

Revisão semana 7


Índice

- 01 Compass
- 02 Queries Complexas
- 03 Índices



01

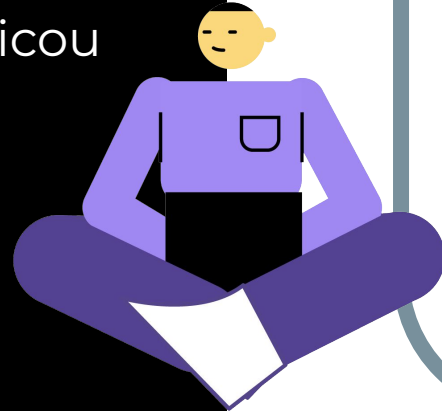
Compass



Que tal revermos quanta coisa importante
aprendemos esta semana?

Aproveite para detectar os pontos que ficou
com dúvida.

Bons estudos!



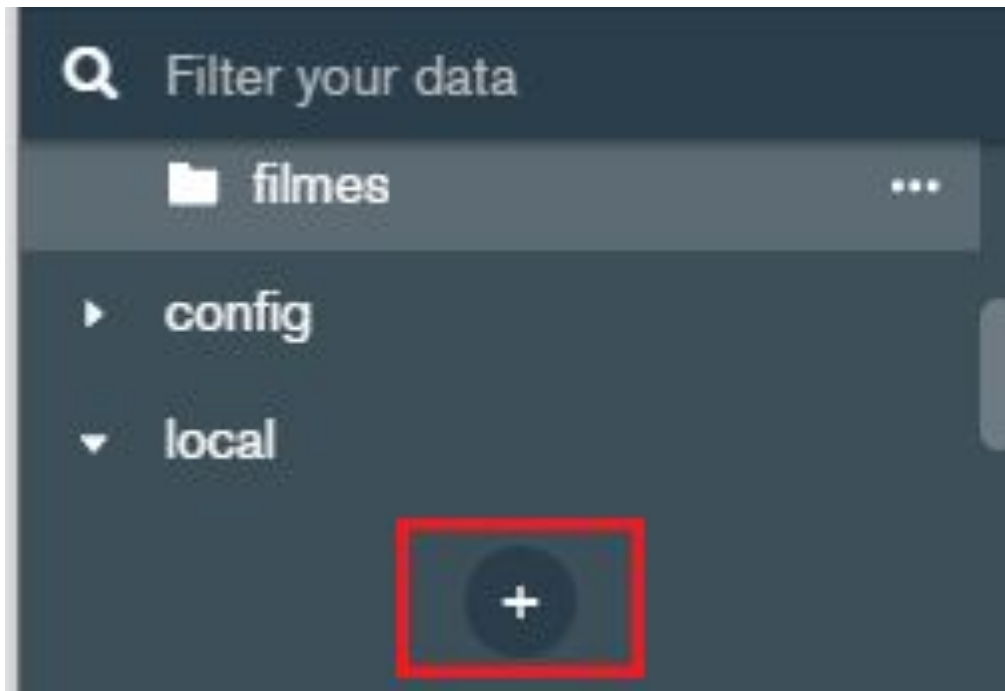
Comandos CRUD

- **db.collection.updateMany()** - Atualiza vários documentos.
- **db.collection.deleteOne()** - Exclui um único documento.
- **db.collection.deleteMany** - Exclui vários documentos.
- **db.collection.find()** - Recupera os documentos de uma coleção.
- **db.collection.insertOne()** - Para inserir um único documento.
- **db.collection.insertMany()** - Para inserir vários documentos.
- **db.collection.find()** - Semelhante ao Select. Recupera documentos de uma coleção.
- **db.collection.updateOne()** - Atualiza um único documento.

Compass

O Compass é a GUI para MongoDB. Intuitivo e flexível, o Compass foi criado para explorar e manipular facilmente seu banco de dados. Fornece visualizações detalhadas de esquemas, métricas de desempenho em tempo real, recursos sofisticados de consulta e muito mais.

Clique no sinal de + para criar um banco de dados novo.



Compass

Ao criar o banco de dados, você também pode criar a coleção.

Create Database

Database Name

Collection Name



Você também pode selecionar um banco de dados e clicar no botão Create collection para criar uma coleção.

