

# Bancos de Dados Não Convencionais, Não Relacionais ou No SQL

Banco de Dados II



# Conceito

O banco de dados não relacional surgiu como uma alternativa para situações em que há uma quantidade exorbitante de dados para armazenar, onde a estruturação em tabelas se torna mais difícil. Sendo definido como um banco de dados que não usa o sistema de tabelas.



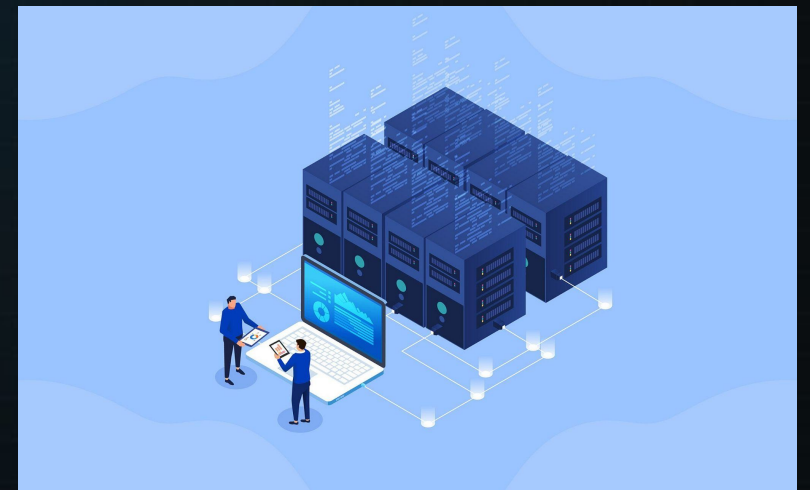
# Diferença entre bancos

**Banco de dados relacional:** Fornece dados íntegros e imutáveis, garantindo um dado consistente, possuindo sistema rígido.

Ex: Oracle, MySQL, PostgreSQL, MariaDB.

**Banco de dados não relacional:** Representa diversos tipos de bancos de dados, não exigindo a rigidez de esquemas para armazenar os dados, sendo possível adicionar novas propriedades.

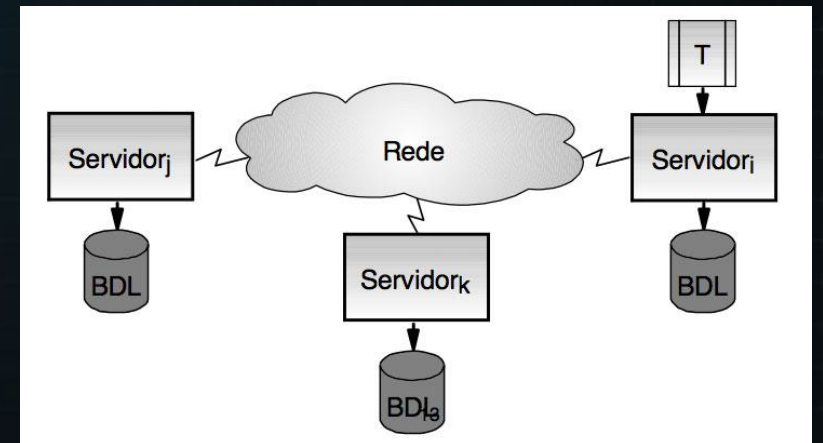
Ex: Memcached, Cassandra, MongoDB.



# Diferença entre bancos

**Data Warehouses:** Permitem a integração de dados de diferentes fontes. Além de manter os dados atuais, também é possível consultar um histórico. Geralmente, o Data warehouse é indicado para criação de relatórios de análise, sendo possível compilar grande quantidade de dados. Ou seja, permitem a consulta e armazenamento de dados em grandes volumes.

