

NoSQL

Fundamentos de Banco de Dados

Modelo de Banco de Relacional

- Vantagens
 - Altamente estruturado (tabelas, SQL)
 - Altamente portável
 - Tecnologia madura
 - Alta performance para Transações ACID

Modelo de Banco de Relacional

- Alta performance para Transações ACID

Atomicidade – uma transação deve ser realizada em sua totalidade ou não ser realizada de forma alguma

Consistência – uma transação deve preservar consistência, levando o BD de um estado consistente para outro, se for completamente executada do início ao fim e sem interferência de outras transações

Isolamento – a execução de uma transação não deve ser interferida por quaisquer outras transações que ocorram simultaneamente

Durabilidade (ou Permanência) – as mudanças aplicadas ao BD por uma transação confirmada precisam persistir no BD (ou seja, não podem ser perdidas em caso de falhas)

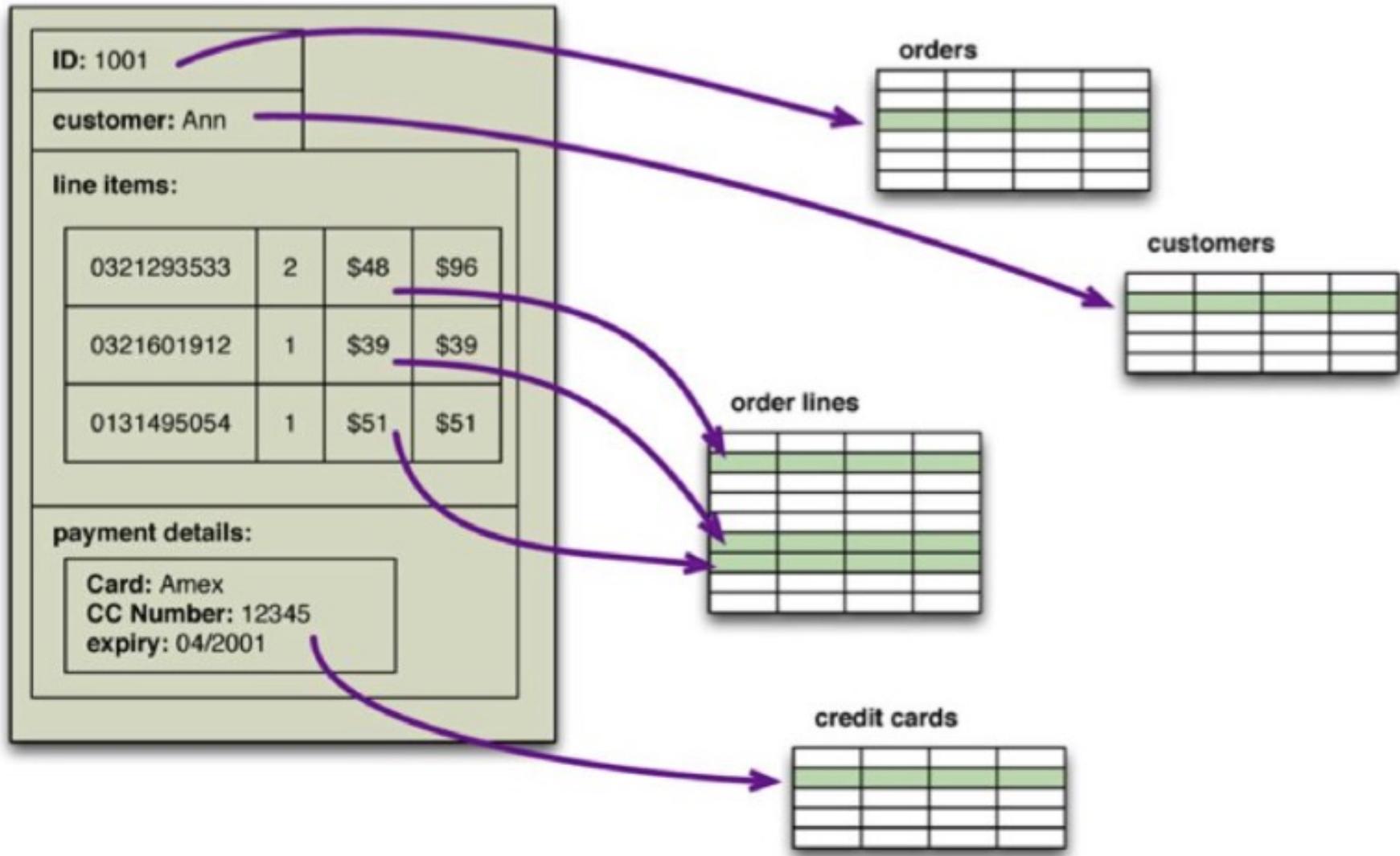
Modelo de Banco de Relacional

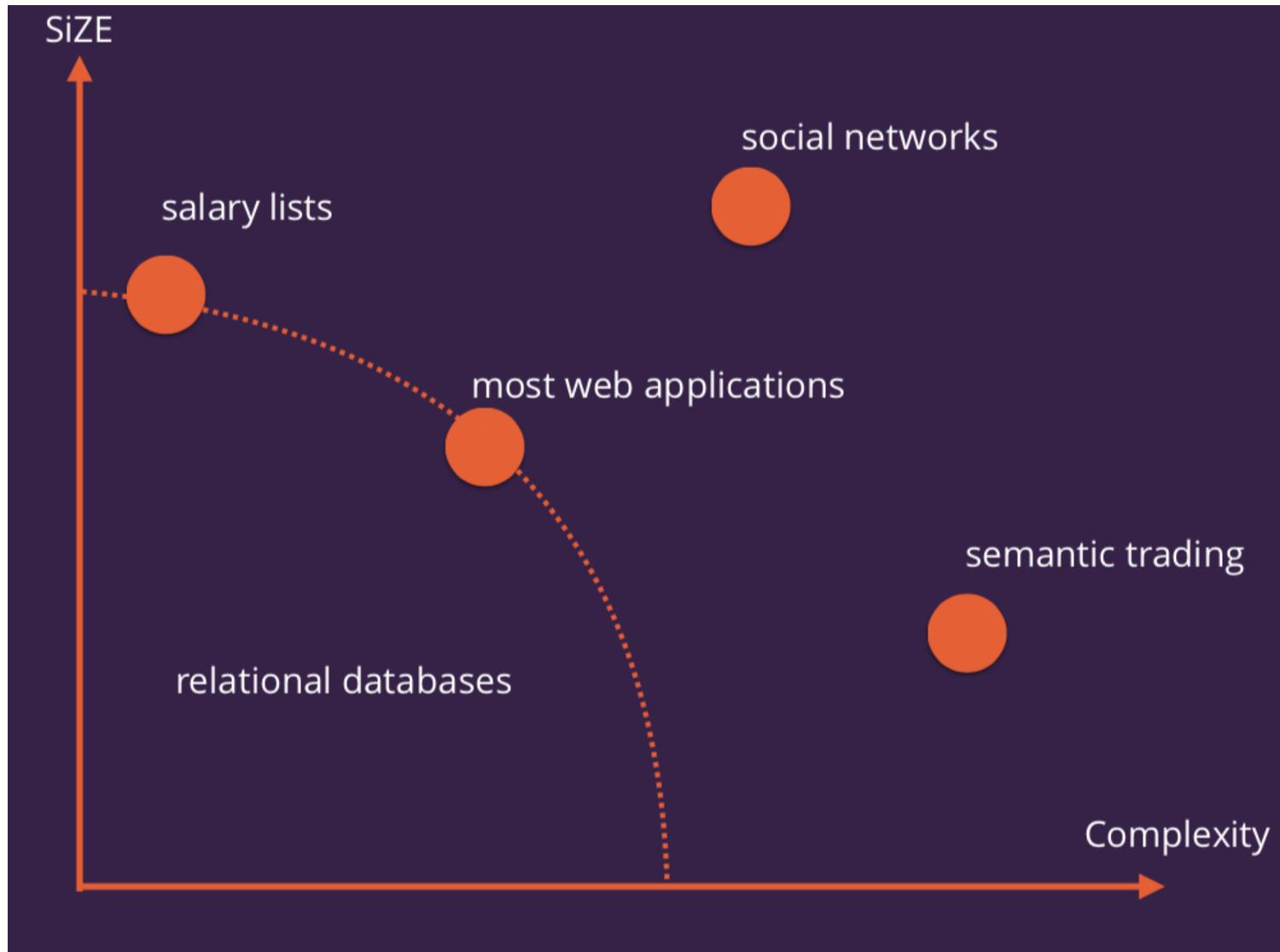
- Desvantagens
 - Impedance mismatch
 - Consultas demoradas para modelos complexos
 - Modelo com estrutura rígida
 - Escalabilidade parcial

Impedance Mismatch

- Principal desvantagem do Modelo Relacional
- BDs relacionais
 - Modelo normalizado – tabelas distintas
 - Sem listas ou registros aninhados
- Tradução entre os modelos
 - Custo
