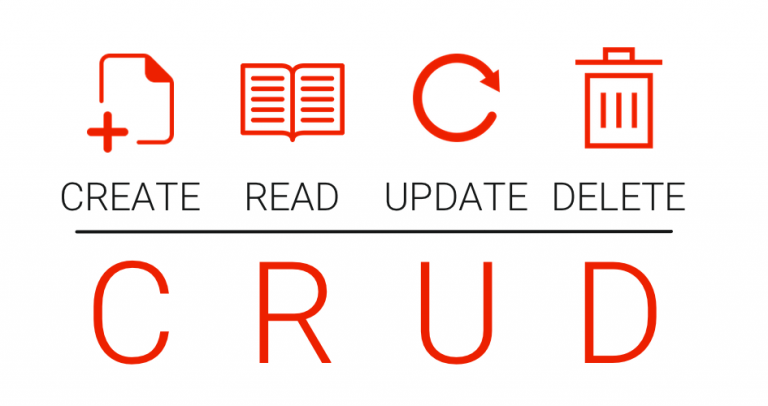
Aula 13 - CRUD no MongoDB

# CRUD

CRUD é um acrônimo que se refere às quatro operações fundamentais de um banco de dados:

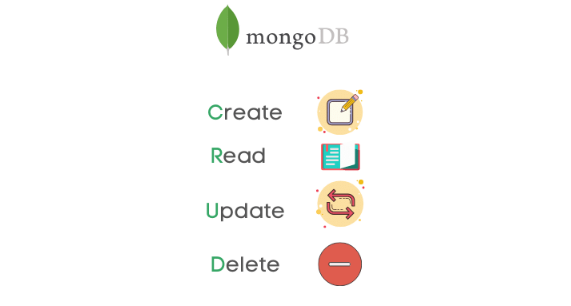


* C: Criar (Cria um dado, insere-o no banco de dados)
* R : Read (Ler um dado, mostrá-lo ao cliente)
* U : Atualizar (Atualizar um dado, alterar suas informações internas)
* D : Excluir (Excluir um dado, removê-lo de nossa coleção

comandos de suporte

Em nosso cliente CLI, temos comandos que, embora não façam parte da classificação CRUD, são úteis para poder gerenciar corretamente nosso banco de dados.

* **show dbs** : Mostra os bancos de dados existentes.
* **use <db name>** : Cria um novo banco de dados (se não existir) e passa o mouse sobre ele
* **db** : Mostra em qual banco de dados estamos posicionados.
* **show collections** : Mostra todas as coleções disponíveis no banco de dados posicionado.
* **db.createCollection(name)** : Cria uma coleção no banco de dados posicionado.
* **db.dropDatabase()** : Descarta o banco de dados atual.
* **db.collection.drop()** : Descarta a coleção do banco de dados descartado.

Primeiros comandos CRUD: CR

* **db.collection.insertOne(doc)** : Adiciona um novo documento à coleção selecionada.
* **db.collection.insertMany(docs)** : Adiciona vários documentos à coleção selecionada (dada uma matriz de documentos).
* **db.collection.findOne(opt)** : Localiza um item que atende aos critérios de pesquisa (opt), retorna o primeiro documento que atende a esses critérios.
* **db.collection.find(opt):** Retorna todos os documentos que atendem a esse critério.
* **db.collection.find(opt).pretty()** : Adicionado para tornar os resultados de um find() mais apresentáveis.

contagem de dados

Os comandos de contagem para determinar o número de documentos em uma coleção são:

* **db.collection.estimatedDocumentCount()** Conta a estimativa mais próxima do número de documentos com base em seus metadados.
* **db.collection.countDocuments(opt)** Conta os documentos que atendem aos critérios definidos nas opções (opt).



## opt (opções): adicionando opções

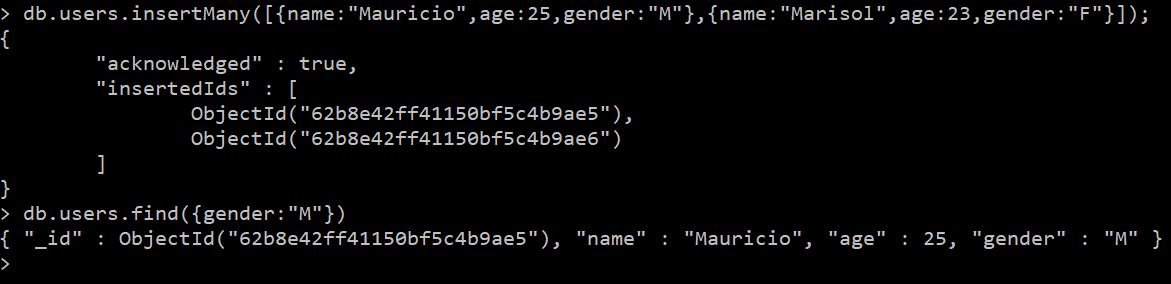
Em muitas consultas encontramos o elemento (opt), isso se refere às opções de filtro de pesquisa que podemos executar ao pesquisar um valor, a sintaxe elementar de um opt é:

***{valor da propriedade}***

******

Exemplo: Inserção de dois dados de sexo diferente

Então, procurando apenas aqueles que atendem a um gênero específico.



