

NoSQL - Introdução

FRANCISCO PAULO DE FREITAS NETO

F.FREITAS@IFPB.EDU.BR

Bancos de dados relacionais

Os bancos de dados relacionais (BDR) desempenham um papel importante desde a sua criação. Os motivos são muitos:

- **Persistência**
- **Concorrência**
- **Integração**
- **Modelo padrão**

Bancos de dados relacionais

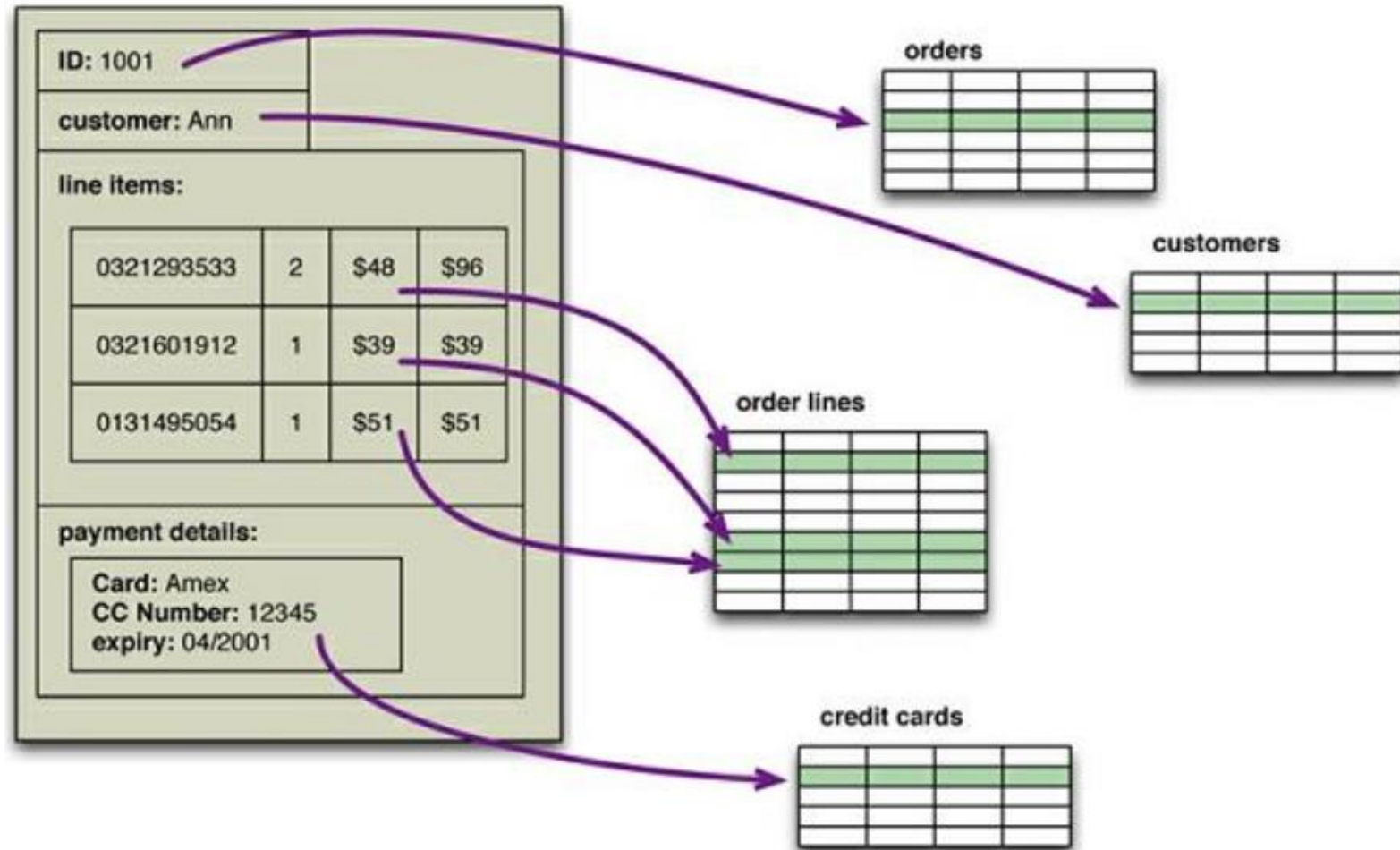
Por outro lado, existem algumas limitações nos bancos relacionais. A principal delas é a incompatibilidade de impedância.

