

INTRUÇÕES PARA O MONGO – CRIAR BANCO E UMA COLEÇÃO:

entrar na pasta do mongo pelo terminal:

cd Program Files\MongoDB\Server\4.4\bin

subir o servidor:

mongod

executa o mongo:

mongo

Checar os bancos já criados:

show databases

para criar um novo banco e/ou selecionar(Checar se esse nome já não existe):

use nomeBanco

após criado o banco, devemos criar a coleção:

db.createCollection("nomeColecao")

caso queira verificar a coleção que está trabalhando no caso que acabou de criar:

show collections

o comando **show databases** irá mostrar todos os bancos q existem no servidor

INTRUÇÕES PARA O MONGO – RESTAURAR O BD DE UM BSON:

Criar um banco e uma coleção conforme os passos anteriores para ter onde restaurar o banco

É necessário ter os configs de dbrestore na máquina para seguir os passos abaixo, em:

C:\Program Files\MongoDB\Server\4.4\bin

Deixar o arquivo .bson de bkp na pasta bin (**C:\Program Files\MongoDB\Server\4.4\bin**)

entrar na pasta do mongo pelo terminal:

cd Program Files\MongoDB\Server\4.4\bin

Fora do mongo, executar o comando para que o restore seja criado no servidor:

**mongorestore --host localhost --port 27017 --db nomeDoBanco --collection nomeColecao
nomeArquivoRestore.bson**

executar o mongo(OBS: Pode-se utilizar dois terminais, um para os comandos de config e outro para os comandos do mongo):

mongo

entrar no banco restaurado (se não souber o nome, executar: **show databases** p/ mostrar os bancos no servidor):

use nomeBancoRestaurado

para verificar os documentos, executar (se não souber a coleção: **show collections**):

db.nomeColecao.find().pretty()

INTRUÇÕES PARA O MONGO – REALIZAR CONSULTAS SIMPLES:

Consultar todos os documentos

db.nomeColecao.find().pretty()

consultar documentos por id ou campo:

db.nomeColecao.find({"_id": 239}).pretty()

consultar todos os documentos que tenham produto cujo prodId = 6610:

db.nomeColecao.find({"lineItems.prodId" : 6610}).pretty()

consultar o produto o primeiro produto cujo id = 239

db.nomeColecao.findOne({"_id" : 239})

INTRUÇÕES PARA O MONGO – FAZER INSERÇÃO:

Para inserir documentos ou um documento:

db.nomeColecao.insert({ "_id": 258, "nome": "José" }, { "_id": 250, "nome": "Maria" })

Para inserir apenas um documento:

db.nomeColecao.insertOne({ "_id": 258, "nome": "José" })

Para inserir mais de um documento:

db.nomeColecao.insertMany({ "_id": 258, "nome": "José" }, { "_id": 250, "nome": "Maria" })

para checar as inserções:

db.nomeColecao.count()

INTRUÇÕES PARA O MONGO – FAZER INSERÇÃO:

Para apagar todos os documentos da coleção:

db.nomeColecao.remove()

Para apagar um documento específico

db.nomeColecao.remove({"_id" : 1405})

mostrar a remoção:

db.bares.count()

INTRUÇÕES PARA O MONGO – CONSULTAS MAIS COMPLEXAS:

Mostrar consulta com o filtro de lineltems, sem mostrar o id do registro do documento:

```
db.api.find({"_id" : 310}, {"lineltems.prodId" : true, "_id": false}).pretty()
```

Consulta com operador and. Selecionar o id dos pedidos que tenham os produtos 4324 e 3220:

```
db.api.find({$and: [ {"lineltems.prodId" : 4324}, {"lineltems.prodId": 3220} ] }, {"_id" : true} )
```

Consulta com operador or. Selecionar o id dos pedidos que tenham os produtos 4324 ou 3220:

```
db.api.find({$or: [ {"lineltems.prodId" : 4324}, {"lineltems.prodId": 3220} ] }, {"_id" : true} )
```

INTRUÇÕES PARA O MONGO – COMANDO UPDATE:

Update altera não só os documentos mais como as coleções e esquema também, por isso deve ter muito cuidado ao usar o comando update:

Para alterar um determinado atributo, é necessário o operador set:

```
db.nomeColecao.update( {"id": ObjectId("5d5dsfsd5f4s5df4")}, {$ set {"nome" : "Zé"}})
```

Para alterar todos o atributo de todos os documentos, utiliza-se da sintaxe:

```
db.nomeColecao.update( {}, {$ set {"departamento" : "RH"}}, {multi: true})
```

Note que é necessário o multi: true, caso não seja passado esse objeto será alterado o primeiro documento somente

*Alterar o valor de um atributo dentro de um array:

```
db.nomeColecao.update(
{ "_id" : 11476 },
{ $ set : { "lineltems.$[elem].prodCount" : 10 } },
{ arrayFilters: [{ "elem.prodId" : 4477 } ] }
)
```

*Operador mul, faz a multiplicação:

```
db.api.update(
{"_id":11476},
{$mul: {"lineltems.$[elem].prodCount":2}},
{arrayFilters: [{"elem.prodId": 4477}]}
)
```

Lembrando que se caso queira fazer a multiplicação para todos deve adicionar o multi:

```
db.api.update(
{"lineltems.prodId ":5697},
{$mul: {"lineltems.$[elem].prodCount":1.5}},
{multi: true, arrayFilters: [{"elem.prodId": 5697}]} ,
)
```

Alterar o esquema da Coleção, removendo atributo:

```
db.api.update(  
  {"_id ":254},  
  {$unset: {"lineItems.$[elem].prodCost": ""}},  
  { arrayFilters: [{"elem.prodId": 5697}]},  
)
```