

Tema 04

Introdução ao MongoDB

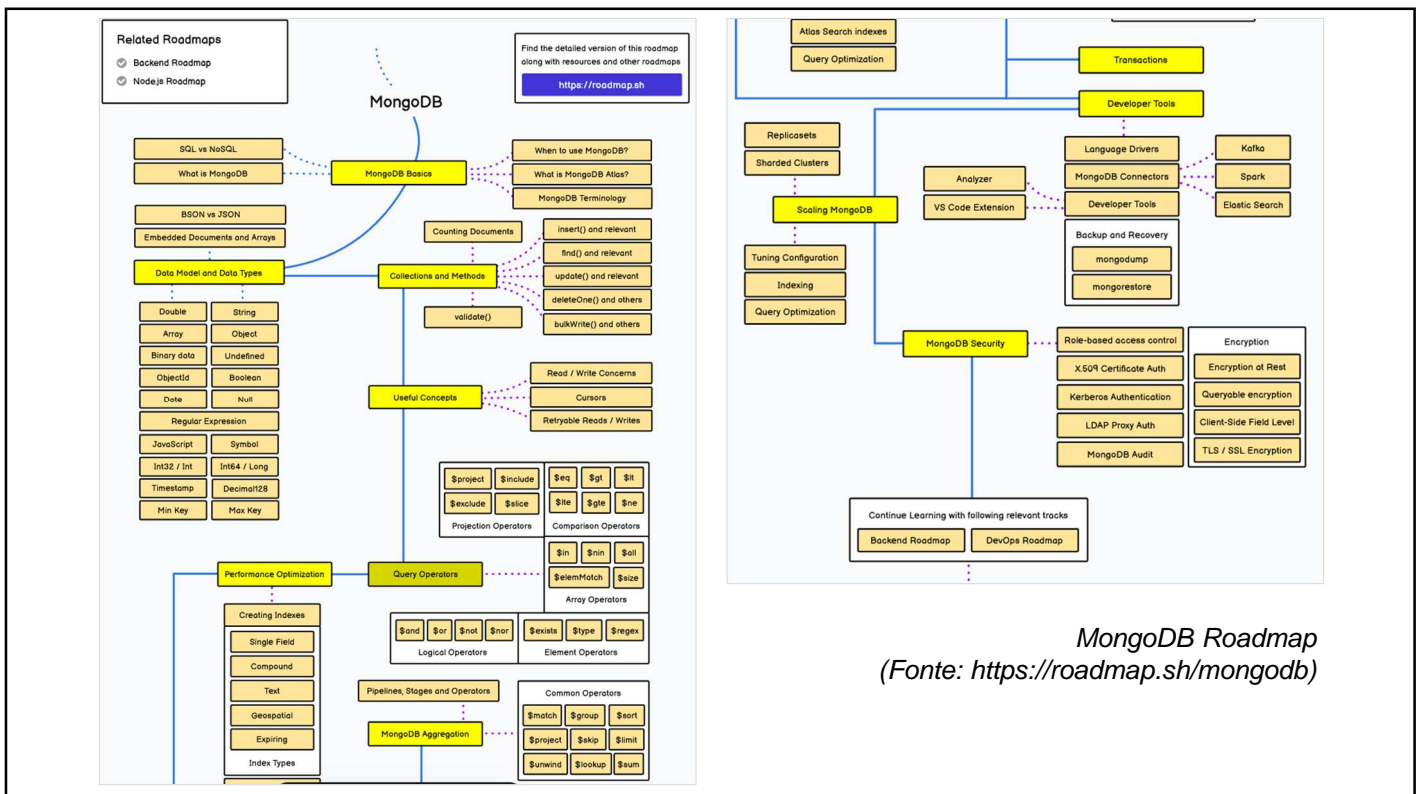
Banco de Dados NoSQL



MongoDB

- **MongoDB** é um programa de banco de dados de base *source-available*, multiplataforma e orientado a documentos
- Classificado como NoSQL, utiliza documentos JSON-like com esquemas opcionais
- Documentação em:
 - <https://www.mongodb.com/pt-br/docs/>





Design de esquema MongoDB

- No MongoDB, um **esquema** define a estrutura real do banco de dados
 - Organização de tabelas (ou coleções), tipos de dados de campos, restrições, índices e outras regras que governam o armazenamento e a recuperação de dados