Os bancos de dados relacionais, consomem e retornam relações (tabelas) conforme a álgebra relacional. Porém as aplicações trabalham sempre com estruturas de dados diferentes

- Modelo de dados \neq Estruturas de dados

Os BDR só trabalham com dados atômicos, não é possível trabalhar diretamente com listas, pilhas, registros aninhados. Então sempre deve haver uma tradução

Bancos de dados relacionais



Bancos de dados relacionais

Acreditava-se que os bancos de dados orientados a objetos (BDOO) substituiriam os bancos relacionais por solucionar o problema da incompatibilidade de impedância

- Mas esses bancos tiveram pouca aceitação pelo mercado

Outra alternativa foram alguns frameworks que abstraíam o mapeamento dos dados entre a aplicação e o banco.



Clusters

Com a WEB, as necessidades das aplicações (e dos seus dados) mudaram bastante. Existem novas necessidades: links, redes sociais, logs...

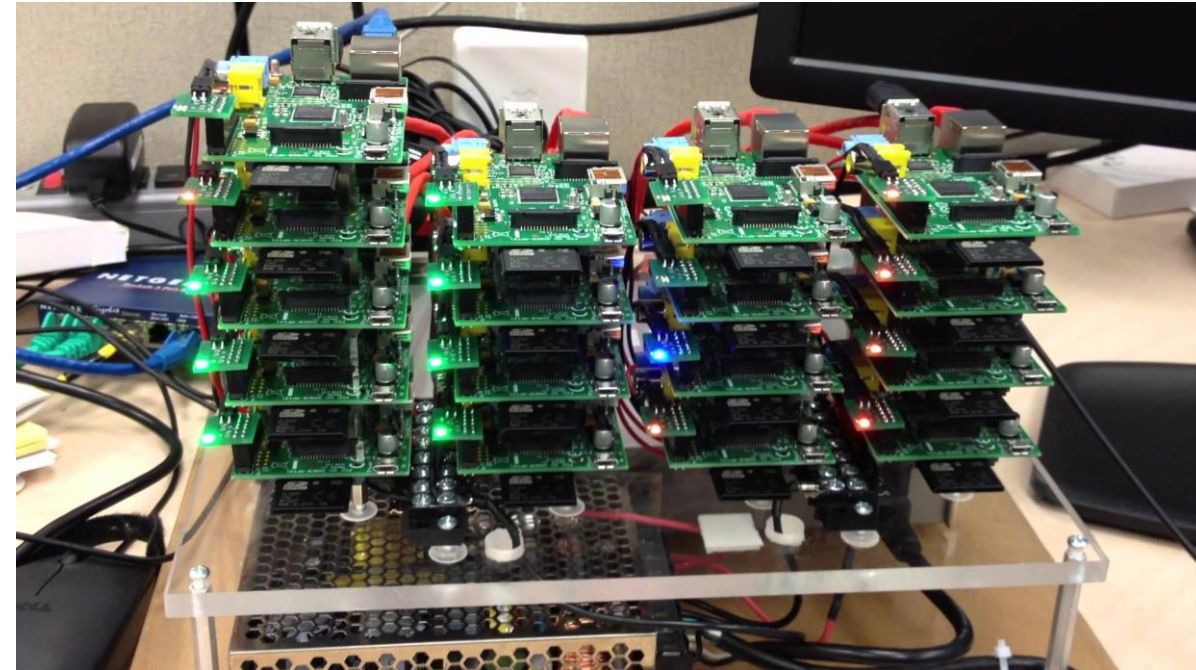
Com isso há um grande aumento no volume de dados gerado e armazenado:

- Segundo a CISCO cada usuário de internet consumiu em média 9,9 gb de dados em 2015. Em 2020 estima-se que esse número chegue a 25,1 gb.

Para lidar com isso, existem duas principais alternativas:

- Escalonamento vertical: adquirir computadores mais potentes e mais caros
- Escalonamento horizontal: investir em clusters

Clusters



Clusters

Com os clusters surgiu também um novo problema:

- Os bancos relacionais não foram projetados para rodar em clusters.

É difícil manter a integridade referencial em um cluster, e um banco relacional deve garantir as propriedades ACID

Alguns bancos como Oracle RAC ou SQL Server rodam em cluster, porém utilizam um sistema de disco compartilhado

- Cria-se um único ponto de falhas

Clusters

Além de tudo isso existem os custos.

Os bancos comerciais voltados para cluster são extremamente caros. Então, torna-se necessário pensar em alternativas.

No início dos anos 2000, Google e Amazon desenvolveram suas próprias soluções em bancos de dados:



NoSQL

O termo NoSQL surgiu no fim da década de 1990 como um banco não relacional e de código aberto.

A forma como utilizamos atualmente passou a ser utilizado a partir de 2009, durante uma reunião organizada por Johan Oskarson com desenvolvedores de bancos não relacionais e de código aberto.

O nome escolhido foi NoSQL Meetup e contou com desenvolvedores dos bancos: Voldemort, Cassandra, Hypertable, CouchDB, HBase, MongoDB, dentre outros.

