

# Aula 2: Modelos de bancos de dados NoSQL

## Apresentação

---

Nesta aula, iniciaremos o estudo dos modelos de bancos de dados NoSQL. Verificaremos os quatro modelos que compõem esse tipo de banco de dados. Estudaremos cada modelo de modo a verificar as características que os definem, assim como avaliar as diferenças entre eles e as vantagens de cada um. Com esse estudo inicial, poderemos conhecer as aplicações adequadas para cada tipo de modelo de dados NoSQL.

# Objetivo

---

- Classificar os diferentes modelos de bancos de dados NoSQL;
- Avaliar as diferenças entre os modelos e suas vantagens;
- Reconhecer as aplicações adequadas para cada tipo de modelo NoSQL.

## Introdução

---

Em termos de definição, não há um conceito único quando se trata do banco de dados NoSQL, pois não há apenas um modelo; pelo menos quatro modelos de dados são usados nesse tipo de banco de dados. Trata-se de um grupo de modelos de dados usados para manipular grandes volumes de dados que não estão limitados apenas para uso com a linguagem estruturada de dados (SQL) para sua manipulação.

Todos os modelos de banco de dados foram desenvolvidos para atender a necessidades específicas do período em que foram lançados; necessidades apresentadas pelas aplicações em uso, como, por exemplo, as redes sociais atualmente. Os dados gerados pelas redes sociais começaram a aumentar de uma forma tão acentuada com o uso de aplicativos que foram necessárias medidas emergenciais para manter o controle e a gerência dos dados.



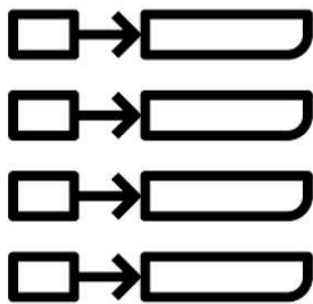
Fonte: Przemyslaw Marczyński / Unsplash.

## Modelos de bancos de dados NOSQL

---

## Banco de dados chave-valor

Além dessa justificativa para o desenvolvimento desse novo tipo de banco de dados, pode ainda ser destacado o fato de que o banco de dados relacional pode apresentar lentidão em consultas com grandes volumes de dados e também não permite a previsão de crescimento do processamento, sendo necessários ajustes na configuração. Quanto aos tipos de dados armazenados, os modelos NoSQL podem ser usados para diferentes formatos, com destaque para o tipo constituído de pares chave-valor para armazenar dados de modo simples; um exemplo desse tipo de formato é o tipo [JSON](#) (JavaScript Object Notation).



Dentre os modelos de dados NoSQL, destaca-se o modelo baseado no armazenamento de dados chave-valor. Esse é um dos modelos mais usados e é conhecido pelo formato como armazena os dados, ou seja, utilizando pares de chaves-valores, sendo denominados como *key-value pair databases* (KVP). O modelo contém conjuntos de pares chave-valor, ou seja, primeiro a definição da chave seguida do valor dado a ela.

 Fonte: O autor.

Os valores armazenados podem ser de qualquer tipo de objeto binário, como dados textuais, documentos no formato de intercâmbio JSON, entre outros. Esses objetos são indexados por chaves, possibilitando a busca por chaves únicas, como no exemplo a seguir, recuperando apenas os valores da chave cliente em um atributo do tipo JSON:

```
INSERT INTO pedidos (detalhes) VALUES('{"cliente":"Ana", "itens":{"produto":"tablet", "qtd":2}}')
SELECT detalhes->'cliente' AS cliente FROM pedidos
```

Com o uso desse modelo de dados, a aplicação tem o controle sobre a forma mais adequada de armazenamento, tornando-a mais flexível. No banco de dados relacional, um registro tem um número determinado de atributos, mas nesse modelo NoSQL um registro é formado por um conjunto de pares do tipo chave-valor que nem sempre tem os mesmos pares. No modelo chave-valor, não há exigência de um esquema, como acontece no relacional. Isso possibilita, além da grande flexibilidade, a facilidade na escalabilidade da base de dados.

### Exemplo

Com exemplos de banco de dados NoSQL do modelo chave-valor, podem ser citados o [Redis](#), [Amazon DynamoDB](#), [Microsoft Azure Cosmos DB](#), todos multimodelos, por admitirem outros modelos de dados. E outros chave-valor puros, como [Memcached](#) e [etcd](#).

