

Modelos de Dados para Sistemas NoSQL

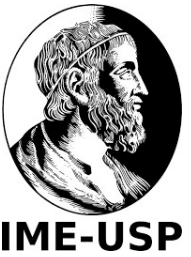
Parte 2

**Bancos de Dados Chave-Valor,
de Documentos e de Famílias de Colunas**

Kelly Rosa Braghetto
kellyrb@ime.usp.br

Departamento de Ciência da Computação

Sistemas NoSQL – Modelos de Dados

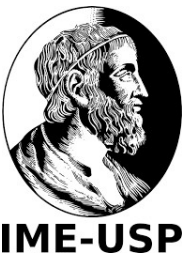


Baseado no livro

***NoSQL Distilled – A Brief Guide to the
Emerging World of Polyglot Persistence***

de Pramod Sadalage e Martin Fowler

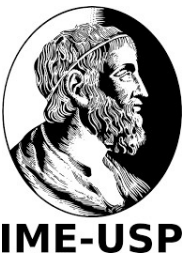
Sistemas NoSQL – Modelos de Dados



- Diferentes tipos de modelos de dados são usados nos sistemas NoSQL
 - Grande heterogeneidade
- Esses modelos costumam ser divididos nas seguintes categorias:
 - **Chave-valor**
 - **Documentos**
 - **Famílias de colunas**
 - **Grafos**

orientados a
agregados

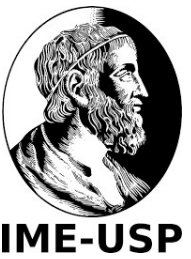
Bancos de Dados Chave-Valor e de Documentos



- São fortemente orientados a agregados
- São constituídos por conjuntos de agregados
 - cada agregado tem uma chave (ou ID), usada na recuperação dos dados
- Diferença entre eles:
 - **BD chave-valor:** os agregados são “opacos” para o SGBD
 - **BD de documentos:** o SGBD pode “enxergar” a estrutura dos agregados

Bancos de Dados

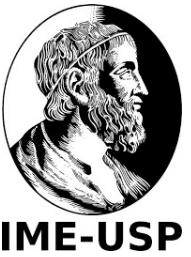
Chave-Valor



- A opacidade traz flexibilidade no armazenamento
 - É possível armazenar qualquer tipo de dado dentro de um agregado
 - O SGBD vê o conteúdo de um agregado apenas como um punhado de bits que precisam ser mantidos juntos
 - Mas pode haver limite de tamanho
- Um agregado é acessado por meio de busca pela sua chave
- O suporte à definição de índices sobre campos do agregado geralmente não é implementado

Bancos de Dados

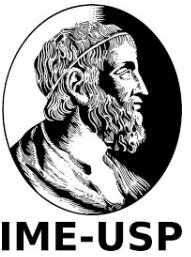
Chave-Valor



- Interface simples
 - Modelo de dados:

pares (chave, valor)
 - Operações:
 - **Insert**(chave,valor)
 - **Fetch**(chave) → busca/recuperação de um valor
 - **Update**(chave,valor)
 - **Delete**(chave)
 - Alguns sistemas permitem que a operação de Fetch seja realizada sobre faixa de valores de chaves
 - Ex.: $2 < \text{chave} < 10$

Bancos de Dados Chave-Valor



- Exemplos

- Redis

- <http://redis.io/>

- Memcached

- <http://memcached.org/>

- Riak

- <http://basho.com/products/riak-kv/>

- OrientDB

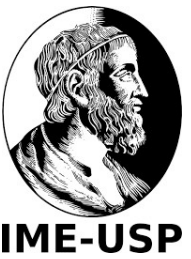
- <http://orientdb.com/>

Bancos de Dados de Documentos



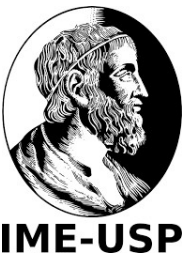
- Vantagem da não-opacidade: flexibilidade no acesso
 - É possível fazer buscas sobre o conteúdo do agregado
 - Também é possível recuperar partes de um agregado (ao invés de recuperá-lo por inteiro)
- O SGBD define quais são as estruturas e tipos de dados permitidos dentro do agregado
 - Há restrições quanto ao que pode ser armazenado no agregado
- É possível definir índices sobre o conteúdo dos agregados

