

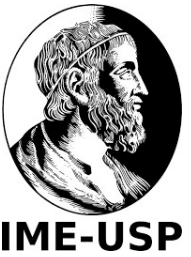
Modelos de Dados para Sistemas NoSQL

Parte 1

Kelly Rosa Braghetto
kellyrb@ime.usp.br

Departamento de Ciência da Computação

Sistemas NoSQL – Modelos de Dados



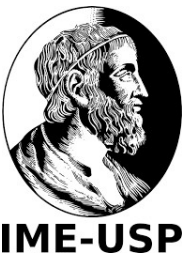
Baseado no livro

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

de Pramod Sadalage e Martin Fowler

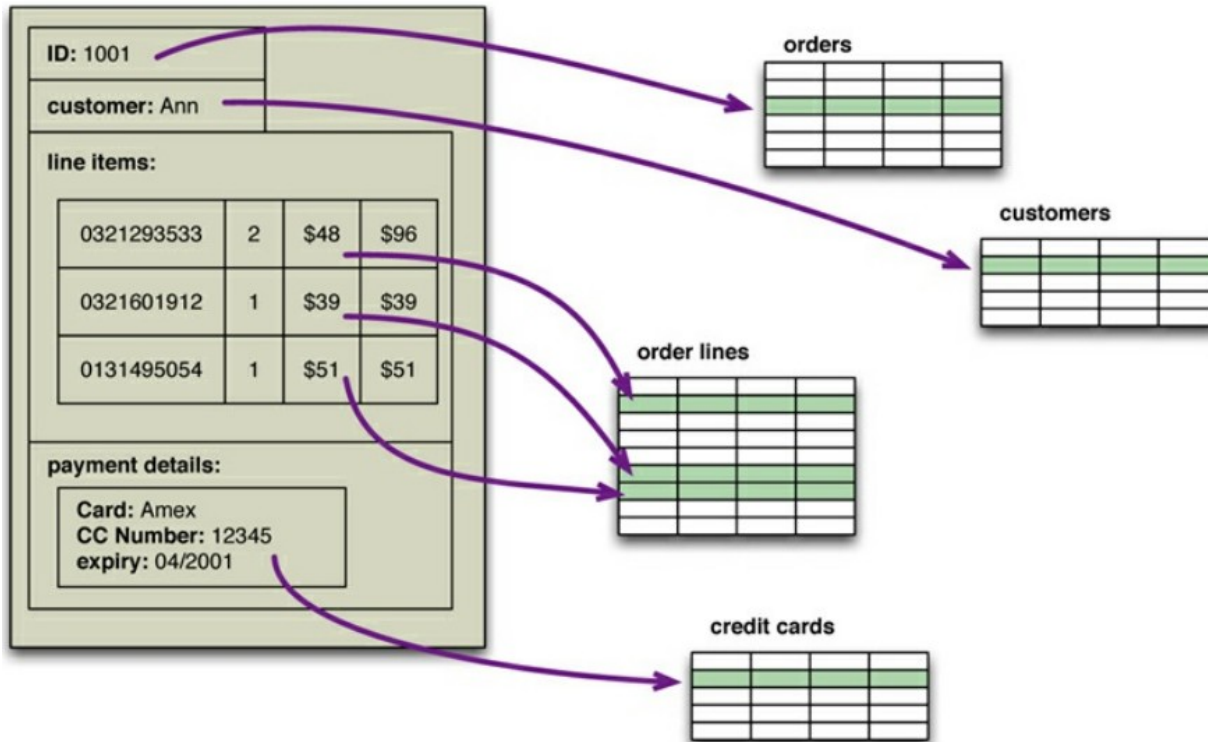
Contexto:

Bancos de Dados Relacionais



- Modelo relacional: 1970
 - Primeiras implementações: a partir 1974
- Um **BD relacional** é um conjunto de **relações**
- Cada **relação** é um conjunto de **tuplas**
- Cada **tupla** é um conjunto de valores para **atributos**
 - Atributos são sempre simples e atômicos
- Uma tupla pode representar uma **entidade** ou **relacionamento**
- Um relacionamento é expresso por um atributo que **referencia** uma tupla da mesma relação ou de uma outra relação
- A SQL é a linguagem de consulta padrão para BDs relacionais

