

Banco de Dados



# Unidade 7

## Dados não estruturados

Profª Eliana Caus Sampaio

Prof. Howard Roatti



# Estruturado, semiestruturado, não estruturado?

De antemão vale registrar que há discussões sobre a validade dessa classificação, pois mesmo os dados ditos não estruturados possuem algum tipo de estruturação própria.

# Dados estruturados

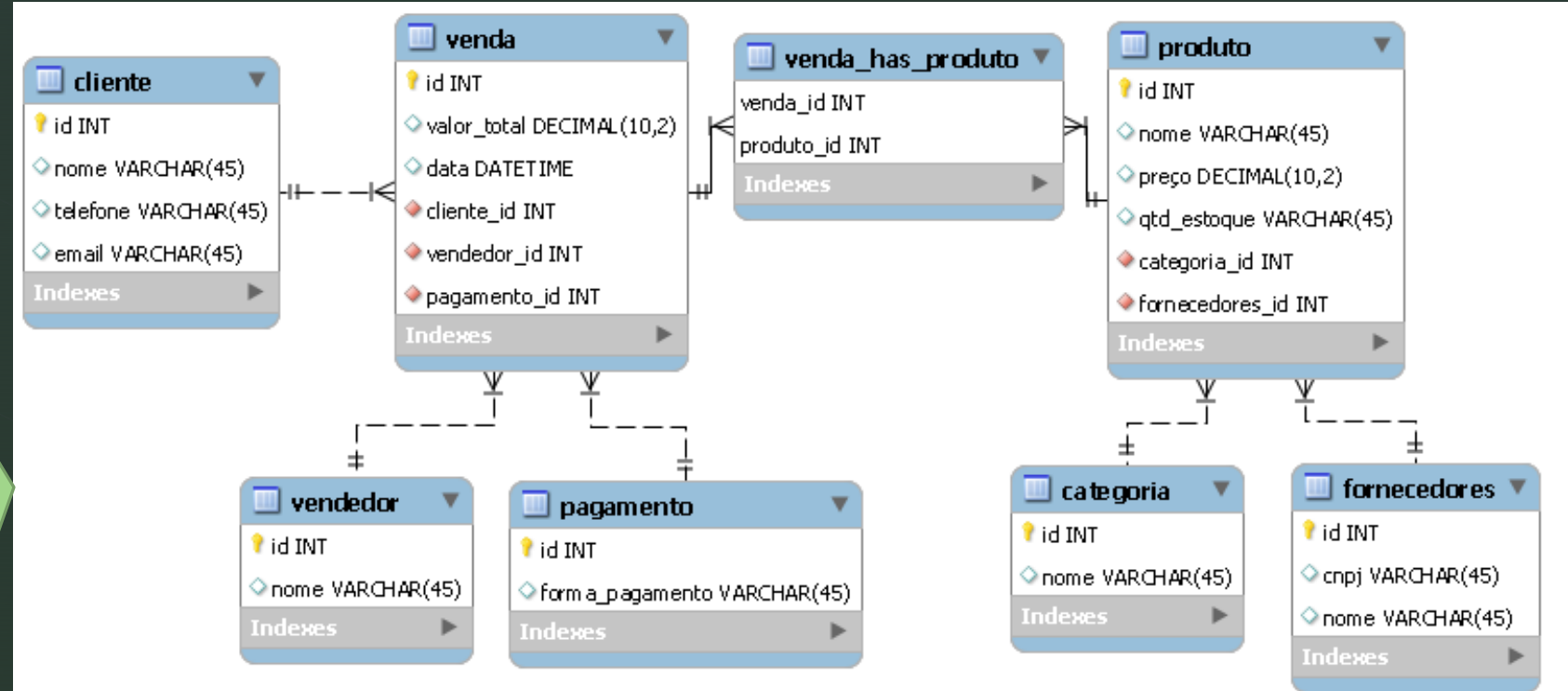
O dado é dito estruturado quando contém uma **organização** para serem recuperados.