NoSQL

Não existe uma definição formal para o termo, em geral, os bancos NoSQL possuem algumas características em comum:

- Não utilizam o modelo relacional
- A maioria são projetos de código fonte aberto
- São voltados para a execução em cluster (com algumas exceções)
- Baseiam-se nas necessidades da WEB
- São bancos sem esquema

NoSQL

O termo NoSQL é geralmente tido como Not Only SQL (não somente SQL)

Ao invés do significado da sigla, é melhor pensar no que esses bancos representam:

- *"Conjunto mal definido de bancos de dados, na sua maioria em código aberto, desenvolvido no século XXI (mundo WEB) e não utilizando o modelo relacional"*
- *"É melhor pensar NoSQL como um movimento ao invés de uma tecnologia"*

Sadalage e Fowler

NoSQL

O NoSQL é mais indicado quando se busca alta disponibilidade ou escalabilidade (muito comum na WEB)

Outros usos:

- Grande volume de dados (big data)
- Esquema flexível para dados flexíveis
- Gerenciamento de dados não transacionais: logs, por exemplo

Agregados de dados

Nos bancos relacionais, tudo é pensado como tuplas. Todos os dados são elementos atômicos, não existem listas, estruturas...

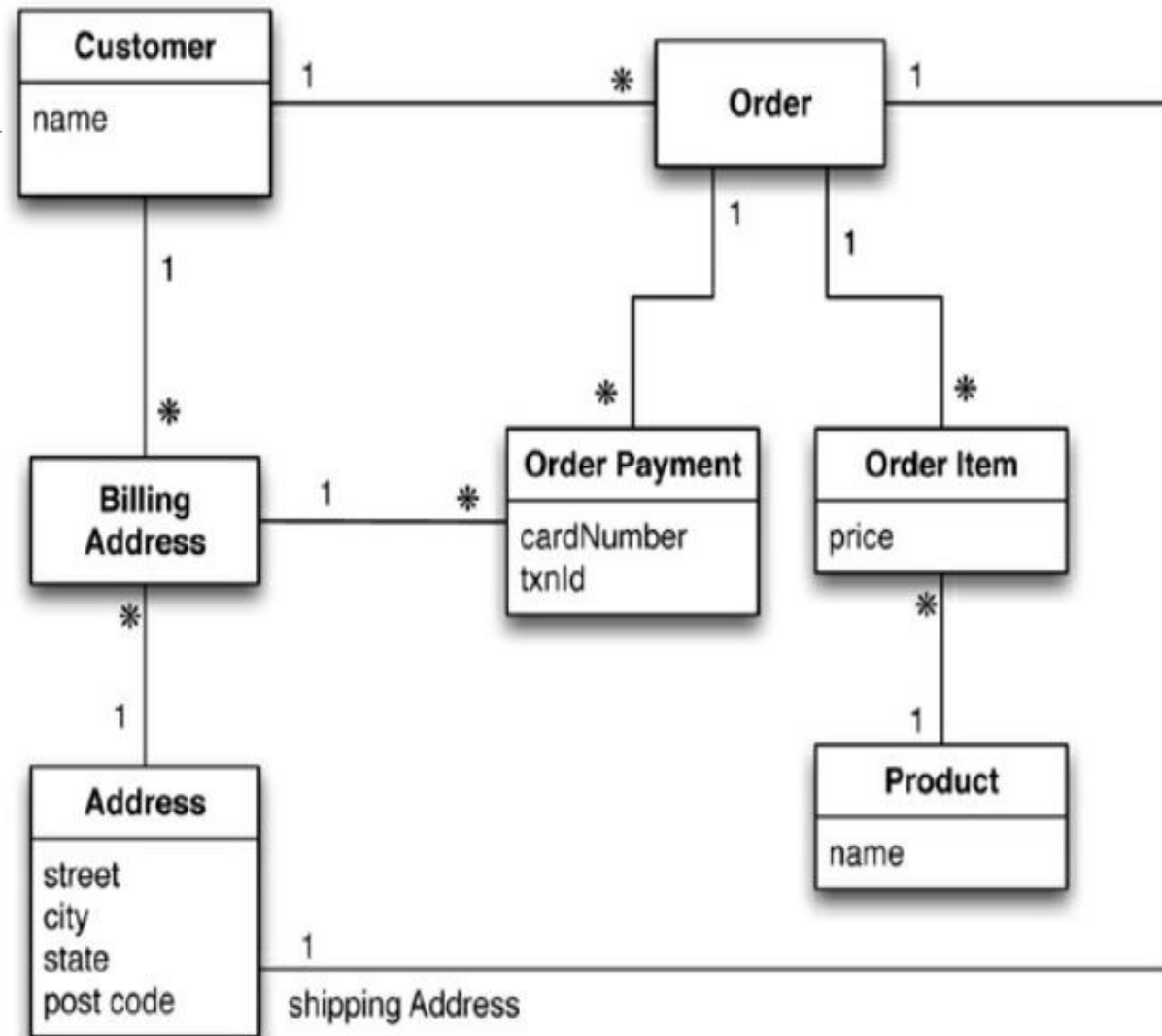
Porém nas aplicações nem sempre se trabalha com elementos tão simples.