Bancos de Dados de Documentos



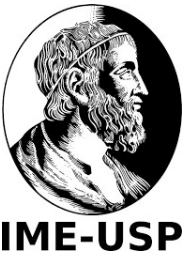
- Interface
 - Modelo de dados: **pares (chave, documento)**
 - **Documento: dados em formatos semiestruturados (JSON, XML, etc.)**
 - Permite a definição de documentos aninhados
 - Documentos são agrupados em **coleções**

Bancos de Dados de Documentos



- Operações básicas:
Insert(chave,documento), **Fetch**(chave),
Update(chave,documento), **Delete**(chave)
- Também fazem **Fetch de documentos buscando pelo seus conteúdos**
 - mas não há uma linguagem de busca padrão
 - cada SGBD tem sua própria API
- Geralmente não implementam bloqueios explícitos
 - Atomicidade em operações envolvendo mais de um agregado precisa ser implementadas nas aplicações

Bancos de Dados de Documentos



- Exemplos

- MongoDB

<http://www.mongodb.org/>

- Couchbase

<http://www.couchbase.com/>

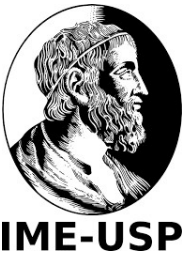
- Amazon DynamoDB

<https://aws.amazon.com/pt/dynamodb/>

- CouchDB

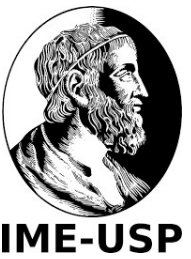
<http://couchdb.apache.org/>

Chave-valor ou documentos?



- Na prática, a separação entre esses dois modelos não é clara
- Exemplos de chave-valor com recursos de documentos:
 - O Riak permite adicionar metadados nos agregados, para apoiar a criação de índices e de relacionamentos entre agregados
 - O Redis permite dividir um agregado em listas ou conjuntos
 - É possível prover consultas baseadas em conteúdo em BDs chave-valor integrando-os a ferramentas de busca, como a Solr e a ElasticSearch
 - <https://lucene.apache.org/solr/>
 - <https://www.elastic.co/pt/elasticsearch/>

Chave-valor ou documentos?



A escolha entre um modelo e outro deve ser guiada pelo tipo de consulta que se espera receber com mais frequência no BD

- Em BDs chave-valor:
 - Recuperação de agregados pela chave
- Em BDs de documentos:
 - Recuperação de agregados a partir de consultas baseadas na estrutura interna do documento (e também na chave)