O modelo chave-valor foi projetado para não oferecer suporte a transações. O controle das aplicações exige que os desenvolvedores desses sistemas codifiquem a parte do controle das transações de forma a efetuar o tratamento dos dados; o controle de réplicas, quando necessário, e da própria tolerância a falhas. Porém, isso foi previamente definido para facilitar o uso do modelo em aplicações na web, em aplicações com sensores onde há a inclusão automática dos dados e em equipamentos móveis, como smartphones e tablets. Em outras palavras, em aplicações nas quais os usuários esperam respostas rápidas para suas solicitações.

Além da velocidade, o modelo chave-valor permite maior escalabilidade e disponibilidade, pois os dados podem ser particionados e replicados em clusters, consequentemente eliminando o ponto único de falha. Esse, então, é o motivo por que o modelo é adequado a esses tipos de aplicações com a adequação ao tipo de armazenamento de dados, colocando a questão do controle transacional em segundo plano em escala de prioridade.

## Banco de dados orientado a documentos

As vantagens principais deste modelo podem ser listadas como a capacidade de suportar conjuntos de pares chave-valor, ou seja, sua característica principal é o armazenamento de documentos. Os documentos podem ser compreendidos como uma forma de organização em **coleções** de documentos com características semelhantes, onde cada elemento pode ser um item simples ou composto. Um registro na base de dados faz relação com um documento. Em sua composição, os documentos normalmente são compostos por um identificador e um valor.

## Comentário

Quanto ao valor pode-se utilizar outros identificadores acompanhados de valores. Como exemplos podem ser apresentados documentos simples ou embutidos em outro (aninhado) e listas. O modelo também possui ainda um bom desempenho em face das exigências para o funcionamento adequado de aplicações que necessitam funcionar no ambiente distribuído como é o caso do baseado em Big Data.

## Exemplo

Como exemplos desses bancos de dados podemos citar o [MongoDB](#), [RavenDB](#), [CouchDB](#) e ainda o próprio [PostgreSQL](#) com o uso do formato [JSONB](#), que é uma versão do JSON muito mais rápida pois permite a criação de índices. Quanto ao formato de um documento, esses podem ser codificados de diferentes formas, como: XML, JSON, [BSON](#) (*Binary JSON*), YAML (*YetetAnother Markup Language*) ou ainda pode ser usado com arquivos em texto simples. O tipo JSON é o mais utilizado atualmente.

A seguir é apresentado um exemplo de uma coleção de documentos no MongoDB, o banco de dados NOSQL mais popular no ranking de popularidade de [SGBDs](#):

```
db.postagem.insert([
  { titulo: 'Banco de dados orientado a documentos',
    detalhes: 'é um dos modelos de bancos de dados NOSQL',
    autor: 'curso NOSQL',
    link_site: 'www.curso_de_nosql.com',
    tags: ['banco de dados', 'NOSQL', 'MongoDB'],
    nr_de_likes: 1000},
  { titulo: 'NoSQL Database',
    detalhes: "CouchDb usa views ao invés de coleções",
    autor: 'curso NOSQL',
    link_site: 'www.curso_de_nosql.com',
    tags: ['banco de dados', 'NOSQL', 'CouchDB'],
    nr_de_likes: 200,
    comentario: [{login: 'usuario1', mensagem: 'post1', data: new Date("2020-08-30"), like: 0}]}])
```

## Banco de dados colunar ou orientado por colunas

Os bancos de dados relacionais são orientados ou organizados em linhas (registros), pois os dados em cada linha de uma tabela são armazenados juntos, por atributos da tabela. Já no banco de dados colunar, ou orientado por colunas, os dados são armazenados em colunas. Embora essa diferença possa parecer extremamente simples, ela é a característica mais importante dos bancos de dados colunares.

### Exemplo

São exemplos desses bancos de dados o [Cassandra](#)<sup>1</sup>, o [HBase](#), o [Google BigTable](#) e o [Hypertable](#).

Uma linha no banco de dados é uma coleção de colunas que tem uma associação a uma determinada chave. Assim, uma família de colunas corresponde a uma coleção de linhas semelhantes.

No caso do Cassandra, como é uma característica do NOSQL de não estar limitado ao uso da SQL, existe a [CQL](#) – Cassandra Query Language, que é uma linguagem que suporta comandos do tipo SQL. Vejamos um exemplo de um comando para criar uma tabela no Cassandra:

```
CREATE COLUMNFAMILY cliente (  
    codigo varchar primary key,  
    nome varchar,  
    cidade varchar )
```

Outro ponto positivo do modelo surge pela maior consistência, pois permite a execução de consultas paralelas. O banco de dados colunar não é indicado em sistemas que necessitam de controle transacional com as propriedades ACID (Atomicidade-Consistência-Isolamento-Durabilidade) e para o uso com consultas envolvendo agregações, como a função SUM (somatório) por conta de sua estrutura.