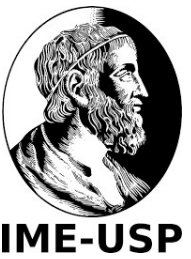
Modelo de Dados Relacional – Limitações



- Diferença entre a representação no modelo relacional e as estruturas de dados em memória principal
- Não projetado para uso em aglomerados de máquinas (*clusters*)

Anos 2000: demanda crescente por mais flexibilidade e escalabilidade, principalmente para dados dos sistemas web

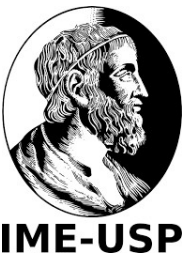
Sistemas NoSQL



Sistemas de banco de dados que não usam o modelo de dados relacional

- Outras características comuns:
 - Bom desempenho em operações simples de leitura e escrita
 - Particionamento e replicação de dados
 - Escalabilidade horizontal
 - Ausência de esquema fixo
 - APIs próprias de cada sistema
 - Ausência de linguagens de consulta e manipulação de dados padronizadas

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

Agregados



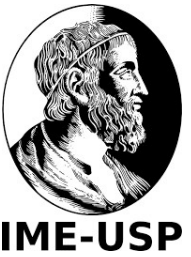
Agregado: conjunto de objetos relacionados, tratados como uma **unidade de dados**

- Origem: *Domain Driven Design*
 - Abordagem para desenvolvimento de software
- Um agregado pode ter uma estrutura mais complexa que a de uma tupla relacional, pois permite **aninhamento**
- Os agregados são as **unidades para manipulação** dos dados e **gerenciamento da consistência** nos sistemas NoSQL
 - A comunicação com o componente de armazenamento dos dados é feita em termos de agregados
 - A modificação de um agregado é implementada como uma operação atômica

Agregados facilitam o funcionamento dos Sistemas NoSQL em aglomerados de máquinas (*clusters*)