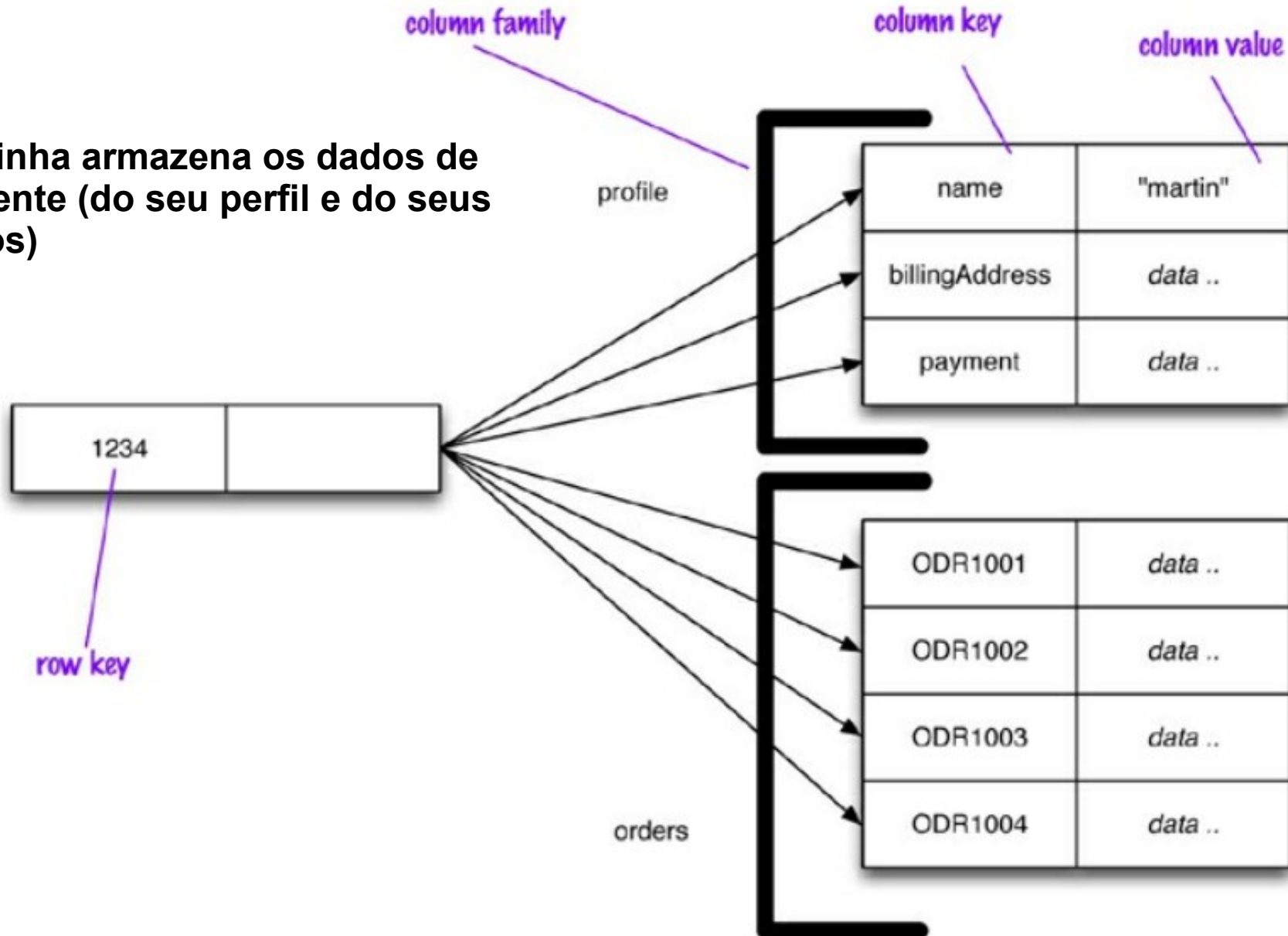
Bancos de Dados de Famílias de Colunas



- **BDs de famílias de colunas** armazenam dados como linhas, onde cada linha possui uma chave e colunas agrupadas em famílias
 - Cada coluna deve fazer parte de uma família
- Dentro de uma mesma família ficam dados que, frequentemente, serão acessados juntos
- As linhas não precisam ter as mesmas colunas
- Novas colunas podem ser adicionadas a qualquer linha em qualquer momento (sem a necessidade de serem adicionadas às demais linhas)
- Uma coluna pode ser uma **supercoluna** (= mapa de **colunas**), ou seja, um contêiner para outras colunas

