



Minicurso: Projeto de Bancos de Dados NoSQL

**Angelo Augusto Frozza
Geomar André Schreiner
Ronaldo dos Santos Mello**

GBD/INE/CTC/UFSC



Quem são os autores?



Quem são os autores?



Geomar André Schneider - Doutor em Ciência da Computação (UFSC), professor na UNOESC e UFFS (Chapecó-SC). Área de atuação: Bds, Bds NoSQL e NewSQL, interoperabilidade e particionamento de dados.

Quem são os autores?



Angelo Augusto Frozza - Doutor em Ciência da Computação (UFSC), professor no IFC – Campus Camboriú. Área de atuação: BD, *data warehouse*, BD geográficos e modelagem de dados.

Quem são os autores?



Ronaldo dos Santos Mello – Doutor em Ciência da Computação (UFRGS), professor na UFSC, tutor do PET Computação e Coordenador do GBD. Área de atuação: modelagem de dados e restrições de integridade, integração e interoperabilidade de dados, BD NoSQL e NewSQL.

Objetivo



O minicurso possui um caráter teórico-prático, envolvendo a interação com **protótipos** desenvolvidos no âmbito do **Grupo de BD da UFSC** para o projeto de BDs NoSQL e **interfaces Web** para acesso e visualização de dados para dois BDs NoSQL: ***MongoDB*** e ***Neo4j***.

Organização

Parte 1:

BDs NoSQL (*Not Only SQL*) e seus principais modelos de dados (chave-valor, orientado a colunas, orientado a documentos e orientado a grafos)



Parte 2:

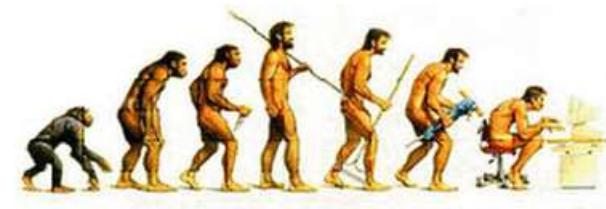
Uma metodologia para projeto lógico de BDs NoSQL a partir de um esquema conceitual definido através do modelo Entidade-Relacionamento (modelo ER)

Parte 3:

Implementação do projeto em sistemas de gerência de BDs (SGBDs) NoSQL

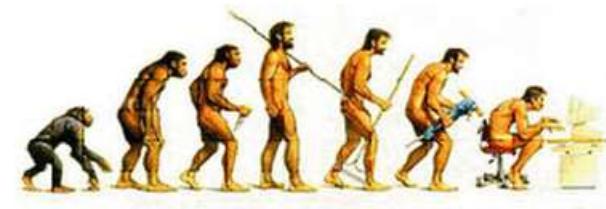
Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- Propriedades dos BDs NoSQL
- Modelos de dados NoSQL
 - Chave-valor
 - Orientado a documentos
 - Orientado a colunas
 - Orientado a grafos



Parte 1: BDs NoSQL e modelos de dados

- **Um pouco de história**
- Propriedades dos BDs NoSQL
- Modelos de dados NoSQL
 - Chave-valor
 - Orientado a documentos
 - Orientado a colunas
 - Orientado a grafos



Um pouco de história

O termo NoSQL

- **1998** - usado pela primeira vez por **Carlos Strozzi**
 - No entanto, fazia referência a um SGBD relacional que não implementava a linguagem SQL.
- **2009** - Evento organizado por **Johan Oskarsson**
 - Motivação: *BigTable (Google)* e *DynamoDB (Amazon)*
 - Apresentação de novos sistemas de banco de dados não relacionais, distribuídos e de código aberto:
 - ✗ *Voldemort, Cassandra, Dynomite, Hbase, Hypertable, CouchDB* e *MongoDB*.

Um pouco de história

O termo NoSQL

- Atualmente, entende-se que:
 - Nem todo problema de gerenciamento e análise de dados é melhor solucionado usando um SGBDR tradicional.



