

# NoSQL

Prof .Maiquel de Brito

`maiquel.b@ufsc.br`

BLU3024

UFSC Blumenau

# Tendências decorrentes do *Big Data*

- É necessário trabalhar com dados inconsistentes
- Dados inseridos não são atualizados
- Crescimento exponencial imprevisível
- Consistência não é algo tão crítico
- Leitura de dados maior que escrita de dados

Bancos de dados relacionais não atendem às tendências atuais

# Limitações dos BDs relacionais

## Incompatibilidade de representação (tradução livre para *impedance mismatch*)

- Modelo relacional representa dados através de relações e tuplas
- Qualquer outra forma de representação deve ser “traduzida” para esse formato
- Exemplos: dados aninhados, objetos, grafos, árvores etc

## Crescimento Vertical

- BDs maiores  $\rightsquigarrow$  máquinas melhores (CPU, memória etc) – crescimento vertical
- Clusters oferecem *crescimento horizontal*
- BDs relacionais não são compatíveis com clusters

# Bancos de dados NoSQL

Bancos de dados não baseados no modelo relacional

~~No SQL~~ → **N**ot **O**nly SQL

# NoSQL - Histórico

- Ideias desenvolvidas “lentamente” por grupos independentes para resolver problemas específicos resultando em diversos bancos de dados existentes hoje
- Google Big Table Paper (2006) [download](#)  
linhas com uma chave única armazenando colunas que têm relação entre si  
acessar dados relacionados entre si sem *join*  
evitar o uso de muitas tabelas

# NoSQL - Algumas características (1)

- *Schema* é menos relevante

Schema: define quais dados o SGBD deve esperar.

Tabelas com colunas fixas.

- NoSQL é não-relacional

modelo relacional: relacionamentos, normalização

NoSQL: não há normalização

O BD “não sabe” como os registros se relacionam entre si

Várias informações em uma única coluna

Vantagens: facilidade de armazenamento e recuperação, velocidade

## NoSQL - Algumas características (2)

- Projetado para *clusters*

NoSQL tem escalabilidade horizontal (mais recursos são obtidos com mais máquinas)

Relacional tem escalabilidade vertical (mais recursos requerem upgrade em uma máquina)

# Modelo de dados em NoSQL

**Data Model:** modelo que define a organização e manipulação dos dados

Modelo de dados mais comum: **relacional**


- Tabelas divididas em colunas
- Cada coluna armazena um valor
- Colunas podem referenciar outras tabelas



# Modelo de dados em NoSQL

NoSQL adota outro modelo de dados  
abandona o modelo relacional

Há diversos modelos de dados NoSQL:



- key-value
- column-family
- document
- graph

- Todos não relacionais
- Todos baseados em agregações

# Modelo de dados em NoSQL - Column family

- Decendente direto do Google Big Table
- Guarda semelhanças com o modelo relacional  
inclui *linhas* e *colunas*
- Foco na velocidade de operações entre colunas  
em vez de focar na velocidade de acesso
- Colunas podem armazenar conjuntos de valores  
em vez de um único valor, como no modelo relacional
- Vantagem: toda a informação pode estar contida em uma  
única linha  
em vez de requerer um SQL complexo com muitos *joins*

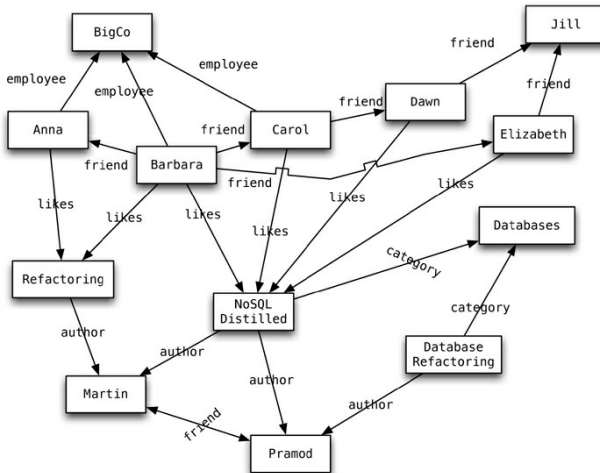
# Modelo de dados em NoSQL - Key Value

- Dados armazenados na forma de pares (*dado, valor*)
- Uma chave (*key*) dá acesso a determinados valores  
não há muita preocupação sobre o que são esses valores
- Metáfora: caixa de correio
- Hash tables

artista:1:nome	AC/DC
artista:1:gênero	Hard Rock
artista:2:nome	Belchior
artista:2:gênero	MPB

