PCS5787 – Ciência dos Dados
Modelagem de Dados não Relacionais

PCS5787 Ciência dos Dados - Modelagem de Dados não Relacionais
Prof. Dr. Pedro Luiz Pizzigatti Corrêa
pedro.correa@usp.br
segundo semestre 2020

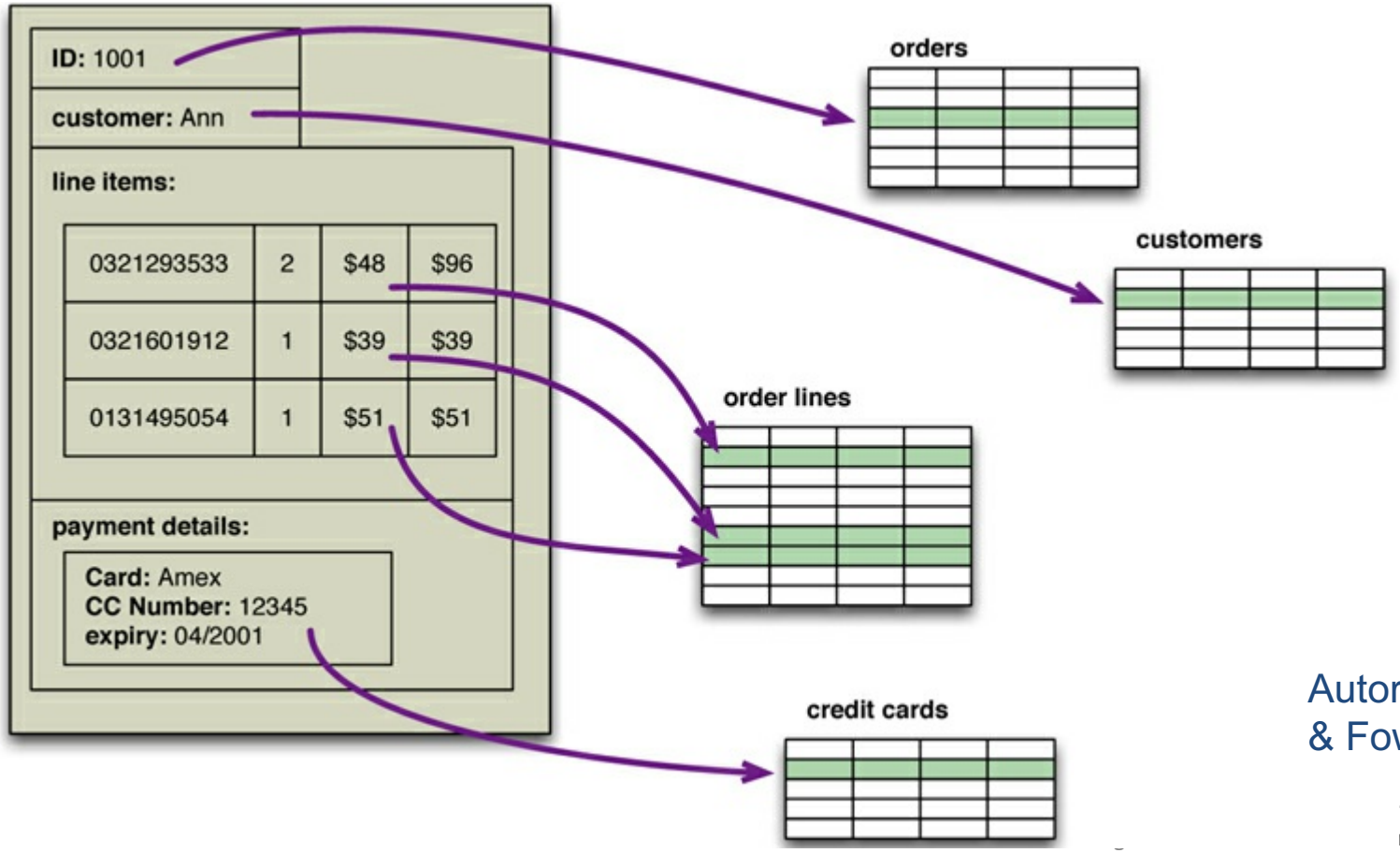
Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Incompatibilidade dos Modelos de Dados de Memória e os Modelos de Armazenamento

- Necessidade de reorganização dos dados manipulados em memória pelos programas para uma representação relacional (banco de dados)
- **Impedância** de modelos usado pelas aplicações e pelos modelo de persitência dos dados.

Incompatibilidade do Modelos de Dados de Memória e os Modelos de Armazenamento



Autor: Sadalage & Fowler, 2013

Características de Banco de Dados Big Data

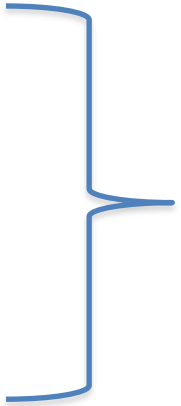
- Não utilizam o Modelo Relacional;
- Demanda de Processamento e Armazenamento em *Clusters/Grid*;
- Apropiados para aplicações WEB;
- Não tem um esquema;
- Também conhecidos como NoSQL.

Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Categorias de Soluções Big Data

- Chave-valor. Ex: Riak, Redis;
- Documento. Ex: MongoDB;
- Famílias de Colunas. Ex: Cassandra, HBase
- Grafos. Ex: FlockDB, Neo4J



Orientação a Agregados

Agregado: conjunto de objetos relacionados que desejamos tratar como unidade.

Caracaterísticas dos Agregados

- Unidade de armazenamento de dados e gerenciamento de consitência;
- Facilita a execução de banco de dados num *cluster*, pois consititui uma unidade natural de fragmentação e replicação;
- São mais simples de serem tratados pelas aplicações uma vez que lidam com os dados por meio de uma estrutura agregada.