 - Limita a produtividade de desenvolvedores

Impedance Mismatch - Exemplo





Modelo de Banco de Dados NoSQL

- Vantagens:
 - Modelo de dados flexível
 - Apropriado para dados complexos
 - Melhor escalabilidade
 - Necessidades das aplicações
 - Melhor performance

Modelo de Banco de Dados NoSQL

- Desvantagens:
 - Falta de padronização (dificulta portabilidade)
 - Uma tecnologia ainda em desenvolvimento

Modelo Relacional x NoSQL

Bancos relacionais	Bancos NoSQL
Usam relacionamentos entre tabelas.	Não usam relacionamentos entre tabelas.
Usam esquemas fixos.	Não usam esquemas.
Pouca flexibilidade de escalonamento.	Fácil de escalar.
Proporcionam grande consistência.	Pouca consistência.
Oferecem dificuldade de disponibilidade.	Atendem maior distribuição/disponibilidade dos dados.

NoSQL

- Primeira aparição do nome NoSQL
 - SGBRelacional open-source Strozzi NoSQL
 - Conjunto de arquivos ASCII
 - Manipulados por scripts e não por SQL
- Aparição oficial do nome NoSQL
 - Reunião Junho/2009/San Francisco/Johan Oskarsson
 - Modelo alternativo de armazenamento de dados
 - Nome coletado em um canal de IRC para a reunião
 - NoSQL Meetup

NoSQL

- Não tem definição formal
- Características comuns das aplicações:
 - Não usa o modelo relacional
 - Necessidade de executarem em clusters
 - Baseados nas necessidades da WEB
 - Sem esquema rígido

