

MOSTRA BANCO DE DADOS EXISTENTE NA CONEXÃO ATIVA
show dbs

APONTA PARA BANCO DE DADOS
use decio

MOSTRA COLEÇÕES EXISTENTES DENTRO DO BANCO DE DADOS APONTADO
show collections

CRIA COLEÇÃO DENTRO DO BANCO DE DADOS APONTADO
db.createCollection("teste")

INSERIR UM DOCUMENTO NA COLEÇÃO DO BANCO DE DADOS
db.teste.insertOne({
 title: "Post Title 1",
 body: "Body of post.",
 category: "News",
 likes: 1,
 tags: ["news", "events"],
 date: Date()
})

Estrutura base: db.COLEÇÃO.updateOne(DOCUMENTO)

OBS: .insertOne() insere apenas um documento na coleção.

INSERIR MÚLTIPLOS DOCUMENTOS NA COLEÇÃO DO BANCO DE DADOS
db.teste.insertMany([
 {
 title: "Post Title 2",
 body: "Body of post.",
 category: "Event",
 likes: 2,
 tags: ["news", "events"],
 date: Date()
 },
 {
 title: "Post Title 3",
 body: "Body of post.",
 category: "Technology",
 likes: 3,
 tags: ["novo", "velho"],
 date: Date()
 },
 {
 title: "Post Title 4",
 body: "Body of post.",
 category: "Event",
 likes: 4,
 tags: ["novo", "velho"],
 date: Date()
 }
])

OBS: .insertMany() insere múltiplos documentos na coleção.

INSERIR UM DOCUMENTO NA COLEÇÃO DO BANCO DE DADOS

```
db.teste.updateOne(  
  { title: "Post Title 5" },  
  {  
    $set:  
    {  
      title: "Post Title 5",  
      body: "Body of post.",  
      category: "Event",  
      likes: 5,  
      tags: ["news", "events"],  
      date: Date()  
    }  
  },  
  { upsert: true }  
)
```

“\$set”: Altera o valor de um campo

“{ upsert: true }”: Se nenhum documento na coleção corresponder ao filtro, o MongoDB inserirá um novo documento com os campos e valores definidos na operação.

Estrutura base: db.COLEÇÃO.updateOne(FILTRO, {\$set: DOCUMENTO}, OPCIONAL)

OBS: .updateOne() atualiza apenas o primeiro documento que corresponde ao filtro.

INSERIR MÚLTIPLOS DOCUMENTOS NA COLEÇÃO DO BANCO DE DADOS

```
db.teste.insertMany([  
  {  
    title: "Post Title 6",  
    body: "Body of post.",  
    date: Date()  
  },  
  {  
    title: "Post Title 7",  
    body: "Body of post.",  
    category: "Technology",  
    likes: 7,  
    tags: [],  
    date: Date()  
  },  
  {  
    title: "Post Title 8",  
    body: "Body of post.",  
    category: "Event",  
    likes: 8,  
    tags: ["news", "events"],  
    date: Date()  
  }  
)
```

EXIBIR TODOS OS DOCUMENTOS

```
db.teste.find()
```

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.find( {tags: null} )
```

EXIBIR CONTAGEM DE DOCUMENTOS COM FILTRO

```
db.teste.count( {tags: null} )
```

EXIBIR CONTAGEM DE DOCUMENTOS COM FILTRO

```
db.teste.find( { category: "Event", "likes": { $lt: 5 } } ).count()
```

\$lt: Menor que (<)

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.find( { category: { $in: ["Event", "News"] } } )
```

\$in: Valor está contido em um array (ou sequencia)

EXIBIR DOCUMENTOS COM FILTRO CONDIÇÃO “AND”

```
db.teste.find( {category: "Event", "likes": { $lte: 5 } } )
```

\$lte: Menor ou igual a (<=)

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.find( { "likes": { $lt: 4 } } )
```

\$lt: Menor que (<)

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.find( { "likes": { $gt: 4 } } )
```

\$gt: Maior que (>)

EXIBIR DOCUMENTOS COM FILTRO CONDIÇÃO “OR”

```
db.teste.find( { $or: [ { "likes": { $lt: 4 } }, { "likes": { $gt: 4 } } ] } )
```