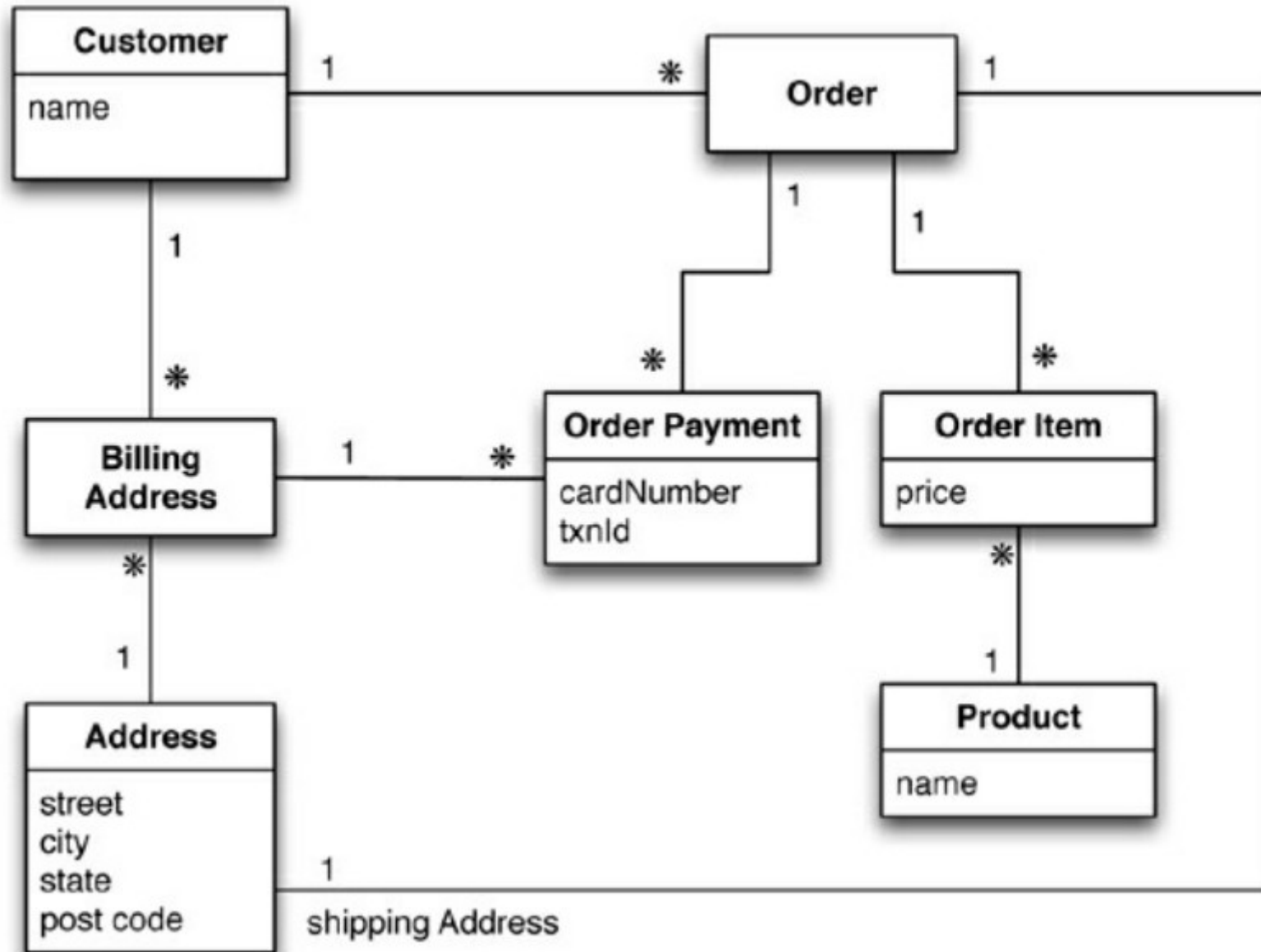
- Nesses sistemas, deseja-se minimizar o número de nós que precisam ser acessados para se responder uma consulta
- Um agregado indica quais dados são manipulados juntos e que, portanto, deveriam ficar em um mesmo nó
 - Unidade natural para replicação e particionamento
- Agregados também facilitam o desenvolvimento de aplicações
 - Possuem uma estrutura mais próxima da dos objetos de dados em memória manipulados por elas

Exemplo: Comércio eletrônico



- Necessidade de armazenar dados de:
 - Clientes
 - Catálogo de produtos
 - Pedidos
 - Endereços de entrega
 - Endereços de faturamento
 - Pagamentos

Exemplo: Esquema relacional do BD (em notação UML)



