

# Bancos não relacionais: SGBD's NoSQL Orientados a Documentos – MongoDB

QXD0099 - Desenvolvimento de Software para Persistência

**Universidade Federal do Ceará - *Campus* Quixadá**

Prof. Francisco Victor da Silva Pinheiro  
victorpinheiro@ufc.br



# Agenda

- Introdução aos Bancos Não Relacionais (NoSQL)
  - O que é NoSQL?
  - Características Principais dos Bancos NoSQL
  - Tipos de Bancos NoSQL
  - Comparação com Bancos Relacionais
  - Casos de Uso
  - Vantagens e Desvantagens
- SGBD Orientados a Documentos: MongoDB
  - Conceito de banco orientado a documentos
  - MongoDB
  - Componentes principais no MongoDB
- Comandos Básicos do MongoDB
  - Operações CRUD básicas

# Introdução aos Bancos Não Relacionais (NoSQL)

O que é NoSQL?  
Características Principais dos Bancos NoSQL  
Tipos de Bancos NoSQL  
Comparação com Bancos Relacionais  
Casos de Uso  
Vantagens e Desvantagens

# O que é NoSQL?

- O termo NoSQL refere-se a sistemas de gerenciamento de banco de dados que não utilizam o modelo relacional tradicional baseado em tabelas. Em vez disso, esses bancos são projetados para lidar com grandes volumes de dados, alta velocidade de leitura e escrita, e escalabilidade horizontal.
- O termo pode ser interpretado como "Not Only SQL", indicando que esses bancos complementam, em vez de substituir, os bancos relacionais.



# Características Principais dos Bancos NoSQL

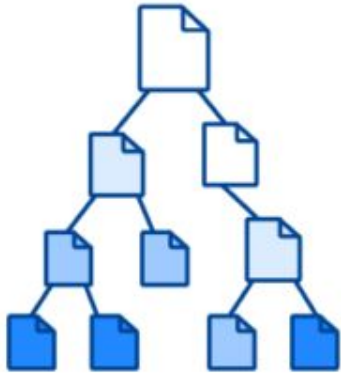
- **Esquema Flexível ou Sem Esquema**
  - Diferentemente dos bancos relacionais, os bancos NoSQL geralmente não exigem a definição de um esquema fixo. Isso permite que diferentes registros em uma mesma coleção armazenem dados com formatos variados.
  
- **Escalabilidade Horizontal**
  - Bancos NoSQL são projetados para crescer adicionando mais servidores (escalabilidade horizontal), ao contrário dos bancos relacionais que frequentemente dependem de hardware mais robusto (escalabilidade vertical).

# Características Principais dos Bancos NoSQL

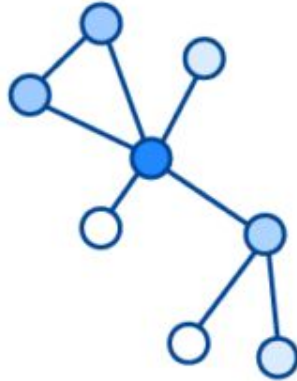
- **Alta Disponibilidade e Tolerância a Falhas**
  - Utilizam estratégias de replicação e particionamento para garantir a disponibilidade dos dados, mesmo em caso de falha de servidores.
- **Modelos de Dados Variados**
  - Os bancos NoSQL suportam diferentes modelos de dados, como documentos, chave-valor, colunares e grafos.

# Tipos de Bancos NoSQL

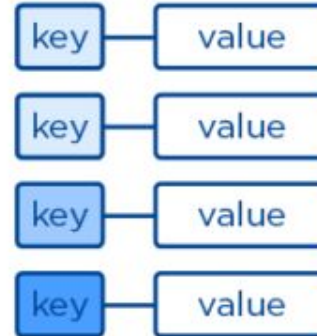
**Document**



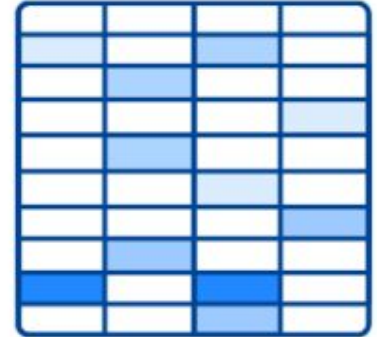
**Graph**



**Key-Value**



**Wide-column**



# Tipos de Bancos NoSQL

- **Documentos**

- Dados são armazenados em documentos, geralmente no formato JSON, BSON ou XML.
  - Exemplo: MongoDB, CouchDB.

- **Chave-Valor**

- Dados são armazenados como pares de chave e valor.
  - Exemplo: Redis, DynamoDB.





# Tipos de Bancos NoSQL

- **Coluna**

- Dados são armazenados em colunas em vez de linhas, otimizando consultas analíticas.
  - Exemplo: Apache Cassandra, HBase.

- **Grafos**

- Focado em relações e conexões entre entidades, utilizando grafos como modelo de dados.
  - Exemplo: Neo4j, Amazon Neptune.

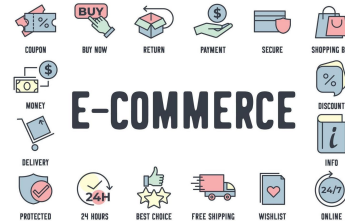


# Comparação com Bancos Relacionais

Característica	Bancos Relacionais	Bancos NoSQL
Modelo de Dados	Tabelas	Documentos, Chave-Valor, Colunas, Grafos
Esquema	Fixo	Flexível ou inexistente
Consultas	SQL	Varia conforme o banco
Escalabilidade	Vertical	Horizontal
Consistência	Forte	Eventual ou configurável
Aplicação Típica	Transações financeiras, ERP	Big Data, IoT, redes sociais

# Casos de Uso

- **Redes Sociais**
  - Modelagem flexível e consultas rápidas para dados relacionados a perfis, postagens e conexões.
- **E-commerce**
  - Escalabilidade para lidar com grandes volumes de dados, como inventário, carrinho de compras e histórico de transações.