São como **etiquetas**, **linhas** e **colunas** que identificam pontos sobre aquela informação que facilitam o trabalho sobre eles.

São utilizados pela maioria das empresas e embora os dados produzidos ao longo do tempo sejam maiores do que os dados que são representados, certamente são eles os utilizados para a tomada de decisão em muitas organizações há muitos anos.

# Dados estruturados

Esses dados possuem estruturas bem definidas, rígidas, pensadas antes da existência do conteúdo que irá povoar aquela estrutura.



Por serem pré-definidas, cada dado só consegue armazenar um conteúdo compatível com a estrutura definida para ele, por exemplo, um dado definido como numérico não aceitará conteúdo que fira essa propriedade, tal como um texto



# Dados estruturados

A análise dos dados estruturados possui uma facilidade implícita na sua existência, já que a sua estrutura não muda com frequência e que os dados carregados seguem padrões predeterminados.

```
SELECT  P.PRO_NOME AS "NOME DO PRODUTO",  
        P.PRO_PRECO,  
        U.UE_DESC  
FROM PRO P,UE U  
WHERE P.UE_COD = U.UE_COD  
      AND P.PRO_PRECO < 20 OR P_PRECO IS NULL  
      AND UPPER(U.UE_DESC) IN ('KILOGRAMA', 'LITRO');
```

Dessa maneira a análise não requer avançadas técnicas de interpretação ou conhecimentos estatísticos.

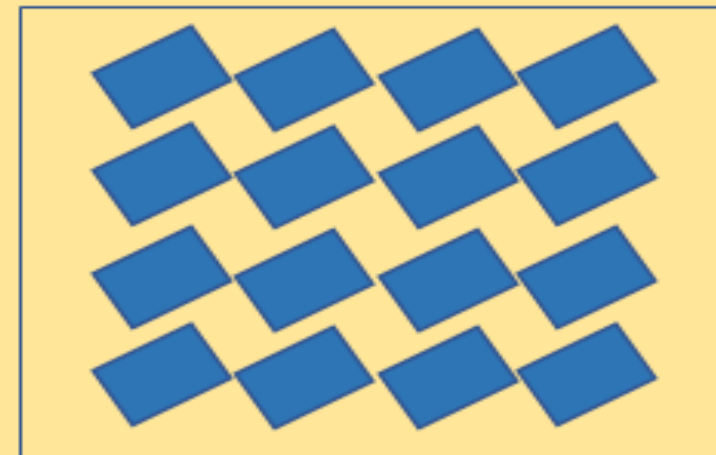
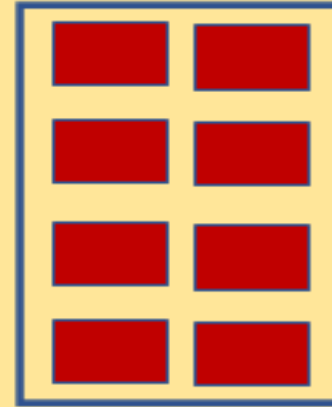
Alguns exemplos de dados estruturados são as planilhas eletrônicas (Excel), banco de dados, arquivo XML e CSV.

# Dados estruturados

Dados estruturados geralmente residem em *Relational Data Base System* – RDBMS – e são gerados tanto por humanos quanto por máquinas desde que sejam criados dentro de uma estrutura relacional.

Para manusear esses dados, utiliza-se *Structured Query Language* – SQL

## Dados estruturados



# Dados semiestruturados

Um meio termo entre estruturados e não estruturados, os semiestruturados apresentam representação estrutural heterogênea, não sendo nem estritamente tipados, nem completamente não-estruturados.

Nesses dados são chamados de auto-descritivos, pois o esquema de representação está presente de forma explícita ou implícita juntamente com o dado. Para que a estrutura seja identificada e extraída, primeiramente deve-se fazer uma análise do dado.



