

# **Curso de Ciência da Computação**

## **Banco de Dados 2** **Aula 07**

**Prof. Dr. rer. nat. Eros Comunello**



**UNIVALI**

# Banco de Dados Não Relacional

O que é NoSQL?

Vantagens e Características NoSQL

Tipos de BD NoSQL

Exemplo de BD NoSQL

# O que é NoSQL?

- ❑ O termo NoSQL foi primeiramente utilizado em 1998 como o nome de um banco de dados não relacional de código aberto;
- ❑ Autor, Carlo Strozzi, alega que o movimento NoSQL:
  - ❑ "é completamente distinto do modelo relacional e portanto deveria ser mais apropriadamente chamado **NoREL**" ou algo que produzisse o mesmo efeito".
- ❑ O nome era uma tentativa de descrever o surgimento de um número crescente de BD não relacionais e fazia uma referência ao esquema de atribuição de nomes dos bancos de dados relacionais como **MySQL**, **PostgreSQL**, etc.

# O que é NoSQL?

- ❑ Bancos de dados relacionais escalam, mas **quanto maior o tamanho, mais custoso** se torna essa escalabilidade:
  - ❑ Custo de novas máquinas; e
  - ❑ Aumento de especialistas nos bancos de dados utilizados.
- ❑ Os não relacionais, permitem uma escalabilidade mais barata e menos trabalhosa;
  - ❑ Não exigem máquinas extremamente poderosas; e
  - ❑ Facilidade de manutenção permite que um número menor de profissionais seja necessário.

# O que é NoSQL?

- ❑ Os BD NoSQL, vão ficando mais populares entre as grandes empresas pois reúnem as características de **poder trabalhar com dados semiestruturados** ou crus vindos de diversas origens (arquivos de log, websites, arquivos multimídia, etc...).

# Vantagens NoSQL

O NoSQL apresenta três vantagens fundamentais:

- ❑ **Volume:** Este ambiente consegue armazenar de forma sustentada o aumento exponencial de dados gerados;
- ❑ **Velocidade:** O objetivo das empresas é cada vez mais reduzir o tempo necessário com a criação, organização e acesso aos dados, sendo este ambiente eficiente comparando com os existentes anteriormente (como as bases de dados relacionais);
- ❑ **Variedade:** Os dados guardados podem estar em diferentes formatos: desde dados em estruturas tabulares a ficheiros XML ou até ficheiro binários, como fotografias ou música.

# Características NoSQL

## ❑ Utilização do processamento paralelo para processamento das informações:

- ❑ Para se atingir uma performance razoável no processamento de grandes volumes de dados, é mais eficiente dividir a tarefa em várias outras menores;
- ❑ Podem ser executadas ao mesmo tempo, distribuindo essas tarefas;
- ❑ Os sistemas precisam atingir um alto grau de maturidade no processamento paralelo.
- ❑ O uso de muitos processadores baratos oferece **melhor performance**;
- ❑ Se torna também uma solução economicamente interessante, pois é possível **escalar horizontalmente** apenas adicionando hardware e não limita a empresa a poucos fornecedores de hardware mais poderoso.