NoSQL

- Modelos de dados
 - Chave-Valor
 - Documento
 - Grafo

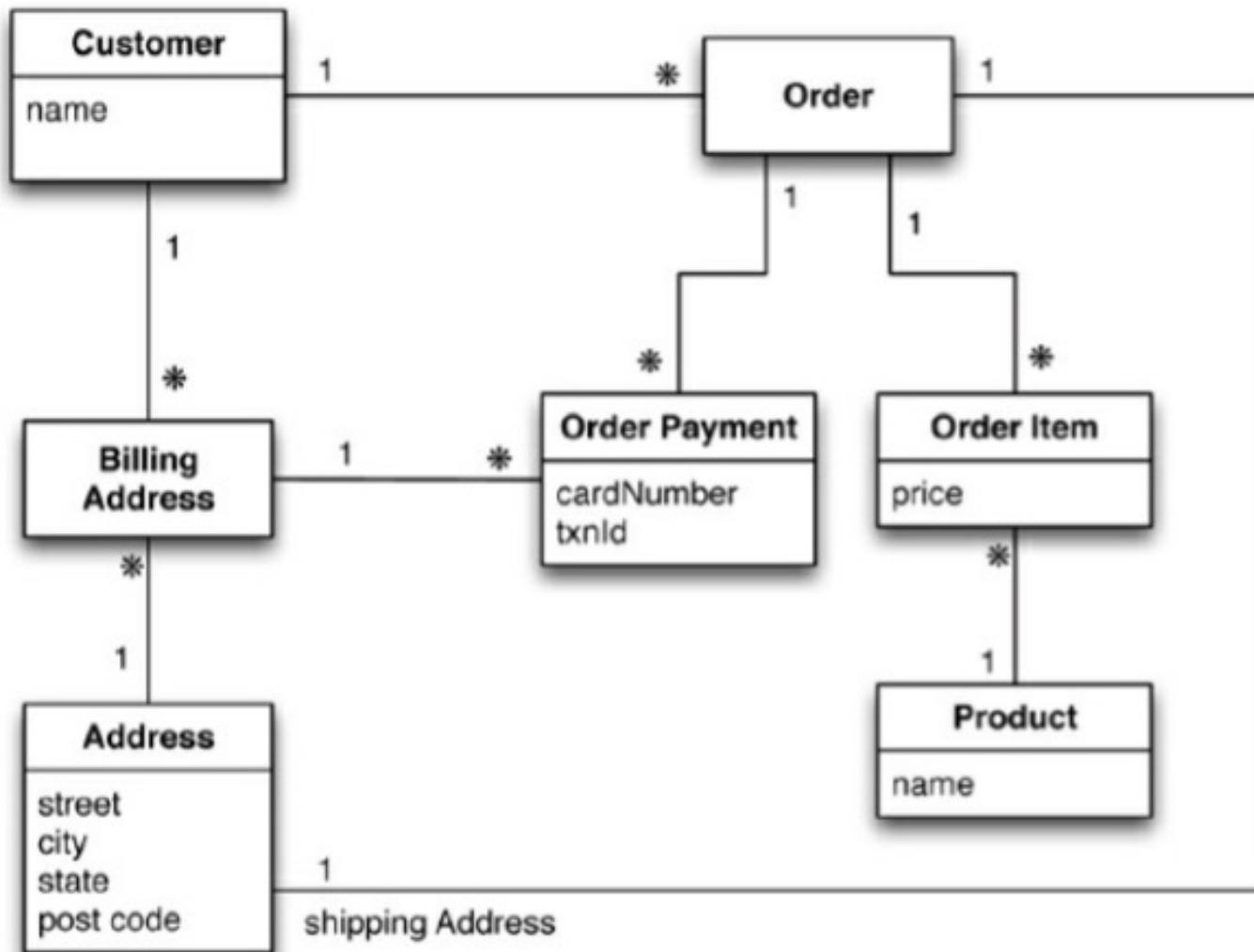
Comparando os modelos

Relacional

X

NoSQL

Modelo Relacional



Modelo Relacional

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

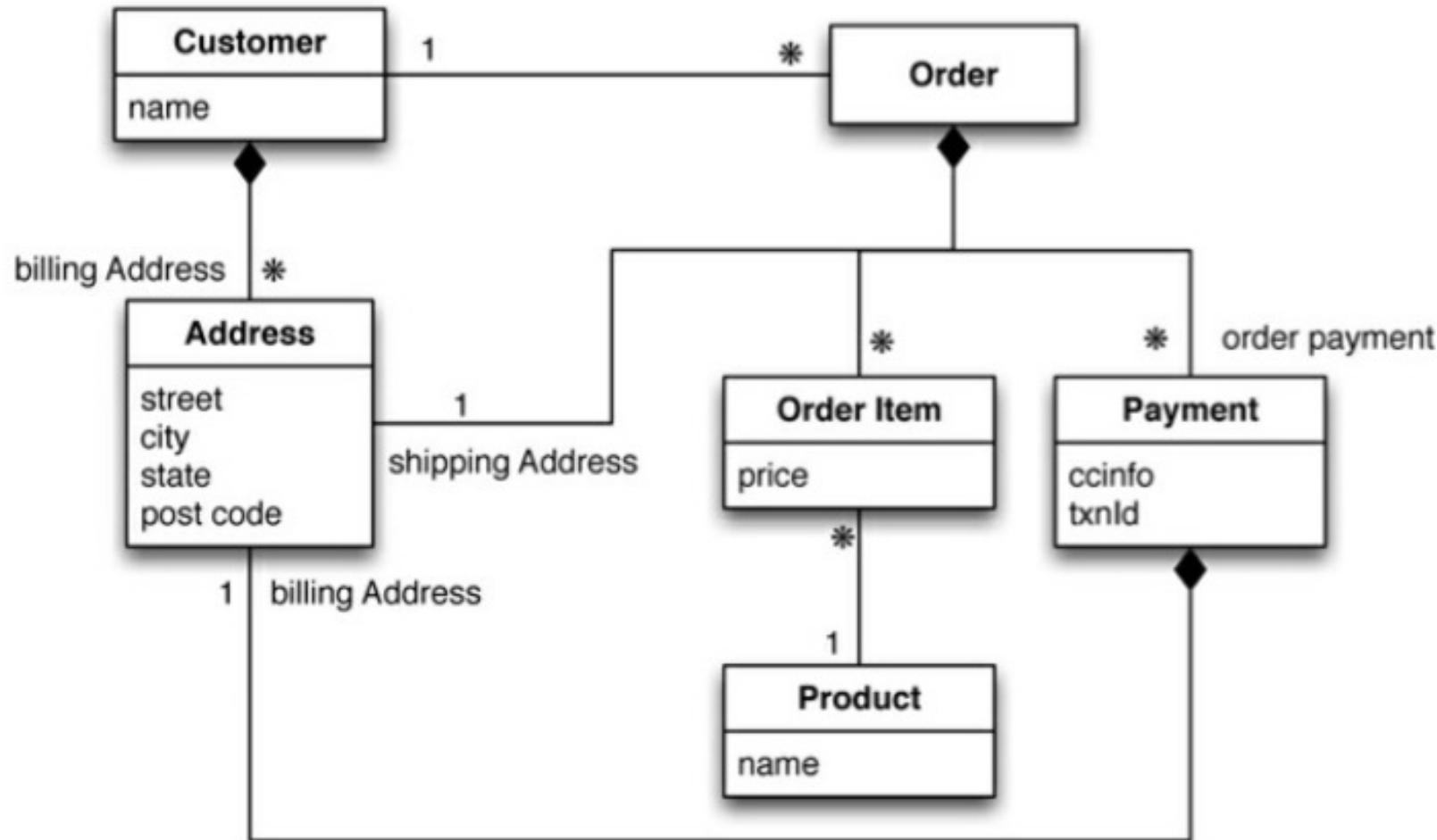
OrderPayment				
Id	OrderId	CardNumber	BillingAddressId	txnId
33	99	1000-1000	55	abelif879rft

