

# Banco de dados NoSQL

Programação Orientada a Objetos

# NoSQL

- NoSQL é um termo genérico
- Bancos de dados modelados de forma diferente das relações tabulares usadas nos bancos de dados relacionais
- Algumas aplicações: Facebook, Google e Amazon.com
- São cada vez mais usados em big data e aplicações web de tempo real
- Não substitui o relacional

# Características

- Simplicidade de projeto
- Escalonamento horizontal mais simples para **clusters de máquinas**
  - Onde seria problema para bancos de dados relacionais
- Controle mais refinado sobre a disponibilidade
- Estruturas de dados
  - Chave-valor
  - Coluna larga
  - Grafo ou documento
  - *outros*

- A adequação particular de um determinado banco de dados NoSQL **depende do problema que ele deve resolver**
- Algumas barreiras
  - Não usa linguagem SQL
  - Não realiza junções entre tabelas (como os relacionais)
  - Falta de interfaces padronizadas (várias soluções)
  - Investimento anterior em relacional

- A maioria dos bancos de dados NoSQL oferece um conceito de **Consistência Posterior** em que as alterações do banco de dados são **propagadas para todos os nós "posteriormente"** (geralmente em milissegundos)
- De forma que as consultas de dados **podem não retornar dados atualizados imediatamente**
- Problema conhecido como **leitura obsoleta**

- Alguns sistemas NoSQL podem exibir gravações perdidas e outras formas de perda de dados
- Para o processamento de transações distribuídas em vários bancos de dados, a consistência de dados é um desafio ainda maior, que é difícil tanto para bancos de dados relacionais quanto NoSQL
- O intuito não é eliminar bancos de dados relacionais, mas oferecer uma alternativa

# Desafios de armazenamento de dados

- Grande volume de informação
- Informações de diversos tipos, em forma
  - Estruturada (ex.: planilha)
  - Não estruturada (ex.: documento de texto)
- Constância de mudanças no software
- Custos de hardware para escalabilidade
- Cenários de redução de custos

# Algumas características do NoSQL

- Independente de esquema
  - Armazenamento não possui uma organização rígida
  - Conjunto de bibliotecas para acesso mais simples
- Livre de junções de tabelas
  - Não são indicadas as junções
  - Mesmo que possíveis em alguns modelos
  - Torna a utilização menos complexa



# Algumas características do NoSQL

- Relativamente fácil crescimento horizontal
  - A maioria das soluções NoSQL são escaláveis
    - Fácil anexar mais computadores
    - Não necessariamente computadores mais caros
- Hardware mais simples
  - Hardware barato pela própria arquitetura da solução

# Algumas características do NoSQL

- Autocontido
  - Cada computador que funciona como armazenamento é independente
  - Levando em diversos casos, cópias das bases de dados
- Controle de usuário
  - Necessita de poucos elementos de controle de
    - Usuário
    - Transação
    - Concorrência

# Modelos de dados para NoSQL

- Quatro modelos mais comuns para organizar a informação dentro de um NoSQL
  - Chave-valor (Key-value)
  - Documento (Document stores)
  - Orientado a colunas
  - Grafo
- O melhor modelo depende da aplicação

# Descrição dos modelos

- **Coluna:** usa tabelas, linhas, e colunas
  - Diferentemente de um banco de dados relacional, os nomes e o formato das colunas podem variar de linha para linha na mesma tabela
- **Documento:** armazena informação em documentos (JSON, BSON, XML, etc.)
- **Chave-valor:** usa arrays associativos (dicionários ou tabelas hash)
- **Grafo:** usa estruturas de grafos

# Modelo *key-value*

- Estrutura mais simples
- Cada conjunto de dados possui apenas uma chave principal

Dados	
Key	Value
1	Aluno: João Curso: Ciência da Computação Conclusão: 2020
2	Aluno: Pedro Curso: Ciência da Computação Conclusão: 2022 Origem: Transferência interna

não é necessário  
padronizar os  
dados

# Modelo *key-value*

- Exemplo: Amazon SimpleDB
- Algumas vantagens:
  - **Flexibilidade na modelagem:** os *Values* (dados) podem ser inseridos e alterados, relacionados a cada registro
  - **Operação simples:** é necessário apenas buscar a *Key* e já se tem acesso aos dados
  - **Escalabilidade e disponibilidade:** pela sua simples estrutura, as técnicas computacionais para aumentar a capacidade e redundância são facilmente aplicadas

# Modelo *Document stores*

- Diversos arquivos formatados para fazer o armazenamento
  - Extensible Markup Language (XML)
  - JavaScript Object Notation (JSON)

exemplo JSON

```
{  
  '_id' : 1,  
  'name' : { 'Nome' : Pedro, 'Sobrenome' : 'Silva' },  
  'curso' : [ 'Ciência da Computação',  
              'Análise e desenvolvimento de sistemas' ]  
}
```

# Modelo *document stores*

- Exemplo: [MongoDB](#)
- Algumas vantagens:
  - **Capacidade organizacional:** junto do JSON é possível armazenar arquivos e outros tipos de dados binários
  - **Forma organizada:** existem grande quantidade de informação já colocada em JSON, dessa forma a migração é algo menos complicado
  - **Informação variada:** pela sua estrutura é possível ter dados em que alguns elementos do registro estão disponíveis e outro não (null).