## Banco de dados orientado a documentos

Veja alguns aspectos sobre o surgimento desse banco de dados:

- 1** Esse banco de dados surgiu nas décadas de 80 e 90 no mesmo período do modelo orientado a objetos. Porém, não foi tão usado como os bancos de dados semiestruturados usando a XML.
- 2** O interesse pelos bancos de dados em grafos voltou com o uso acentuado da web e, mais especificamente, com a Web de Dados, na qual a internet é usada como plataforma para a troca de dados.
- 3** Com a popularização das redes sociais, o interesse nos bancos de dados em grafos aumentou ainda mais, pois o formato gráfico do esquema (conjunto de tabelas relacionadas em uma aplicação) usado por esse modelo facilitou o entendimento da aplicação e o desenvolvimento de consultas mais elaboradas envolvendo ainda várias tabelas.

Além disso, eles podem definir naturalmente as relações existentes em uma base de dados. Os bancos de dados gráficos foram criados especificamente para manter e melhor explorar os relacionamentos existentes nas aplicações, navegando por eles graficamente e por meio de linguagem de consulta específica. Os bancos de dados em grafos são compostos em sua modelagem por nós que representam as entidades de dados, arestas que indicam os relacionamentos existentes entre os nós, e propriedades que descrevem os dados armazenados.

**Sobre a estrutura, podemos definir, ainda, que é um conjunto de vértices e ramos na representação dos dados, no lugar do uso de tabelas ou pares chave-valor. A desvantagem dessa característica é a possibilidade da existência de dados com redundância ou com incoerência.**

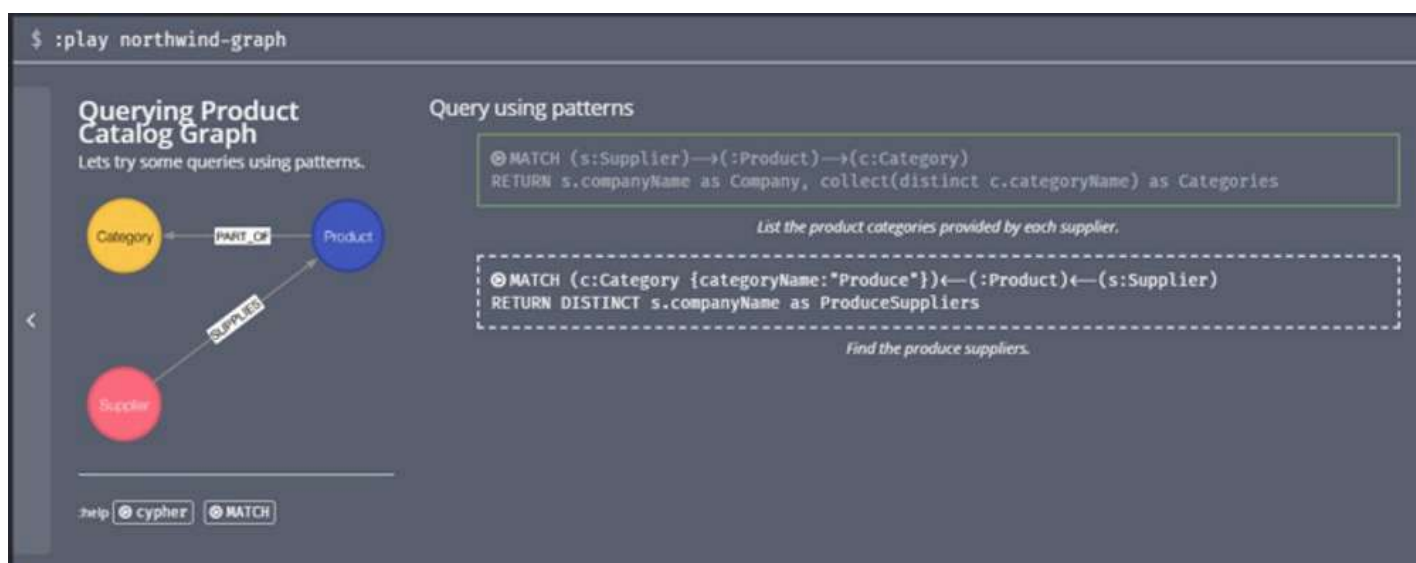
Exemplo

Alguns exemplos de bancos de dados em grafos: [Neo4J](#), [AllegroGraph](#), [OrientDB](#), [InfiniteGraph](#) e [Amazon Neptune](#).