Modelo a ser escolhido

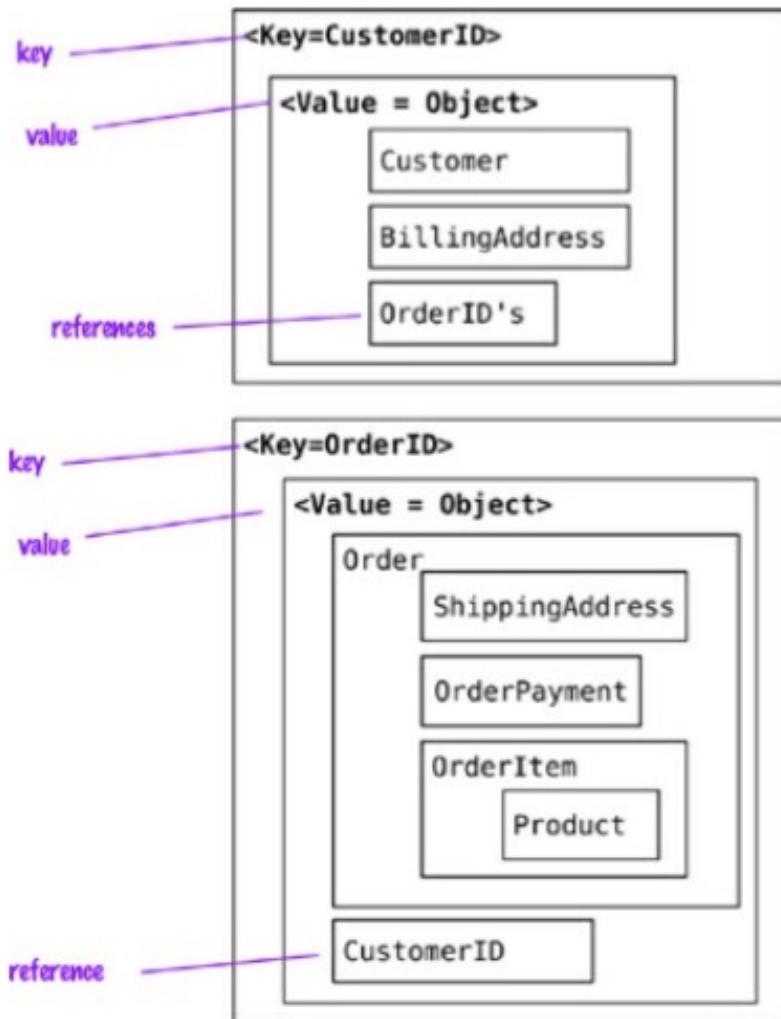
- Escolha:
 - necessidades da aplicação/distribuição/desempenho
 - forma de acesso aos dados

Chave-valor

Modelo Agregado A



Chave-valor



```
// 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"}
    }
  ],
}
```