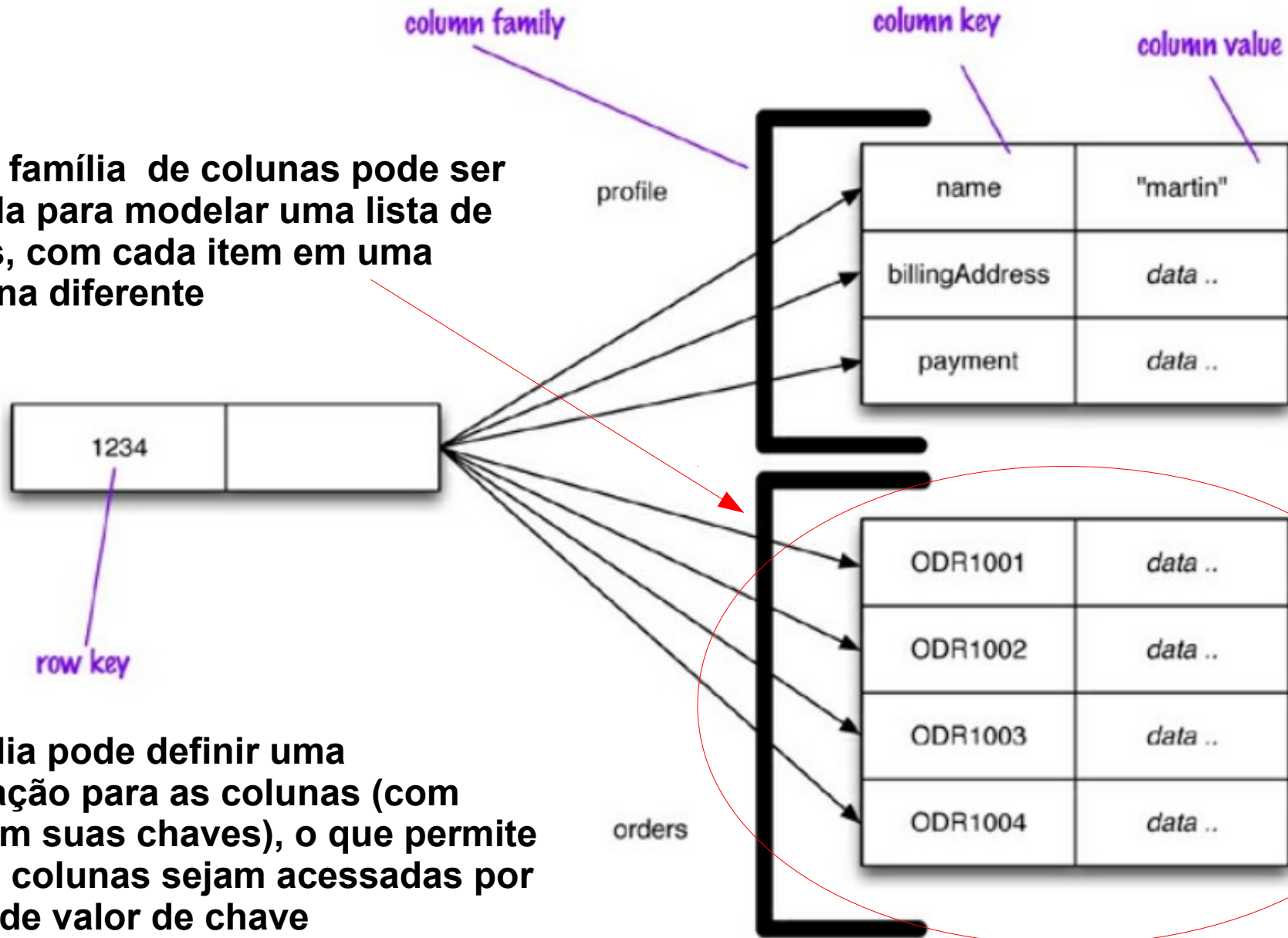
Famílias de Colunas – Exemplo

Cada linha armazena os dados de um cliente (do seu perfil e do seus pedidos)



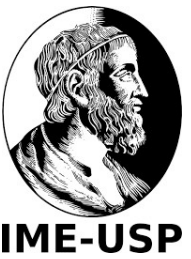
Famílias de Colunas – Exemplo

Uma família de colunas pode ser usada para modelar uma lista de itens, com cada item em uma coluna diferente



