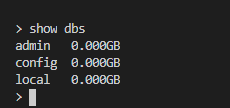
Aula MongoDB

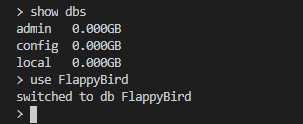
***Usando MongoDB***

Com ambos os terminais abertos, iremos, no segundo, começar a digitar alguns comandos e manipular um pouco nosso banco de dados Mongo.

Para mostrar todos os nossos bancos, usamos o comando ***show dbs***, do Ingles, Mostrar Banco de Dados (dbs = Data Bases)



Para acessar um banco de dados, usamos o comando ***use*** ***nomeDoBanco***, caso ele não exista, será criado



Iremos criar nossa primeira coleção dentro do nosso banco, para isso digite ***db.nomeDaColecao.insert(argumentos)***. O comando ***insert*** é um método que recebe argumentos, um desses argumentos são os dados da coleção, no nosso caso, um documento JSON, veja o exemplo:



Criei uma coleção chamada jogadores e inserir um documento JSON nele com nome e pontuação, simulando o nosso esquema visto mais cedo. O retorno ***nInserted : 1*** mostra que deu tudo certo com a operação

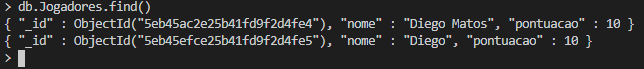
Agora, para realmente nos certificar que os dados foram registrados na coleção Jogadores, podemos usar o método ***find()***, que também recebe argumentos, mas se deixarmos em branco, ele irá retornar todos os documentos dessa coleção



Como você pode observar, nosso jogador foi registrado com sucesso, e também possui uma propriedade a mais chamada ID, essa propriedade é gerada automaticamente pelo mongo, e é única.

Também podemos passar um JavaScript

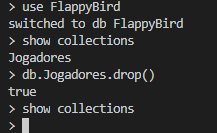




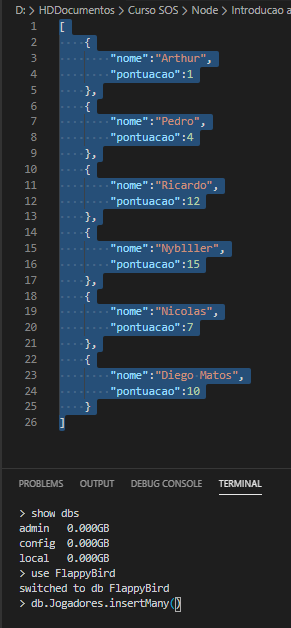
Podemos usar o comando ***show collections*** para ver todas as coleções do nosso banco de dados



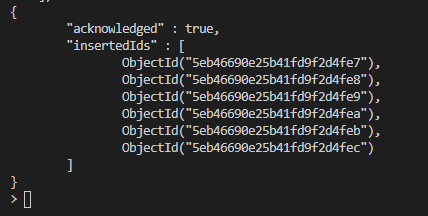
Para deletarmos uma coleção, basta usar o método ***drop()***

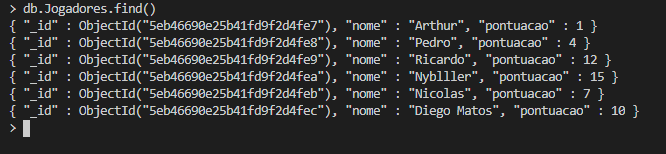


Agora a coleção jogadores foi deletada. Mas iremos cria-la de novo e continuar com os exercícios. Dessa vez, iremos adicionar muitos dados de uma só vez, usando o método ***insertMany()***

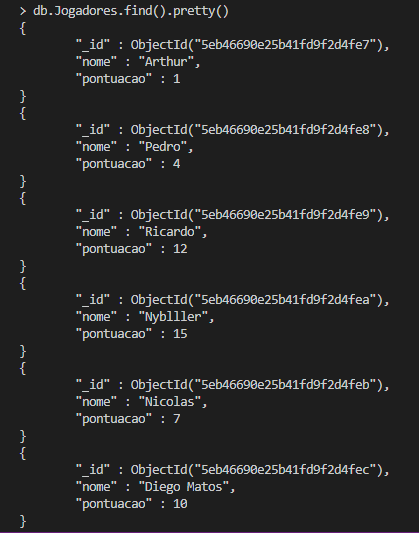


Basta copiar e colar dentro do parênteses



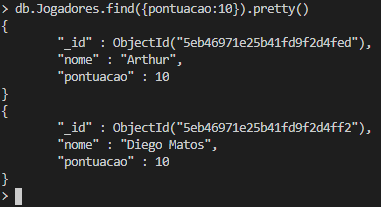


Podemos usar o método ***pretty()*** no final do ***find()*** para deixar os dados mais fácil de ler