Exemplo: o BD relacional



IME-USP

Customer	
Id	Name
1	Martin

Orders		
Id	CustomerId	ShippingAddressId
99	1	77

Product	
Id	Name
27	NoSQL Distilled

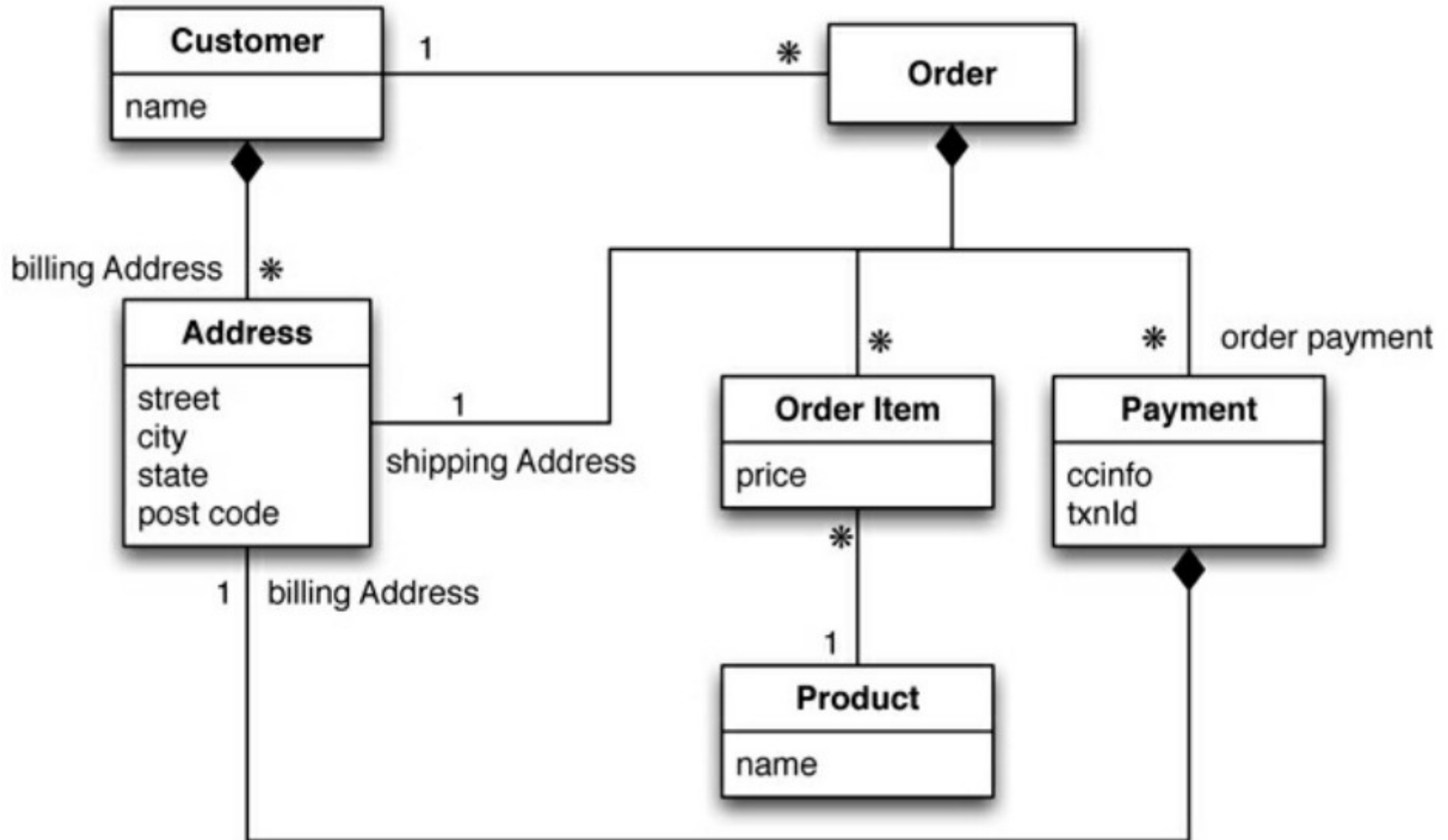
BillingAddress		
Id	CustomerId	AddressId
55	1	77