# Dados semiestruturados

**Como principais características de dados semiestruturados temos:**

1. Definição a posteriori:.
2. Estrutura irregular:
3. Estrutura implícita:
4. Estrutura parcial:.
5. Estrutura extensa:
6. Estrutura evolucionária:
7. Estrutura descritiva e não prescritiva:
8. Distinção entre estrutura e dados não é clara:



# Dados semiestruturados

Como exemplo de dados semiestruturados temos:

1. **Linguagem de marcação XML:** conjunto de regras de codificação de documentos, orientado por tags, que define um formato legível tanto para humanos quanto para máquinas.
2. **JSON (*Java Script Object Notation*) de padrão aberto:** formato de intercâmbio de dados semiestruturado reconhecido não só pelo Java. Está estruturado em pares nome/valor (ou objeto / tabela hash) e uma lista de valores ordenada (ou matriz, sequência, lista). Por ser uma estrutura intercambiável entre as linguagens, o JSON se destaca na transmissão de dados entre aplicativos da Web e servidores.
3. **Dados semiestruturados NoSQL:** nesses bancos de dados não há separação da organização (esquema) dos dados o que o torna a melhor alternativa para armazenar informações que não cabem facilmente no formato de registro e tabela como textos com comprimento variável.

# Dados semiestruturados

	Dados estruturados	Dados semiestruturados
Esquema	Pré-definido	Nem sempre há um esquema pré-definido
Estrutura	Regular	Irregular
	Independente do dado	Embutida no dado
	Reduzida	Extensa
	Fracamente evolutiva	Fortemente evolutiva
	Prescritiva	Descritiva
Distinção entre estrutura e dado	Clara	Não é clara

# Dados não estruturados

- Nem todas as informações possíveis podem estar contidas dentro dos bancos de dados.
- Além disso, os dados estruturados precisam ser preenchidos para que o uso deles se dê de forma automatizada.
- No entanto, documentos de texto, não são vistos na plenitude.
- Muitas informações contidas nesses documentos podem não estar sendo devidamente analisados por falta de condições de percorrer o seu conteúdo, justamente pela dificuldade que seria classificar cada palavra do texto e relacioná-las com contextos, momentos, pessoas ou citações.