Chave-valor

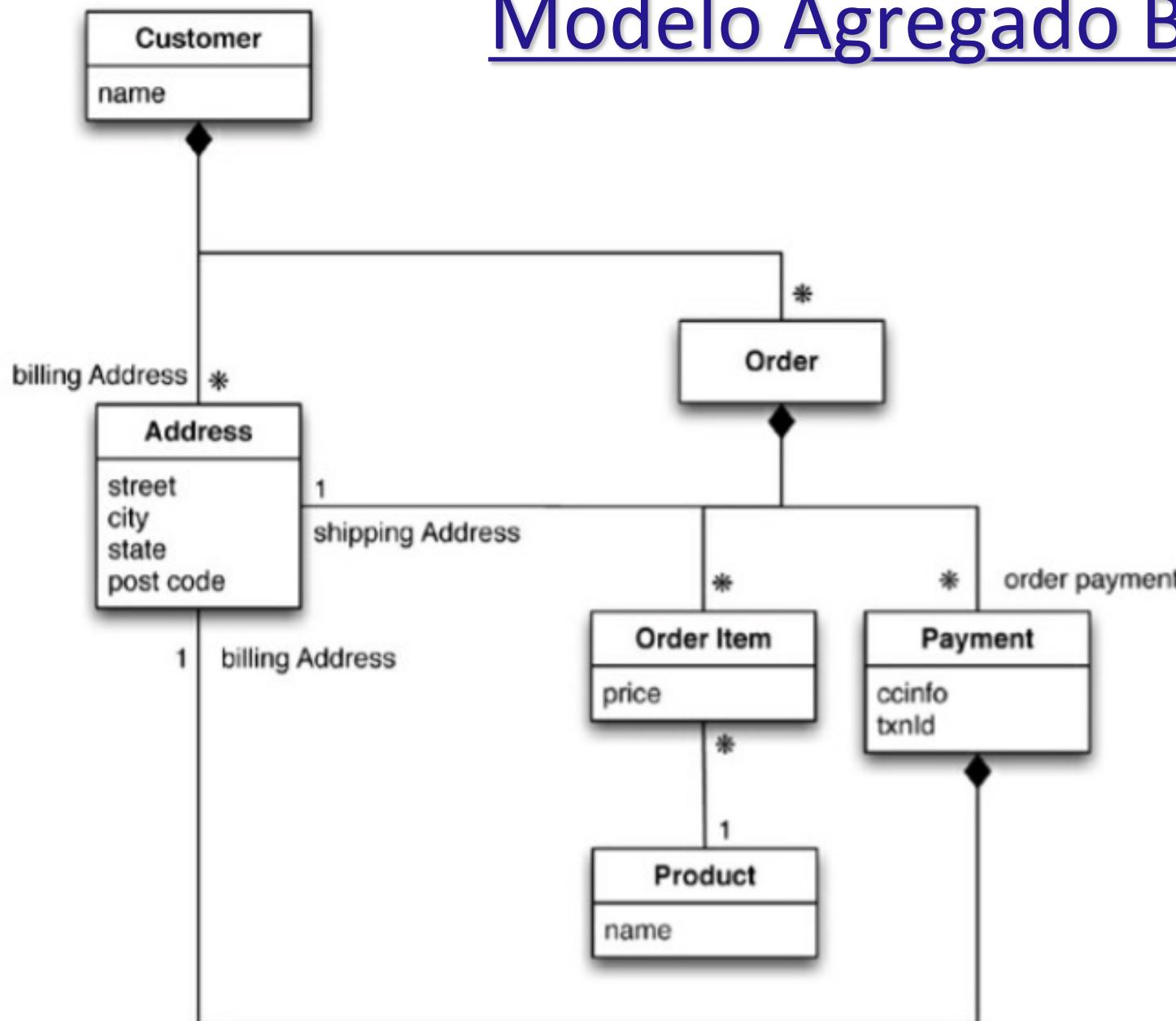
Modelo Agregado A

*JSON: JavaScript Object Notation

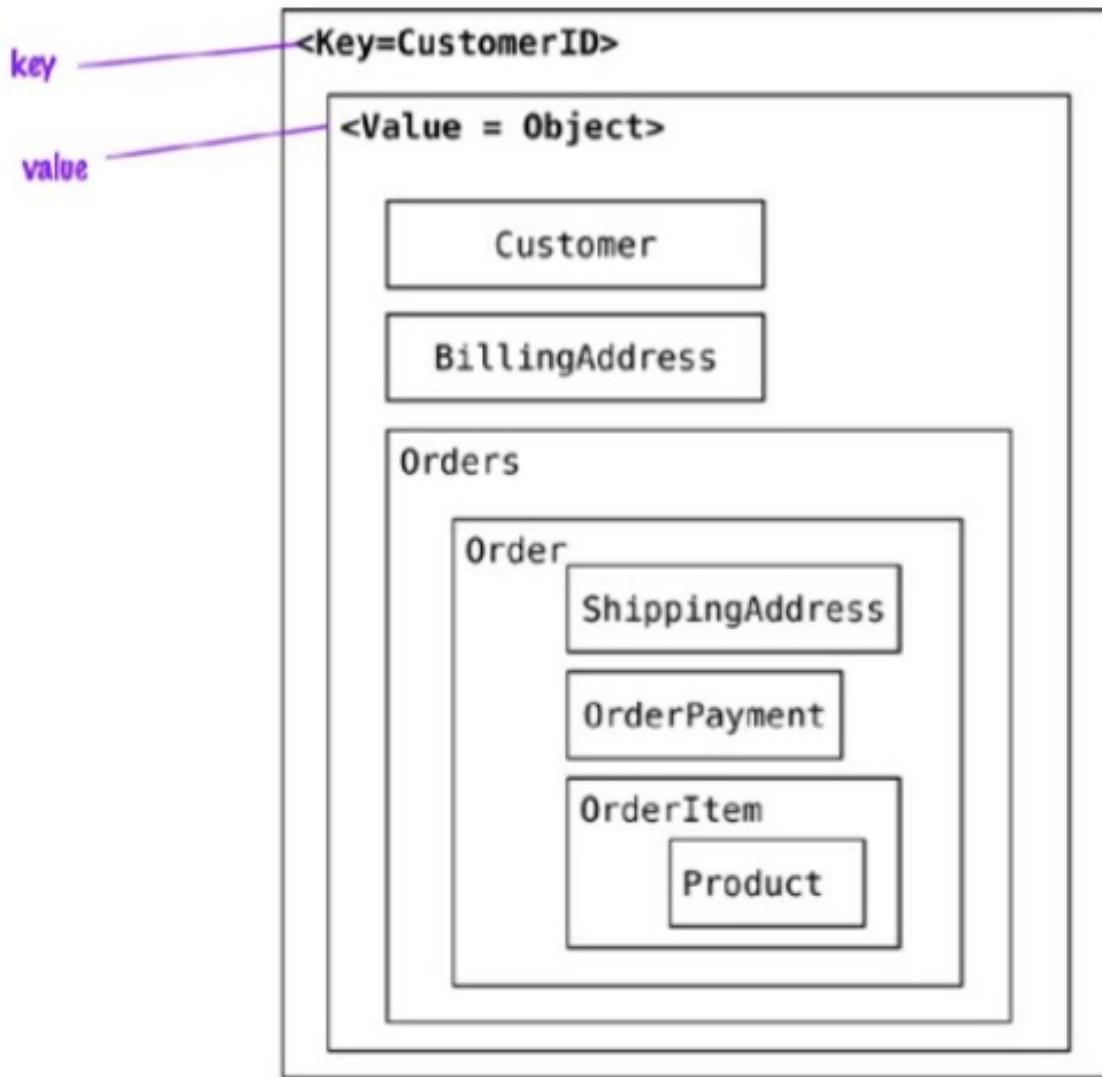
formato de padrão aberto, em texto legível para humanos, para transmitir dados que consistem de pares atributo/valor.

Chave-valor

Modelo Agregado B



Chave-valor Modelo Agregado B



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

Chave-valor Modelo Agregado B

*JSON: JavaScript Object Notation

formato de padrão aberto, em texto legível para humanos, para transmitir dados que consistem de pares atributo/valor.

Amazon DynamoDB

```
{  
    TableName : "Music",  
    KeySchema: [  
        {  
            AttributeName: "Artist",  
            KeyType: "HASH", //Partition key  
        },  
        {  
            AttributeName: "SongTitle",  
            KeyType: "RANGE" //Sort key  
        }  
    AttributeDefinitions: [  
        {  
            AttributeName: "Artist",  
            AttributeType: "S"  
        },  
        {  
            AttributeName: "SongTitle",  
            AttributeType: "S"  
        }  
    ProvisionedThroughput: {  
        ReadCapacityUnits: 1,  
        WriteCapacityUnits: 1  
    }  
}
```