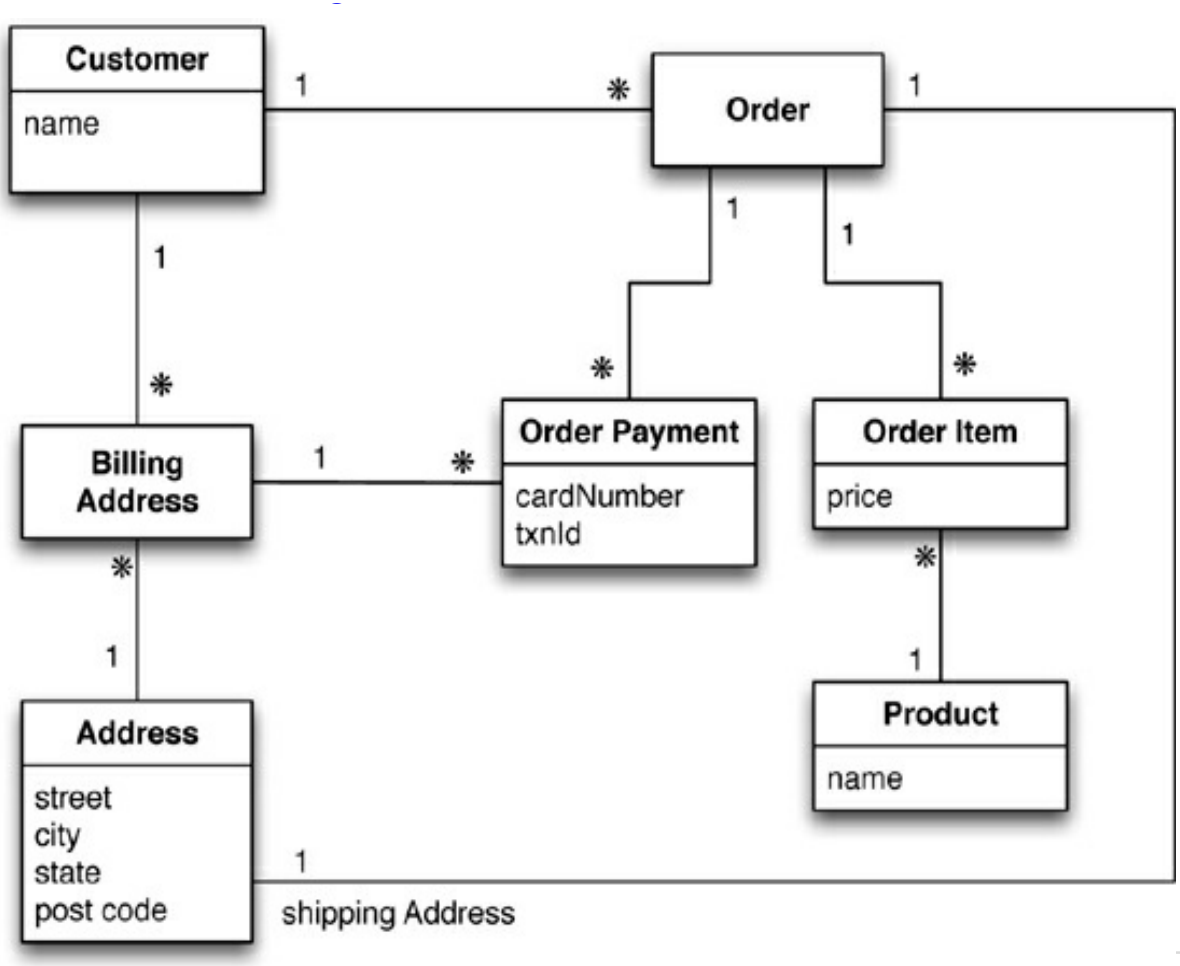
Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Cenário: website de Comércio Eletrônico

- Aplicação de venda de ítens pela web. Teremos que armazenar dados sobre usuários, catálogo de produtos, pedidos, as remessas, os endereços de envio, os endereços de cobrança e os dados sobre o pagamento.

Diagrama de Classe



Autor: Sadalage & Fowler, 2013

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

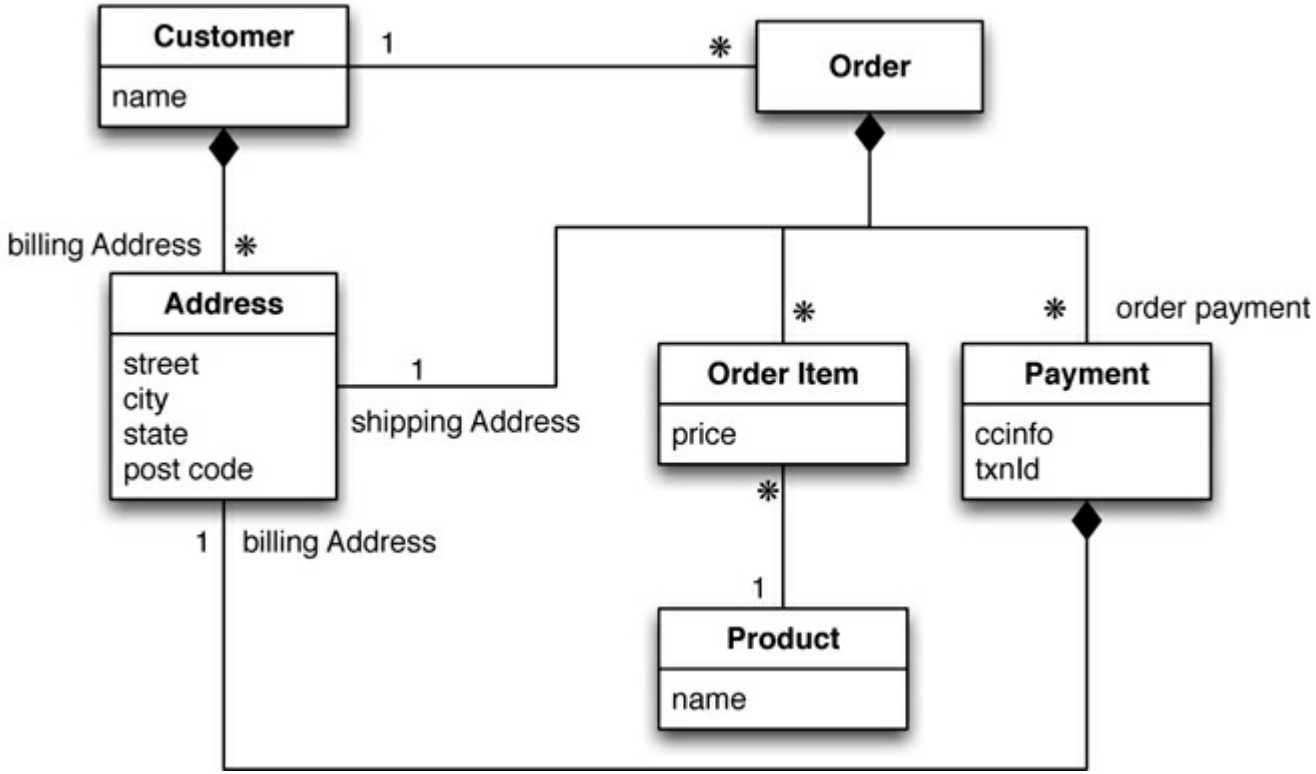
Autor: Sadalage & Fowler, 2013

Considerações sobre o Modelo Relacional do Exemplo

- Tabelas estão Normalizadas (nenhum dado repete-se em múltiplas tabelas);

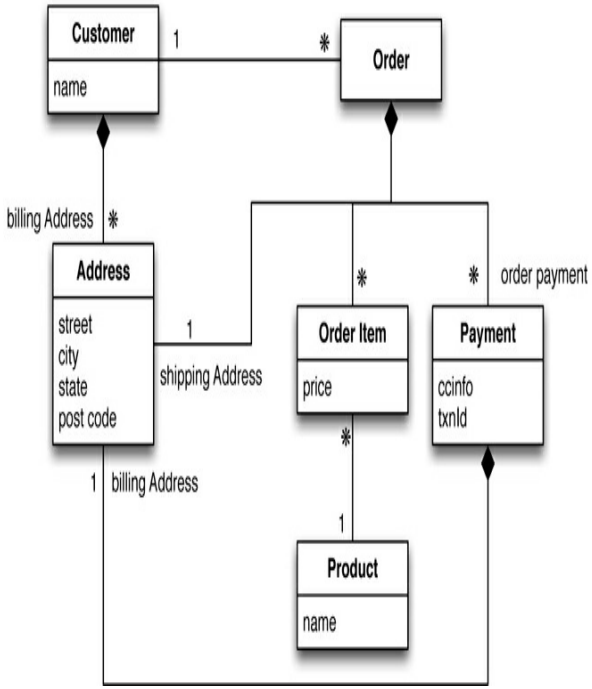
- Integridade referencial (todas as chaves estrangeiras das tabelas, são identificadores únicos de registros em outras tabelas);