# Modelo orientado a colunas

- Semelhante ao modelo de *Document stores*
- Usa estruturas separadas para juntar informações pertinentes a uma entidade
- Uso relacionado a sistemas com grande volume de dados já estruturados e com necessidade de processamento rápido

Família de super coluna: Clientes

RowID: 1

Super Coluna: Nome

Nome: João

Sobrenome: Silva

Super Coluna: Curso

Nome: Ciência da Computação

Conclusão: 2020

RowID: 2

Super Coluna: Nome

Nome: Pedro

Sobrenome: Silva

Super Coluna: Curso

Nome: Engenharia de Computação

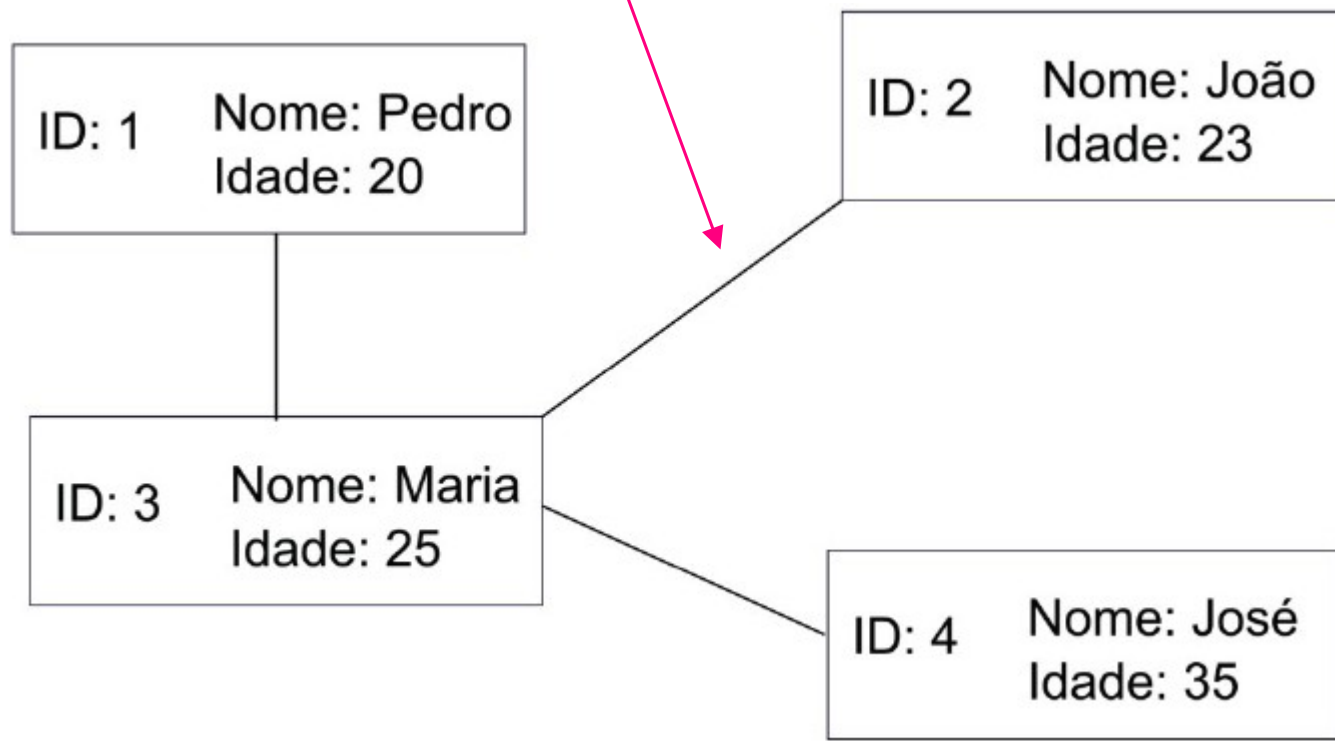
Conclusão: 2020

# Modelo orientado a colunas

- Exemplo: [Apache Cassandra](#)
- Usado em
  - Netflix
  - Organização Europeia para Pesquisa Nuclear (CERN)

# Modelo orientado a grafos

- A informação pode ter uma estrutura inicial de *key-value*
- Existe algum tipo de relacionamento entre cada uma das entradas dos dados



# Algumas implementações

- Há várias abordagens para se classificar bancos de dados NoSQL, sendo que algumas delas se sobrepõem
  - **Coluna:** Accumulo, Cassandra, Druid, HBase, Vertica
  - **Documento:** Apache CouchDB, Elasticsearch, ArangoDB, BaseX, Clusterpoint, Couchbase, Cosmos DB, IBM Domino, MarkLogic, MongoDB, OrientDB, Qizx, RethinkDB
  - **Chave-valor:** Aerospike, Apache Ignite, ArangoDB, Couchbase, Dynamo, FoundationDB, InfinityDB, MemcacheDB, MUMPS, Oracle NoSQL Database, etc.
  - **Grafo:** AllegroGraph, ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso
  - **Multi-modelo:** Apache Ignite, ArangoDB, Couchbase, FoundationDB, InfinityDB, MarkLogic, OrientDB

- Exemplo de como armazenar os dados no MongoDB

JSON

```
{  
  '_id' : 1,  
  'nome' : { 'Nome' : João, 'Sobrenome' : 'Silva' },  
  'curso' : [ 'Ciência da computação'],  
  'notaMedia' : [ 9.8]  
}
```