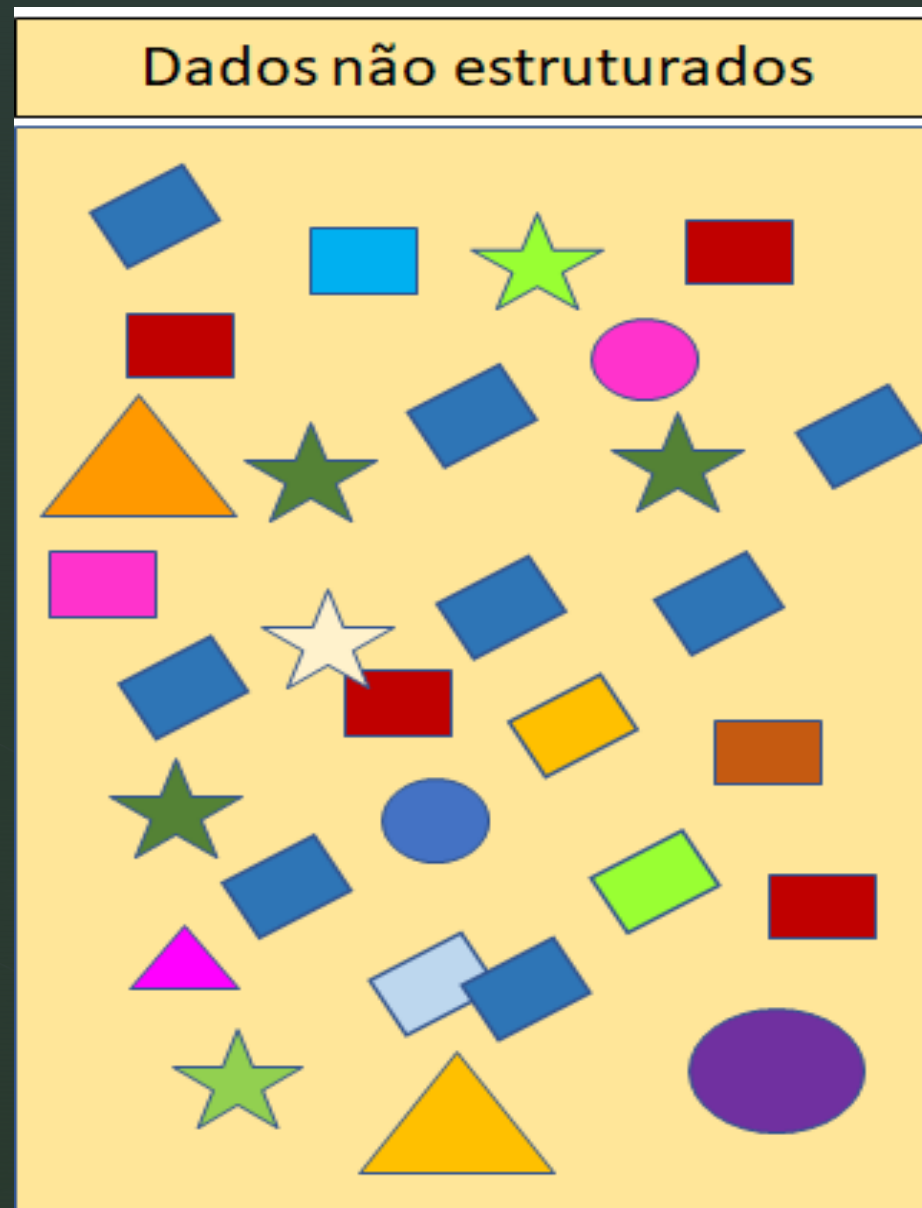
Pense em todas as palavras contidas em um bloco de notas, um e\_mail ou um documento de um editor de texto e verá a dificuldade de se relacioná-las a algum contexto significativo. Soma-se a isso a produção oriunda das redes sociais, onde as próprias emoções estão contidas nas manifestações de seus usuários.

Essa incapacidade é ainda maior em dados contidos em vídeos ou áudios.

## Dados não estruturados

Possuem estruturas indefinidas, desalinhadas, não padronizadas, podendo ser compostos por diversos elementos diferentes.

São gerados e serão consumidos de maneira diferente daqueles utilizados em dados estruturados.





# Dados não estruturados

- Dados estruturados estão atrelados a um contexto o que facilita a extração de informações, porém, somente 10% dos dados gerados no mundo são estruturados.
- O problema reside em como extrair informações em dados não estruturados, sem um contexto específico ou com estruturas conhecidas.
- Analisar dados não estruturados requer um esforço maior pois são mais complexos, naturalmente. Requer ensinar a máquina (***machine learning***) a compreender, interpretar e calcular as características ou padrões que se deseja encontrar. E é na aprendizagem da máquina de transformar uma equação matemática em algo interpretável, replicável e com acurácia suficiente, que reside a complexidade de analisar os dados.

# Dados não estruturados

Nesse universo de análise de dados há uma máxima que diz que 80% dos dados tanto nos nossos dispositivos pessoais quanto em soluções empresariais, são não estruturados, ou seja, há uma predominância desses dados, gerados por todos nós, em nosso cotidiano, de análise complexa o que os leva a serem considerados ativos valiosos para as empresas.