Orientação a agregados



Autor: Sadalage & Fowler, 2013

Exemplo de Dados no formato JSON



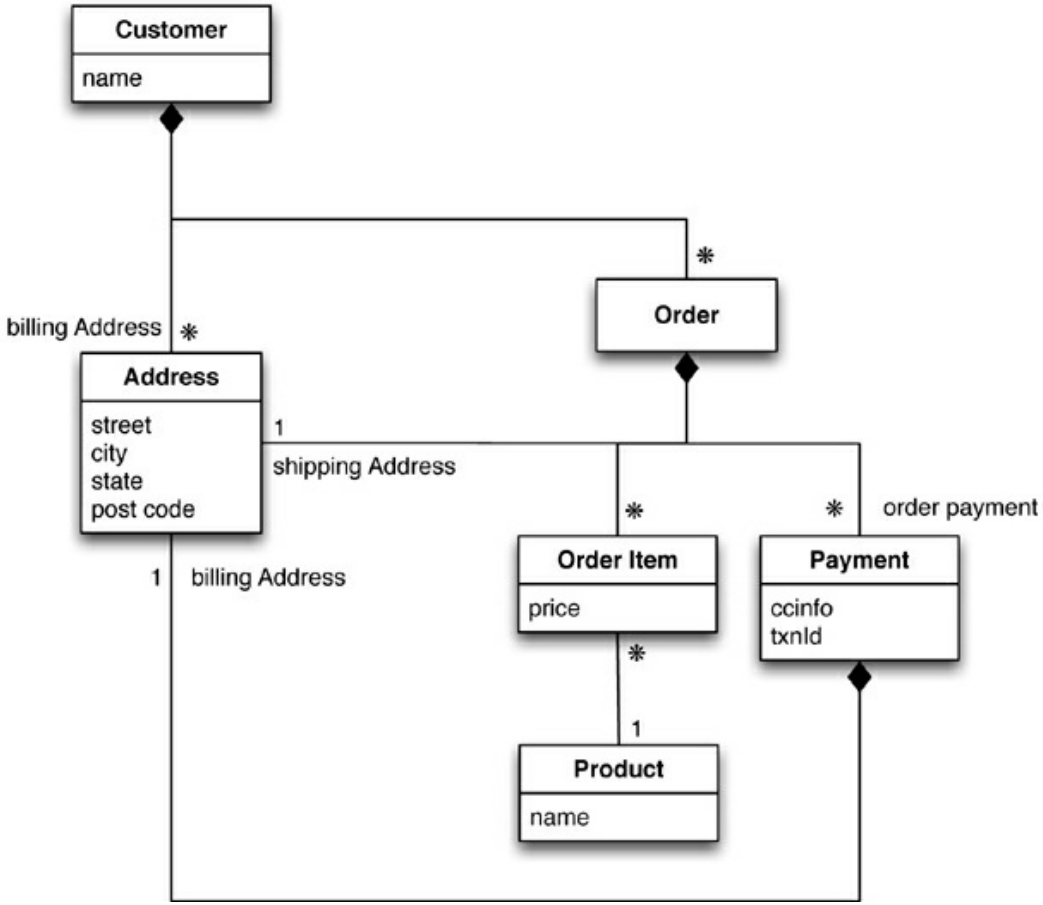

```
// in customers
{
"id":1,
"name":"Martin",
"billingAddress":[{"city":"Chicago"}]
}

// 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"}
  }
  ],
}
```

Autor: Sadalage & Fowler, 2013

Considerações sobre o Agregado

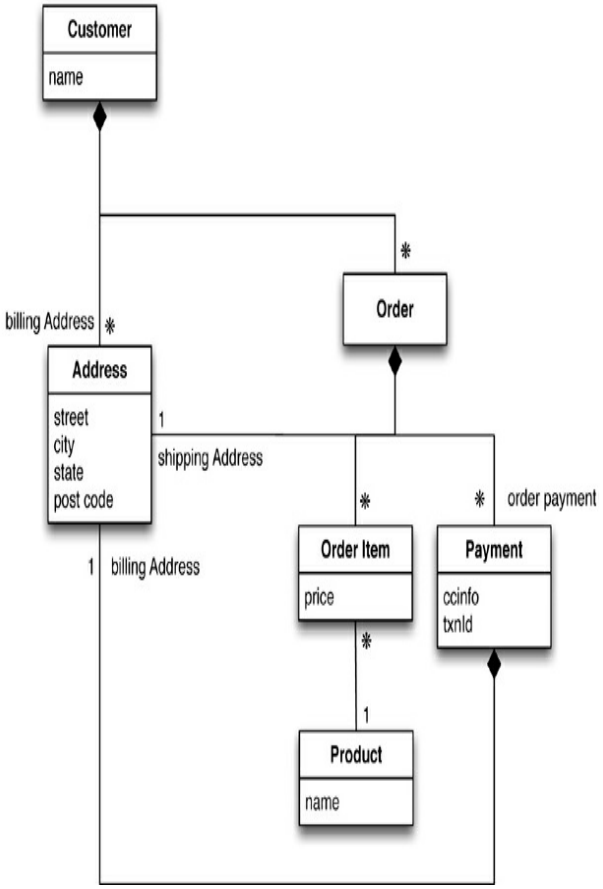
- Temos 2 agregados: Cliente e Pedido;
- Losango preto (Símbolo do Diagrama de Classes UML) indica um agregado;
- Um único endereço aparece 3 vezes, ao invés de usarmos IDs. Esse registro é tratado como valor copiado;
- A conexão entre os agregados Cliente e Pedido estabelece um relacionamento de agregados, e não é um agregado. Da mesma forma que acontece entre Item de Pedido e Produto. O nome do produto foi incluído em Item de Pedido para minimizar o número de agregados que acessamos durante o acesso aos dados.
- Os limites dos agregados são definidos com base no modelo de acesso das principais aplicações.

Alternativa de Modelo de Agregado



Autor: Sadalage & Fowler, 2013

Alternativa de Dados no formato JSON




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

Autor: Sadalage & Fowler, 2013

Alternativa de Modelo de Agregados

Observações:

- Foi definido um agregado para Cliente, considerando que o cenário de acesso a todos os pedidos de um cliente ao mesmo tempo;
- Se o cenário fosse acesso somente a pedidos, então deve-se adotar a estratégia de agregados separados para cada pedido (modelo anterior).

Observações gerais sobre Agregados

- O Modelo Relacional não diferencia relacionamentos que representam agregações, daqueles que não são agregações. Recurso necessário para armazenamento distribuído dos dados;
- O modelo de agregados é adequado para uso com Cluster/Grid Computacional;
- Agregados permitem suportar transações com características de ACID (Atômicas, Consistentes, Isoladas e Duráveis) num único agregado por vez.

Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- **Modelo de dados de chave-valor e de documentos**
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Caraterísticas dos Banco de Dados Chave-Valor e de Documentos

Exemplos dos tipos de Banco de dados:

- Chave-valor. Ex: Riak, Redis;
- Documento. Ex: MongoDB;

} Orientação a Agregados

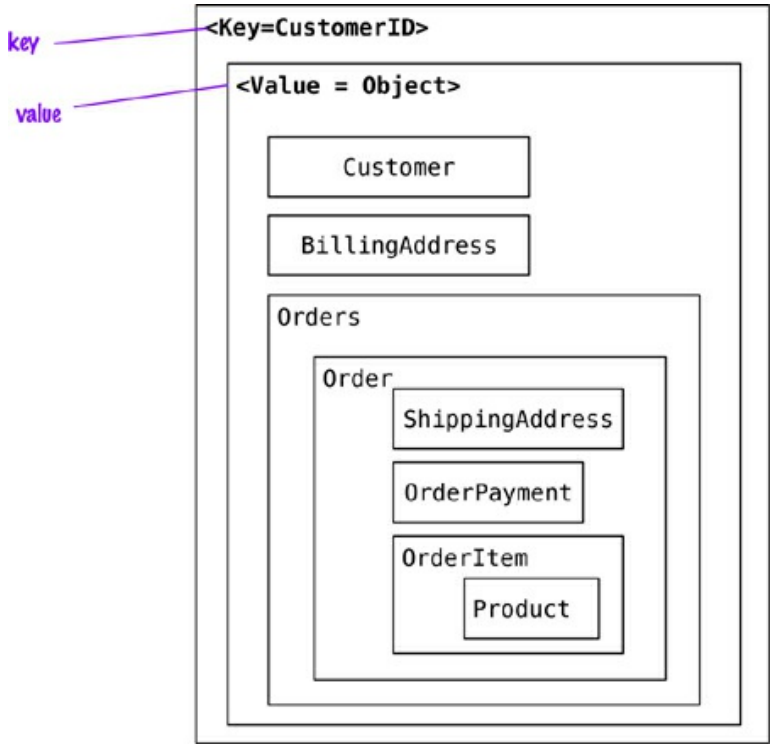
Agregado: conjunto de objetos relacionados que desejamos tratar como unidade.

- Os dois tipos manipulam agregados, sendo uma chave ou ID para obter os dados.

Caraterísticas dos Banco de Dados Chave-Valor e de Documentos

- Diferenças entre os tipos de Banco de dados:
  - **Chave-valor**: agregado é opaco – sem estrutura prévia;
  - **Documento**: pode tratar o agregado como uma estrutura.
- Em Banco de Dados de Documento, pode-se buscar e indexar por campos e conteúdos do agregado;
- As consultas podem ser manipuladas por ferramentas de pesquisa, como ***Solr***, buscando quaisquer agregado que esteja armazenado como estruturas XML ou JSON.

Exemplo1 de um agregado Cliente

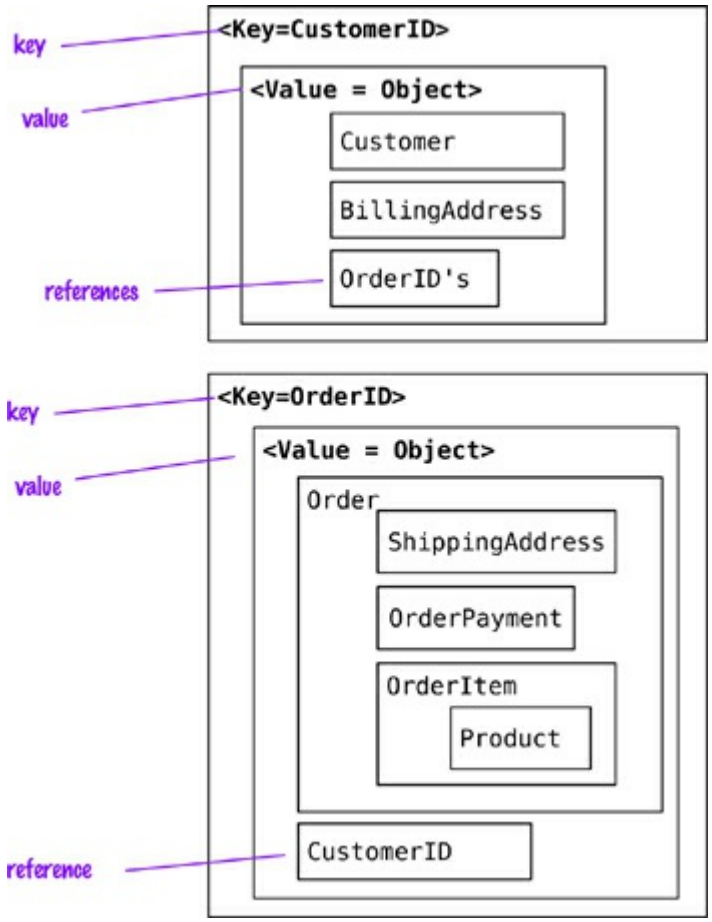


Cenário Principal: Acessar todos os dados a partir do Cliente.

Possível padrão de aplicação
- Acesso a produtos vendidos em cada pedido (deve-se acessar todo o Objeto e ler cada pedido os produtos associados).

Autor: Sadalage & Fowler, 2013

Exemplo2 agregados Cliente e Pedidos



- Cenário Principal: Encontrar pedidos (*Orders*) independentes do cliente (*Customer*)
- Exemplo (JSON):

```
# Customer object
{
"customerId": 1,
"customer": {
  "name": "Martin",
  "billingAddress": [{"city": "Chicago"}],
  "payment": [{"type": "debit","ccinfo": "1000-1000-1000-1000"}],
  "orders":[{"orderId":99}]
 }
}

# Order object
{
"customerId": 1,
"orderId": 99,
"order":{
  "orderDate":"Nov-20-2011",
  "orderItems":[{"productId":27, "price": 32.45}],
   "orderPayment":[{"ccinfo":"1000-1000-1000-1000",
           "txnId":"abelif879rft"}],
   "shippingAddress":{"city":"Chicago"}
   }
}
```