# Características NoSQL

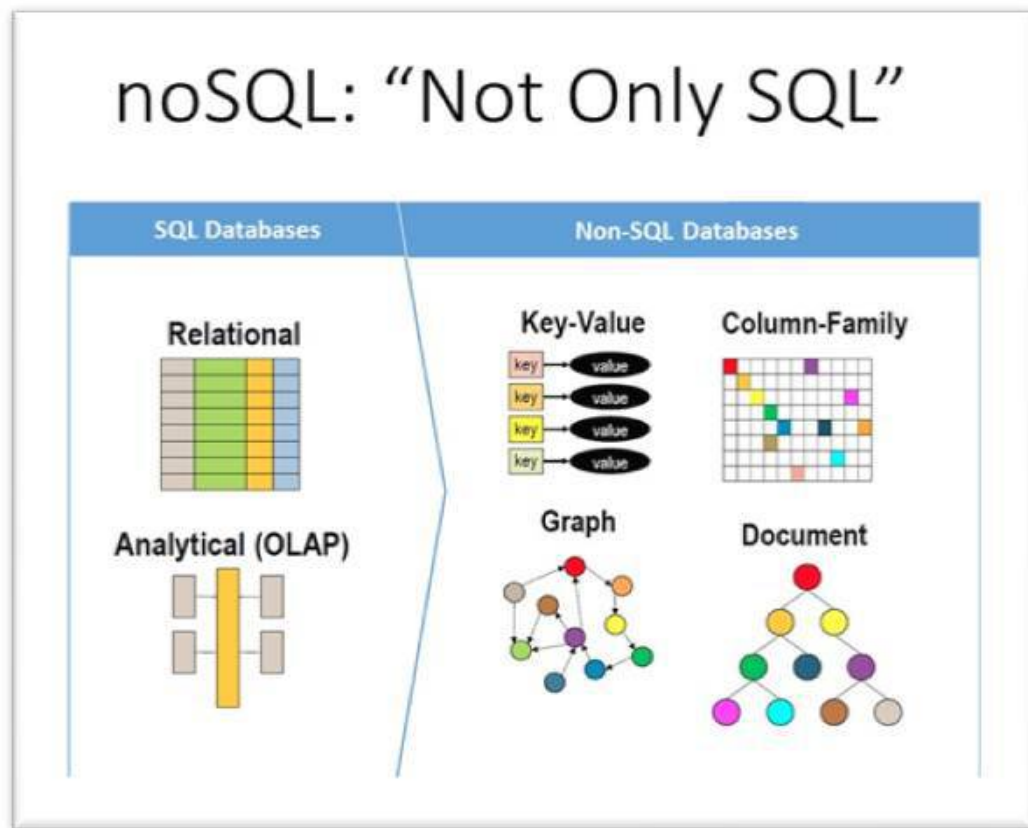
## ❑ Distribuição em escala global:

- ❑ Para atender seus usuários de forma eficiente, algumas empresas utilizam vários data centers, localizados em diversas partes do país ou do mundo;
- ❑ Com isso, uma série de questões sobre **disponibilidade** e **performance** são levantadas ao construir os sistemas;
- ❑ A distribuição deles combinada com o hardware barato, impõe ao sistema a necessidade de ser **robusto** o suficiente para **tolerar** falhas constantes e imprevisíveis;
- ❑ Essas falhas podem ser de **hardware** ou da **infraestrutura** do lugar onde o data center se encontra.



# Tipos de BD NoSQL

- ❑ Orientada a documentos
- ❑ Key-Value
- ❑ Família de colunas
- ❑ Grafos



# Tipos de BD NoSQL

## ❑ Bancos de dados orientados a documentos:

- ❑ Os documentos dos bancos dessa categoria, são **coleções de atributos e valores**, onde um atributo pode ser multivalorado;
- ❑ Em geral, eles **não possuem esquema**, ou seja, os documentos armazenados não precisam possuir estrutura em comum;
- ❑ Essa característica faz deles boas opções para o **armazenamento de dados semiestruturados**;
- ❑ Alguns bancos que utilizam esse padrão são: **MongoDb**, CouchDB, RavenDb, etc.

```
{
  "id": 49,
  "Pais": "Alemanha",
  "Regiao": "Europa",
  "Populacao":
  "PrincipaisCidades": [
    {
      "NomeCidade": "Berlin",
      "Populacao": 3610156,
    },
    {
      "NomeCidade": "Hamburg",
      "Populacao": 1746342,
    }
  ]
}
```

# Tipos de BD NoSQL

## ❑ Banco de dados que trabalham no esquema chave/valor (Key/Value):

- ❑ Sistemas distribuídos nessa categoria, são também conhecidos como **tabelas de hash distribuídas**,
  - ❑ armazenam objetos **indexados por chaves**, e
  - ❑ possibilitam a busca por esses objetos a partir de suas chaves.
- ❑ Alguns bancos que utilizam esse padrão são: DynamoDb, Couchbase, Riak, Azure Table Storage, Redis, Tokyo Cabinet, Berkeley DB, etc...

