id** de forma crescente

**db.alunos.find().sort({ nome: 1}).pretty()**

**db.alunos.find().sort({ \_id: -1}).pretty()** - ordena pelo campo **\_id** de forma decrescente

**db.alunos.find().sort({ nome: -1}).pretty()**

**db.alunos.find().sort({ nome: -1}).limit(1).pretty()** - ordena pelo campo **nome** de forma crescente e pega o primeiro registro

**db.alunos.find().limit(1).pretty() –** seleciona somente o primeiro registro

## Create

**-- inserir apenas um registro na instrução**

db.alunos.insert(

{

"nome": "Helena A",

"dateOfBirth": "1993-03-18T08:30:00.000Z",

"nasceuEmSp": false,

"livros": [

{"titulo":"MySQL" , "leu" : true},

{"titulo": "MongoDB", "leu" :true},

{"titulo": "Java", "leu":true},

{"titulo": "Clean Architecture", "leu" :true}

]

}

)

**-- inserir vários registros em uma única instrução**

db.alunos.insertMany([

{

"nome": "Helena B",

"dateOfBirth": "1993-03-18T08:30:00.000Z",

"nasceuEmSp": true,

"livros": [

{"titulo":"C#" , "leu" : true},

{"titulo": "JavaScript", "leu" :true},

{"titulo": "Java", "leu":true},

{"titulo": "Clean Architecture", "leu" :true}

]

}

,

{

"nome": "Helena C",

"dateOfBirth": "1993-03-18T08:30:00.000Z",

"nasceuEmSp": false,

"livros": [

{"titulo":"NodeJS" , "leu" : true},

{"titulo": "Java", "leu":true},

{"titulo": "Clean Architecture", "leu" :true}

]

}

])

-- verificar se os registros foram inseridos corretamente.

**db.alunos.find({}).sort({ \_id : -1 }).limit(3).pretty()**

#### Exercícios

1. Crie 5 novos registros de alunos.

Deve conter os critérios abaixo. 3 em sp e 2 que não nasceram em sp.

Deve conter pelo menos alguns dos livros colocados: HTML, CSS, Design, JavaScript, Java, C#, MySQL, MongoDB.

A data de nascimento deve ser inferior a data atual.

## Update

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

db.alunos.update(

{ "nome" : "Helena C" },

{ "nasceuEmSp" : false }

)

**db.alunos.find({}).sort({ \_id : -1 }).limit(1).pretty()**

db.alunos.update(

{ "nome" : "Helena B" },

{

$set : { "nasceuEmSp" : true }

}

)

#### Exercícios

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

Deixe todos os registros criados acima que estiverem como true para false, e vice-versa.

Atualize a data de nascimento de 2 alunos.

Inclua pelo menos 3 livros que elas tenham lido.

db.alunos.update(

{ "\_id" : ObjectId("5dab90e10d61146c420264aa") },

{ $push: { “livros”: { “titulo” : “Design”, “leu” : true } } }

## Delete

**db.alunos.remove({ "\_id" : ObjectId("5dab90e10d61146c420264aa") })**

**db.alunos.remove({ "\_id" : ObjectId("5dab90e10d61146c420264ab") })**

#### Exercícios

1. Exclua 5 itens dada uma determinada condição.

Exemplo:

Por ordenação de inserção, os dois primeiros que foram inseridos.

Por ordem alfabética, os dois primeiros.

Por ordem de signo decrescente, o primeiro signo.

**Desafio Extra**

Por ordem de habilidades, a que possui o maior registro de habilidades.

db.alunos.aggregate([

{

$project: {

item: 1,

numeroDeHabilidades: { $cond: { if: { $isArray: "$habilidades" }, then: { $size: "$habilidades" }, else: "NA"} }

}

}

] )

Verificar o registro que possui a maior quantidade de habilidades.

db.alunos.find({ \_id : ObjectId("5d811ade31c542921b14417b") }).sort({ \_id : -1 }).limit(3).pretty()

Com ordenação

db.alunos.aggregate([

{

$project: {

item: 1,

numeroDeHabilidades: { $cond: { if: { $isArray: "$habilidades" }, then: { $size: "$habilidades" }, else: "NA"} }

}

},

{ $sort: { numeroDeHabilidades: 1 } }

] )