Esses dados podem ser chamados de **agregados**:

- Conjunto de objetos relacionados que são tratados como uma única unidade pelo banco
- Funciona bem em clusters

Agregados de dados

Por exemplo, em um banco relacional



Agregados de dados

As tabelas seriam como a seguinte imagem

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

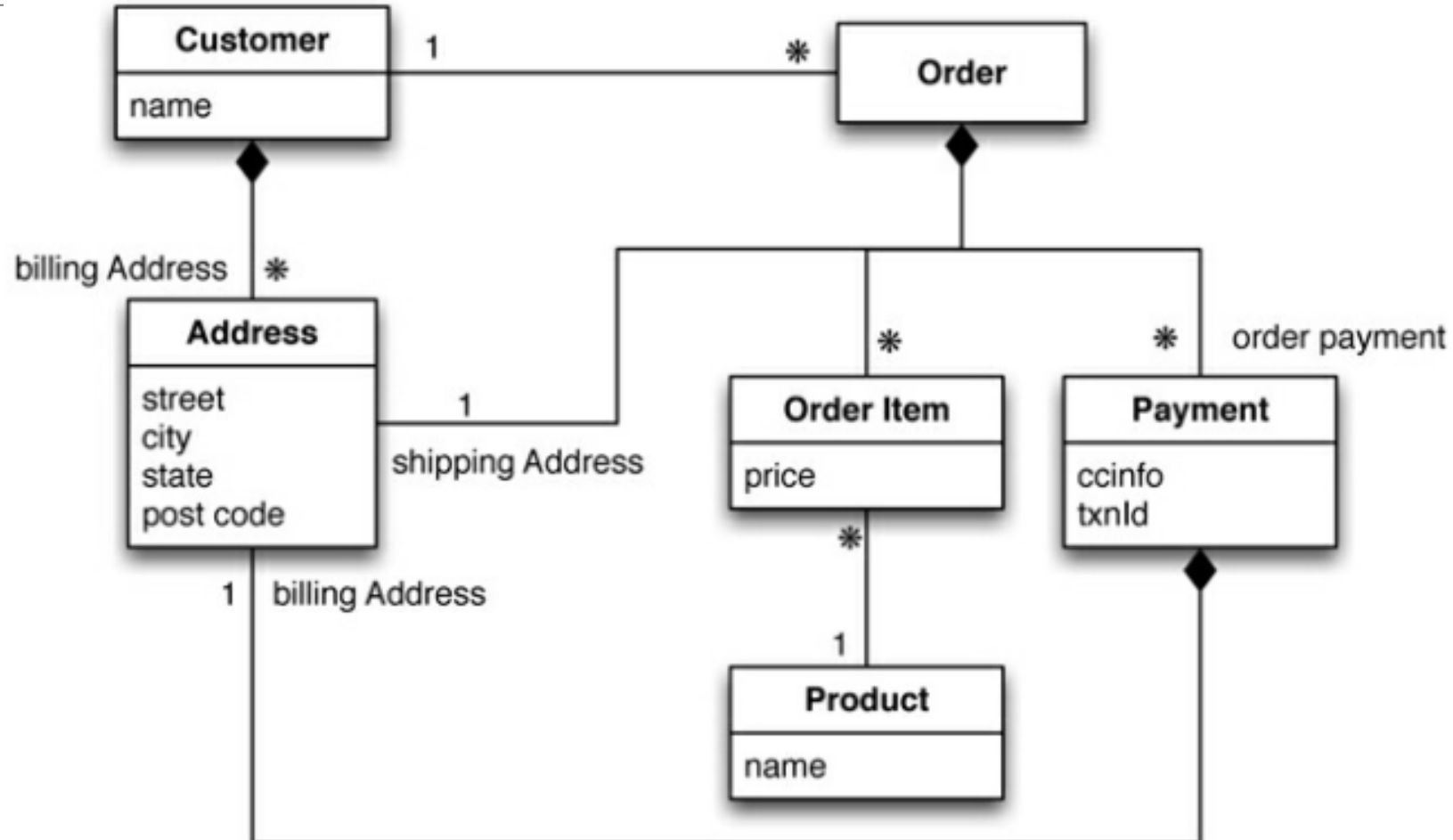
OrderItem			
Id	OrderId	ProductId	Price
100	99	27	32.45