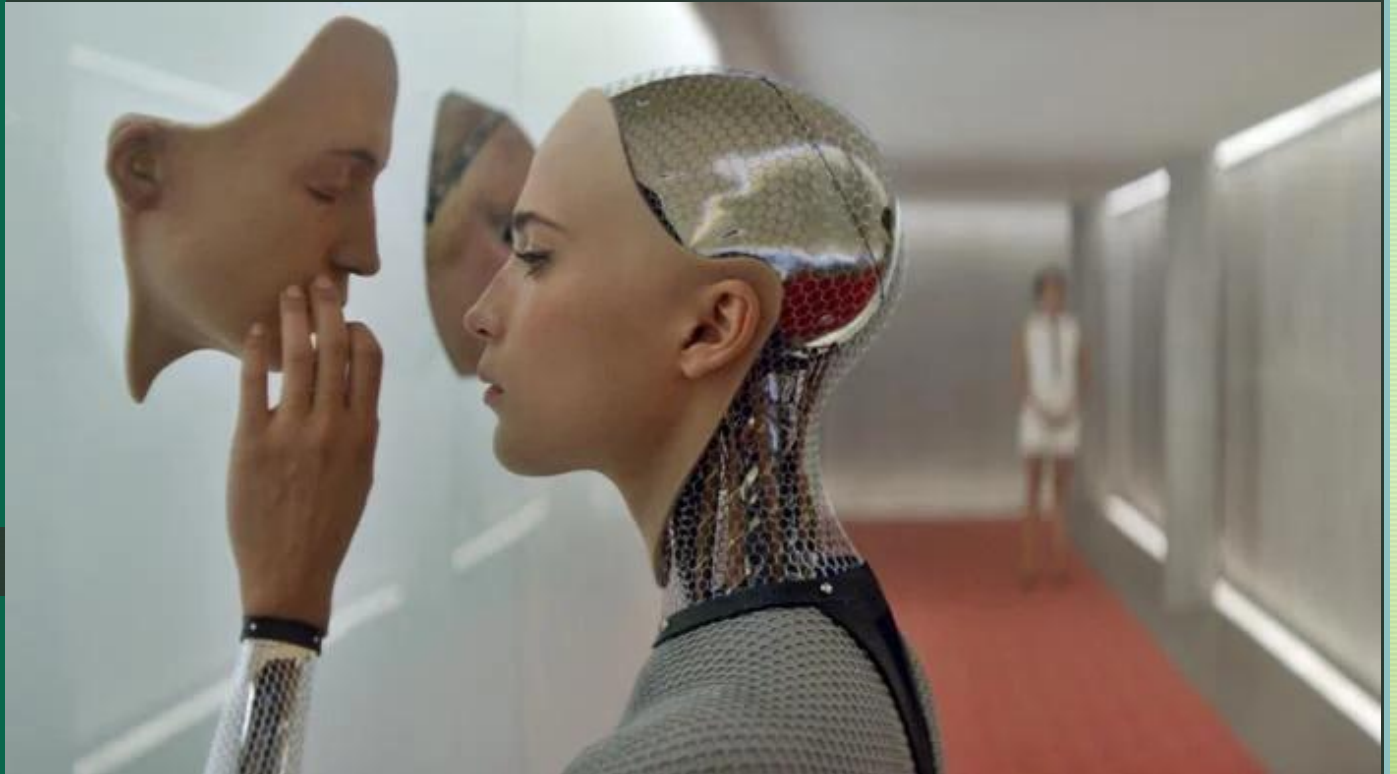
# Dados não estruturados

**Os dados não estruturados gerados por humanos são:**

1. Textos:
2. E\_mail:
3. Dados móveis:
4. Mídia social:
5. Site:
6. Mídia:
7. Comunicações:

**Os dados não estruturados gerados por máquinas são:**

1. Imagens de satélite:
2. Dados científicos:
3. Vigilância digital:



# Dados não estruturados

	Dados estruturados	Dados não estruturados
<b>Características</b>	<ul style="list-style-type: none"><li>- Modelo de dados pré-definido.</li><li>- Usualmente somente texto</li><li>- Fácil de consultar</li></ul>	<ul style="list-style-type: none"><li>- Modelo de dados não pré-definido.</li><li>- Podem ser texto, vídeo, som, imagens ou outros formatos.</li><li>- Dificuldade na pesquisa.</li></ul>
<b>Armazenamento</b>	<ul style="list-style-type: none"><li>- Banco de Dados Relacional</li><li>- <u>Data Warehouses</u></li></ul>	<ul style="list-style-type: none"><li>- Aplicações</li><li>- Banco de Dados <u>NoSQL</u></li><li>- <u>Data Warehouses</u></li><li>- <u>Data Lakes</u></li></ul>
<b>Gerados por</b>	<ul style="list-style-type: none"><li>- Humanos e máquinas</li></ul>	<ul style="list-style-type: none"><li>- Humanos e máquinas</li></ul>
<b>Aplicação típica</b>	<ul style="list-style-type: none"><li>- Sistema de reserva de passagens.</li><li>- Controle de estoque.</li><li>- Sistemas CRM.</li><li>- Sistemas ERP.</li></ul>	<ul style="list-style-type: none"><li>- Processamento de palavras.</li><li>- Software de apresentação.</li><li>- e-mail de clientes</li><li>- Ferramentas de visualização ou edição de mídias.</li></ul>
<b>Exemplos</b>	<ul style="list-style-type: none"><li>- Datas</li><li>- Numero de telefone</li><li>- Número do seguro social</li><li>- Número do cartão de crédito</li><li>- Nome do cliente</li><li>- Endereço</li><li>- Número e nome do produto</li><li>- Informações de transações</li></ul>	<ul style="list-style-type: none"><li>- Arquivos de texto</li><li>- Relatórios</li><li>- Mensagens de <u>e_mail</u></li><li>- Arquivos de áudio</li><li>- Arquivos de vídeo</li><li>- Imagens</li><li>- Imagens de vigilância</li></ul>