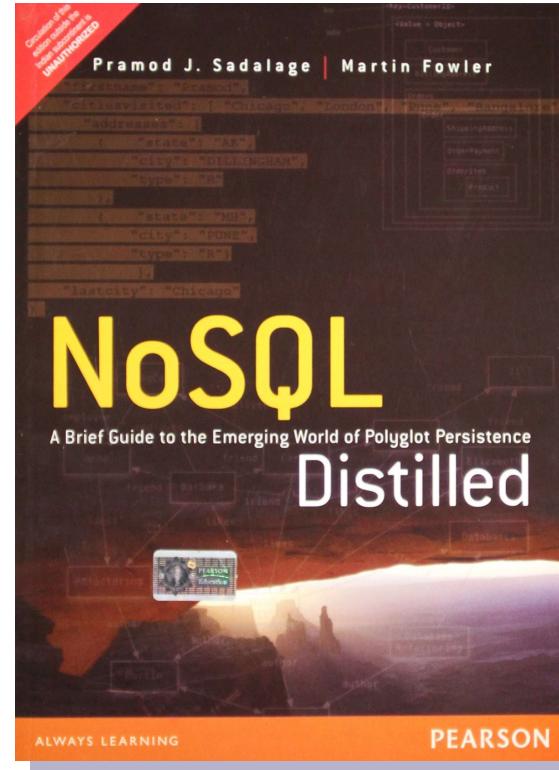
Um pouco de história

O termo NoSQL

NoSQL => “***Not only SQL***”
(SADALAGE e FOWLER, 2013)

• BDs NoSQL

- não seguem o modelo de dados relacional (modelo de tabelas)
- não pretendem substituir o modelo relacional
 - são uma opção quando é necessária uma maior flexibilidade na estruturação do banco.



Um pouco de história

O termo NoSQL

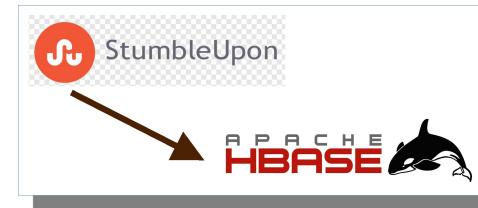
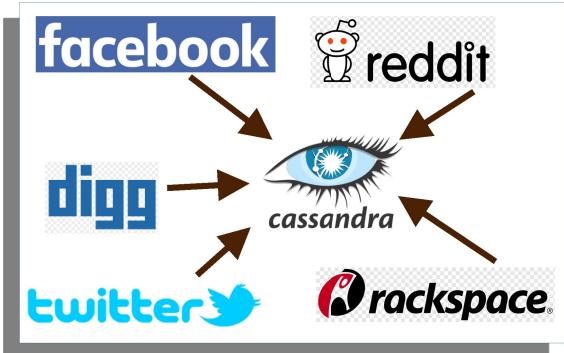
- De acordo com **Stonebreaker**, duas razões principais motivam a troca de um SGBDR por um sistema NoSQL:
 - ✓ Necessidade de um **desempenho** melhor
 - ✓ Necessidade de mais **flexibilidade**



STONEBREAKER, Michael. *SQL Databases vs. NoSQL Databases*. **Communications of the ACM**, v. 3, i. 4, April 2010, 9. p. 10-11.
<https://dl.acm.org/doi/10.1145/1721654.1721659>.

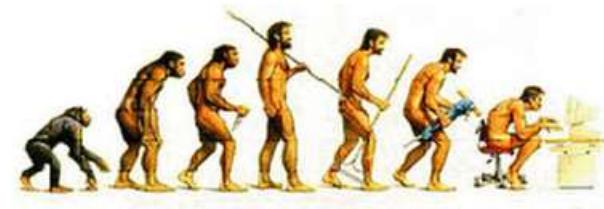
Um pouco de história

Quem usa NoSQL?



Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- **Propriedades dos BD NoSQL**
- Modelos de dados NoSQL
 - Chave-valor
 - Orientado a documentos
 - Orientado a colunas
 - Orientado a grafos



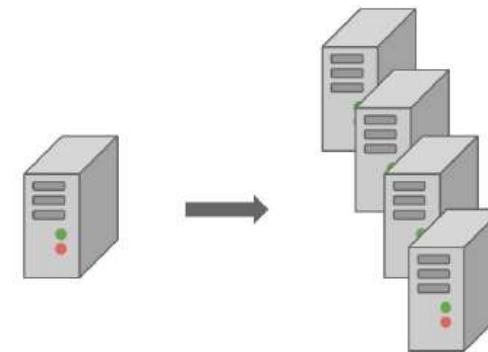
Propriedades dos BDs NoSQL

- Escalabilidade horizontal e armazenamento distribuído de dados complexos
- Interface de acesso simples
- Alta disponibilidade e relaxamento das propriedades ACID
- Esquemas flexíveis ou opcionais

Propriedades dos BDs NoSQL

- **Escalabilidade horizontal e armazenamento distribuído de dados complexos**

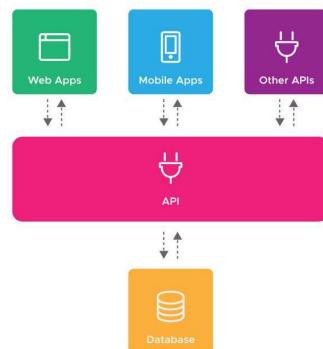
- ✓ Escalabilidade pelo **aumento no número de máquinas** disponíveis para o armazenamento e processamento de dados.
- ✓ É uma solução mais viável que a escalabilidade vertical.
- ✓ Requer que diversos processos (*threads*) de uma tarefa sejam criados e distribuídos (dificuldade nos BD relacionais).



Propriedades dos BDs NoSQL

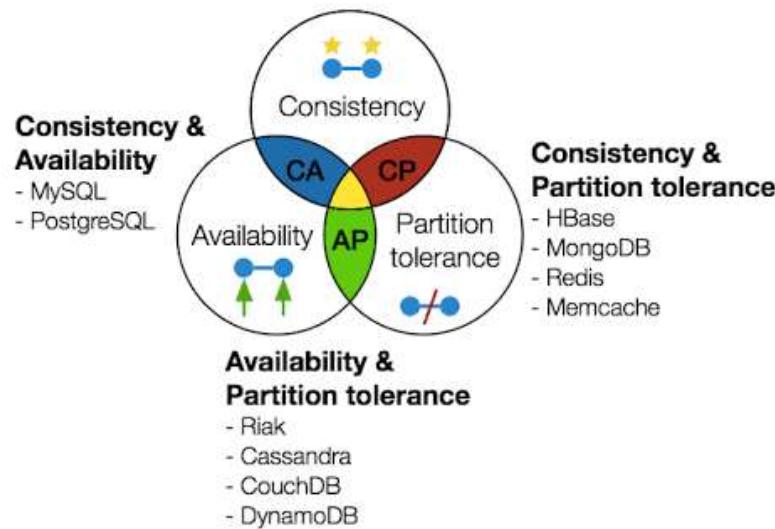
- **Interface de acesso simples**

- ✓ Geralmente utilizando **APIs** (*Application Programming Interfaces*) específicas com acesso baseado em chaves.
- ✓ O foco está na recuperação rápida e eficiente dos dados.



Propriedades dos BDs NoSQL

- Alta disponibilidade e relaxamento das propriedades ACID



- BD NoSQL seguem as propriedades **BASE**:

- ✓ **BA – Basically Available**: Disponibilidade é prioridade, mesmo com falhas parciais
- ✓ **S – Soft-State**: Não precisa ser consistente o tempo todo
- ✓ **E – Eventually Consistent**: Consistente em momento indeterminado.