# Pesquisas mais complexas: filtros

Pesquisas do mundo real nem sempre exigem que um valor seja **igual** a outro. Às vezes precisamos que seja menor, maior, diferente, entre outras coisas.

Filtros podem ser adicionados dentro dos elementos critérios (opt) com a ajuda do símbolo **$** , além disso, podemos adicionar mais de um filtro para garantir que o documento se encaixe em critérios muito específicos.

Assim, a sintaxe geral será:

**db.coll.find( {key: {$operator: val}} )**

MongoDB: operadores para filtros de consulta

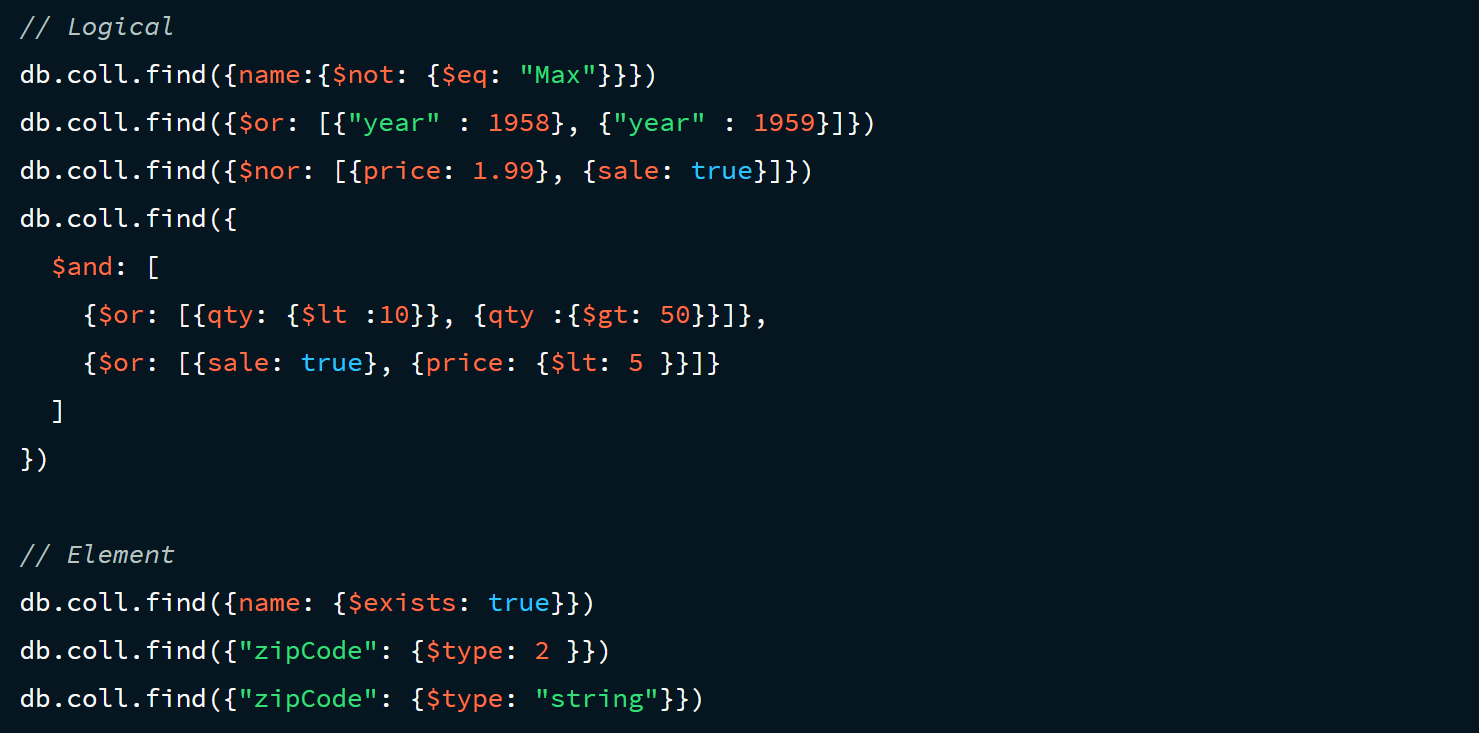
* **$and :** Execute a operação AND -> sintaxe: {$and: [ {},{} ] }
* **$or :** Execute a operação OR -> sintaxe: {$or: [ {},{} ] }
* **$lt:** Corresponde a valores menores que um valor especificado.
* **$lte :** Corresponde a valores menores ou iguais a um valor especificado.
* **$gt** : Corresponde a valores maiores que um valor especificado.
* **$gte** : Corresponde a valores maiores ou iguais a um valor especificado.
* **$ne:** Corresponde a valores que não são iguais a um valor especificado.
* **$eq** : Seleciona os documentos que são iguais a um valor especificado.
* **$exists** : Seleciona documentos com base na existência de um campo.
* **$in :** Seleciona os documentos especificados em uma matriz.