Autor: Sadalage & Fowler, 2013

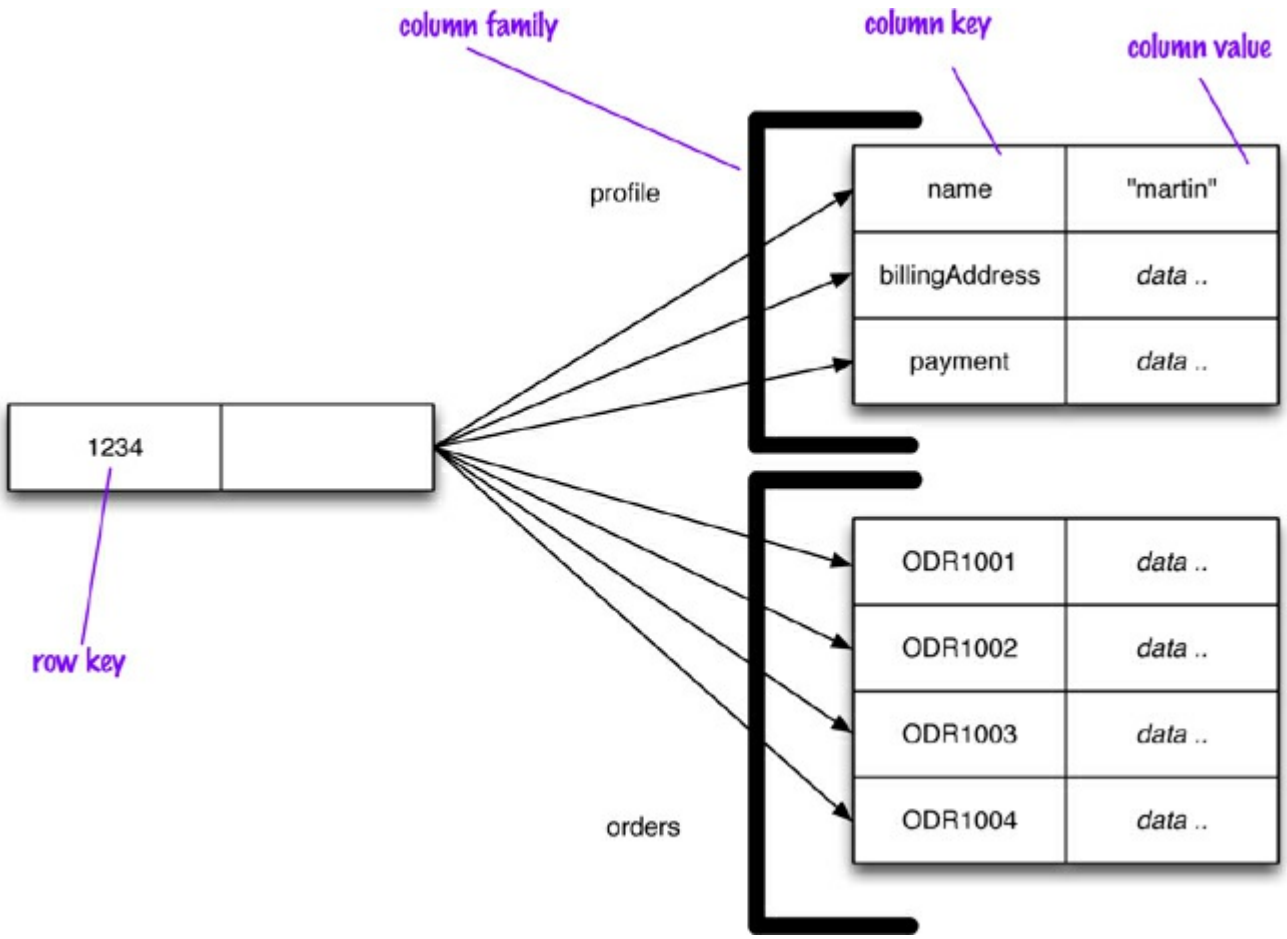
Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- **Armazenamento de Famílias de Colunas**
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Caraterísticas dos Banco de Dados Famílias de Colunas

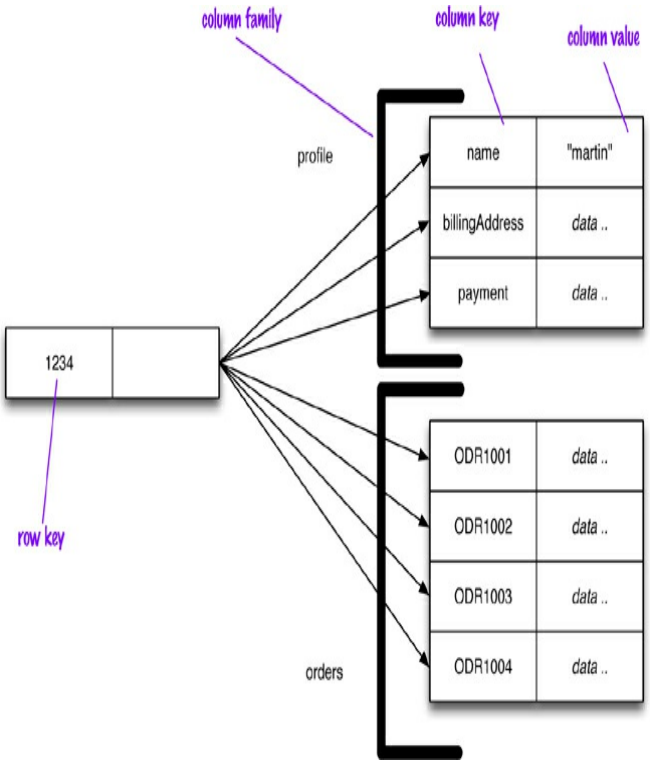
- Colunas esparsas e sem esquema. Exemplo: BigTable (Google), Hbase e Cassandra;
- Há cenários em que as colunas podem ser agrupadas como uma unidade básica de armazenamento – por isso o nome **Famílias de Colunas**;
- Quando tratado como Família de Colunas o acesso ocorre em dois níveis: acesso ao agregado e em seguida à Família de Colunas

Exemplo de um agregado Cliente, representado em Famíias de Colunas



Autor: Sadalage & Fowler, 2013

Exemplo de um agregado Cliente, representado em Famíias de Colunas

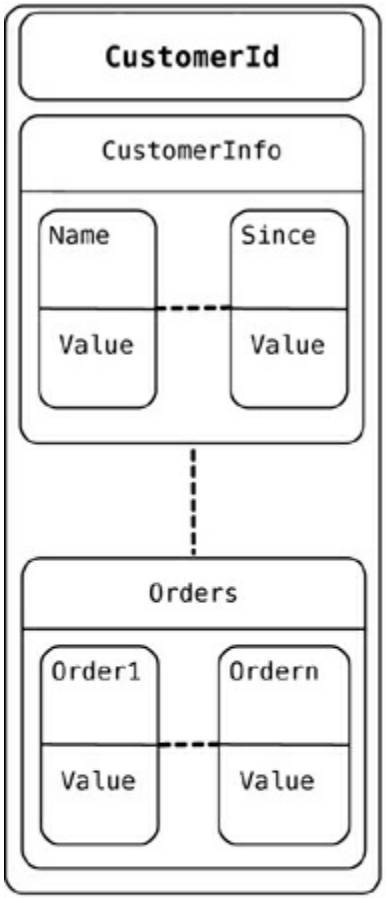


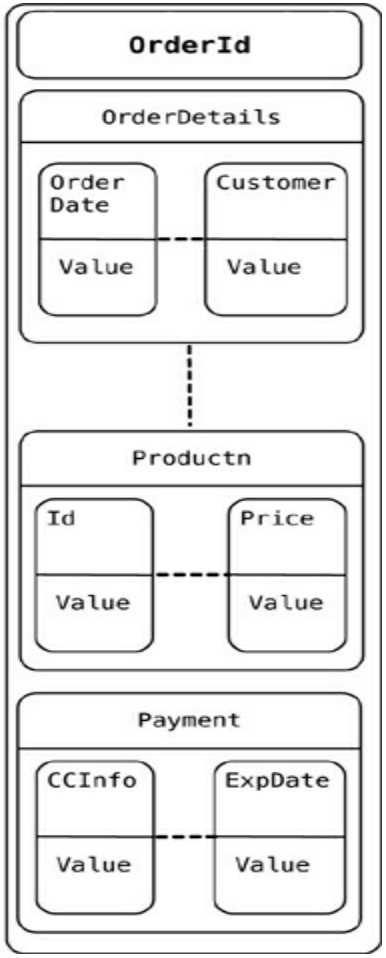
- A primeira chave é o identificador de “Linha”, identificando o agregado (Ex: “1234”);
- Idenfificador do Agregado acessa os detalhes do Agregado de segundo nível (Colunas);
- Além de acessar a “Linha” como um todo do agregado, pode-se acessar um coluna em particular. Exemplo: obter um nome de um determinado cliente:

  get(‘1234’, ‘name’);
- Cada coluna faz parte de uma única família de colunas que geralmente são acessados em conjunto (agregado);

Autor: Sadalage & Fowler, 2013

Visão Conceitual de alternativas de Modelos Físicos de Famílias de Colunas





Autor: Sadalage & Fowler, 2013

Caraterísticas dos Banco de Dados Famílias de Colunas

- Tipos de Acessos:
  - Orientado a “Linhas”: cada linha é um agregado (ex: ID: ‘1234’), com acesso estruturado as famílias de Colunas (Perfil e Pedidos);
  - Orientado a Colunas: cada Família de Colunas define um tipo de registro (ex: Perfil) com linhas. Cada linha é uma Família de Colunas.

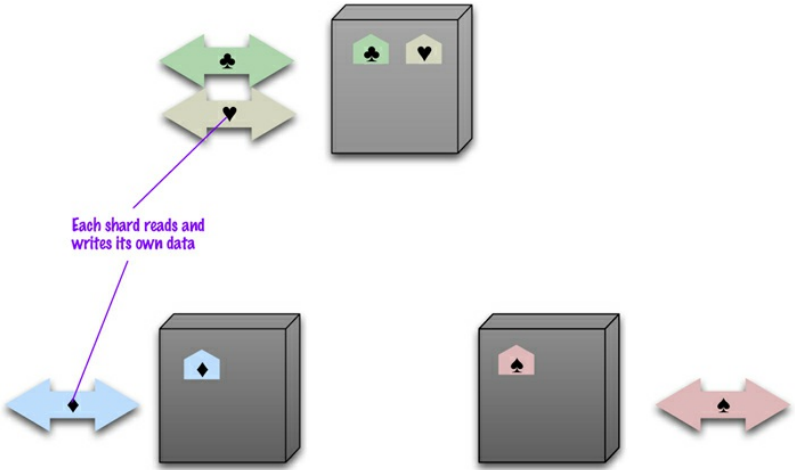
Caraterísticas dos Banco de Dados Famílias de Colunas

- Tipos de Colunas (Cassandra):
  - **Linhas Estreitas (*Skinny*)**: poucas colunas, sendo as mesmas colunas utilizadas por todas as linhas;
  - **Linhas Largas (*Wide*)**: cada linha possui muitas colunas, sendo que as colunas de cada linha podendo ser diferentes entre sí. Família de Colunas Largas modela o conceito de lista de colunas.
    - Exemplo: Poderíamos definir pedidos como sendo uma Família de Colunas Largas, ordenados pelo seu id (concateção de Data + ID): 20160728-1001

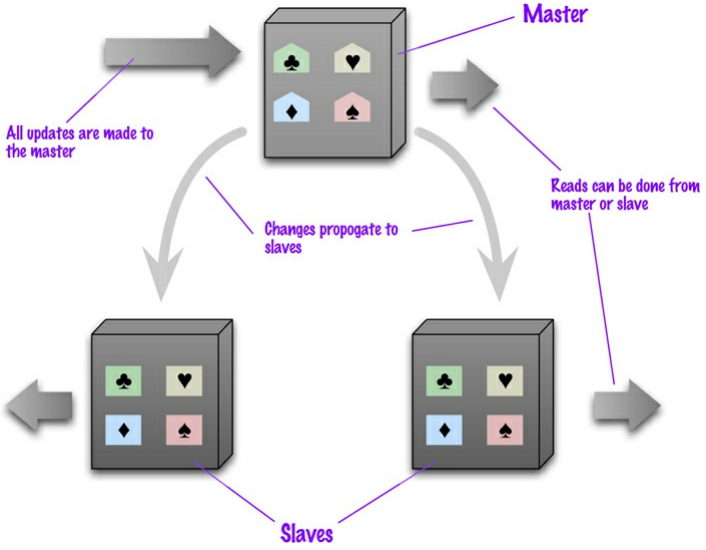
Modelos de Distribuição de agregados

- Agregados podem ser distribuídos num *Cluster/Grid*, para aumentar à escalabilidade, eficiência e disponibilidade;
- Uso de Fragmentos (subconjuntos de Agregados) e Réplicas (Cópias);
- Desvantagens: aumento da complexidade.

Modelos de Distribuição de agregados: Exemplos:



Fragmentos Distribuídos em Vários Nós.



Fragmementos podem ser replicados em Nós *Master* e *Slave*.

Autor: Sadalage & Fowler, 2013

Agregados - resumo

- Agregados definem um conjunto de dados que acessamos como unidade. Limite para operações ACID;
- Banco de Dados Chave-Valor, Documentos e Famílias de Colunas – Banco de Dados Orientado a agregados;
- Agregados facilitam o armazenamento de dados em Cluster/Grid;
- Facilitam o acesso quando realizado no mesmo agregado.
- Banco de dados sem agregados são adequados quando as interações ocorrem em muitos padrões diferentes.

Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

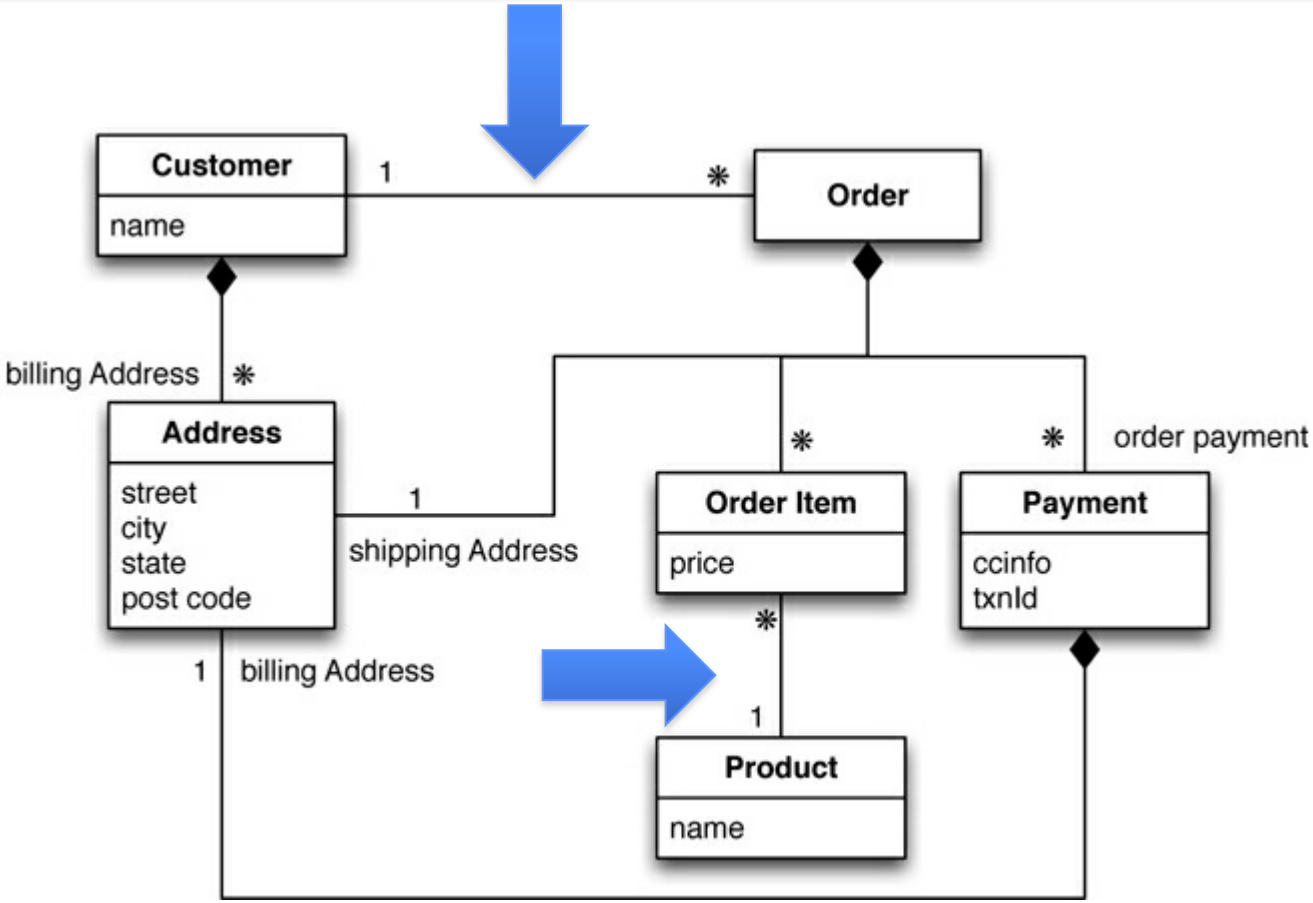
Categorias de Soluções Big Data

- Chave-valor. Ex: Riak, Redis;
- Documento. Ex: MongoDB;
- Famílias de Colunas. Ex: Cassandra, Hbase

Orientação a Agregados

Relacionamentos simples

Orientação a agregados – relacionamentos simples entre agregados



Autor: Sadalage & Fowler, 2013

Categorias de Soluções Big Data

- Chave-valor. Ex: Riak, Redis;
- Documento. Ex: MongoDB;
- Famílias de Colunas. Ex: Cassandra, Hbase
- Grafos. Ex: FlockDB, Neo4J

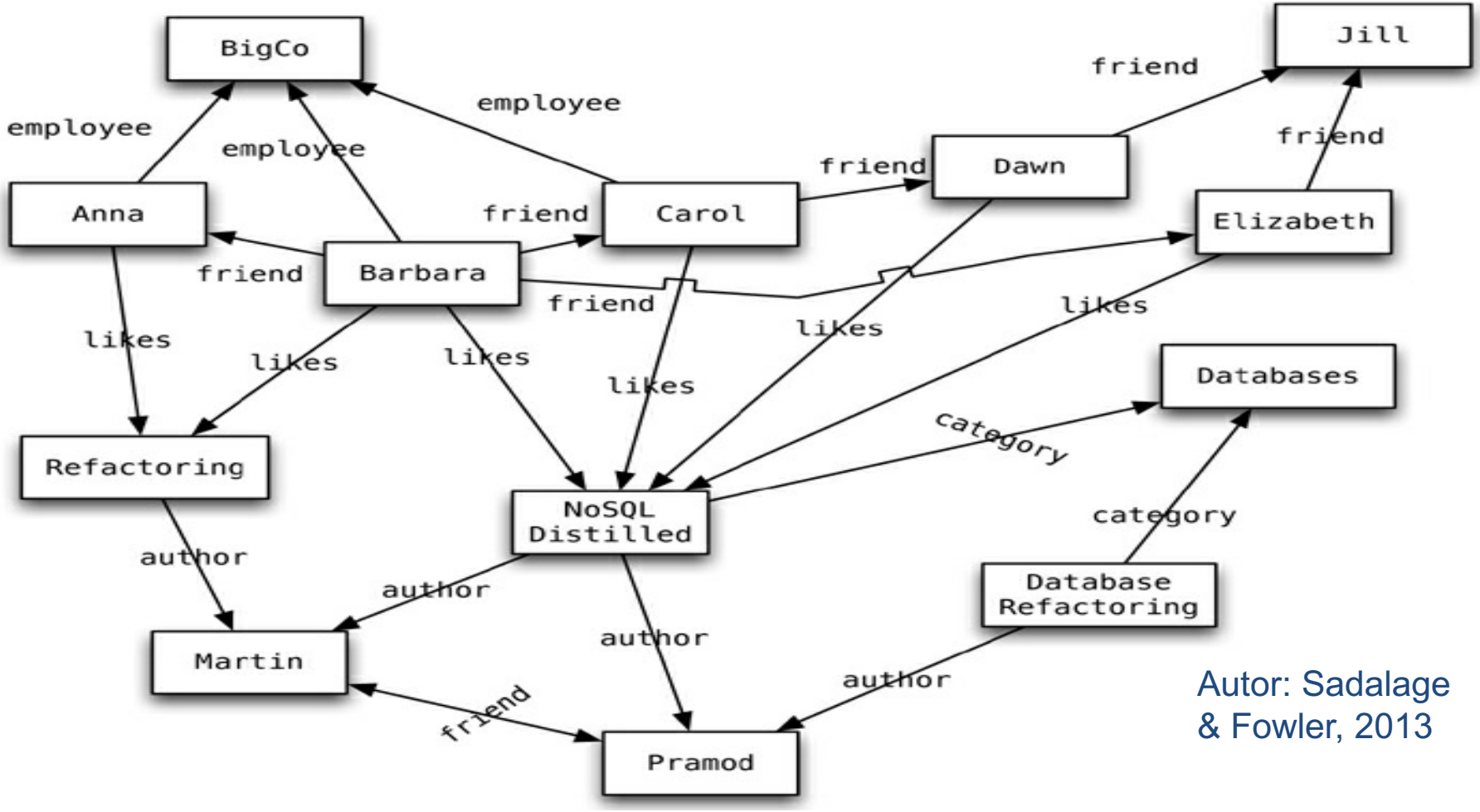
Orientação a Agregados

Relacionamentos simples

Orientação a Relacionamentos

Banco de Dados orientado a Grafos: registros pequenos com interconexões complexas.