# Modelo de dados em NoSQL - Graph

- Armazena *entidades* e *relacionamentos* entre entidades



Fonte: [Sadalage and Fowler, 2013]

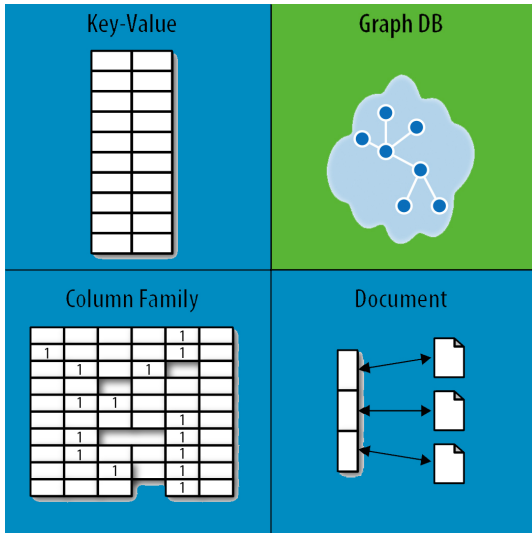
# Modelo de dados em NoSQL - Document

- Armazena cada **registro** em um **documento** (XML, JSON etc)
- Sem *schema* fixo  
o próprio documento descreve sua estrutura de dados
- Diferentes registros podem ter **diferentes atributos**

```
{firstName: ''Alice'',  
  lastName: ''Johnson'',  
  position: ''CFO'',  
  officeNumber: ''2-120'',  
  officePhone: ''555-222-3456''}  
  
{firstName: ''Bob'',  
  lastName: ''Wilson'',  
  position: ''Manager'',  
  officeNumber: ''2-130'',  
  officePhone: ''555-222-3478'',  
  hireDate: ''1-Feb-2010'',  
  terminationDate: ''12-Aug-2014''}
```

- Consultas pelos valores dos atributos dentro dos documentos  
ex.: `db.employees.find( position:'Manager' )`

# Modelo de dados em NoSQL



fonte: <https://neo4j.com/blog/why-nosql-databases/>

# NoSQL na prática - Cassandra

- Cada entidade é armazenada em uma **linha** com um identificador único (*primary key*)
- Cada linha é composta de uma série de colunas
- *Column family*: grupo de colunas que têm relação entre si  
Ex.: endereço: rua, número, cep, bairro, cidade...
- Cada linha é um conjunto de  $n > 0$  column families

# Agregações

Modelo relacional: baseado em tuplas

cada tupla armazena uma instância de uma única entidade

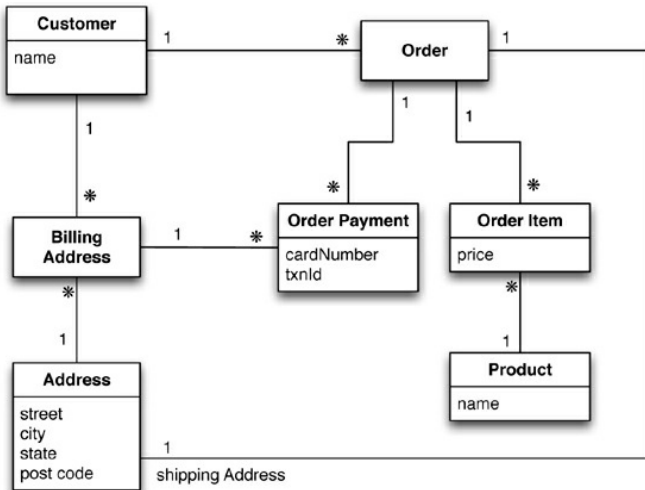
não há tuplas aninhadas

Agregação: estruturas mais complexas, tuplas aninhadas

Definição: *coleção de objetos relacionados que podem ser tratados como uma unidade*