sintaxe: {chave:{$in: [matriz de valores] } }

* **$nin:** não corresponde a nenhum dos valores especificados em uma matriz.
* **$size** : Corresponde ao número de elementos especificados.
* **$all** : Corresponde a todos os valores definidos dentro de um array.
* **$elemMatch** : Corresponde a algum valor definido na consulta.

[Operadores de consulta e projeção](https://docs.mongodb.com/manual/reference/operator/query/) | mongodb

Exemplo de pesquisas CRUD com filtros combinados

****

MongoDB: Pesquisa Avançada

* **db.coll.distinct(val)**

retorna um array com os diferentes valores que um determinado campo assume nos documentos da coleção.

* **db.coll.find({doc.subdoc:value})**

É usado para filtrar subdocumentos.

* **db.coll.find({nome: /^Max$/i})**

filtrar usando expressões regulares

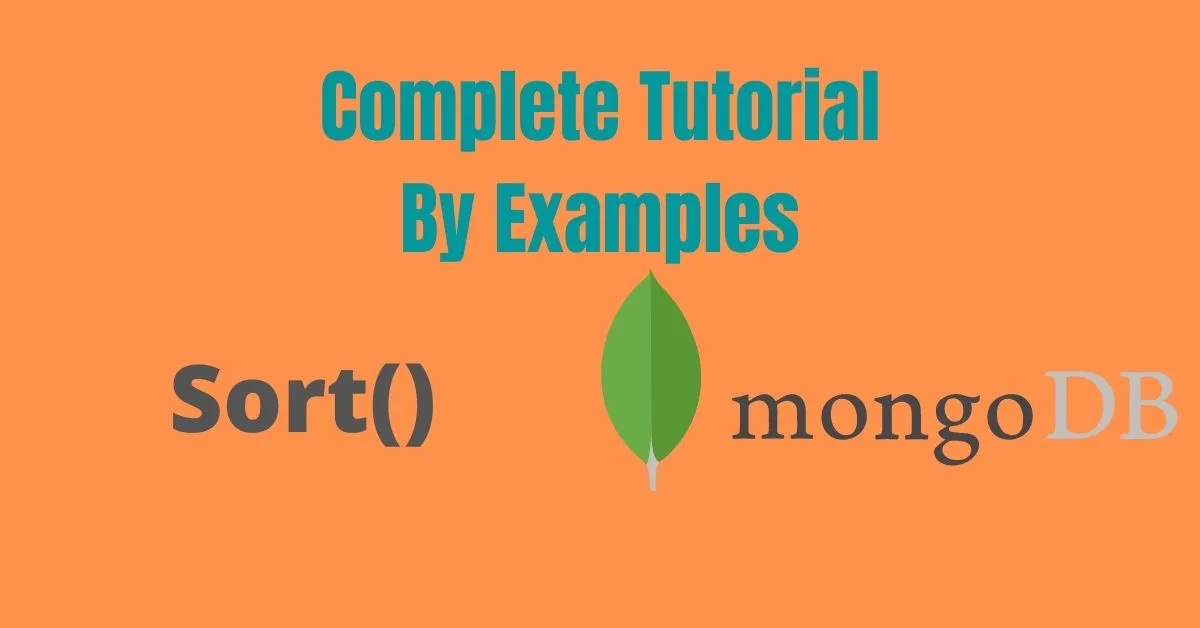
## Projeções, classificações, saltos e limites

projeções

Às vezes, não precisamos de todas as informações em um documento. Se tivermos um documento com 100 propriedades, podemos definir apenas as propriedades que queremos obter.

Uma projeção é incluída ao fazer uma pesquisa (sempre como segundo argumento) e equivale a dizer ao banco de dados: “ **eu só preciso disso** ”

Então, poderíamos dizer **db.users.find({},{name:1});** O que indica que o campo "nome" é o único que precisamos obter do documento, economizando espaço e complexidade no resultado.

Organizar

É usado para organizar as informações. A classificação é definida como **1 ou -1** para classificar em ordem crescente ou decrescente, respectivamente.

A sintaxe é: **db.collection.find().sort({val\_A:1,val\_B:-1})**

A razão pela qual podemos adicionar vários valores de ordenação é que, no caso de dois documentos terem o mesmo valor, podemos ordená-los sob outro critério

Ignorar e Limitar

**Skip** : Ignora o número de documentos indicados: Podemos usá-lo quando paginarmos, quando precisarmos ignorar um valor que sabemos ser desnecessário, etc.