OrderItem			
Id	OrderId	ProductId	Price
100	99	27	32.45

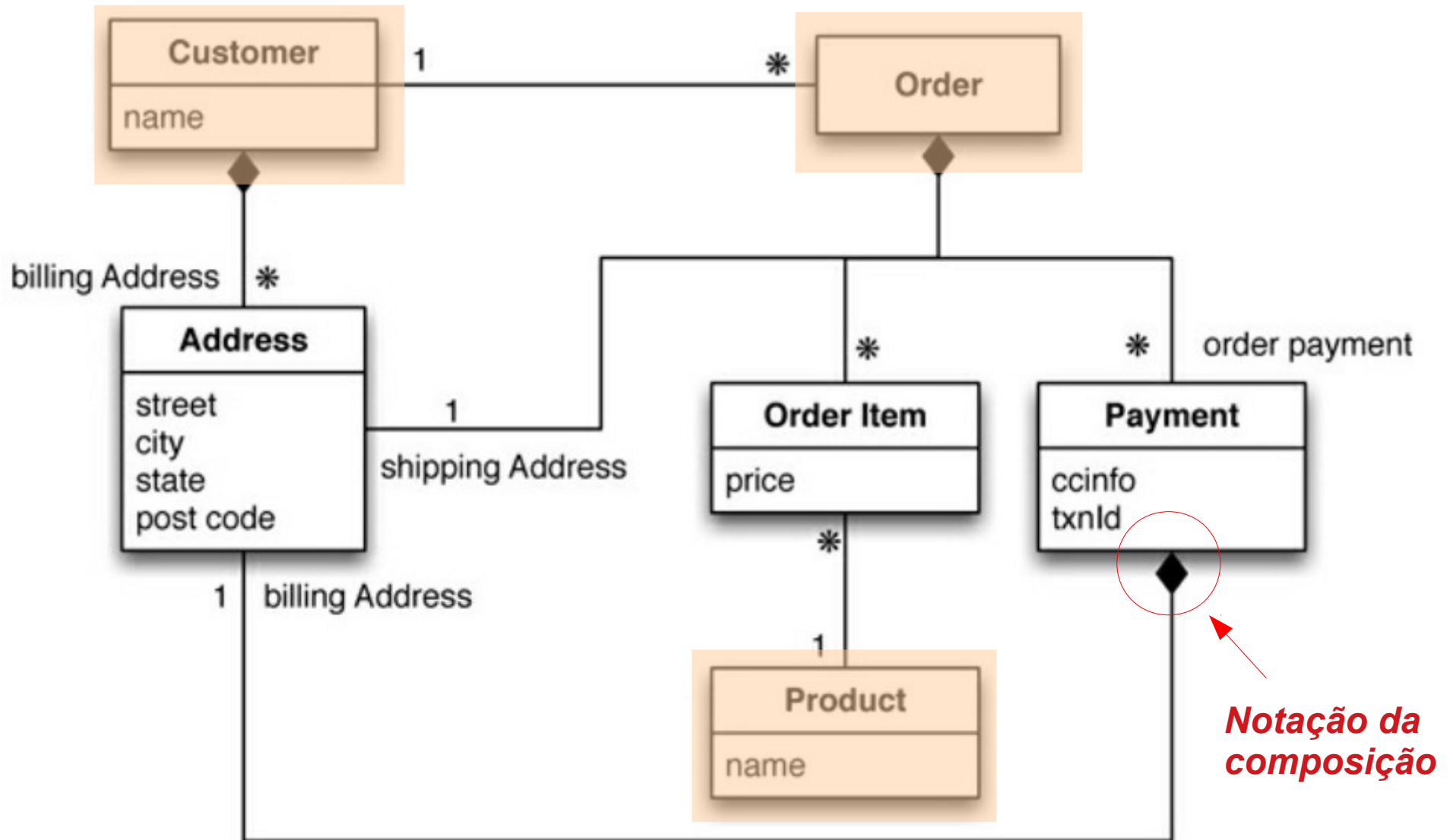
Address	
Id	City
77	Chicago

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

Exemplo: Esquema de agregados (em notação UML)



Exemplo: Esquema de agregados (em notação UML)



Três tipos de agregados: **Customer**, **Order** e **Product**

Instância do esquema de agregados



IME-USP

```
// in customers
{
  "id":1,
  "name":"Martin",
  "billingAddress":[{"city":"Chicago"}]
}

// in orders
{
  "id":99,
  "customerId":1,
  "orderItems":[
    {
      "productId":27,
      "price": 32.45,
      "productName": "NoSQL Distilled"
    }
  ],
  "shippingAddress":[{"city":"Chicago"}]
  "orderPayment":[
    {
      "ccinfo":"1000-1000-1000-1000",
      "txnId":"abelif879rft",
      "billingAddress": {"city": "Chicago"}
    }
  ],
}
```