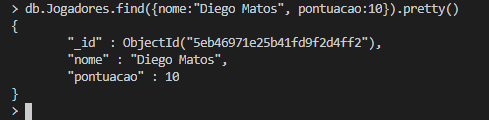


***Busca***

Podemos procurar por ***todos*** os documentos que possuírem um valor, simplesmente passando um a propriedade e o valor no ***find()***

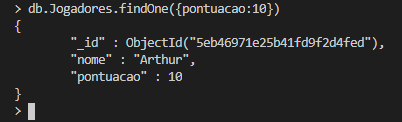


Esses são todos os jogadores que possuem 10 pontos. Podemos ser mais específicos ainda e passar mais uma propriedade



Assim, esses são todos os jogadores que possuem essa propriedade, no nosso exemplo, apenas um.

Também podemos usar o método ***findOne()***, que irá retornar o primeiro valor encontrado. Recriando o primeiro exemplo de busca:

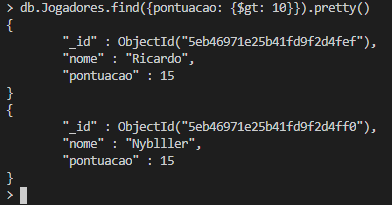


***Obs***: o método pretty() não funciona no findOne(), pois ele retornar apenas um documento

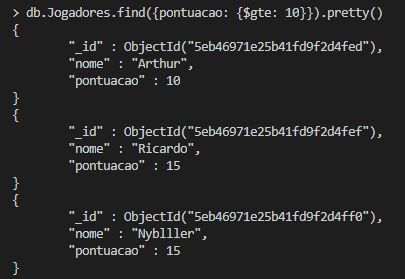
***Operadores Query***

Basicamente podemos realizar buscas passando uma “***relação lógica***” como parâmetro, esse relação é um objeto e é um pouco diferente de como estamos acostumados.

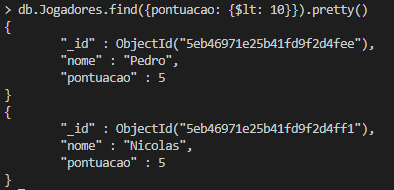
**Maior que:** O operador query é ***$gt***. Abreviatura para ***Greater Than***. Exemplo: Buscar todos os jogadores com pontuação maior que 10



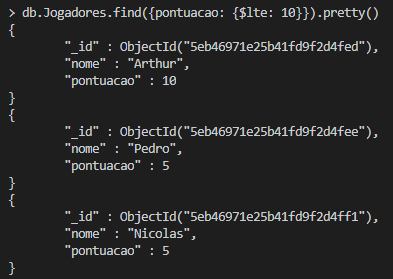
**Maior que ou igual:** O operador query é ***$gte***. Abreviatura para ***Greater Than or Equal***. Exemplo: Buscar todos os jogadores com pontuação maior que ou igual à 10



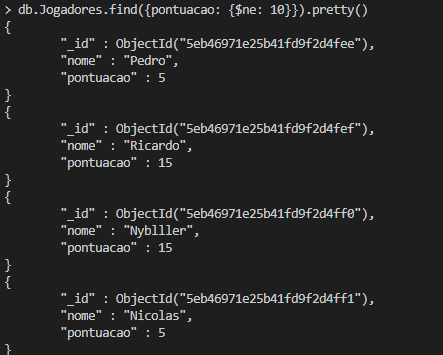
**Menor que:** O operador query é ***$lt***. Abreviatura para ***Less Than***. Exemplo: Buscar todos os jogadores com pontuação menor que 10



**Menor que ou igual:** O operador query é ***$lte***. Abreviatura para ***Less Than or Equal***. Exemplo: Buscar todos os jogadores com pontuação menor ou igual à 10



**Não é igual:** O operador query é ***$ne***. Abreviatura para ***Not Equal***. Exemplo: Buscar todos os jogadores que não tem uma pontuação de 10

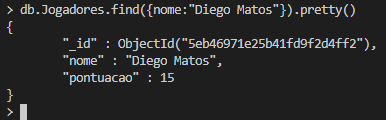


***Atualizar***

Digamos que um dos jogadores teve sua pontuação alterada, ele jogou nosso jogo e realizou mais pontos, agora, precisaremos alterar a pontuação dele no banco de dados, para isso iremos utilizar os métodos ***update()*** (comando ultrapassado/desatualizado), ***updateOne()*** e ***updateMany()***