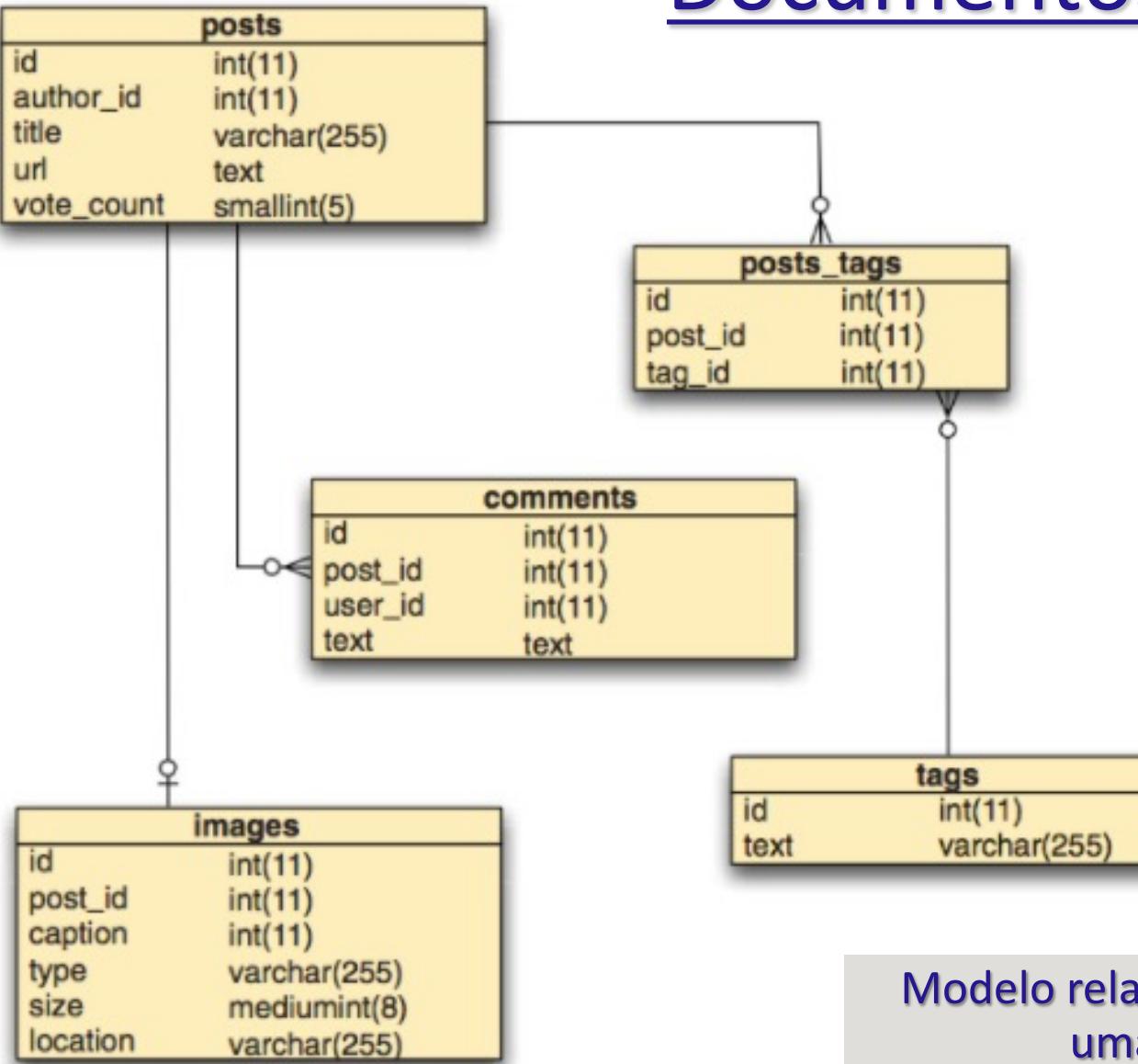
Chave-valor

- Principais usos:
 - Dados acessados por uma chave única
 - Sessão em um sistema
 - Perfil de usuário
 - Carrinho de compras

```
// in order collection
{
  "customerId":12345,
  "orderId":67890,
  "orderDate":"2012-12-06",
  "items": [
    {
      "product": {
        "id":112233,
        "name":"Refactoring",
        "price":"15.99"
      },
      "discount":"10%"
    },
    {
      "product": {
        "id":223344,
        "name":"NoSQL Distilled",
        "price":"24.99"
      },
      "discount":"3.00",
      "promo-code":"cybermonday"
    },
    ...
  ]
}
```

Documentos

Documentos



Modelo relacional para posts em
uma rede social

imagem retirada do livro MongoDB in Action

Documentos

Listing 1.1 A document representing an entry on a social news site

```
{ _id: ObjectId('4bd9e8e17cefd644108961bb'),  
  title: 'Adventures in Databases',  
  url: 'http://example.com/databases.txt',  
  
  author: 'msmith',  
  vote_count: 20,  
  tags: ['databases', 'mongodb', 'indexing'],  
  image: {  
    url: 'http://example.com/db.jpg',  
    caption: '',  
    type: 'jpg',  
    size: 75381,  
    data: "Binary"  
  },  
  comments: [  
    { user: 'bjones',  
      text: 'Interesting article!'  
    },  
  
    { user: 'blogger',  
      text: 'Another related article is at http://example.com/db/db.txt'  
    }  
  ]  
}
```