→ Denormalização de *Product*

Ainda sobre o exemplo



Como ocorre em qualquer tipo de BD, um dado agrupamento de dados pode favorecer algumas operações enquanto atrapalha outras

- **Por exemplo, o agregado “order”:**
 - Ajuda quando um usuário precisa consultar os detalhes de um pedido seu ou quando o comerciante está “processando” o pedido para a entrega
 - **Recuperando um só agregado se responde à consulta**
 - Atrapalha quando deseja-se saber o histórico de vendas dos produtos:
 - Para responder essa consulta, pode ser preciso percorrer todos os agregados de pedidos armazenados no BD

Instância do esquema de agregados

```
// in customers
{
  "id":1,
  "name":"Martin",
  "billingAddress":[{"city":"Chicago"}]
}

// in orders
{
  "id":99,
  "customerId":1,
  "orderItems":[
    {
      "productId":27,
      "price": 32.45,
      "productName": "NoSQL Distilled"
    }
  ],
  "shippingAddress":[{"city":"Chicago"}]
  "orderPayment":[
    {
      "ccinfo":"1000-1000-1000-1000",
      "txnId":"abelif879rft",
      "billingAddress": {"city": "Chicago"}
    }
  ],
}
```

► **Detalhes dos itens inclusos
são recuperados junto com
os demais dados do pedido**