Entre as vantagens quando em comparação aos bancos de dados relacionais, podemos dar destaque para o uso da estrutura em grafo que facilita a forma de apresentação dos objetos do banco de dados e das suas propriedades. Assim como no caso da teoria de conjuntos, a teoria dos grafos é bem conhecida e testada, fazendo uso de implementações de algoritmos para percorrer os grafos à procura de nós com os dados, isso de forma bastante otimizada pelos recursos que utiliza. Com o uso da tecnologia baseada em grafos, o banco de dados permite mais flexibilidade na recuperação de dados e é possível incluir ou excluir vértices e/ou ramos sem maiores dificuldades.

Na Figura 1, é apresentada uma consulta executada no banco de dados com o tradicional exemplo Northwind no Neo4J onde pode-se observar a forma gráfica de apresentação dos resultados de uma consulta na linguagem Cypher.

**Figura 1 - Exemplo de consulta no Neo4J**




Fonte: <https://neo4j.com/>.

## Tipos de aplicações adequadas para os bancos de dados NoSQL

É sempre bom lembrar que as tecnologias surgem pela necessidade de solução aos problemas que ocorrem de acordo com a demanda da sociedade. Agora, apresentaremos as indicações de aplicações para os modelos de bancos de dados NoSQL:

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

 Clique nos botões para ver as informações.



O modelo chave-valor é indicado para aplicações que necessitam de alto desempenho, sendo que isso também apresenta como consequência a redução total nos gastos operacionais de um sistema. São bons exemplos desses tipos de aplicações:

- A análise em tempo real dos serviços financeiros como a bolsa de valores, onde há um grande volume de dados a ser processado e que necessita de uma rápida resposta nas requisições dos usuários;
- Em aplicações bancárias que receberam um aumento da carga de transações devido às operações realizadas pelos aplicativos de pagamento;
- Para o controle de dados de sessões web que possuem um conjunto de dados de entrada (put) ou recuperação (get) e esses dados podem ser armazenados de forma única, agilizando o processamento;
- Tratamento de dados de carrinhos de compras no comércio eletrônico pode ter uma melhor performance com o uso do modelo chave-valor;
- Em aplicações para comunidades interessadas em jogos eletrônicos online ou dados obtidos de sensores em sistemas baseados em Internet das Coisas (IoT). Como exemplos, podemos listar o uso do DynamoDB pela Samsung para efetuar o backup dos dados de seus aplicativos pelo grande volume dos dados. A Netflix também faz uso do DynamoDB para a execução de testes com dados personalizados de experiências de streaming para milhões de seus clientes.

## Orientado a documentos



A flexibilidade na estrutura permite que o modelo baseado em documentos seja indicado para alguns tipos de aplicações, tais como:

- São indicados em aplicações onde o banco de dados orientado a documentos pode servir para centralizar os dados de diferentes tipos de eventos de logs de sistemas. Uma forma de controle é registrar a origem do evento e manter chaves indicando o tipo de evento e o nome do aplicativo;
- São úteis em aplicativos de comércio eletrônico por conta das diferenças de atributos para as diversas categorias de produtos existentes. Sua utilização elimina a necessidade de constantes alterações na estrutura das tabelas;
- Pode auxiliar em sistema de gerenciamento de conteúdo onde os documentos armazenados não têm um formato pré-estabelecido, como comentários de usuários em blogs e suas respectivas postagens.

Agora apresentaremos exemplos de aplicações apropriadas para o modelo baseado em colunas:

- Pode ser útil na função de contadores de visitantes em aplicativos. O Cassandra dispõe do tipo `CounterColumnType` para esse fim;
- Sistema de registro de logs tendo em vista a tendência de um grande volume de dados nesse tipo de aplicação;
- Nas aplicações com dados dos usuários produzidos pelas redes sociais, o Facebook usa o MySQL para os dados estruturados e o Cassandra e o HBase para o controle de mensagens;
- Plataforma de blogs podem fazer uso do banco colunar para manter os comentários em colunas diferentes para as categorias apresentadas na aplicação.