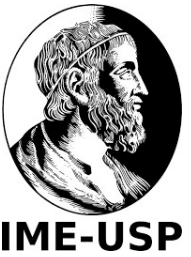
A família pode definir uma ordenação para as colunas (com base em suas chaves), o que permite que as colunas sejam acessadas por faixas de valor de chave

Bancos de Dados de Famílias de Colunas



- O SGBD “enxerga” a organização das colunas em famílias e a usa a seu favor no armazenamento e recuperação dos dados
- Modelo de escalabilidade: dividir tanto linhas quanto colunas em diversos nós
 - Linhas são divididas entre os nós de acordo com o valor da chave
 - Colunas são distribuídas entre os nós de acordo com suas famílias (pré-definidas pelo programador)

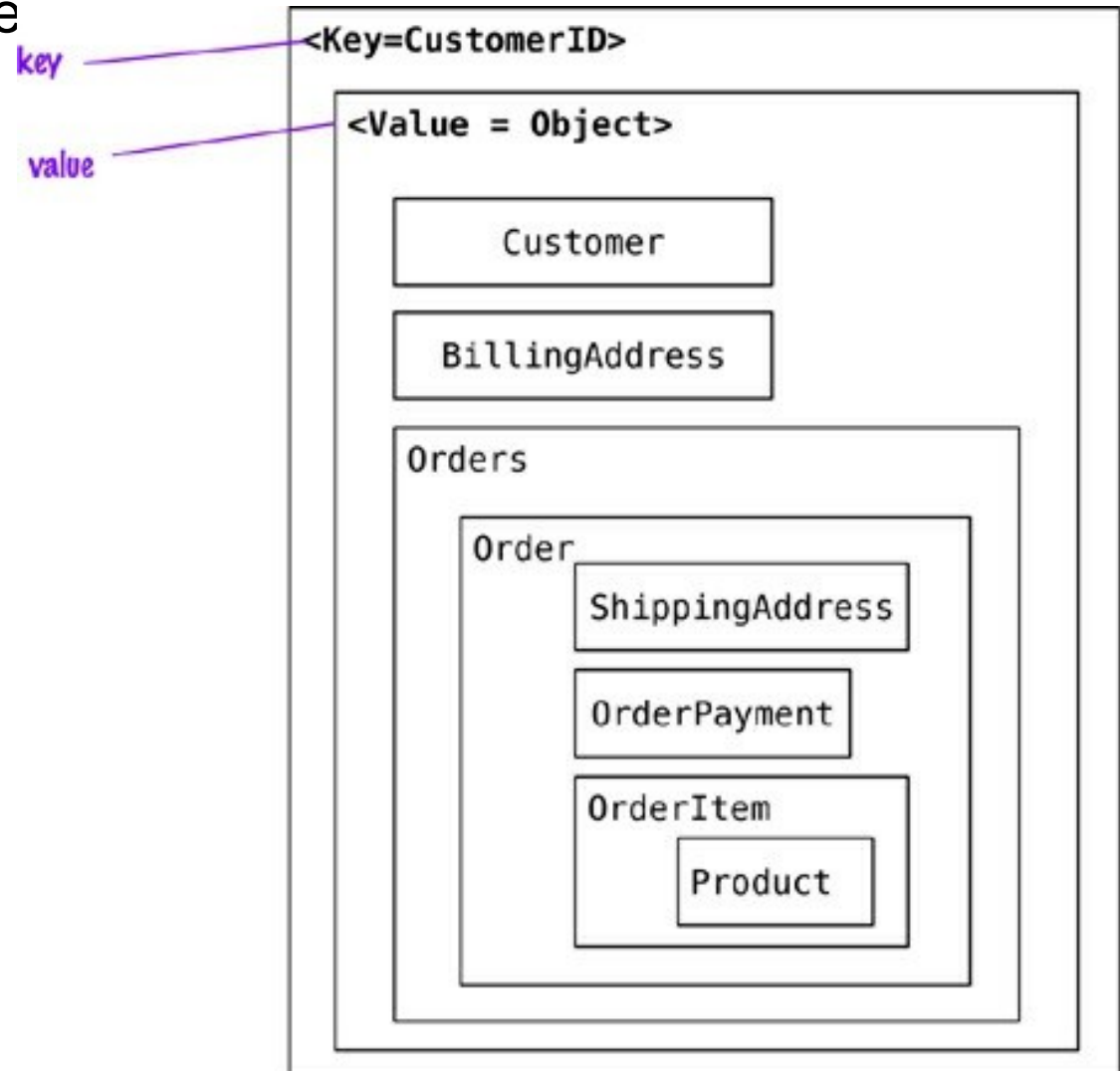
Bancos de Dados de Famílias de Colunas



- Exemplos:
 - Google Bigtable → Precursor
<https://cloud.google.com/bigtable/>
 - Cassandra
<https://cassandra.apache.org/>
 - Hbase
<https://hbase.apache.org/>
 - HyperTable
<http://hypertable.org/>
 - MonetDB
<https://www.monetdb.org/>

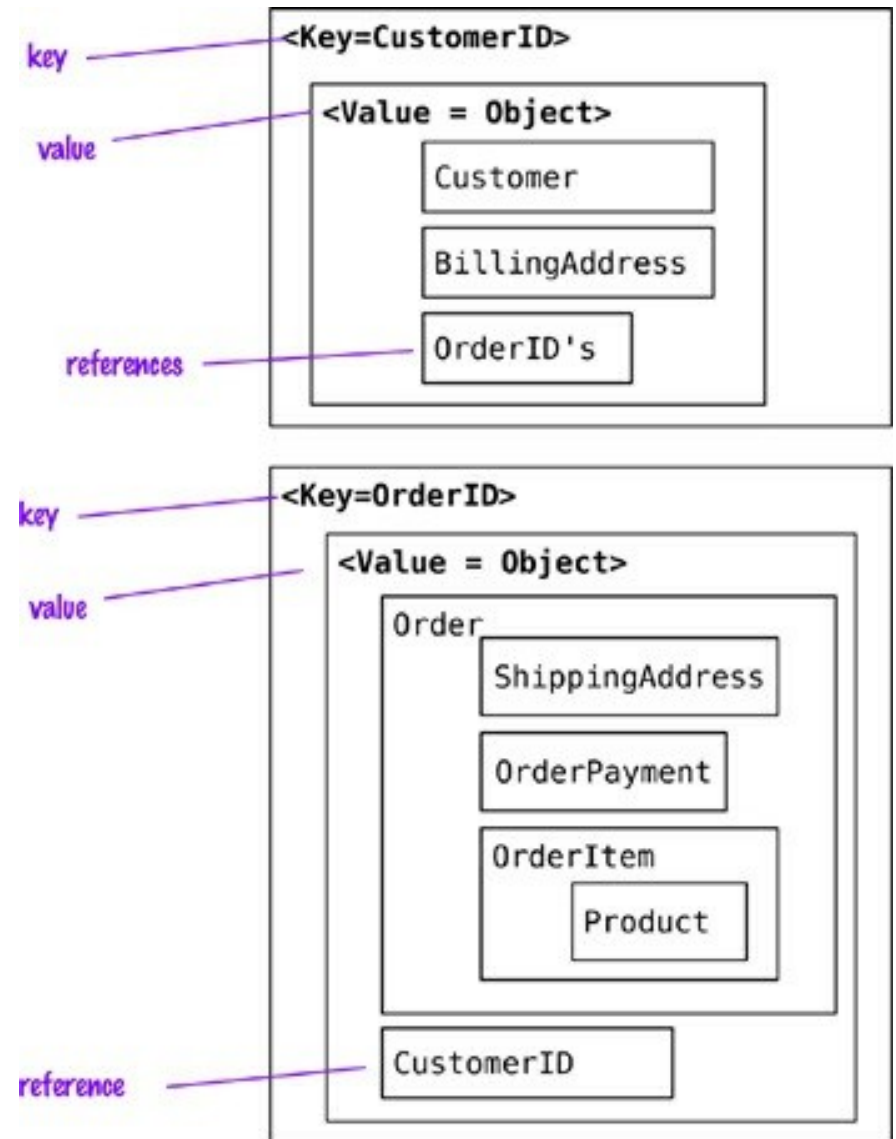
Exemplo: BD Chave-valor do Comércio Eletrônico