# NoSQL

- Devido as cada vez mais frequentes demandas por **escalabilidade** e, também pela natureza estruturada do Modelo Relacional, os desenvolvedores começaram a perceber as dificuldades em se organizar os dados desse modelo em um sistema distribuído trabalhando com particionamento de dados.
- Nesse ponto as soluções não-relacionais ganham foco.
- Uma emergente e popular classe de bancos de dados não-estruturados chamados *not only SQL* – NoSQL.

## Bancos de dados não relacionais - características:

1. **Escalabilidade:** capacidade de gravar dados em múltiplos armazéns de dados simultaneamente sem levar em conta a limitação física da infraestrutura.
2. **Modelo de consulta e dados:** usam frameworks especializadas para armazenar dados com um conjunto de APIs para acessar os dados de maneira inteligente.
3. **Design de persistência:** Devido a alta velocidade, variedade e volume do *big data*, esses bancos de dados usam diferentes mecanismos para a persistência de dados.
4. **Diversidade de interface:** a maioria suporta APIs RESTful como o ponto de partida de interface, além de oferecer uma variedade de mecanismos de conexão para programadores e administradores incluindo ferramentas de análise, relatórios e visualizações.
5. **Consistência eventual:** enquanto os RDBMS usam ACID como mecanismo de garantia de consistência nos dados, bancos não relacionais usam BASE (*Basically Available, Soft state, and Eventual Consistency*). Destes a Consistência Eventual é mais importante porque é responsável pela resolução de conflitos quando dados estão em movimento entre nós de uma aplicação distribuída.

# NoSQL

## Orientação Agregada

- A **orientação agregada** surge da percepção de que o usuário deseja, frequentemente, trabalhar com dados na forma de unidades com uma estrutura mais complexa do que um conjunto de linhas.
- Tipo um registro mais complexo que permita que listas e outras estruturas de dados sejam aninhadas dentro dele. Essa característica pode ser observada em bancos de dados chave/valor, de documento e família de colunas.
- Como não existe um consenso para identificar esse tipo de registro, vamos chamá-lo de **agregado** que vem do *Domain-Driven Design* (Projeto Orientado a Domínio), que considera o agregado como um conjunto de objetos relacionados que é tratado como uma unidade. Isso permite definir a unidade de manipulação e de consistência desse dado.