Organização de Coleções

- No MongoDB, dados são armazenados em **coleções**
 - Análogas às tabelas em bancos de dados relacionais
 - Cada coleção contém documentos que representam registros individuais
- Ex.: Coleção “estudantes” que armazena informações de alunos

```
// Exemplo de coleção de alunos
{
  "_id" : ObjectId (
    "609fcb7f5824d05b7d9b827c" ),
  "nome" : "John Doe" ,
  "idade" : 25 ,
  "serie" : "A"
}
```

Tipos de Dados e Campos

- MongoDB suporta vários **tipos de dados** para **campos** dentro de documentos
 - Campos podem armazenar strings, números, datas, arrays e outros tipos de dados

```
// Ex. de diferentes tipos de campos
{
  "_id" : ObjectId ( "609fcb7f5824d05b7d9b827d" ),
  "nome" : "Alice" ,
  "idade" : 30 ,
  "aniversario" : ISODate ( "1994-03-15" ),
  "disciplinas" : [ "Math" , "Science" ],
  "notas" :
  {
    "matematica" : "A" ,
    "ciencias" : "B"
  }
}
```

- MongoDB usa **BSON (Binary JSON)** para armazenar documentos oferecendo suporte para uma gama diversificada de **tipos de dados**

- **String** – Tipo de dado mais comumente usado para armazenar os dados. No MongoDB deve ser UTF-8 válido
- **Integer** – Armazena um valor numérico. Integer pode ser de 32 bits ou 64 bits, dependendo do seu servidor
- **Boolean** – Armazena um valor booleano (verdadeiro/falso)
- **Double** – Armazena valores de ponto flutuante
- **Chaves Min/Max** – Compara um valor com os elementos BSON mais baixos e mais altos
- **Matrizes** – Armazena matrizes, listas ou múltiplos valores em uma chave
- **Timestamp** – Pode ser útil para registrar quando um documento foi modificado ou adicionado
- **Objeto** – Usado para documentos incorporados
- **Nulo** – Armazena um valor Nulo
- **Símbolo** – Usado de forma idêntica a uma string; no entanto, geralmente é reservado para linguagens que usam um tipo de símbolo específico
- **Date** – Armazena a data ou hora atual no formato de hora UNIX. Você pode especificar sua própria data e hora criando o objeto Date e passando dia, mês e ano para ele
- **ID do objeto** – Armazena o ID do documento
- **Dados binários** - Armazena dados binários
- **Código** – Armazena código JavaScript no documento
- **Expressão regular** - Armazena expressões regulares



Restrições e Índices

- O MongoDB permite a definição de restrições e índices para impor a integridade dos dados e melhorar o desempenho da consulta

– Ex.: Criação de um índice no campo "nome" da coleção "estudantes"

```
// Exemplo de criação de índice  
db.students.createIndex ({ "name" : 1 })
```



Regras para Armazenamento e Recuperação

- O esquema especifica regras para armazenar e recuperar dados de forma eficiente
 - Inclui diretrizes para tamanho de documento, níveis de aninhamento e estratégias de otimização de consulta

– Ex.: Desnormalização e incorporação são técnicas comuns usadas para otimizar a recuperação de dados

// Exemplo de dados incorporados

```
{
  "_id" : ObjectId ( "609fcb7f5824d05b7d9b827e" ),
  "nome" : "Bob" ,
  "idade" : 28 ,
  "endereco" : {
    "cidade" : "Nova York" ,
    "cep" : "10001"
  }
}
```



Modelagem de dados no MongoDB

- A **modelagem de dados do MongoDB** organiza dados não estruturados de um evento do mundo real em um modelo de dados lógico em um banco de dados
 - Não há necessidade de construir um esquema antes de adicionar dados ao banco de dados MongoDB porque o MongoDB é flexível
 - Isso significa que o MongoDB suporta um esquema de banco de dados dinâmico



Modelagem de dados no MongoDB

- Para criar um modelo de dados perfeito no MongoDB, equilibre:
 - Necessidades do aplicativo
 - Recursos de desempenho
 - Padrões de recuperação de dados
- Ao criar modelos de dados, leve em consideração:
 - Os usos dos dados pela aplicação
 - Consultas, atualizações e processamento de dados
 - O design fundamental dos dados em si

Projetos de modelo de dados do MongoDB

- Para modelar dados no MongoDB, duas estratégias estão disponíveis
 1. *Modelo de Dados Incorporados*
 2. *Modelo de Dados Normalizado*