- Leitura de todos os dados de um cliente por meio da sua chave
- Se for necessário consultar dados por pedido, processamento terá que ser feito via aplicação



Exemplo: BD Chave-valor do Comércio Eletrônico

- Alternativa: separar cliente e pedido em dois tipos de agregados
- Custo adicional: ter que atualizar a lista de pedidos em cliente a cada vez que um pedido for inserido
- Usando um BD de documentos, a lista de pedidos não precisaria ser mantida em Cliente

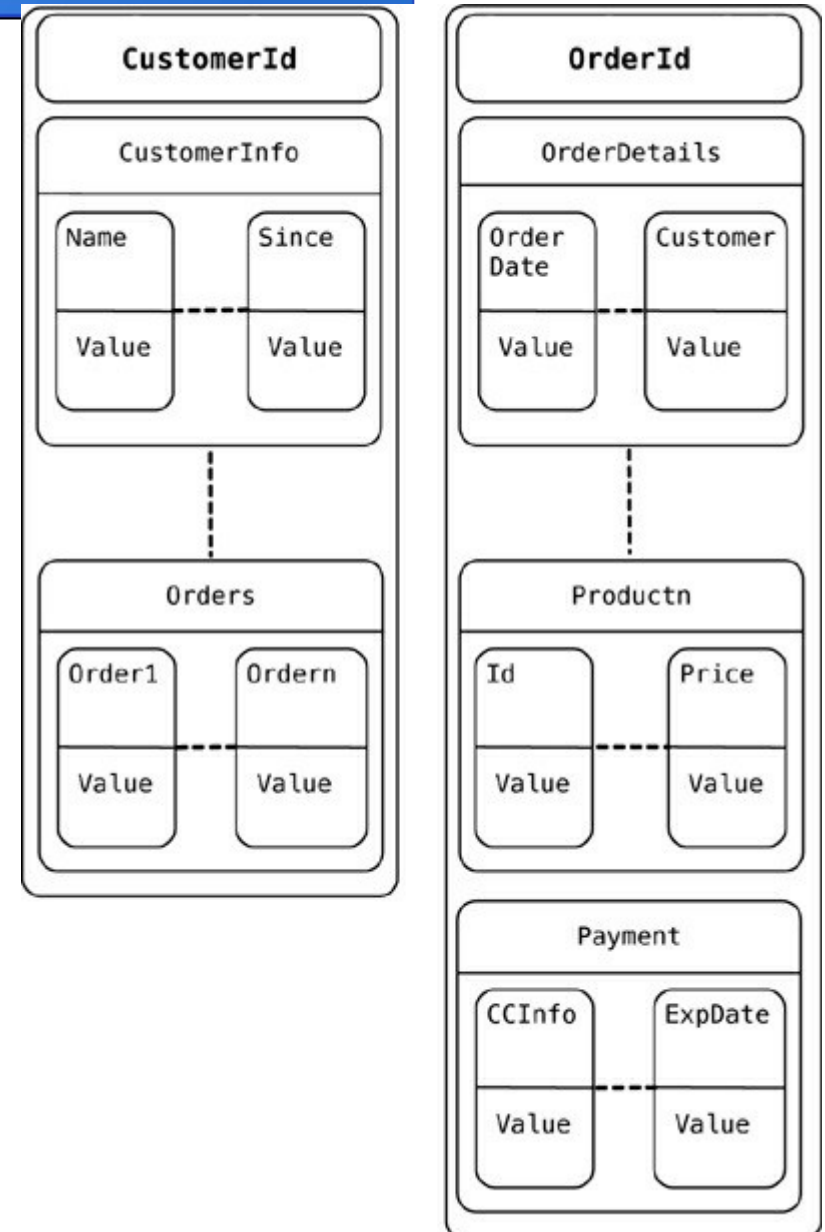


Exemplo: BD de Família de Colunas do Comércio Eletrônico

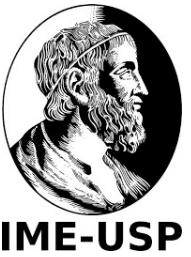


IME-USP

- Primeiras colunas são carregadas antes
- Algum nível de denormalização é bom em alguns casos

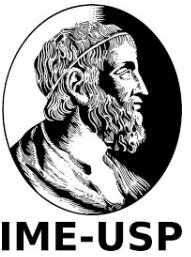


Sobre a ausência de esquema fixo



- Liberdade e flexibilidade
 - Estrutura é alterada conforme ganha-se mais conhecimento sobre o projeto (novos atributos são adicionados quando são descobertos)
 - Pode-se deixar de armazenar determinado tipo de dados, sem perder os dados do mesmo tipo que já estão no BD
 - Facilidade de lidar com falta de uniformidade

Sobre a ausência de esquema fixo



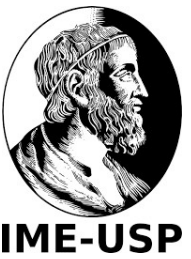
- Problema: esquema implícito → embutido nas aplicações que acessam o BD
 - Necessidade de vasculhar códigos para entender o BD
 - SGBD não é “ciente” do esquema e, portanto, não pode fazer otimizações específicas para o BD, nem ajudar a manter a consistência dos dados
 - Manutenção difícil quando múltiplas aplicações acessam o BD