Propriedades dos BDs NoSQL

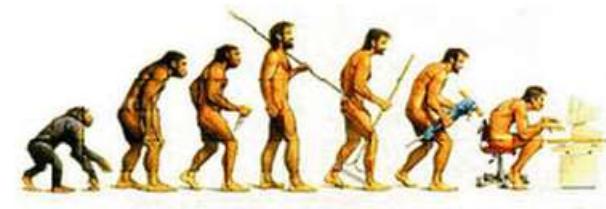
- **Esquemas flexíveis ou opcionais**

- BDs NoSQL são *schemaless*.
 - Quando esquemas existem, esses são flexíveis e permitem alterações.
 - Ser *schemaless* facilita a alta disponibilidade e alta escalabilidade, mas não garante a integridade dos dados.
- Modelos de dados NoSQL
 - Não suportam junções
 - A maioria não suporta integridade referencial e relacionamentos entre dados.

FROZZA, A. A. Uma abordagem para extração de esquemas de bancos de dados NoSQL com ênfase em dados geográficos.
Tese (Doutorado) - UFSC/PPGCC, Florianópolis, 2021.

Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- Propriedades dos BDs NoSQL
- **Modelos de dados NoSQL**
 - Chave-valor
 - Orientado a documentos
 - Orientado a colunas
 - Orientado a grafos



Modelos de Dados NoSQL

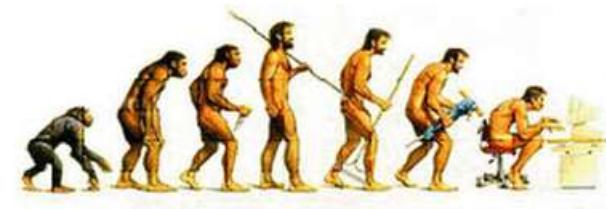
Um **modelo de dados** é o modelo por meio do qual percebemos e manipulamos os dados.

(SADALAGE e FOWLER, 2012)

- Ele descreve como interagimos com os dados no banco de dados.
- É diferente de um **modelo de armazenamento**, que descreve como o BD armazena e manipula os dados internamente.

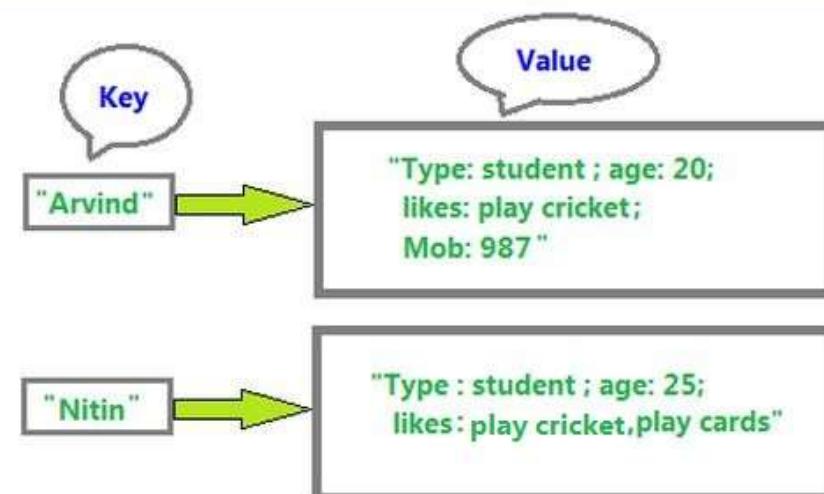
Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- Propriedades dos BDs NoSQL
- Modelos de dados NoSQL
 - **Chave-valor**
 - Orientado a documentos
 - Orientado a colunas
 - Orientado a grafos



Modelo Chave-Valor (key-value)

- ✓ Modelo simples similar a uma estrutura de indexação (p.ex. tabela *hash*)
 - Chave identifica um conteúdo mono ou multivalorado
- ✓ API simples
 - *get(key)*
 - *put(key, value)*
 - *delete(key)*
- ✓ Não suporta
 - Definição de esquemas
 - Relacionamentos entre dados
 - Linguagem de consulta