\$or: Retorna documentos que atendem pelo menos um critério

\$lt: Menor que (<)

\$gt: Maior que (>)

EXIBIR DOCUMENTOS COM FILTRO CONDIÇÃO “OR”

```
db.teste.find( { $or: [ { "likes": { $lt: 3 } }, { "category": null } ] } )
```

\$or: Retorna documentos que atendem pelo menos um critério

\$lt: Menor que (<)

EXIBIR DOCUMENTOS COM FILTRO CONDIÇÃO “AND” e “OR”

```
db.teste.find( { category: "Event" , $or: [ { "likes": { $lt: 3 } }, { "category": null } ] } )
```

\$or: Retorna documentos que atendem pelo menos um critério

\$lt: Menor que (<)

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.find( {category: "Event"} )
```

EXIBIR DOCUMENTOS COM FILTRO COM SELEÇÃO DE CAMPO (CHAVE)

```
db.teste.find( {category: "Event"}, {category: true} )
```

retorna “_id” e campo “category”

{category: true} ou {category: 1}

```
# EXIBIR DOCUMENTOS COM FILTRO COM SELEÇÃO DE CAMPO (CHAVE)
db.teste.find( {category: "Event"} , {category: false} )
# retorna “_id”, exceto o campo “category”
# {category: false} ou {category: 0}
```

```
# INSERIR MULTIPLOS DOCUMENTOS NA COLEÇÃO DO BANCO DE DADOS
```

```
db.teste.insertMany([
  {
    title: "Post Title 2",
    body: "Body of post.",
    category: "Event",
    likes: 2,
    tags: ["news", "events"],
    date: Date()
  },
  {
    title: "Post Title 3",
    body: "Body of post.",
    category: "Technology",
    likes: 3,
    tags: ["novo", "velho"],
    date: Date()
  },
  {
    title: "Post Title 4",
    body: "Body of post.",
    category: "Event",
    likes: 4,
    tags: ["novo", "velho"],
    date: Date()
  },
  {
    title: "Post Title 7",
    body: "Body of post.",
    category: "Technology",
    likes: 7,
    tags: [],
    date: Date()
  },
  {
    title: "Post Title 8",
    body: "Body of post.",
    category: "Event",
    likes: 8,
    tags: ["news", "events"],
    date: Date()
  }
])
```

EXIBIR DOCUMENTOS COM FILTRO CONDIÇÃO “AND”

```
db.teste.find( {category: "Event", likes: 2} )
```

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.findOne( {category: "Event"} )
```

OBS: .findOne() retorna apenas o primeiro documento que corresponde à consulta

EXIBIR DOCUMENTOS COM FILTRO LIMITADO PARA 2 DOCUMENTOS

```
db.teste.find().limit(2)
```

OBS: db.COLEÇÃO.find().limit(QUANTIDADE)

EXIBIR DOCUMENTOS COM FILTRO ORDEM CRESCENTE

```
db.teste.find().sort( {likes: 1} )
```

```
db.teste.find().sort( {title: 1} )
```

EXIBIR DOCUMENTOS COM FILTRO ORDEM DECRESCENTE

```
db.teste.find().sort( {likes: -1} )
```

```
db.teste.find().sort( {title: -1} )
```

EXIBIR DOCUMENTOS COM FILTRO

```
db.teste.find( {title: "Post Title 6"} )
```

ATUALIZAR DOCUMENTOS COM FILTRO

```
db.teste.updateOne( {title: "Post Title 6"}, { $set: {category: "Nova"} } )
```

\$set: Altera o valor de um campo

OBS: .updateOne() atualiza apenas o primeiro documento que corresponde ao filtro

ATUALIZAR DOCUMENTOS COM FILTRO

```
db.teste.updateOne( {title: "Post Title 6"}, { $set: {"tags": null} } )
```

\$set: Altera o valor de um campo

ATUALIZAR DOCUMENTOS COM FILTRO

```
db.teste.updateOne( {title: "Post Title 7"}, { $set: {"tags": ["old"]} } )
```

\$set: Altera o valor de um campo

ATUALIZAR DOCUMENTOS COM FILTRO

```
db.teste.updateMany({}, { $set: {tags: ["novo", "venho"]} } )
```

\$set: Altera o valor de um campo

OBS: .updateMany() atualiza todos os documentos na coleção.