The screenshot displays the MongoDB Compass interface. On the left, the 'Local' database is selected, showing a list of collections: 'admin', 'catalogo', 'atorFilme', 'filmes', 'config', 'local', and 'startup_log'. The 'catalogo' collection is expanded, showing its sub-collections: 'atorFilme' and 'filmes'. On the right, the 'Collections' tab is active, showing a 'Create collection' button and a 'View' button. Below these, two collection cards are displayed: 'atorFilme' and 'filmes'. Each card shows the storage size (20.48 kB) and the number of documents (43 for 'atorFilme' and 19 for 'filmes').

Collection Name	Storage size	Documents
atorFilme	20.48 kB	43
filmes	20.48 kB	19

Selecione uma coleção para visualizar ou inserir documentos.

Para inserir documentos, ao clicar no botão ADD DATA, podemos escolher entre importar um arquivo csv ou Json ou Inserir documento no modo Json ou editor campo-a-campo.

The screenshot displays the MongoDB Atlas web interface. On the left, a dark sidebar shows the 'Local' database structure with '4 DBS' and '4 COLLECTIONS'. The 'catalogo' database is expanded, showing collections like 'admin', 'catalogo', 'atorFilme' (selected), 'filmes', 'config', 'local', and 'startup_log'. The main panel shows the 'catalogo.atorFilme' collection with the 'Documents' tab active. A filter bar is present with the text '{ field: 'value' }'. Below the filter, there is a green 'ADD DATA' button and a 'VIEW' button. The document list shows two entries:

```
> {
  "_id": 1,
  "nome": "Sam",
  "sobrenome": "Worthington",
  "filme": "Avatar"
}
```

```
{
  "_id": 2,
  "nome": "Zoe",
  "sobrenome": "Saldana",
  "filme": "Avatar"
}
```

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

▼

📁

+

▼

► COLLATION

Untitled

SAVE

▼

📄

☒ SAMPLE MODE

☒ AUTO PREVIEW



▼

📄 6 Documents in the Collection



Preview of Documents in the Collection

A guia Aggregations permite criar pipelines, ou seja, estágios para nossa consulta.

Esses pipelines podem ser exportados para JAVA, C#, Node.js ou Python, para inserirmos no nosso projeto, reduzindo o tempo de desenvolvimento.

```
_id: 1
nome: "Ana Luísa"
idade: 22
cidade: "Campo Grande"
estado: "MS"
telefone: "(67) 3032-2233"
```

|||

▼

Select...

☒

🗑️

+

A sample of the aggregated results from this stage will be shown below

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation



COLLATION

Untitled

SAVE



SAMPLE MODE



AUTO PREVIEW



6 Documents in the Collection



Preview of Documents in the Collection

Podemos ainda salvar o pipeline criado para utilização posterior ou criar uma view, que poderemos consultar sempre que preciso.

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

```
_id: 1  
nome: "Ana Luísa"  
idade: 22  
cidade: "Campo Grande"  
estado: "MS"  
telefone: "(67) 3032-2233"
```



Select...



A sample of the aggregated results from this stage will be shown below

Guias do Compass

- **Schema** - Fornece uma visão geral do tipo de dados e da forma dos campos em uma coleção específica.
- **Validation** - Sua função é definir regras de validação para o seu schema.
- **Explain Plain (Planos de Execução)** - Exibe o comportamento de uma consulta. Esses dados são importantes para identificarmos problemas de performance ou utilização de índice.
- **Indexes** - Exibe os índices que foram criados em uma coleção e quais foram usados. Essa é uma informação importante, pois pode sugerir alterações na sua query ou no índice:

02

Queries Complexas

Operadores e funções do MongoDB para criar consultas complexas

\$exists - Verifica se na coleção existe ou não determinado campo.

Sintaxe:

```
db.collection.find( { campo: { $exists: true, $nin: [ "parametro1", "parametro2" ] } } )
```

Like - Utilizamos // - semelhante ao curinga % do MySQL.

Sintaxe: `db.collection.find({campo: /^a/})`

\$group - Semelhante ao Group By no MySQL. Agrupa dados correspondentes a um parâmetro.

Sintaxe: `db.collection.aggregate([{$group: {_id: "$campo", Total: {$sum: 1}}}])`

\$in - Semelhante ao MySQL. Lista os documentos que correspondem a algum parâmetro definido no \$in.

Sintaxe: `db.collection.find({ campo: { $in: ["parametro1", "parametro2"] } })`

Operadores e funções do MongoDB para criar consultas complexas

\$nin - Semelhante ao not in do MySQL.