1 **Tags stored as array of strings**

2 **Attribute points to another document**

3 **Comments stored as array of comment objects**

Documento representando um post na rede social

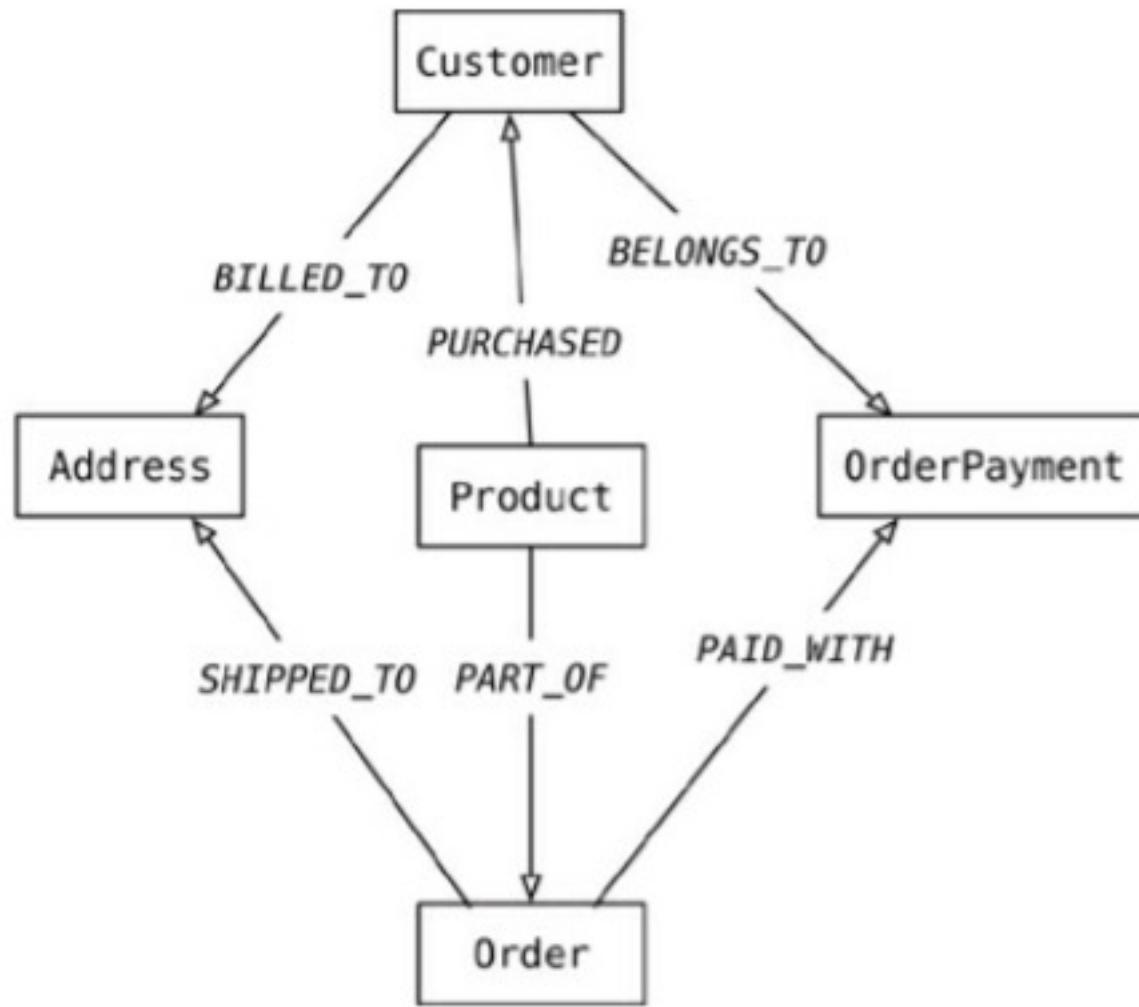
```
// in order collection
{
  "customerId":12345,
  "orderId":67890,
  "orderDate":"2012-12-06",
  "items": [
    {
      "product": {
        "id":112233,
        "name":"Refactoring",
        "price":"15.99"
      },
      "discount":"10%"
    },
    {
      "product": {
        "id":223344,
        "name":"NoSQL Distilled",
        "price":"24.99"
      },
      "discount":"3.00",
      "promo-code":"cybermonday"
    }
  ]
}
```

SQL	Document Database Query
select * from order	db.order.find()
select * from order where customerId = 12345	db.order.find({ "customerId":12345 })
select orderId, orderDate from order where customerId = 12345	db.order.find({"customerId":12345}, {"orderId":1,"orderDate":1})
select * from order o join orderItem oi on o.orderId = oi.orderID join product p on oi.productId = p.Id where p.name like '%Refactoring%'	db.order.find({ "items.product.name": "/Refactoring/" })

Documentos

- Principais usos:
 - Histórico de eventos
 - Cada evento com seus atributos
 - Gerenciamento de conteúdo
 - Plataformas de blogs