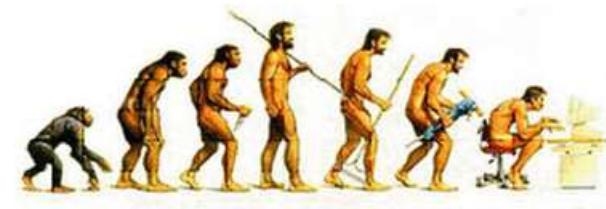


Modelo Chave-Valor (*key-value*)



Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- Propriedades dos BDs NoSQL
- Modelos de dados NoSQL
 - Chave-valor
 - **Orientado a documentos**
 - Orientado a colunas
 - Orientado a grafos

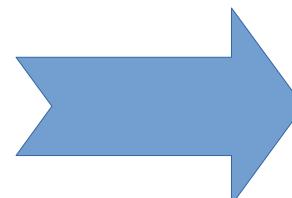
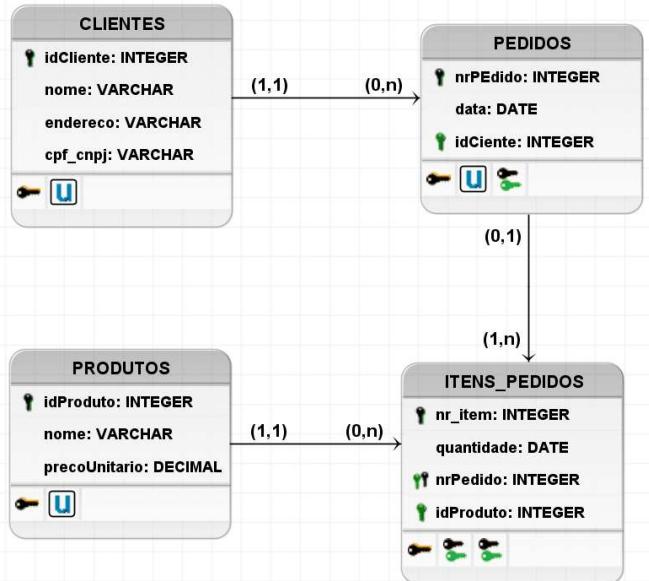


Modelo Orientado a Documentos

- ✓ Modelo adequado à representação de objetos complexos
 - Um objeto (“documento”) possui uma chave e um conjunto de atributos
 - Atributos podem ter domínios atômicos ou complexos (listas, *tuplas*, conjuntos)
- ✓ Possuem APIs proprietárias ou linguagens de consulta simples
- ✓ Não suporta relacionamentos entre dados
- ✓ Falta de padronização
 - *MongoDB* (JSON)
 - *Amazon SimpleDB* (*Domínio* → *Item* → *Atributo* → {*Valor*})

```
{ "_id": "discussion_tables",
  "_rev": "D1C946B7",
  "Sunrise": true,
  "Sunset": false,
  "FullHours": [1,2,3,4,5,6,7,8,9,10],
  "Activities": [
    {"Name": "Football", "Duration": 2, "DurationUnit": "Hours"},
    {"Name": "Breakfast", "Duration": 40, "DurationUnit": "Minutes",
      "Attendees": ["Jan", "Damien", "Laura", "Gwendolyn", "Roseanna"] }
  ]
}
```

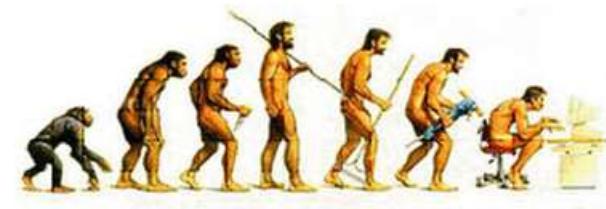
Modelo Orientado a Documentos



```
{
  "id": "nrPedido",
  "cliente": {
    "idCliente": "integer",
    "nome": "string",
    "endereco": "string",
    "cpf_cnpj": "string"
  },
  "pedido": {
    "nrPedido": "integer",
    "data": "string",
    "itensPedido": [
      {
        "nr_item": "integer",
        "quantidade": "decimal",
        "produto": {
          "nome": "string",
          "precoUnitario": "decimal"
        }
      }
    ]
  }
}
```