| Chave           | Valor  |
|-----------------|--------|
| carro_3345_cor  | preto  |
| carro_3345_pneu | 17     |
| carro_3365_cor  | branco |
| carro_3365_pneu | 15     |
| carro_4560_peso | 1215   |
| carro_4715_ano  | 2016   |

# Tipos de BD NoSQL

## ❑ Bancos de dados de famílias de colunas:

- ❑ Bancos relacionais normalmente guardam os registros das tabelas **contiguamente** no disco;
  - ❑ Por exemplo, caso se queira guardar id, nome e endereço de usuários em um sistema de cadastro, os registros seriam:
    - ❑ Id1, Nome1, Endereço1;
    - ❑ Id2, Nome2, Endereço2.
- ❑ Essa estrutura torna a **escrita muito rápida**, pois todos os dados de um registro são colocados no disco com uma única escrita no banco;
- ❑ Essa estrutura também é eficiente caso se queira ler registros inteiros;
- ❑ Mas para situações onde se quer ler algumas poucas colunas de muitos registros, essa estrutura é pouco eficiente, pois muitos blocos do disco terão de ser lidos.

# Tipos de BD NoSQL

## ❑ Bancos de dados de famílias de colunas:

- ❑ Para esses casos onde se quer otimizar a leitura de dados estruturados, bancos de dados de famílias de colunas são mais interessantes, pois eles guardam os dados contiguamente por coluna.
- ❑ O exemplo anterior em um banco de dados dessa categoria ficaria:
  - ❑ Id1, Id2;
  - ❑ Nome1, Nome2;
  - ❑ Endereço1, Endereço2.

| Tabela                          |                                 |                                 |                                 |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Família de coluna 1             |                                 | Família de coluna 2             | Família de coluna 3             |
| Coluna 1                        | Coluna 2                        | Coluna 3                        | Coluna 4                        |
| #1 {Chave: Valor, Chave: Valor} | #1 {Chave: Valor, Chave: Valor} | #1 {Chave: Valor, Chave: Valor} | #1 {Chave: Valor, Chave: Valor} |
| #2 {Chave: Valor, Chave: Valor} | #2 {Chave: Valor, Chave: Valor} | #2 {Chave: Valor, Chave: Valor} | #2 {Chave: Valor, Chave: Valor} |

# Tipos de BD NoSQL

## ❑ Bancos de dados de famílias de colunas:

- ❑ Desvantagem de um banco de dados de famílias de colunas:
  - ❑ A escrita de um novo registro é bem mais custosa do que em um banco de dados tradicional;
- ❑ Num primeiro momento, os bancos tradicionais são mais adequados a processamento de transações online (OLTP);
- ❑ Já os bancos de dados de famílias de colunas são mais interessantes para processamento analítico online (OLAP);
- ❑ O Bigtable é uma implementação da Google dessa categoria de bancos de dados;
- ❑ Outros bancos de dados que são orientados a coluna: Hadoop, Cassandra, Hypertable, Amazon SimpleDB, etc.