# NoSQL Agregados

## Banco de dados de par chave/valor

São de longe os bancos de dados mais simples. Não requerem um esquema como os RDBMS o que oferece grande flexibilidade e escalabilidade. Não oferece suporte as propriedades ACID e por isso requerem que os desenvolvedores pensem em posicionamento de dados, replicação e tolerância a falhas, uma vez que não são expressamente controladas pela tecnologia. Os dados não são tipados, e a maior parte são armazenados como *string*. O agregado é opaco, sendo apenas um amontoado de bits sem significado, porém, justamente essa opacidade permite armazenar qualquer coisa no agregado, porém o acesso é dependente da sua chave.



# NoSQL Agregados

## Banco de dados de par chave/valor

Buscando diminuir a opacidade sobre os seus dados, podem permitir estruturas de dados, como faz o Riak, através do qual é possível adicionar metadados a agregados para indexação e conexões entre interagregados. Já o Redis permite que o agregado seja dividido em listas e conjuntos. Também poderá suportar consultas integrando ferramentas de pesquisa como o Solr.

Exemplo desse tipo de banco de dados: **Riak**, *Redis*, *Mencached DB*, *Berkeley DB*, *HamsterDB*, *AmazonDynamoDB* (não é open source), *ProjectVoldemort* (open source do *AmazonDynamoDB*).

# NoSQL Agregados

## Banco de dados orientado a documento

Existem dois tipos de banco de dados de documento. O **primeiro** é frequentemente descrito como um repositório de conteúdo de estilo de documento completo (arquivos Word, páginas Web completas). O segundo é um banco de dados para armazenar componentes de documentos para o armazenamento permanente como uma entidade estática ou para montagem dinâmica das partes do documento. A estrutura do documento e suas partes são fornecido por *JavaScript Object Notation* (JSON) and/or *Binary JSON* (BSON).

Banco de dados de documento são mais comumente utilizados quando você tem que produzir muitos relatórios que precisam ser montados dinamicamente a partir de elementos que mudam frequentemente. Um bom exemplo são o preenchimento completo dos documentos da saúde, onde a composição do conteúdo irá variar em função do perfil do sujeito (idade, residência, nível de renda), plano de saúde e elegibilidade de programas governamentais.

# NoSQL Agregados

## Banco de dados orientado a documento

Nesse banco de dados é possível ver uma estrutura no agregado o que impõe limites sobre o que podemos inserir nele, definindo quais as estruturas e os tipos permitidos. O que torna esse banco mais restritivo proporciona maior flexibilidade de acesso. Diferente do Chave/Valor cujo acesso só pode ser feito pela chave, no banco de dados de documento pode-se submeter consultas baseadas nos campos do agregado, podendo recuperar parte do agregado em vez dele todo e ainda com a possibilidade de criar índices baseando-se no conteúdo do agregado.

Exemplo desse tipo de banco de dados: *CouchDB*, ***MongoDB***, *Terrastore*, *OrientDB*, *RavenDB* e o *LotusNotes*.

# NoSQL Agregados

## Banco de dados orientado a colunas

Banco de dados relacional é orientado a linha uma vez que os dados em cada linha de uma tabela são armazenados juntos. Em um banco de dados orientado a colunas os dados são armazenados em linhas. Embora possa parecer trivial, essa é característica subjacente mais importante dos banco de dados orientado a colunas. É muito fácil adicionar colunas e elas podem ser adicionadas linha a linha, oferecendo grande flexibilidade, performance e escalabilidade. Quando você tem volume e variabilidade de dados, você pode usar esse tipo de banco de dados.

Exemplo desse tipo de banco de dados são: **Cassandra**, *HBase*, *Hypertable* e *AmazonDynamoDB*, *BigTable*.