Sobre a ausência de esquema fixo



- Manutenção difícil quando múltiplas aplicações acessam o BD
 - Problema pode ser amenizado com as seguintes estratégias:
 - encapsulando todo acesso ao BD em uma única aplicação (e.g., um serviço web) que será usada pelas demais
 - delineando as partes dos agregados que podem ser acessadas por cada aplicação

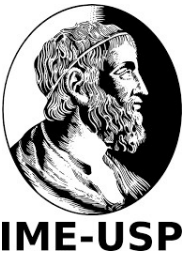
Sobre a ausência de esquema fixo



Com qualquer tipo de modelo de dados,
é preciso cuidado na organização dos dados

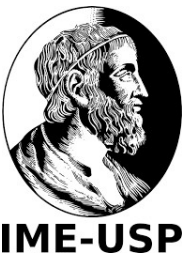
- Pode-se mudar esquemas de BDs relacionais de forma *ad hoc*:
 - Adicionar novas colunas e tabelas para armazenar dados não uniformes
- É preciso ter controle ao modificar a estrutura de BDs sem esquema fixo:
 - Garantir que tantos os dados no formato novo quanto os dados previamente existentes no formato antigo poderão ser facilmente acessados

Resumo



- Modelos de dados de sistemas NoSQL baseados em agregados
 - BDs chave-valor
 - Agregados opacos, busca por chave
 - BDs de documentos
 - Agregados não-opacos, busca e indexação por chave e conteúdo
 - BDs de famílias de colunas
 - Agregados organizados como agrupamentos de (super)colunas
 - Armazenamento e distribuição dos dados é ciente dos agrupamentos

Referências bibliográficas



- Sadalage, P. J., & Fowler, M. (2013). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.

Leitura extra indicada:

- Atzeni, P., Bugiotti, F., Cabibbo, L., & Torlone, R. (2020). *Data modeling in the NoSQL world*. Computer Standards & Interfaces, 67, 103149.