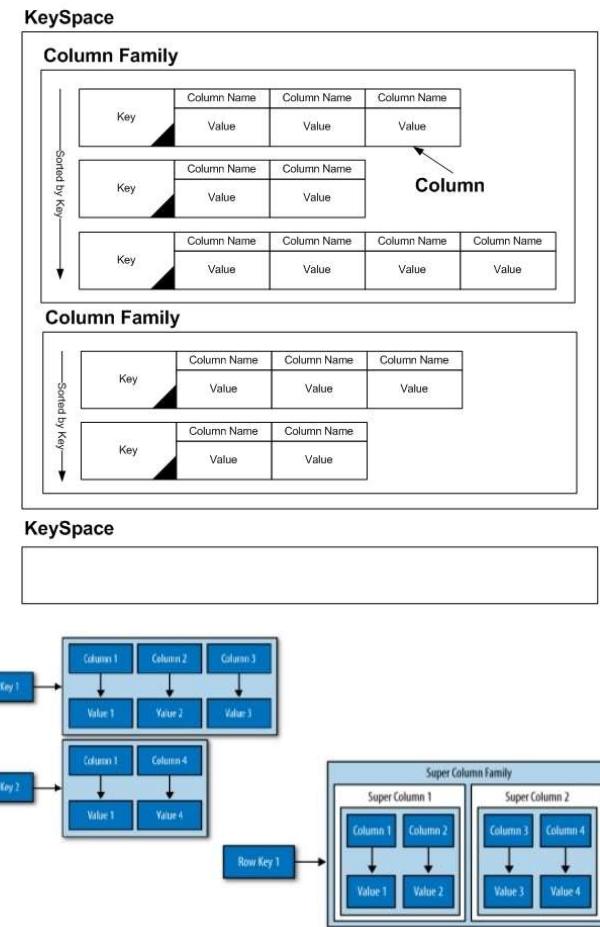
Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- Propriedades dos BDs NoSQL
- Modelos de dados NoSQL
 - Chave-valor
 - Orientado a documentos
 - **Orientado a colunas**
 - Orientado a grafos

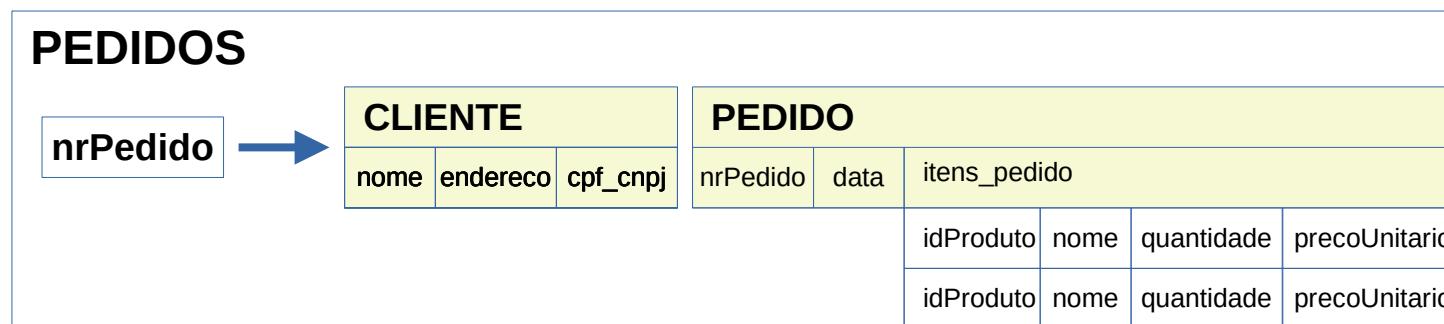
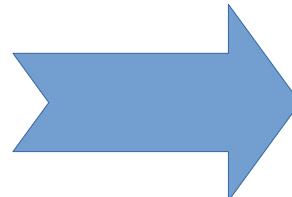
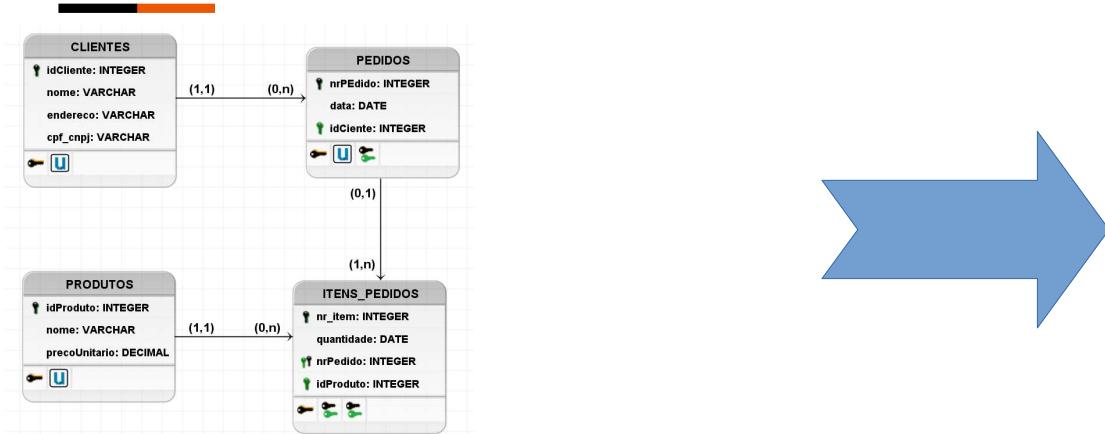