***update()***: Recebe o argumento de busca no documento e depois o modo de atualização. Nesse caso será o ***$set***

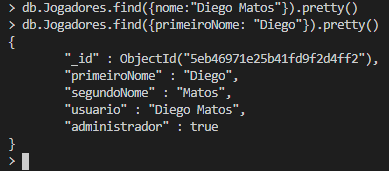




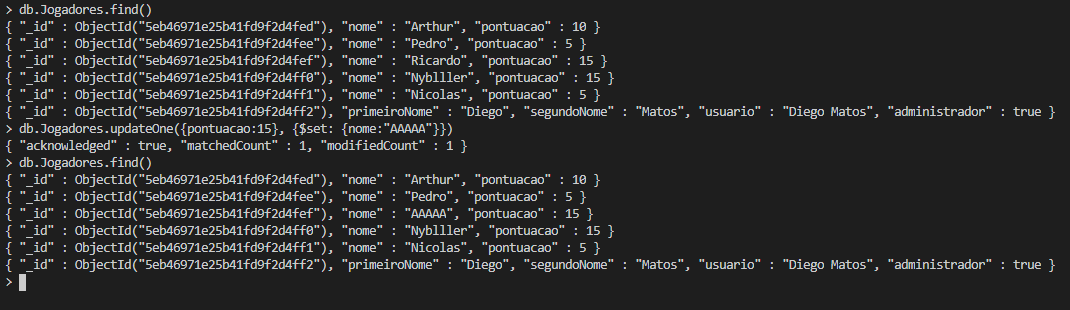
Muita atenção com os objetos !!! Observe que são dois objetos passados por parâmetro, o primeiro possui as propriedades de busca, o segundo, o método, e dentro do segundo objeto, sua propriedade também é um objeto !!!

Eu também tenho a opção de substituir o documento inteiro, eu apenas preciso passar um novo documento no segundo parâmetro, observe:



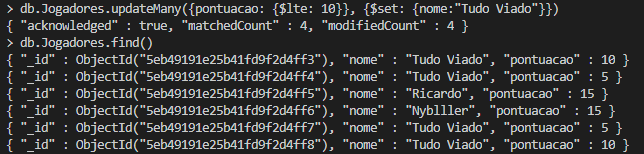


Podemos observar que agora ele não retorna nada quando eu busco por Diego Matos

 ***updateOne()***: Irá atualizar o primeiro documento encontrado.

Podemos observar que há dois jogadores com 15 pontos, como exemplo, mudei o nome do primeiro para “AAAAA”

***updateMany():*** Irá atualizar vários documentos de uma vez

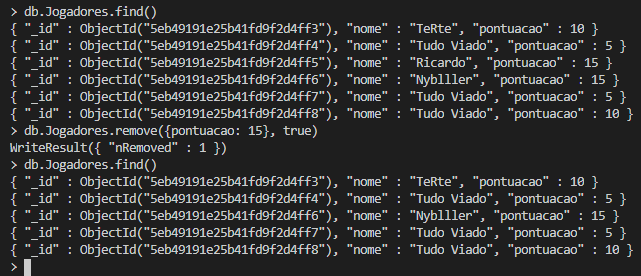


***Remover***

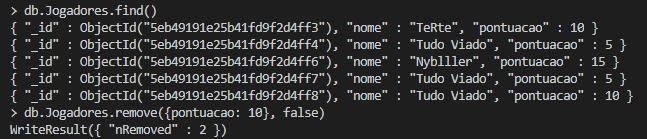
Esses comandos irão remover documentos dentro de uma coleção, são eles: ***remove() (ultrapassado)***, ***deleteOne()***, ***deleteMany()***

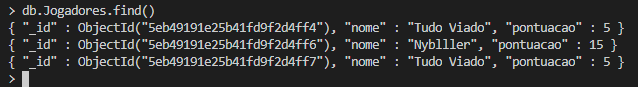
***Remove():*** O método remove recebe dois argumentos, o primeiro é a query de busca e o segundo é uma booleana, false se você deseja deletar todos ou true para apenas o primeiro.

Exemplo com true:

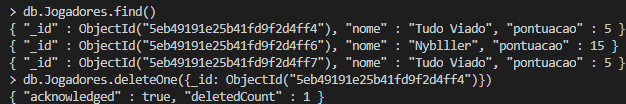


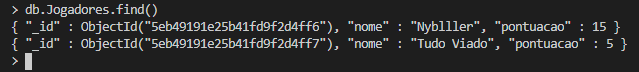
Exemplo com false, esse é o default, não é necessário passar o parâmetro:





***deleteOne()***: Deleta o primeiro documento encontrado





***deleteMany():*** Deleta todos os documentos que satisfazer a busca