## Banco de dados baseado em grafos

A estrutura fundamental para banco de dados de grafos é chamada de relacionamento de no (*node-relationship*), que é mais utilizada quando você deve lidar com dados altamente conectados. Nodes e relacionamentos suportam propriedades, um par de chave-valor onde o dado é armazenado. O banco de dados é navegado seguindo os relacionamentos. Esse tipo de armazenamento e navegação não é possível num RDBMS devido a rígida estrutura da tabelas e a inabilidade de seguir as conexões entre os dados.

Exemplo desse tipo de banco de dados são: **Neo4J**, *InfoGrid*, *FlockDB*, *HyperGraphDB*, *InfiniteGraph*, *OrientDB*.

## Banco de dados Espacial

Interage-se com dados espaciais todo dia. Se você usa um smartphone ou *Global Positioning System* (GPS) para obter instruções sobre um lugar específico, ou se você procura pela localização de um restaurante próximo a um endereço físico ou marco, você está usando aplicações com dados espaciais. Os próprios dados espaciais são padronizados pelos esforços do *Open Geospatial Consortium* (OGC) que estabelece o *OpenGIS* (*Geographic Information System*) e uma série de outros padrões para dados espaciais.

Isso é importante, porque os bancos de dados espaciais são implementados do padrão OGC e, uma empresa pode ter necessidades específicas que podem (ou não) ser atendidas pelo padrão. Um banco de dados espacial tornou-se importante quando as empresas começam a usar várias dimensões diferentes de dados para tomar decisões. Por exemplo, um meteorologista fazendo pesquisa pode querer armazenar e analisar dados de um furacão, incluindo temperatura, velocidade do vento e humidade do ar, e o modelar esses resultados em 3 dimensões (3D).

## Banco de dados Espacial

Banco de dados espaciais armazenam dados em objetos de 2, 2,5 e 3 dimensões (2D, 2,5D e 3D respectivamente). Estamos familiarizados com dados em 2D (cumprimento e largura) e em 3D (cumprimento, largura e profundidade). Não estamos familiarizados com uma dimensão 2,5D, pois são tipos especiais de dados espaciais. São objetos 2D com uma elevação extra de 'metade' da dimensão. A maioria dos bancos de dados espaciais 2.5D contém informação de mapeamento e são frequentemente referenciados como *Geographic Information System* (GIS).

Um elemento atômico de um banco de dados espacial são linhas, pontos e polígonos.

## Persistência Poliglota

Esse termo quer definir um conjunto de aplicativos que usam várias tecnologias de banco de dados, e esse é o resultado mais provável do seu planejamento de implementação em *Big Data*. Vai ser difícil escolher um estilo de persistência, não importa quão estreita seja sua abordagem para *Big Data*. Um banco de dados de persistência poliglota é usado quando é necessário resolver problemas complexos quebrando esse problema em segmentos e aplicando diferentes modelos de banco de dados



# Bibliografia

- PRAMOD, J. Sadalage, FOWLER, Martin. **NoSQL**. Um Guia Conciso para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec, 2013.
- HURWITZ, Judith, NUGENT, Alan, HALPER, Dr. Fern, KAUFMAN, Marcia. **Big Data for dummies**. New Jersey: John Wiley & Sons, 2013.
- SABHARWAL, Navin, GUPTA EDWARD, Shakuntala. **Big Data, NoSQL. Architecting MongoDB**. Laverne, TN: 2015.
- [https://pt.wikibooks.org/wiki/SQL/Dados\\_Estruturados,\\_Semi-Estruturados\\_e\\_N%C3%A3o\\_Estruturados](https://pt.wikibooks.org/wiki/SQL/Dados_Estruturados,_Semi-Estruturados_e_N%C3%A3o_Estruturados)
- <https://www.digitalhouse.com/br/blog/diferenca-dados-estruturados-e-nao-estruturados>
- <http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/>
- <https://www.datamation.com/big-data/structured-vs-unstructured-data.html>
- <https://document.onl/documents/no-sql-558930b5f1ecc.html>
-