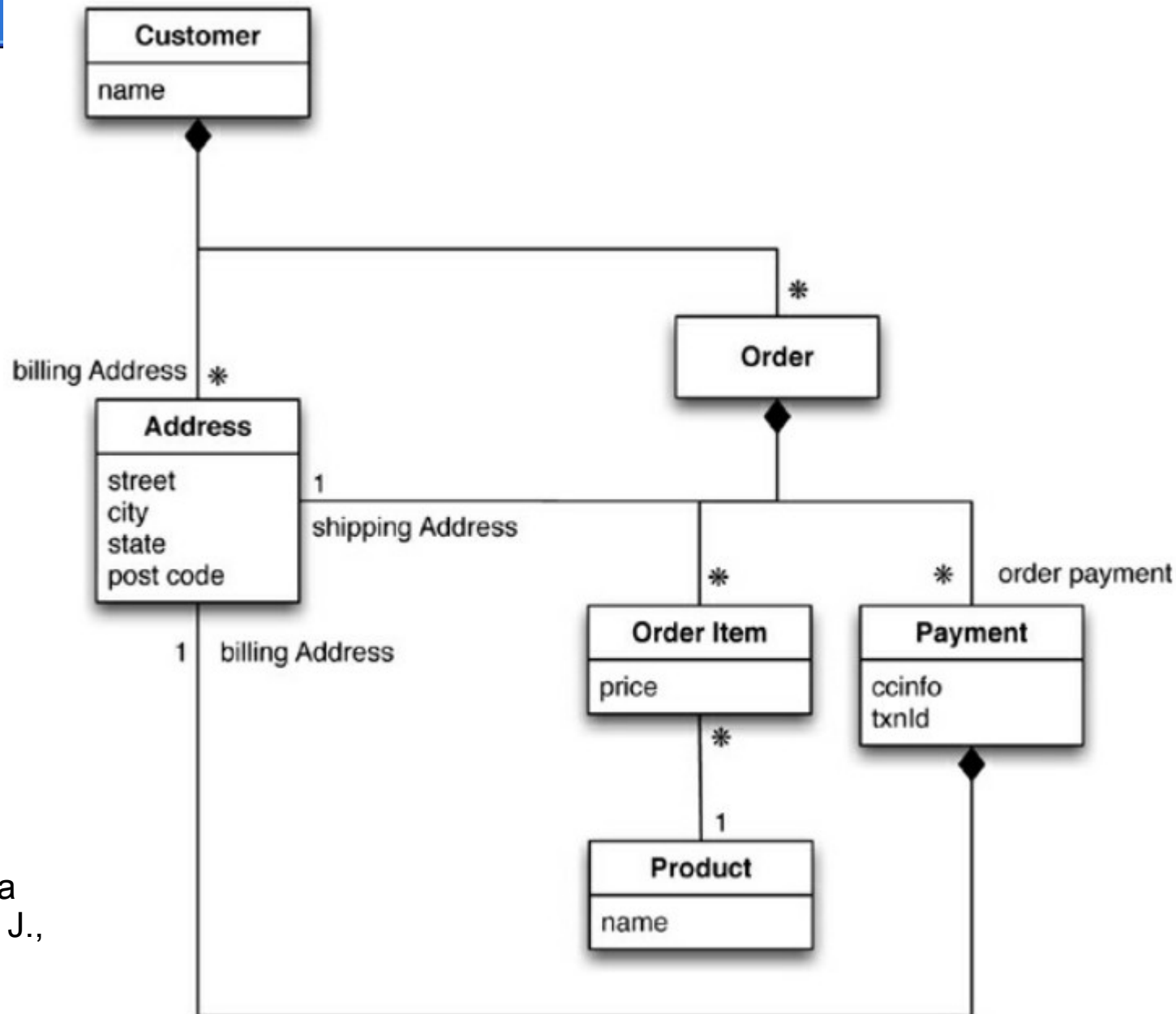
Ainda sobre o exemplo



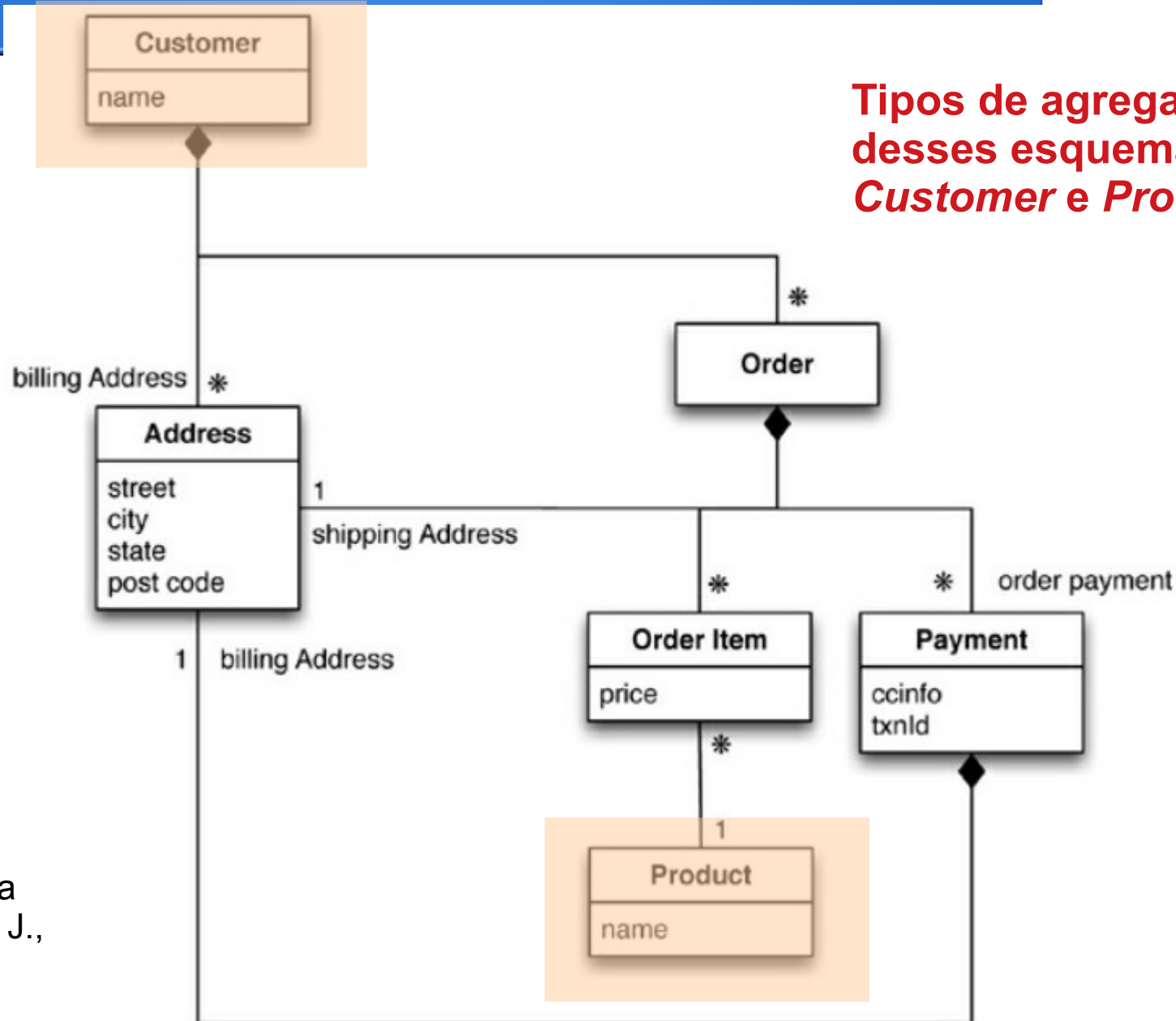
Como ocorre em qualquer tipo de BD, um dado agrupamento de dados pode favorecer algumas operações enquanto atrapalha outras