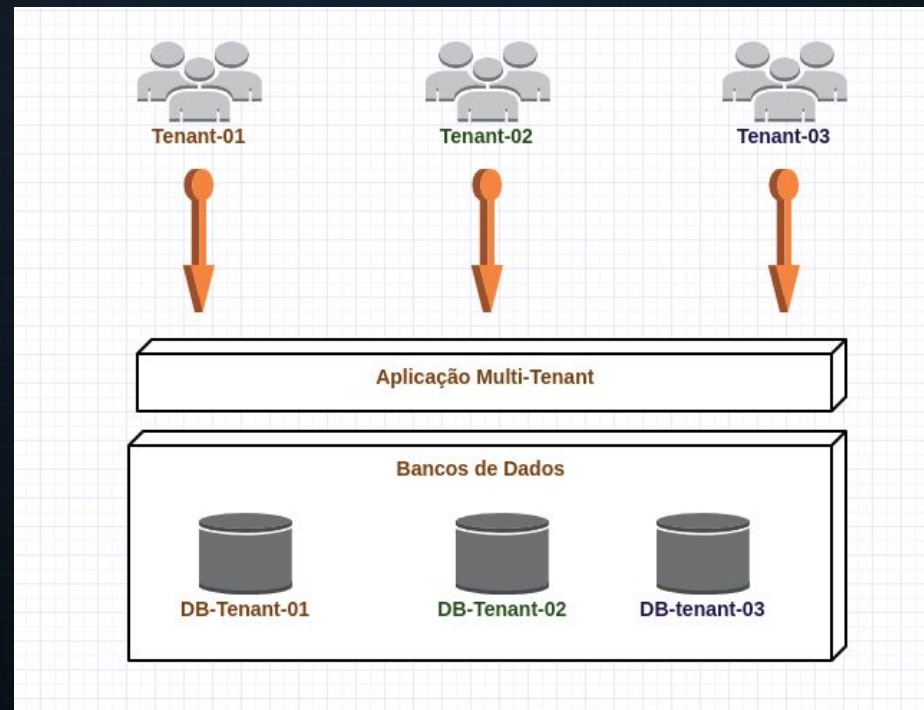
**Banco de dados distribuídos:** São compostos por uma rede de nós, em que cada nó é um computador em um local diferente, se comunicando via rede internet, com o seu principal diferencial sendo a distribuição de informações.





# Diferenças entre bancos

Também existem outros tipos de bancos, como os autônomos, OLTP, JSON, gráficos, os bancos de dados na nuvem e multi modelo, sendo esse último reunindo uma combinação de bancos de dados em um único back-end, o que o torna extremamente flexível.



# Banco NoSQL



O termo 'NoSQL' se refere a tipos não relacionais de bancos de dados que não usam o SQL para consultas.

Os bancos de dados NoSQL (não-relacionais) surgiram a partir da necessidade de utilizar soluções mais fáceis de ampliar ou reduzir a infraestrutura em que há uma quantidade exorbitante de dados para armazenar, onde a estruturação em tabelas se torna mais difícil conforme as empresas crescem.

# Banco NoSQL



Os bancos de dados NoSQL são amplamente reconhecidos por:

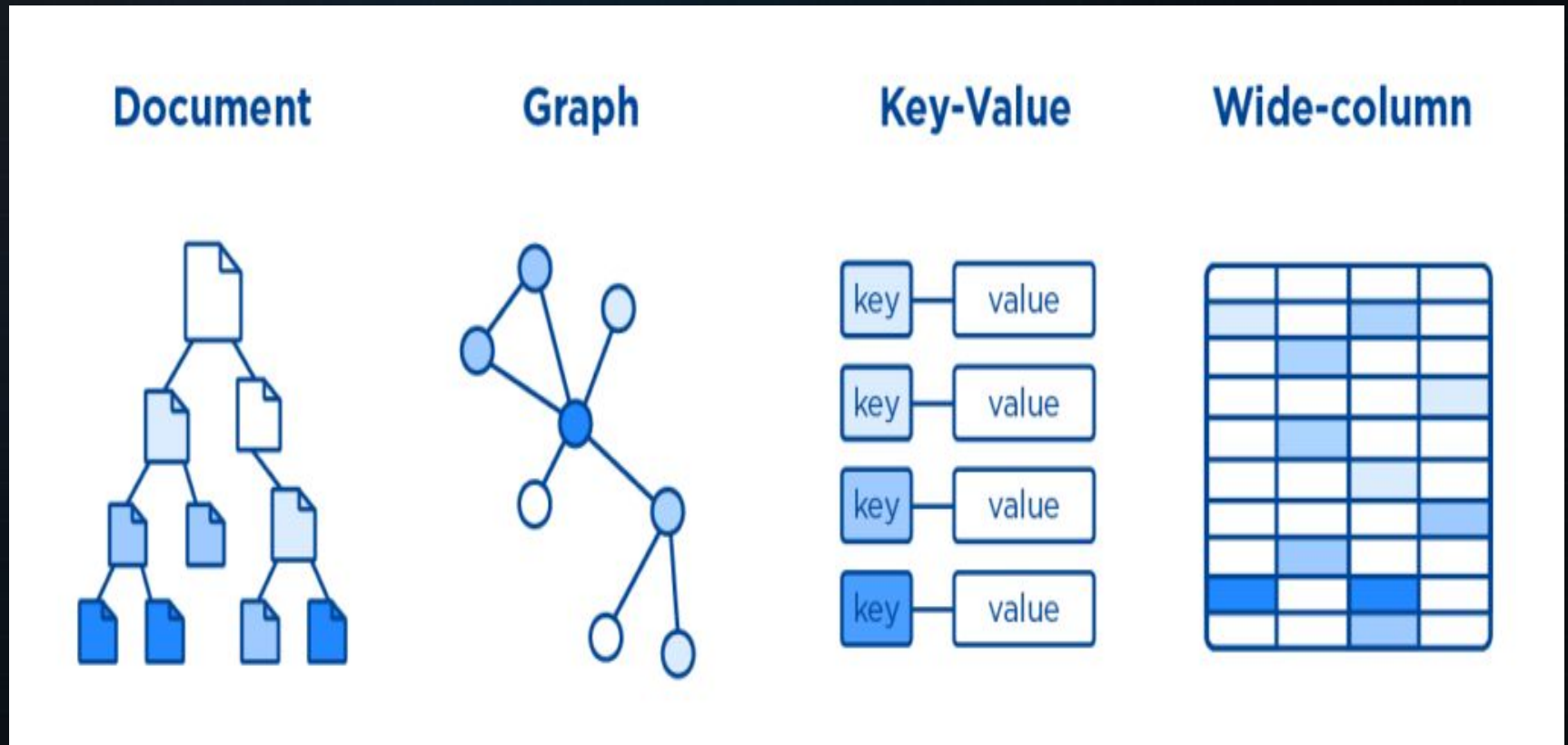
- Sua facilidade de desenvolvimento;
- Escalabilidade;
- Compatibilidade com Big Data;
- Flexíveis;
- Alta performance;
- Funcionais.

Aceitam um paradigma de desenvolvimento ágil, adaptando-se rapidamente aos requisitos em constante mudança, ajudando a gerenciar os novos desafios da diversidade crescente de tipos de dados e modelos e de serem altamente eficazes ao processar dados imprevisíveis.

# Tipos de Banco NoSQL

## Tipos de bancos de dados NoSQL

- Chave-valor;
- Documento;
- Gráfico;
- Colunas.





# Banco de Dados Não Convencionais

**Chave-valor:** Os armazenamentos de chave-valor fazem o pareamento de chaves e valores usando uma tabela de hash. Os tipos de chave-valor são melhores quando uma chave é conhecida e o valor associado dela é desconhecido.

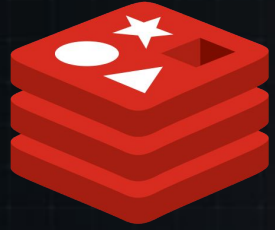
**Documento:** Os bancos de dados de documentos ampliam o conceito do banco de dados chave-valor organizando documentos inteiros em grupos chamados coleções. Eles são compatíveis com os pares chave-valor aninhados e permitem consultas em qualquer atributo em um documento.