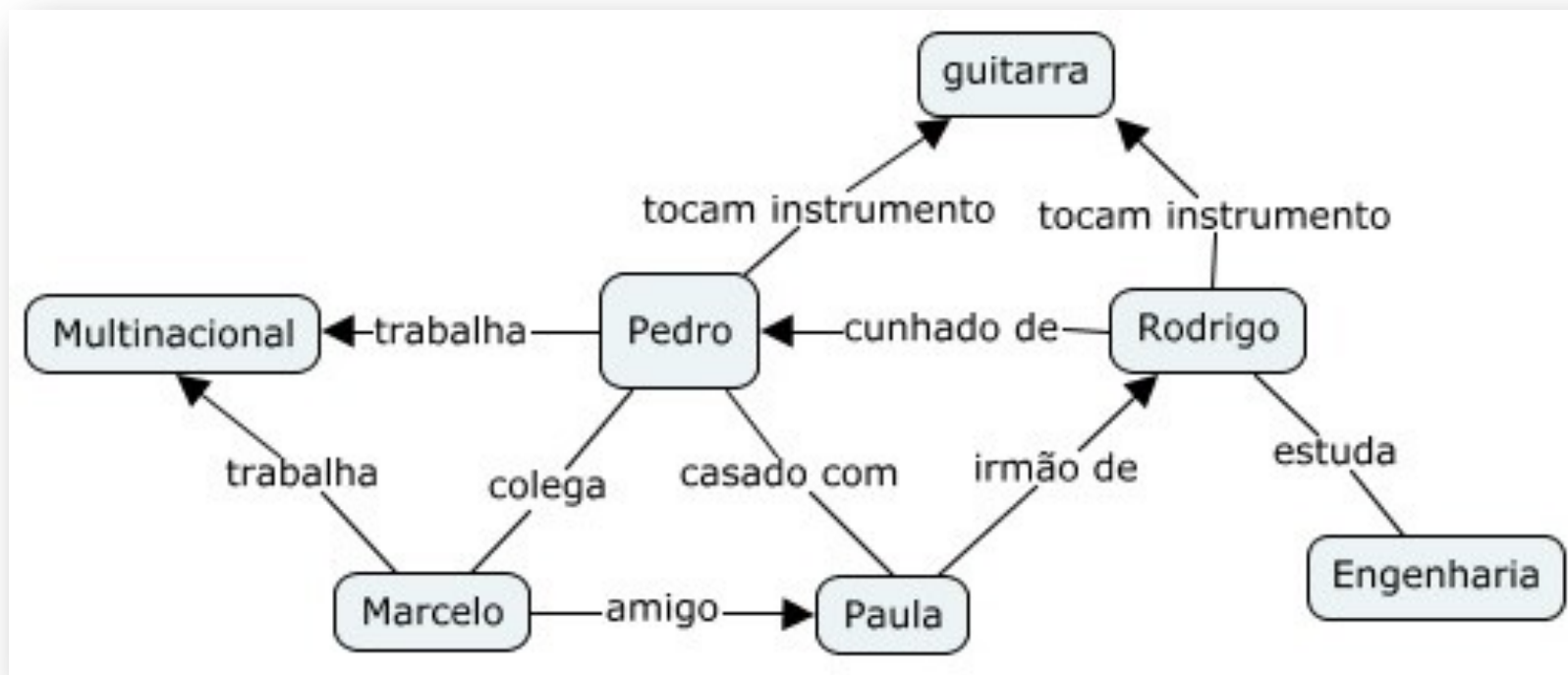
# Tipos de BD NoSQL

## ❑ Bancos de dados de grafos:

- ❑ Diferentemente de outros tipos de bancos de dados NoSQL, esse está diretamente relacionado a um modelo de dados estabelecido, o modelo de grafos;
- ❑ A ideia desse modelo é representar os dados e / ou o esquema dos dados como grafos dirigidos, ou como estruturas que generalizem a noção de grafos;
- ❑ Ele é mais interessante que outros quando informações sobre a interconectividade ou a topologia dos dados são mais importantes, ou tão importante quantos os dados;
- ❑ O modelo orientado a grafos possui três componentes básicos:
  - ❑ os **nós** (são os vértices do grafo),
  - ❑ os **relacionamentos** (são as arestas) e
  - ❑ as **propriedades** (ou atributos) dos nós e relacionamentos.