Address	
Id	City
77	Chicago

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

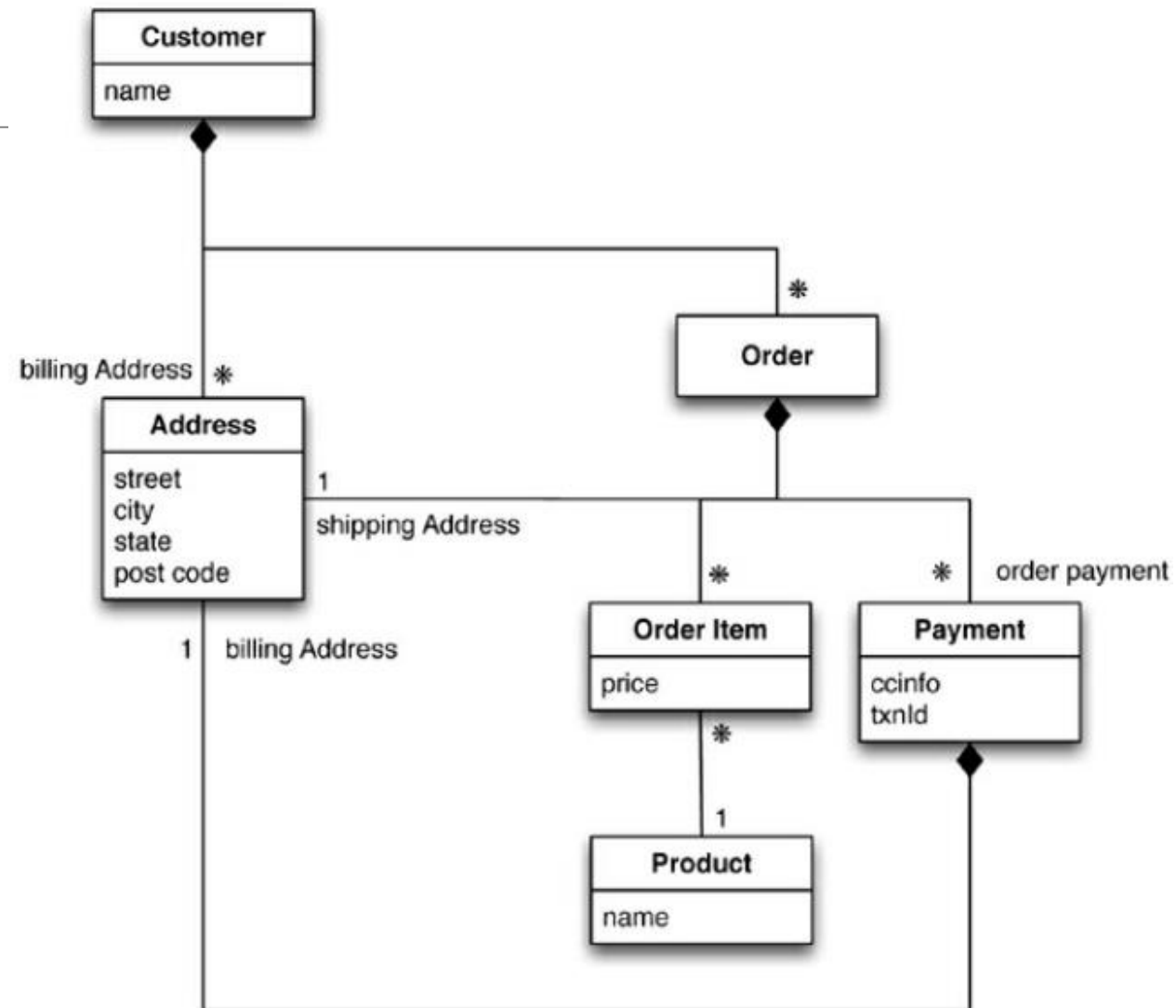
Agregados de dados

O mesmo exemplo utilizando um modelo agregado



Agregados de dados

Não existe somente uma forma de organizar os dados, o mesmo exemplo poderia ser representado da seguinte forma:



Consequência dos agregados

Os BDs relacionais não possuem nenhuma noção de agregados

Quando trabalhamos com agregados, temos uma semântica mais clara, com foco na unidade de interação.

Devemos nos preocupar em **como os dados são manipulados pelas aplicações**.