# Agregações



Fonte: [Sadalage and Fowler, 2013]

# Agregações

Customer	
Id	Name
1	Martin

Orders		
Id	CustomerId	ShippingAddressId
99	1	77

Product	
Id	Name
27	NoSQL Distilled

BillingAddress		
Id	CustomerId	AddressId
55	1	77

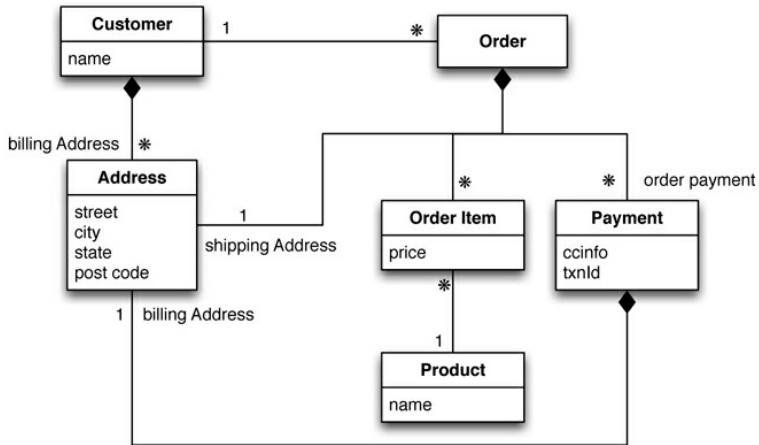
OrderItem			
Id	OrderId	ProductId	Price
100	99	27	32.45

Address	
Id	City
77	Chicago

OrderPayment				
Id	OrderId	CardNumber	BillingAddressId	txnId
33	99	1000-1000	55	abelif879rft

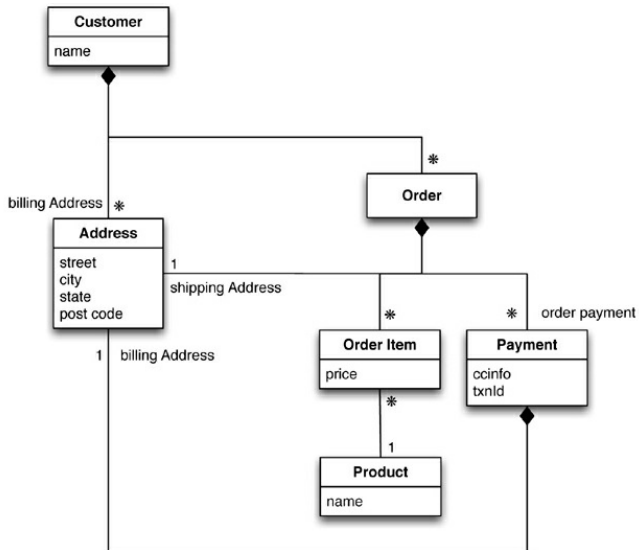
Fonte: [Sadalage and Fowler, 2013]

# Agregações



Fonte: [Sadalage and Fowler, 2013]

# Agregações



Fonte: [Sadalage and Fowler, 2013]

# Modelo de dados em NoSQL - Column family - Retomando...

- Decendente direto do Google Big Table
- Semelhante ao modelo relacional mas..  
inclui *linhas* e *colunas*
- Foco na velocidade de operações entre colunas  
em vez de focar na velocidade de acesso
- Colunas podem armazenar conjuntos de valores  
em vez de um único valor, como no modelo relacional
- Vantagem: toda a informação pode estar contida em uma  
única linha  
em vez de requerer um SQL complexo com muitos *joins*

# Apache Cassandra

- Banco de dados Column Family
- <http://cassandra.apache.org>
- Implementado em Java
- Multiplataforma



SADALAGE, P.J.;FOWLER, M. J. **NoSQL distilled : a brief guide to the emerging world of polyglot persistence.** [S.l.]: Pearson, 2013.