Modelo Orientado a Colunas

- ✓ Modelo mais complexo que o chave-valor:
 - *keyspace* (\equiv BD), *column family* (\equiv tabela) e um **conjunto de colunas** (\equiv registro)
 - Uma coluna possui um *nome* e um *valor*
 - Um conjunto de colunas é acessado por uma *chave*
 - Itens de dados (“registros”) podem ter colunas diferentes
 - Suporte a colunas multivaloradas e super-colunas
- ✓ APIs proprietárias ou linguagens de consulta simples
- ✓ Não suporta relacionamentos entre dados

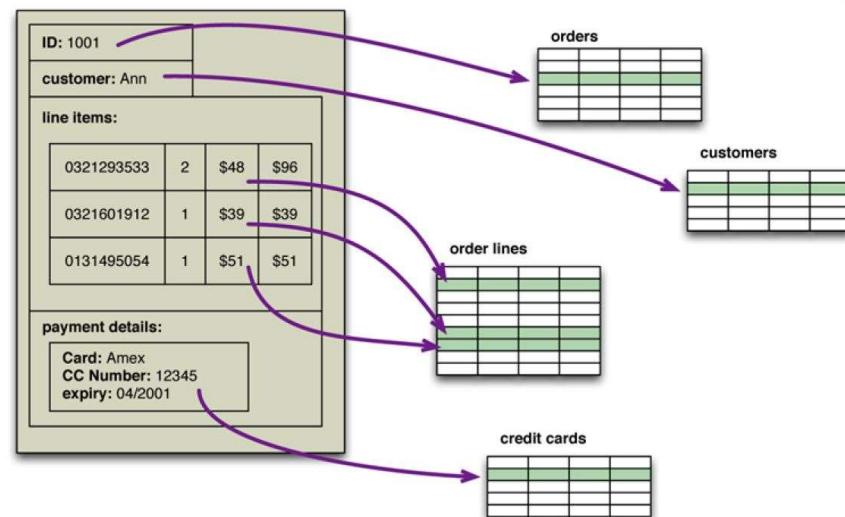


Modelo Orientado a Colunas



Modelo Orientado a Agregados

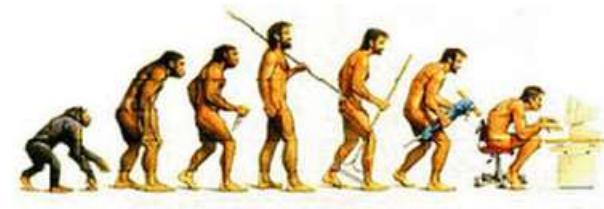
- Os modelos *chave-valor*, *orientado a documentos* e *orientado a colunas* são também classificados como **modelos orientados a agregados**. (SADALAGE e FOWLER, 2013)