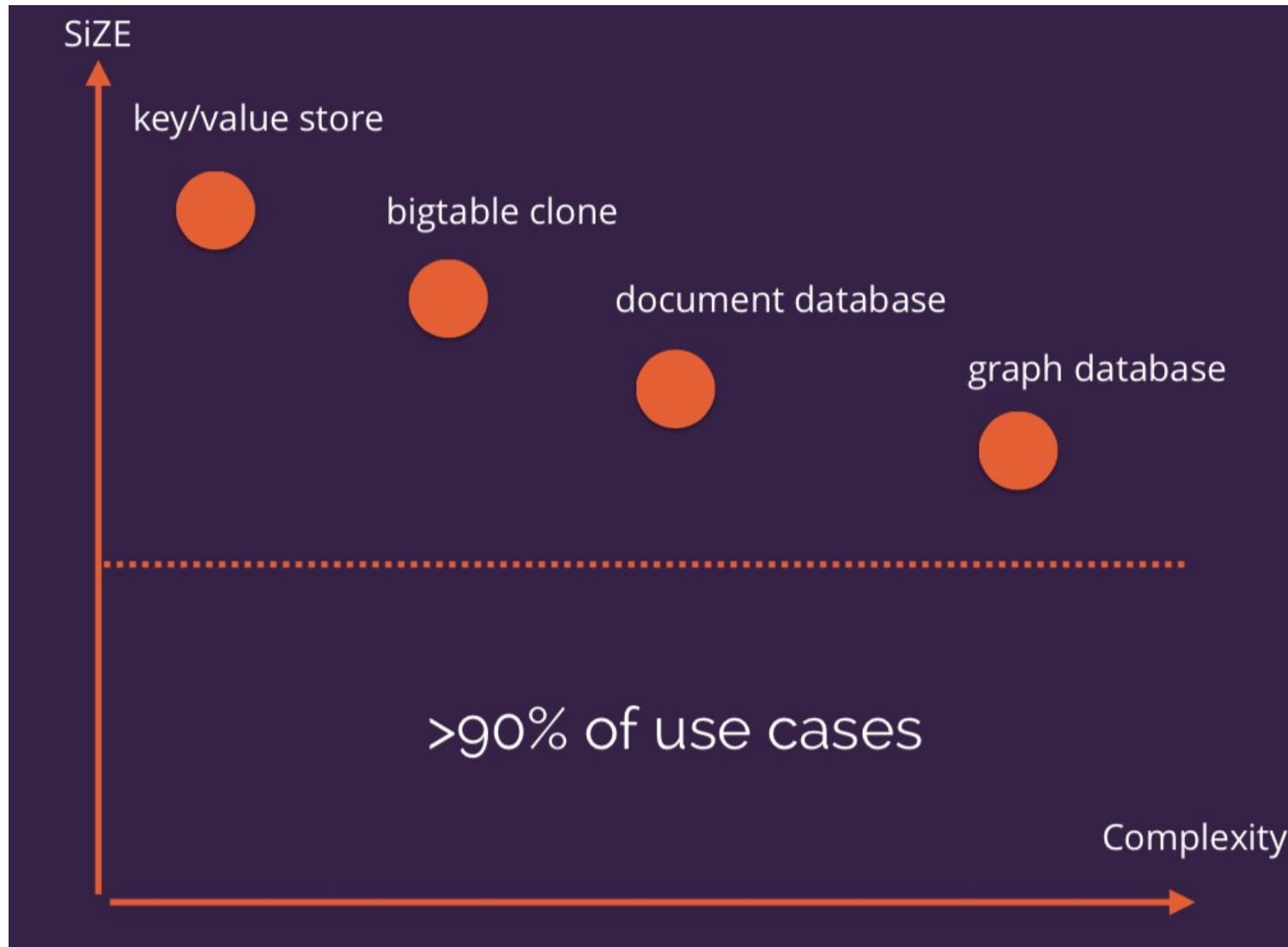
Grafo



Grafos

- Principais usos:
 - Redes de amigos e redes sociais
 - Sistemas de recomendação

E agora? Qual escolher?



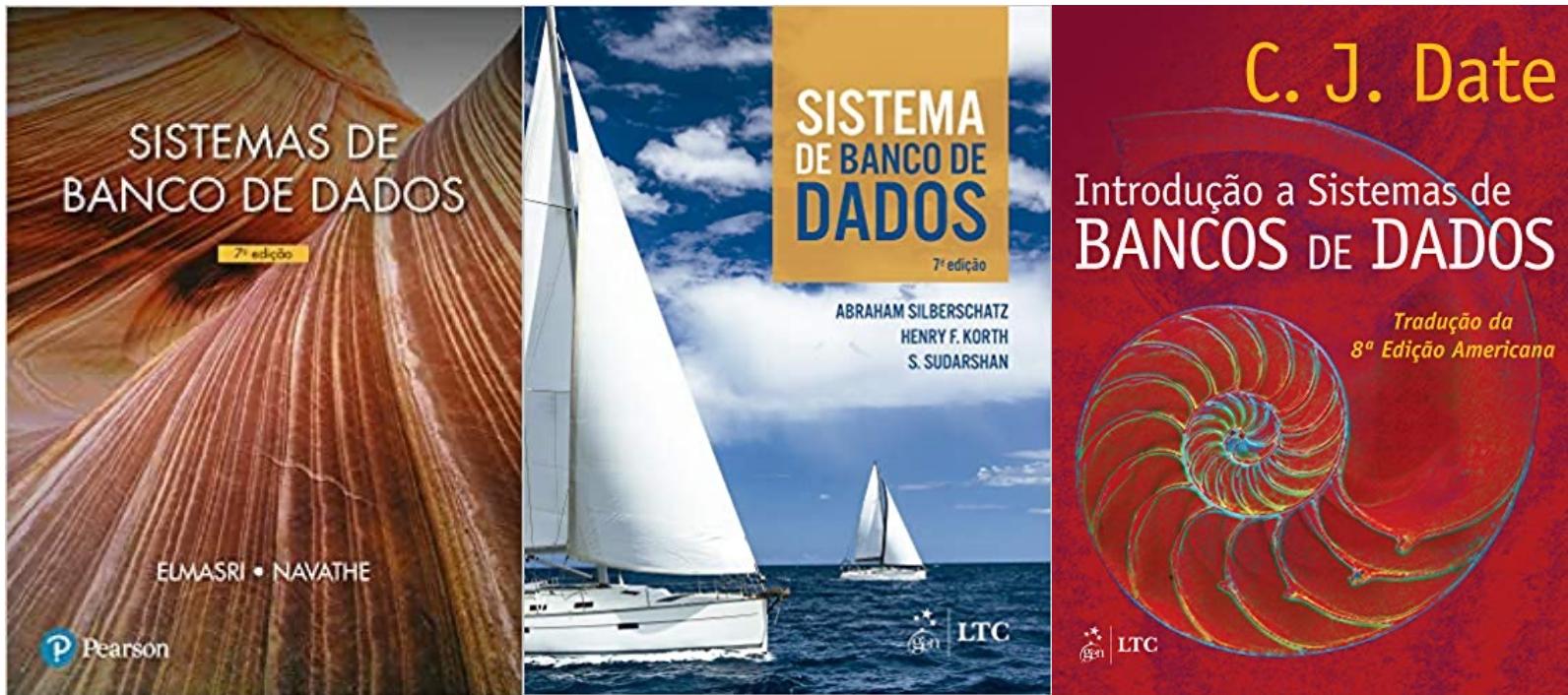
SGBDs

- Chave-valor
 - Redis
 - DynamoDB
(<https://aws.amazon.com/p/t/dynamodb/>)
 - Microsoft Azure Cosmos DB
 - Memcached
- Documentos
 - MongoDB
 - Amazon DynamoDB
 - Databricks
 - Microsoft Azure Cosmos DB
 - Couchbase
- Grafo
 - Neo4j
 - Microsoft Azure Cosmos DB
 - Virtuoso
 - ArangoDB
 - OrientDB

DB-Engines (*ranking*)

<http://db-engines.com/en/ranking>

Referências



- ▢ Bandeira, Denise. Fundamentos de Banco de Dados. Notas de Aula. Unisinos, 2022.
- Muller, Gilberto Irajá. Disciplina de Bando de Dados I. Revisado e atualizado por Prof. João Tavares. Unisinos, 2022.
- Costa, Cristiano André da; Roehrs, Alex. Disciplina de TI Aplicada a Medicina, notas de aula. Unisinos, 2023.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. 3^a. Ed. São Paulo: Makron Books, 2010.

NoSQL

Fundamentos de Banco de Dados