Lista os documentos que não correspondem a algum parâmetro definido no \$nin.

Sintaxe: `db.collection.find({ campo: { $nin: ["parametro1", "parametro2"] } })`

\$lookup - Função equivalente ao left join.

\$push - Agrega uma matriz do atributo selecionado.

Sintaxe:


`db.collection.aggregate([{ $group: { _id: "$campo", alias: { $sum: 1 }, alias1: { $push: "$campo" } } }])`

O MongoDB oferece muitos outros operadores e funções para facilitar a manipulação de dados.

\$unwind - transforma o array em objeto, facilitando a recuperação dos dados.

03

Índices




Os índices no MongoDB funcionam de maneira similar com os índices dos bancos de dados relacionais.

São gerados na forma de **B -Tree**, ou seja, os dados são armazenados em forma de árvore, mas mantendo os nós balanceados.

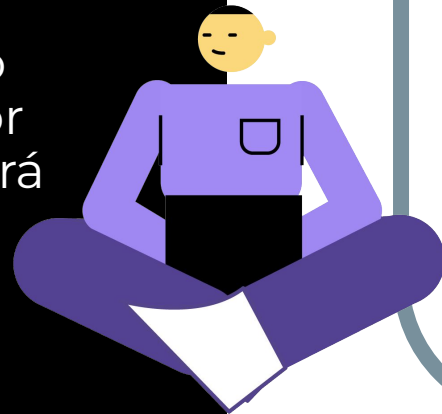
Isso aumenta a velocidade na busca e também no retorno de resultados já ordenados.





Para melhorar a eficiência dos índices, recomenda-se que eles tenham uma alta **cardinalidade**.

Cardinalidade é a quantidade de elementos que não se repetem. Quanto mais valores únicos o campo tiver, maior será a cardinalidade. E mais eficiente será o índice.



Tipos de Índices

- **Índices simples ou de campo único** - Esses índices se aplicam a um único campo em uma coleção.

Sintaxe:

```
db.collection.ensureIndex({ "_id":  
1})
```

- **Índices compostos** - Neste caso o índice será gerado em vários campos e podemos usá-los para consultar um ou mais campos, sem ter que incluir todos eles.

Sintaxe:

```
db.collection.ensureIndex( { campo1: 1, campo2:-1 }  
)eles.
```

Tipos de índices

- **Índices únicos** - São criados para valores únicos. Adicionamos o parâmetro exclusivo ao criá-los.

Sintaxe:

```
db.collection.ensureIndex( { _id : 1  
}, {"unique":true} )
```

- **Índices esparsos** - Incluem todos os documentos. Não há necessidade de que o campo exista em todos os documentos.

Sintaxe: `db.collection.ensureIndex({ "campo" : 1 }, {"sparse":true})`

Tipos de índices

- **Índices de Texto** - Os índices de texto devem ser uma string ou uma array de elementos de string.

Sintaxe:

```
db.collection.createIndex({campo:  
"text"})
```

- **Índice curinga** - Usando o especificador curinga (\$**), você tem permissão para criar vários campos de índices de texto. O MongoDB indexa todo e qualquer campo que contenha dados de string em todos os documentos presentes na coleção fornecida.

Sintaxe:

```
db.collection.createIndex( { "$**": "text" } )
```

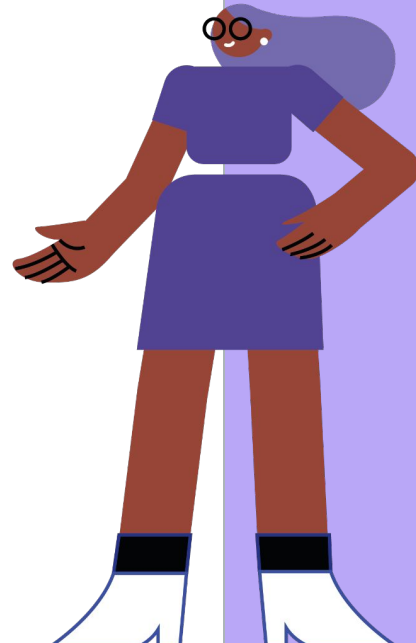
Conclusões

Viu quanta coisa descobrimos nesta semana?

Agora temos muito o que estudar e praticar.

Não se esqueça de anotar todas as dúvidas e discutir com colegas e professores, afinal, a troca de informações e experiências também é conhecimento.

Bons estudos!



Muito obrigado!