Consequência dos agregados

Nem todo banco NoSQL possui o conceito de agregado

- Os bancos de dados de grafo seguem o ACID

As vezes é difícil estabelecer os limites do agregado

Os agregados podem ser úteis, mas **dependendo da modelagem podem se tornar um obstáculo.**

Consequência dos agregados

A principal vantagem dos agregados é facilitar a operação em cluster

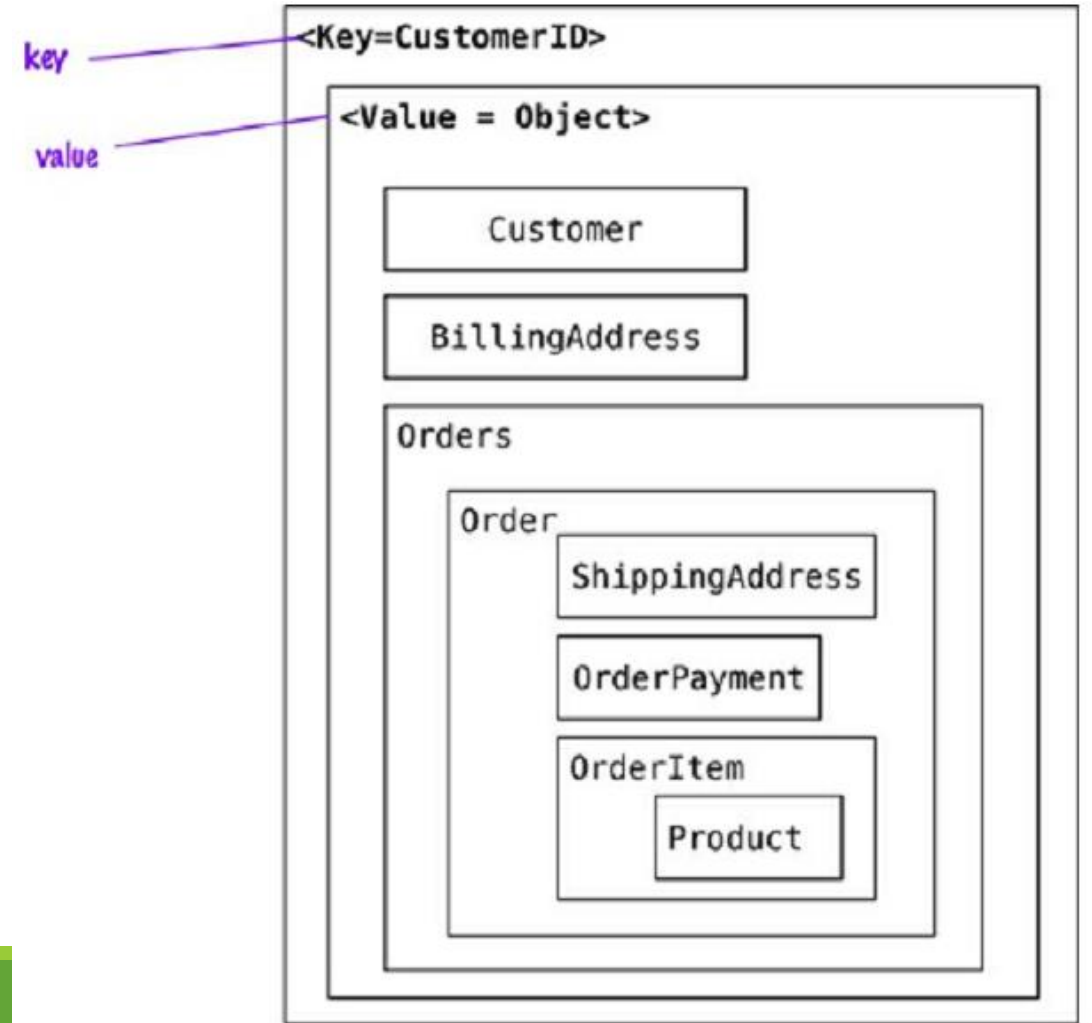
Os bancos orientados a agregado não suportam o ACID, **sendo as operações realizadas em um agregado por vez.**

Modelando para acesso aos dados

Quando trabalhamos com agregado precisamos nos preocupar em como os dados serão **lidos**.

Exemplo, pedidos dentro do agregado cliente:

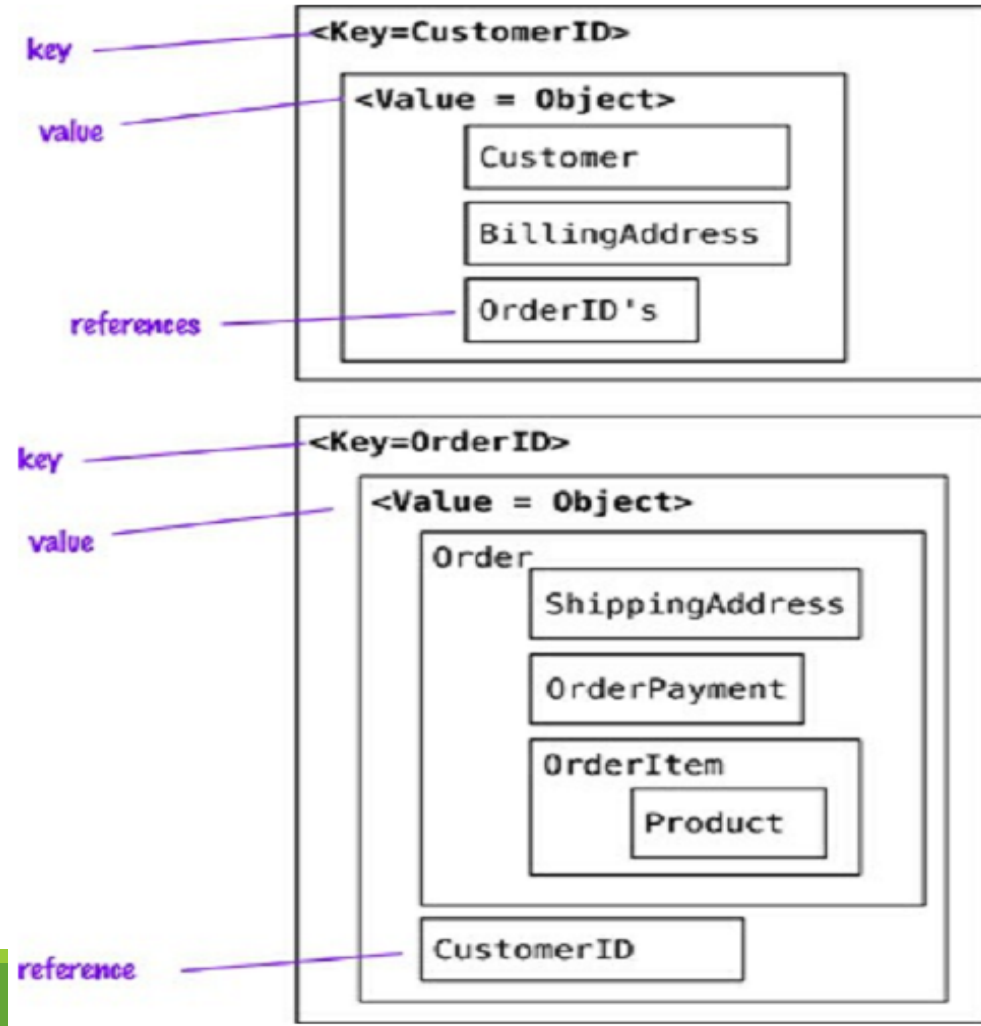
- A aplicação lê todas as informações de cliente a partir da sua chave.



Modelando para acesso aos dados

Poderíamos também pensar em outra modelagem, onde pedidos e clientes fossem agregados separados

Essa modelagem auxiliaria na leitura de informações de pedidos, mas cria uma referência.



Modelando para acesso aos dados

Resumindo: Existem inúmeras formas de modelar os dados utilizando agregados.

Ao fazer a modelagem, devemos nos preocupar em como a leitura será realizada.

Bancos de dados sem esquema

Uma outra propriedade da maioria dos bancos NoSQL é que são bancos sem esquema.

Diferente do relacional, onde o esquema deve ser definido no início;

Em bancos com esquema, você deve definir a estrutura antes de utilizar, no bancos sem esquema há uma **maior flexibilidade**.

- Podemos dizer que os bancos têm esquema flexível

Bancos sem esquema

Não predefinir um esquema gera alguns benefícios, mas trás alguns problemas.

Perdemos um pouco de padronização:

- Exemplo: em um campo idade podemos usar diferentes tipos.

Bancos sem esquema

Na prática não existe banco sem esquema, pois há um **esquema implícito** que é dado pela aplicação.

A responsabilidade de gerenciar o esquema passa para a aplicação

Existem algumas soluções para auxiliar:

- Encapsular a interação do banco em um webservice
- Delimitar diferentes áreas do agregado para serem acessadas por aplicações diferentes.

Modelos de dados

Os bancos NoSQL são classificados em quatro modelos de dados:

- Chave-valor
- Documentos
- Família de Colunas
- Grafos

Os três primeiros são orientados a agregados

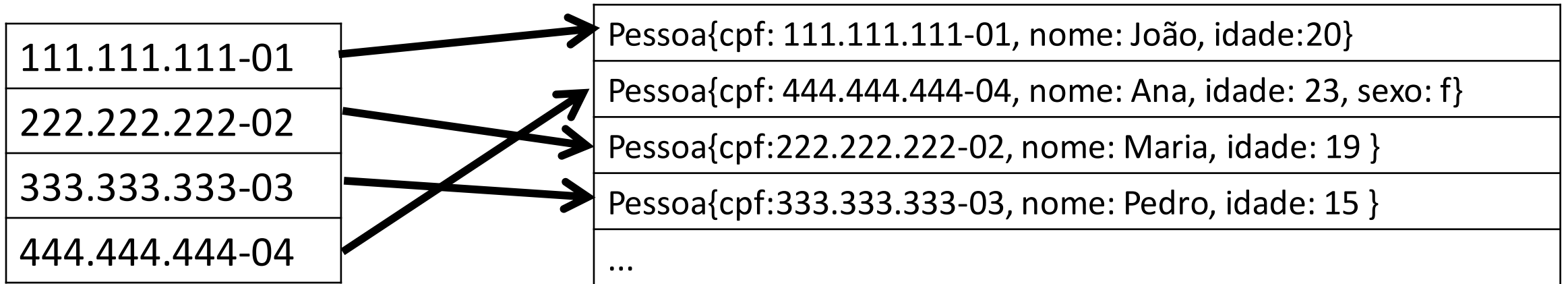
Chave-valor

Funciona como uma espécie de mapa, onde uma chave guarda referência a um agregado de dados

As chaves são utilizadas para recuperar os valores.

O banco não enxerga estrutura no agregado

Chave-valor



Documentos

A unidade básica de armazenamento é o **documento**.

Os documentos são indexados e há mecanismos de consulta, diferentemente dos bancos chave-valor

O documento pode ser um JSON, BSON, XML, ...

Documentos

Exemplo de documento:

"Assunto" : "Banco de Dados"

"Autor" : "Paulo Freitas"

"Data" : "01/02/2017"

"Palavras-chave" : ["NoSQL", "bancos de dados", "MongoDB"]

"Body" : "O MongoDB é um exemplo de banco de documentos"

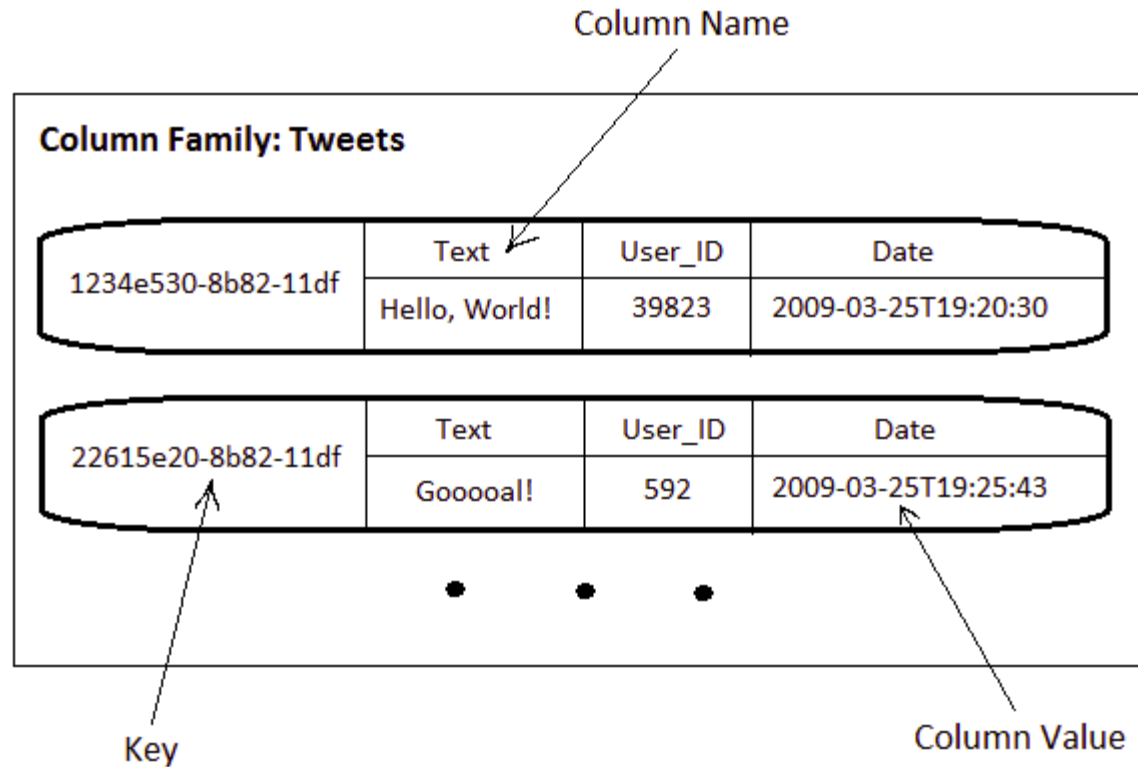
Família de colunas

Os dados agregados são representados como famílias de colunas

As famílias de colunas podem ser diferentes

Há uma chave de linha que leva à coluna.

Família de colunas



Grafos

Os bancos de grafo são uma categoria a parte dos bancos NoSQL

Há um foco maior nos relacionamentos entre os dados

O banco armazena além das informações, os relacionamentos.

Grafos