1. Modelo de dados incorporados

- **1. Modelo de dados incorporados** (ou *modelo de dados desnormalizado*):
 - *Incorpora todos os documentos relacionados em um único documento*
 - Esses documentos aninhados também são chamados de subdocumentos
- Exemplo:
 - Se obtivermos informações do aluno em três documentos diferentes, **Detalhes_pessoais**, **Contato** e **Endereço**, podemos incorporar todos os três em um único documento



```
{
  _id: ,
  Std_ID: "987STD001"
  Detalhes pessoais: {
    Primeiro nome: "Rashmika",
    Sobrenome: "Sharma",
    Data de nascimento: "1999-08-26"
  },
  Contato: {
    e-mail: "rashmika_sharma.123@gmail.com",
    telefone: "9987645673"
  },
  Endereço: {
    cidade: "Karnataka",
    Área: "BTM2ndStage",
    Estado: "Bengaluru"
  }
}
```



2. Modelo de dados normalizado

- **2. Modelo de dados normalizado:**

- *Referências de objetos expressam os relacionamentos entre documentos e objetos de dados*
- Como essa abordagem reduz a duplicação de dados, é relativamente simples documentar relacionamentos muitos-para-muitos sem ter que repetir conteúdo
- Técnica mais eficaz para modelar grandes dados hierárquicos com relacionamentos de coleta cruzada

- **Exemplo:**

- Criação de várias coleções para armazenar dados de alunos que estão vinculados com `_id`



```
{
  _id: <IdAluno101>,
  ID_Padrao: "10025AE336"
}
```

Estudante

```
{
  _id: <IdAluno103>,
  alunoDocID: "IdAluno101",
  e-mail: "rashmika_sharma.123@gmail.com",
  telefone: "9987645673"
}
```

Detalhes pessoais

```
{
  _id: <IdAluno102>,
  alunoDocID: "IdAluno101",
  Primeiro_Nome: "Rashmika",
  Último_Nome: "Sharma",
  Data_De_Nascimento: "1999-08-26"
}
```

Contato

```
{
  id: <IdAluno104>,
  alunoDocID: "IdAluno101",
  cidade: "Karnataka",
  área: "BTM2ndStage",
  estado: "Bengaluru"
}
```

Endereço



Exercício de aula 1

- Imagine que você está desenvolvendo um sistema para uma pequena livraria online
- O sistema precisa armazenar: Informações sobre os livros, os autores e os pedidos dos clientes
- Com base nas informações fornecidas a seguir, modele a estrutura de dados usando MongoDB, escolhendo entre um modelo de dados incorporado ou normalizado
- Justifique sua escolha

- 1. Livros:
 - Cada livro tem um título, uma descrição, um preço, uma data de publicação, um número de páginas e uma lista de autores
 - Um livro pode ter mais de um autor
- 2. Autores:
 - Cada autor tem um nome, uma biografia e uma lista de livros que ele escreveu
- 3. Pedidos:
 - Cada pedido tem um cliente associado, uma data de pedido e uma lista de livros comprados
 - Um cliente pode fazer vários pedidos e comprar vários livros em um único pedido

- Requisitos:

- Estruturar os dados de maneira eficiente para consultas comuns, como:

1. Listar todos os livros de um autor específico.
2. Recuperar detalhes de um pedido específico, incluindo os livros comprados e as informações do cliente.

```
{
  "_id": ObjectId("60a1be3d9f53b5eeeb9c4e5b"),
  "title": "MongoDB: A Beginner's Guide",
  "description": "A comprehensive guide to MongoDB.",
  "price": 29.99,
  "publication_date": ISODate("2023-01-15"),
  "pages": 320,
  "authors": [
    { "name": "John Smith", "author_id": ObjectId("60a1be3d9f53b5eeeb9c4e5c") },
    { "name": "Jane Doe", "author_id": ObjectId("60a1be3d9f53b5eeeb9c4e5d") }
  ]
}
```

Livros

Referências Adicionais

- <https://afzalbadshah.medium.com/data-modeling-and-schema-design-in-mongodb-24a786b1375d>
- <https://www.geeksforgeeks.org/mongodb-data-modelling/>
- https://www.tutorialspoint.com/mongodb/mongodb_datatype.htm



Melhor Centro Universitário
privado do Rio de Janeiro

Melhor EAD do Brasil
Conceito 5 do MEC