# Banco de Dados Não Convencionais

**Gráfico:** Os bancos de dados de grafo usam um modelo baseado em nós e bordas para representar dados interconectados – como relações entre pessoas em uma rede social – e oferecem armazenamento simplificado e navegação por meio de relações complexas.

**Colunar:** Bancos de dados colunares, de coluna larga ou de famílias de colunas armazenam dados de modo eficiente, consultam linhas de dados esparsos e são vantajosos ao consultar em colunas específicas no banco de dados.

# Principais Bancos NoSQL No Mercado



redis

APACHE  
HBASE

neo4j



MEMCACHED



cassandra



amazon  
DynamoDB



mongoDB

# Banco MongoDB

O MongoDB é orientado a documentos, ou seja, os dados são armazenados como documentos, ao contrário de bancos de dados de modelo relacional, onde trabalhamos com registros em linhas e colunas. Os documentos podem ser descritos como dados no formato de chave-valor, no caso, utilizando o formato JSON (JavaScript Object Notation).



# Características

- Sintaxe para consultas
- Indexação
- Escalabilidade horizontal
- Permite executar consultas executando código JavaScript.

# Vantagens

- Ideal para ambientes com poucos recursos, onde qualquer servidor pode atender você;
- Custo baixo: É uma ferramenta com custo baixo, pois por ser de código aberto, você não precisa pagar uma licença;
- Complemento perfeito para o JavaScript.

# Desvantagens

- Não se adequa para aplicações com transações complexas. Neste caso os banco de dados relacionais talvez sejam mais ideais;
- Ainda é uma tecnologia nova, apesar de ser amplamente utilizada;
- Não utiliza 'joins' para consultas, ou seja, aquelas consultas nas quais combinamos tabelas diferentes. No MongoDB temos que executar esse tipo de consulta de maneira diferente.

# Situações de uso vantajosas

## Banco de Dados Não-Convencionais

**Flexibilidade:** Por conter certa flexibilidade, a utilização de um banco NoSQL pode vir a ser por diversas razões, mas assim como todas as situações podem apresentar prós e contras, tem que analisar caso a caso.

No caso do NoSQL, pode se tornar útil e atrativo aos olhos de quem utiliza desenvolvimento ágil, pois dessa forma elimina a complexidade de mudança dos bancos de dados, assim tendo um bom suporte e adaptações rápidas.



# Situações de uso vantajosas

## Banco de Dados Não-Convencionais

**Eficiência:** Outro fator que pode trazer vantagem, é a grande eficiência de busca por conta de sua arquitetura, assim podendo fazer buscas ainda mais rápidas para certa quantidade de dados.



# Situações de uso vantajosas

## Banco de Dados Não-Convencionais

**Baixo custo:** Com os bancos relacionais, é preciso manter mega computadores, mainframes e suportes muito caros, e dependendo do banco ainda o pagamento do mesmo.

Com o NoSQL, isso é reduzido de forma drástica, visto que como grande maioria é open source, o seu custo é muito abaixo, pois não necessita especificamente de um mainframe, pois pode ser desenvolvido para trabalhar em ambientes distribuídos.

# Conclusão

Ao escolher qual banco de dados usar em seu próximo aplicativo, considere usar um banco de dados NoSQL.

Não é à toa que a popularidade do NoSQL cresceu tão rapidamente nos últimos anos.

Além de optar por usar NoSQL, há uma variedade de bancos de dados NoSQL com diferentes recursos que podem ser muito úteis durante o desenvolvimento.

Logo, se você planeja desenvolver um software que possa crescer massivamente, a melhor alternativa é o NoSQL.

The logo for NoSQL, with 'nosql' in a lowercase, sans-serif font. The 'no' is red and the 'sql' is blue.