# Tipos de BD NoSQL

## ❑ Bancos de dados de grafos:





# Tipos de BD NoSQL

## ❑ Bancos de dados de grafos:

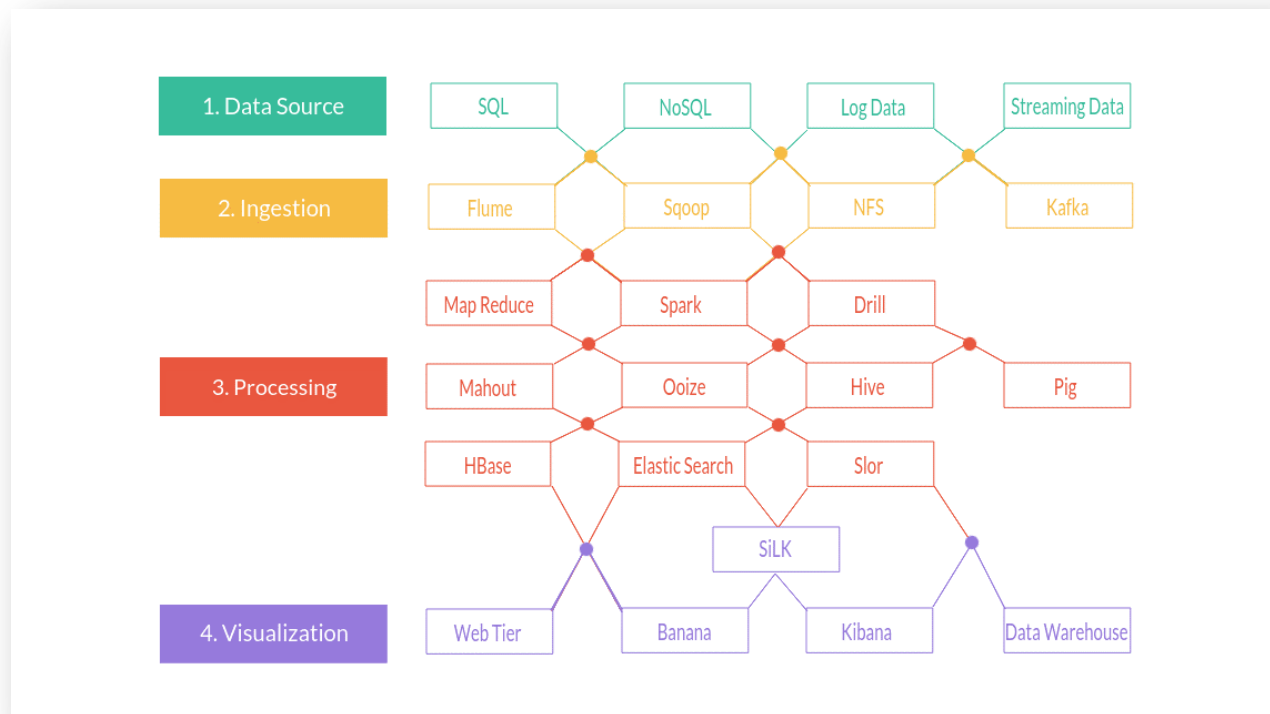
- ❑ Neste caso, o banco de dados pode ser visto como um multigrafo rotulado e direcionado, onde cada par de nós pode ser conectado por mais de uma aresta;
- ❑ Um exemplo pode ser: “Quais cidades foram visitadas anteriormente (seja residindo ou viajando) por pessoas que viajaram para Floripa?”
- ❑ No modelo relacional esta consulta poderia ser muito complexa devido a necessidade de múltiplas junções, o que poderia acarretar uma diminuição no desempenho da aplicação;
- ❑ Porém, por meio dos relacionamentos inerentes aos grafos, estas consultas tornam-se mais simples e diretas;
- ❑ Alguns bancos que utilizam esse padrão são: Neo4J, InfiniteGraph, InforGrid, HyperGraphDB, OrientedDB, GraphBase, etc.,

# Exemplo BD NoSQL - Hadoop

- ❑ Hadoop, que é um framework paralelo no processamento de dados que tem sido usado para redução de mapas e Jobs;
- ❑ Diferentemente do Spark que armazena os dados em memória, o Hadoop armazena em disco e utiliza a técnica de replicação para garantir tolerância a falhas;
- ❑ Funciona em um único servidor até um cluster com milhares de máquinas;
- ❑ Foco em detectar e tratar falhas na camada de aplicação, fornecendo um serviço de alta disponibilidade baseado em um grid de computadores.

# Exemplo BD NoSQL - Hadoop

- ❑ No entanto o Hadoop possui uma grande latência para as consultas;
- ❑ Há dois componentes principais do Hadoop: Hadoop Distributed File System (HDFS) e o Mapreduce.



# Links complementares

- ❑ <https://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044>
- ❑ <https://micreiros.com/tipos-de-bancos-de-dados-nosql/>