DELETA DOCUMENTOS COM FILTRO

```
db.teste.remove( {category: "Technology"}, {justOne: true} )
```

{justOne: true}: Elemento opcional que limita a exclusão a apenas um documento, caso omitido será deletado múltiplos documentos.

DELETA DOCUMENTOS COM FILTRO

```
db.teste.deleteOne( { title: "Post Title 1" } )
```

OBS: .deleteOne() deleta apenas o primeiro documento que corresponde ao filtro.

DELETA DOCUMENTOS COM FILTRO

```
db.teste.deleteMany( { category: "Event" } )
```

OBS: .deleteMany() deleta todos os documentos na coleção.

APAGA TODA A COLEÇÃO
db.teste.drop()

===== Lista de Operadores MongoDB =====

Operadores de Consulta:

- \$eq: Igualdade (=)
- \$ne: Diferença (<> ou !=)
- \$gt: Maior que (>)
- \$lt: Menor que (<)
- \$gte: Maior ou igual a (>=)
- \$lte: Menor ou igual a (<=)
- \$in: Valor está contido em um array (ou sequencia)
- \$nin: Valor não está contido em um array
- \$all: Todos os valores de um array estão presentes em outro array
- \$elemMatch: Um array contém um documento que atende a todos os critérios

Operadores de Projeção:

- \$include: Inclui campos na projeção
- \$exclude: Exclui campos da projeção
- \$slice: Limita o número de documentos retornados
- \$sort: Ordena os documentos retornados
- \$limit: Limita o número de documentos retornados

Operadores de Agregação:

- \$group: Agrupa documentos por um campo ou campos
- \$sum: Soma valores de um campo
- \$avg: Calcula a média de valores de um campo
- \$max: Retorna o valor máximo de um campo
- \$min: Retorna o valor mínimo de um campo
- \$first: Retorna o primeiro valor de um campo
- \$last: Retorna o último valor de um campo

Operadores Lógicos:

- \$and: Combina vários critérios em um único
- \$or: Retorna documentos que atendem a pelo menos um critério
- \$not: Inverte o resultado de um critério

Operadores de Expressão:

- \$abs: Retorna o valor absoluto de um número
- \$add: Soma números ou datas
- \$ceil: Retorna o menor número inteiro maior ou igual ao especificado
- \$divide: Retorna o resultado da divisão do primeiro número pelo segundo
- \$exp: Aumenta e para o expoente especificado

Operadores de Tipo:

- \$type: Retorna o tipo de um valor

Operadores de Modificação:

- \$set: Altera o valor de um campo
- \$unset: Remove um campo
- \$inc: Incrementa o valor de um campo
- \$mul: Multiplica o valor de um campo

Operadores de Texto:

- \$concat: Concatena strings
- \$substr: Retorna uma substring
- \$toLowerCase: Converte uma string para minúsculas
- \$toUpperCase: Converte uma string para maiúsculas

Operadores de Data:

- \$date: Converte um valor para uma data
- \$year: Retorna o ano de uma data
- \$month: Retorna o mês de uma data
- \$day: Retorna o dia de uma data

Operadores de Array:

- \$push: Adiciona um elemento a um array
- \$pop: Remove um elemento de um array
- \$size: Retorna o tamanho de um array

Operadores de Geospatial:

- \$near: Retorna documentos próximos a um ponto
- \$within: Retorna documentos dentro de uma área

Operadores de Atualização:

- \$set: Altera o valor de um campo
- \$unset: Remove um campo
- \$inc: Incrementa o valor de um campo
- \$mul: Multiplica o valor de um campo

Operadores de Upsert:

- \$setOnInsert: Define um valor para um campo se o documento não existir

Operadores de Merge:

- \$merge: Combina documentos de várias coleções

Operadores de Indexação:

- \$createIndex: Cria um índice
- \$dropIndex: Remove um índice

Operadores de Texto Completo:

- \$text: Realiza uma pesquisa de texto completo

Operadores de Validação:

- \$validator: Valida um documento