- Um **agregado** é uma unidade de dados que possui uma estrutura mais complexa do que um conjunto de *tuplas*.
 - É um registro complexo que permite que listas e outras estruturas de registro sejam aninhadas dentro dele e sejam tratadas como uma unidade.
 - Um agregado forma uma unidade mais natural para replicação e fragmentação.

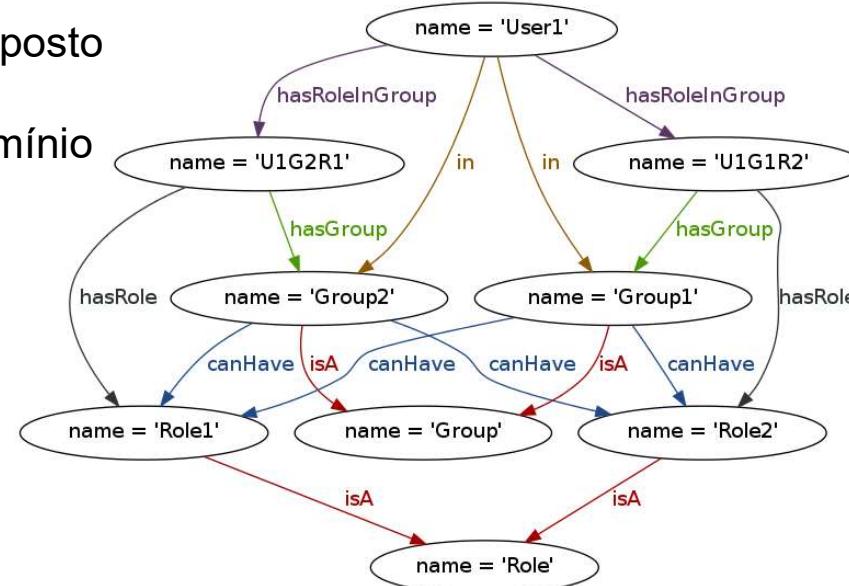
Parte 1: BDs NoSQL e modelos de dados

- Um pouco de história
- Propriedades dos BDs NoSQL
- Modelos de dados NoSQL
 - Chave-valor
 - Orientado a documentos
 - Orientado a colunas
 - **Orientado a grafos**



Modelo Orientado a Grafos

- ✓ Modelo composto por:
 - **Nodos**: um item de dado (“registro”) composto por atributos
 - **Arestas**: relacionamento entre nodos, composto por um rótulo e atributos opcionais
 - **Atributos**: composto por nome e valor (domínio atômico ou multivalorado)
- ✓ O acesso aos dados é feito por meio de APIs proprietárias ou linguagens de consulta simples



Modelos de Dados para NoSQL

Concluindo...

- **Nenhum modelo de dados NoSQL é superior a outro.**
 - ✓ Um modelo NoSQL pode ser mais adequado para ser utilizado em certas situações do que outros.
- Exemplos:
 - ✓ Para dados que são **escritos frequentemente**, mas são **pouco lidos** (p.ex. um contador de *hits* na Web), pode ser usado um BD orientado a documentos, como o *MongoDB*;
 - ✓ Para aplicações que demandam **alta disponibilidade** pode-se utilizar um BD orientado a colunas, como o *Cassandra*;
 - ✓ Para aplicações que exigem um **alto desempenho em consultas, com muitos agrupamentos**, pode-se utilizar um BD orientado a grafos, como o *Neo4j*.





Minicurso: Projeto de Bancos de Dados NoSQL

PERGUNTAS?