# Casos de Uso

- **Análise de Big Data**
  - Processamento eficiente de grandes volumes de dados semi-estruturados ou não estruturados.
- **Internet das Coisas (IoT)**
  - Armazenamento e consulta de dados sensoriais com alta frequência.



# Vantagens e Desvantagens

- Vantagens
  - Alta performance para grandes volumes de dados.
  - Flexibilidade no modelo de dados.
  - Escalabilidade horizontal mais simples e econômica.
  - Adequado para aplicações modernas e distribuídas.

# Vantagens e Desvantagens

- Desvantagens
  - Falta de padronização (cada banco tem sua API e linguagem de consulta).
  - Consistência eventual pode não ser adequada para todos os casos.
  - Ferramentas e suporte menos maduros em comparação aos bancos relacionais.

# SGBD Orientados a Documentos: MongoDB

- **Conceito de Banco Orientado a Documentos**
  - Bancos de dados orientados a documentos armazenam dados em estruturas semiestruturadas, geralmente no formato JSON, BSON (uma versão binária de JSON) ou XML. Cada registro é armazenado como um documento, que pode conter campos e valores, aninhados ou não.
  - Diferentemente dos bancos relacionais, onde dados são organizados em tabelas e colunas, os documentos podem ter esquemas flexíveis, permitindo que diferentes documentos em uma mesma coleção tenham estruturas variadas.

# SGBD Orientados a Documentos: MongoDB

- **Principais Características**

- Flexibilidade: Não é necessário um esquema fixo; novos campos podem ser adicionados sem necessidade de alterar documentos existentes.
- Estruturas Hierárquicas: Documentos podem conter subdocumentos e arrays.
- Consulta Rápida: Suporte a consultas complexas usando estruturas internas de documentos.



# MongoDB

- O MongoDB é um dos sistemas de gerenciamento de banco de dados orientados a documentos mais populares.
- Ele foi projetado para lidar com grandes volumes de dados, permitindo alta escalabilidade e flexibilidade.



# Principais Características do MongoDB

- **Armazenamento de Documentos**
  - Os dados são armazenados como documentos BSON dentro de coleções, que funcionam como "tabelas" em bancos relacionais.
- **Escalabilidade Horizontal**
  - O MongoDB suporta sharding, ou seja, divide dados em diferentes servidores para aumentar o desempenho e a capacidade de armazenamento.
- **Índices**
  - Suporte a criação de índices simples, compostos e geoespaciais para consultas mais rápidas.

# Principais Características do MongoDB

- **Consultas Flexíveis**
  - Permite consultas avançadas utilizando filtros, projeções, e agregações.
- **Alta Disponibilidade**
  - Utiliza replicação para garantir que os dados estejam disponíveis mesmo em caso de falha de um servidor.



# Componentes Principais no MongoDB

- **Banco de Dados (Database)**
  - Conjunto de coleções. Cada banco de dados possui seu próprio conjunto de arquivos no sistema.
- **Coleção (Collection)**
  - Um conjunto de documentos, similar a uma tabela em bancos relacionais. As coleções não exigem esquema fixo.
- **Documento (Document)**
  - Unidade básica de dados, armazenada em formato BSON. Cada documento representa um registro e pode conter subdocumentos e arrays.
- **Índice (Index)**
  - Estruturas que melhoram a eficiência das consultas. O MongoDB permite criar índices em qualquer campo.

# Componentes Principais no MongoDB

- **Shards**
  - Fragmentos que permitem armazenar grandes volumes de dados em diferentes servidores.
- **Replica Set**
  - Conjunto de instâncias do MongoDB que garantem alta disponibilidade e tolerância a falhas.
- **Aggregation Framework**
  - Ferramenta para processamento de dados em várias etapas, como filtro, transformação e agrupamento.
- **Mongo Shell**
  - Interface de linha de comando usada para interagir com o MongoDB.

# Operações Básicas no MongoDB

- **Inserção**

- Adicionar novos documentos a uma coleção.

```
db.collection.insertOne({ name: "Alice", age: 25 });
```

- **Consulta**

- Recuperar documentos com base em filtros.

```
db.collection.find({ age: { $gt: 20 } });
```

# Operações Básicas no MongoDB

- **Atualização**

- Modificar documentos existentes.

```
db.collection.updateOne({ name: "Alice" }, { $set: { age: 26 } });
```

- **Remoção**

- Excluir documentos de uma coleção.

```
db.collection.deleteOne({ name: "Alice" });
```





# Referências

- Curso completo de FastAPI por Eduardo Mendes
  - <https://fastapidozero.dunossauro.com/>
  - <https://github.com/dunossauro/fastapi-do-zero>
  - [Playlist no YouTube](#)
- <https://www.mongodb.com/docs/>
- MongoDB: <https://www.mongodb.com/pt-br>
- FastAPI - <https://fastapi.tiangolo.com/>
- Pydantic - <https://pydantic.dev/>
- SQLAlchemy - <https://www.sqlalchemy.org/>
- SQLAlchemy - <https://sqlmodel.tiangolo.com>



# Obrigado!

## Dúvidas?



**Universidade Federal do Ceará - *Campus* Quixadá**

**Prof. Francisco Victor da Silva Pinheiro**  
victorpinheiro@ufc.br