## Grafos



Em relação a indicações de aplicações para os bancos de dados em grafos, eles podem facilitar o uso da interconectividade dos dados e, por isso, são vantajosos em casos de uso como:

- Em sistemas de redes sociais, pois é mais simples percorrer um grafo de associações de uma forma visual do que fazer uma consulta complexa para analisar os relacionamentos existentes entre os usuários da rede;
- No uso com sistemas de recomendação, que são aqueles que apresentam sugestões para as compras mais acertadas de produtos de acordo com o perfil do usuário. Com isso, aumenta-se o nível de satisfação dos clientes e, consequentemente, as chances de novas compras serem realizadas;
- Em sistemas de detecção de fraudes, onde é necessário criar relacionamentos entre os dados e consultar rapidamente tal base de conhecimento para evitar prejuízos no momento em que estão ocorrendo;
- O BigTable do Google é usado em sistemas que exigem uma personalização de conteúdo, seleção de anúncios, com mídias digitais e IoT.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

## Atividade

1. Assinale a afirmativa incorreta quanto às características dos modelos de dados NoSQL:

- a) O modelo chave-valor é o mais usado e é conhecido pelo formato como armazenam os dados, ou seja, utiliza pares de chaves-valores, sendo denominados como *key-value pair databases* (KVP).
- b) Os bancos de dados em grafos são compostos em sua modelagem por nós que representam as entidades de dados, arestas que representam os relacionamentos existentes entre os nós, e propriedades que descrevem os dados armazenados.
- c) Assim como no modelo orientado a colunas, o modelo relacional possibilita a escolha da orientação do armazenamento em colunas, semelhantemente ao armazenamento em linhas.

- d) Uma linha no banco de dados orientado a colunas, ou colunar, é uma coleção de colunas que tem uma associação a uma determinada chave, e uma família de colunas corresponde a uma coleção de linhas semelhantes.
- e) A vantagem principal do modelo orientado a objetos é que ele pode ser listado como a capacidade de suportar conjuntos de conjuntos de pares chave-valor, ou seja, sua característica principal é o armazenamento de documentos.
- 

2. Assinale a opção que indica de forma incorreta exemplos de SGBDs para cada modelo NoSQL:

- a) MongoDB –documentos.
  - b) Neo4J - grafos.
  - c) Hypertable - chave-valor.
  - d) CouchDB - documentos.
  - e) Hbase – colunar.
- 

3. Assinale a opção falsa em relação às indicações de aplicações de uso ou categorias de sistemas de informação mais adequados para cada um dos modelos de bancos de dados NoSQL apresentados:

- a) Grafos - uso com sistemas de recomendação para a indicação mais acertada de produtos de acordo com o perfil do usuário.
  - b) Colunar - plataformas de blogs e contadores de visitantes em aplicativos.
  - c) Documentos - em aplicativos de comércio eletrônico por conta das diferenças de atributos para as diversas categorias de produtos existentes.
  - d) Chave-valor- no controle de dados de sessões web.
  - e) Grafos - análise em tempo real dos serviços financeiros como a bolsa de valores onde há um grande volume de dados a ser processado.
- 

## Notas

### Cassandra <sup>1</sup>

---

O Cassandra foi criado pelo Facebook, que disponibilizou o código-fonte ainda em 2008, e nele uma coluna corresponde a um par de nome-valor, onde o nome funciona como a chave.

<https://cassandra.apache.org/>

## Referências

---

Elmasri, R.; Navathe, S.B.; **Sistemas de Banco de Dados**. 6ª. Edição; São Paulo: Pearson Education do Brasil; 2018.

Pramod J. Sadalage, Martim Fowler. **NoSQL Essencial - Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**.

1. São Paulo: Novatec, 2013.

## Próxima aula

— Na próxima aula, daremos continuidade aos estudos vistos aqui, abordando os seguintes conteúdos:

---

- Características dos bancos de dados baseados em chave-valor;
- Conceito *schemaless* (sem esquema);
- Escrever comandos NoSQL para a definição da estrutura do banco de dados.

## Explore mais

---

Para aprofundar seus conhecimentos fica a sugestão de leitura do livro *Sistema de banco de dados* de Elmasri, R.; Navathe, S.B.; Sistemas de Banco de Dados no Cap. 24.1.2: Características dos sistemas NOSQL

De modo e entender com mais detalhes o Teorema CAP, faça a leitura da Seção 5.3.1 do livro: Pramod J. Sadalage, Martin Fowler. **NoSQL Essencial - Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**. 1. São Paulo: Novatec, 2013.