Sua sintaxe é: **.skip(offset)**

**Limite** : Limita o número de documentos retornados. Assim podemos fazer diferentes níveis de paginação (Sua página pode retornar 5 elementos por página, ou 100, você decide).

Sua sintaxe é: **.limit(num)**

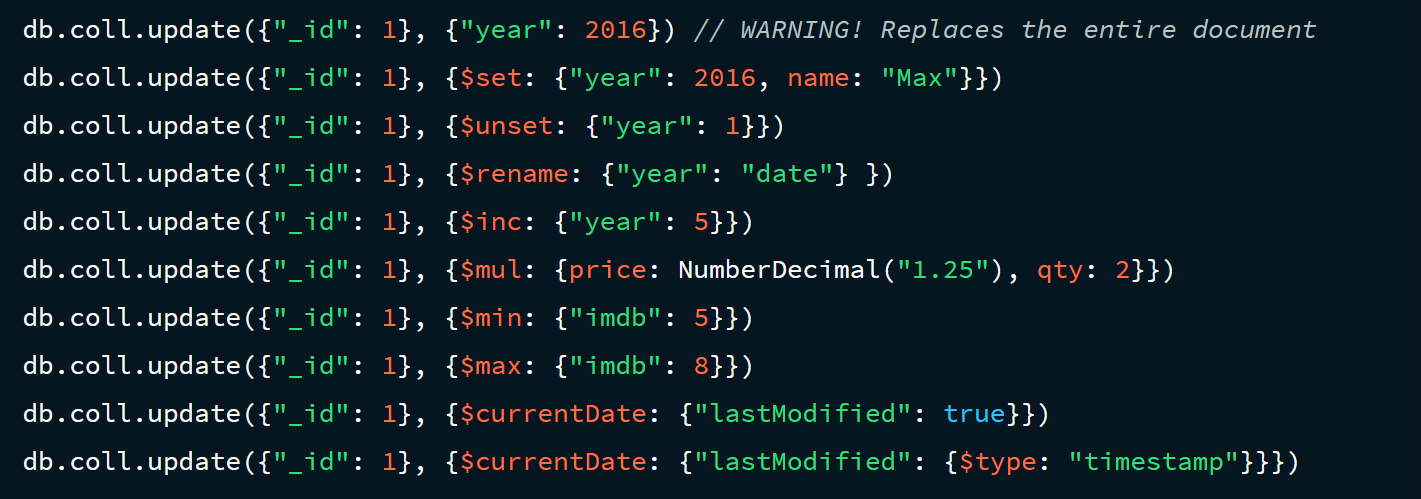
# CRUD - UD: atualizar e excluir

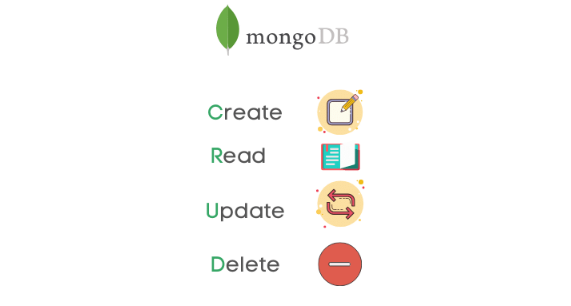
CRUD: U (atualização)

As operações de atualização podem ser executadas de duas maneiras: atualizar um documento ou atualizar vários documentos.

* **db.collection.updateOne(consulta, atualização, opção)**
  + query: usado para filtrar quais elementos atualizar (use os mesmos filtros que find)
  + update: Seção para indicar o que atualizar dos documentos que cumprem com o filtro. Update tem seus próprios operadores como $set, $unset, $inc, $rename, $mul, $min, $max
  + opção: Opções a ter em conta para a atualização (como upsert, que insere o valor caso o documento a atualizar nem exista).
* **db.collection.updateMany(query,update,options)** Atualiza vários documentos que atendem aos critérios.

Lista de operadores para atualização

Notamos como, antes de definir qual operação realizar, colocamos um filtro para saber quais documentos me interessam atualizar.

CRUD : D (Excluir)

Nossa última operação é excluir dados, embora existam muitas variantes de exclusão, veremos apenas as duas principais.

* **db.collection.deleteOne({key:val})** : Exclui apenas o primeiro elemento que atende aos critérios, usados principalmente para encontrar IDs específicos. Recomenda-se não utilizá-lo se soubermos que o valor a ser buscado não se repete.
* **db.collection.deleteMany({key:val}) :** Exclui todos os documentos que atendem aos critérios, é usado quando sabemos que mais de um valor terá aquele valor e precisamos fazer uma limpeza geral.