- **Por exemplo, o agregado “order”:**
 - Ajuda quando um usuário precisa consultar os detalhes de um pedido seu ou quando o comerciante está "processando" o pedido para a entrega
 - Recuperando um só agregado se responde à consulta
 - **Atrapalha quando deseja-se saber o histórico de vendas dos produtos:**
 - **Para responder essa consulta, pode ser preciso percorrer todos os agregados de pedidos armazenados no BD**

Exemplo: Esquema de agregados alternativo



Exemplo: Esquema de agregados alternativo



Instância do esquema de agregados alternativo



```
{
  "customer": {
    "id": 1,
    "name": "Martin",
    "billingAddress": [{"city": "Chicago"}],
    "orders": [
      {
        "id": 99,
        "orderItems": [
          {
            "productId": 27,
            "price": 32.45,
            "productName": "NoSQL Distilled"
          }
        ],
        "shippingAddress": [{"city": "Chicago"}],
        "orderPayment": [
          {
            "ccinfo": "1000-1000-1000-1000",
            "txnId": "abelif879rft",
            "billingAddress": {"city": "Chicago"}
          }
        ]
      }
    ]
  }
}
```

→ **Denormalização de *Products***

Relacionamentos e agregados



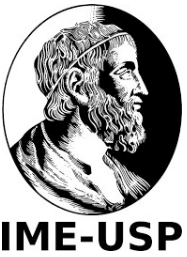
- Em BDs relacionais, não é possível distinguir os relacionamentos de agregação dos demais
 - Consequência: SGBDs relacionais não tem conhecimento sobre os agregados para apoiar o armazenamento e distribuição dos dados
- Em BDs orientados a agregados, o foco é a unidade de interação com o gerenciador de armazenamento de dados
 - O agregado não representa uma organização lógica “intrínseca” aos dados
 - Expressa a forma como os dados são usados pelas aplicações

Mais sobre os agregados



- Não há um conceito de “chave estrangeira” para impor integridade aos relacionamentos entre agregados
- A estrutura interna dos agregados não é “vista” nas decisões sobre armazenamento e distribuição dos dados
- Em muitos sistemas NoSQL, não é possível manipular múltiplos agregados de forma atômica
- Quando há muitas aplicações acessando o mesmo BD NoSQL, pode ser difícil estruturar agregados que beneficiem todas ao mesmo

Resumo



- Sistemas NoSQL usam diferentes tipos de modelos de dados
- O modelo empregado pela maioria pode ser generalizado pelo modelo de agregados
- Um agregado reúne objetos relacionados, que precisam ser tratados como uma unidade
- A distribuição, o armazenamento e o controle de consistência em sistemas NoSQL são feitos em termos de agregados
- O modelo de agregados facilita o uso de aglomerado de máquinas

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.