Exemplo de um modelo de grafos

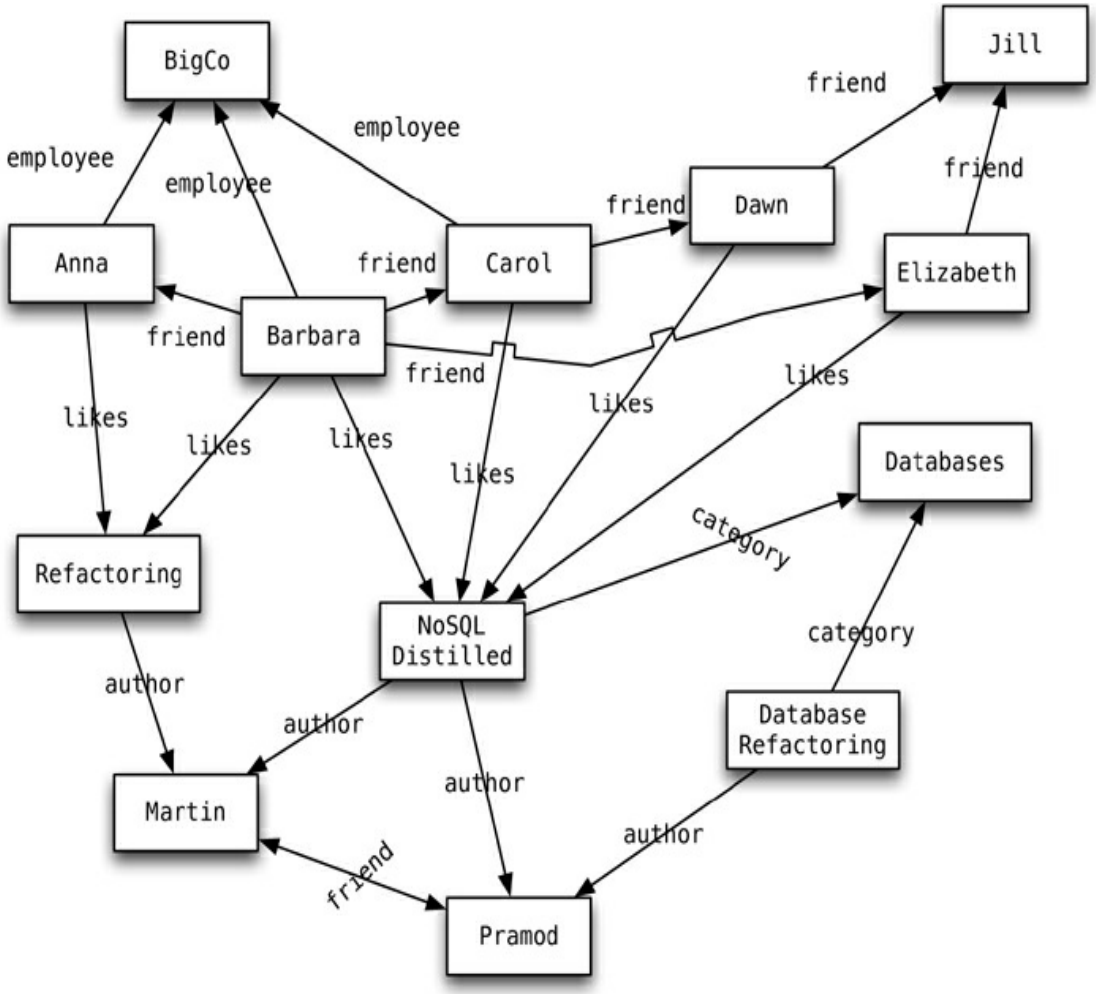


Autor: Sadalage & Fowler, 2013

Caraterísticas dos Banco de Dados de Grafos

- Grafo: Nós conectados por Arestas (arcos);
- Nós tem poucos dados (*Nome*), mas é rico em relacionamentos;
- Aplicações que envolvam relacionamentos complexos. Exemplo: redes sociais, preferências de produtos, dentre outras.
- Permitem associar a arestas e nós objetos, que são subtipos de Nós e Arestas (Neo4J, Infinete Graph);
- As Buscas são iniciadas pela localização de um Nó e então segue-se pelas Arestas.

Exemplo1: Banco de Dados de Grafos – Rede Social

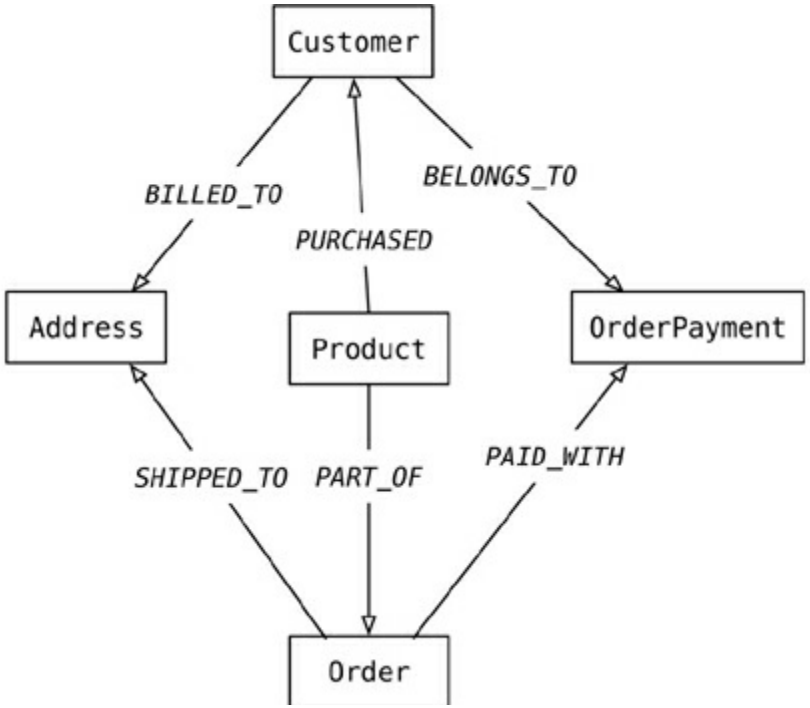


Exemplo: Infome tudo que Anna e Barbara Gostam:

- Procurar Pessoas (Nós) com nome Anna e Carol;
- Percorrer as Arestas (Gosta) dos Nós Anna e Carol;

Autor: Sadalage & Fowler, 2013

Exemplo2: Comércio Eletrônico com Grafos



Exemplo: Que Cliente comprou um determinado produto “Caneta”:

- Procurar o Nó “Caneta”;
- Percorrer as Arestas (Comprou ou *Purchased*) dos Nós *Customers*.

Especialmente útil para recomendar produtos a usuários ou encontrar padrões de compras realizadas por ele.

Autor: Sadalage & Fowler, 2013

Banco de Dados de Grafos

- Banco de Dados orientados a Grafos são propensos a funcionar num único servidor do que distribuído em *Clusters*;
- Transações ACID precisam cobrir múltiplos Nós e Arestas para manter a consistência;
- São mais eficientes que Banco de Dados Relacionais, pois evitam múltiplos *Joins* para percorrer relacionamentos entre linhas (registros) de diferentes tabelas.

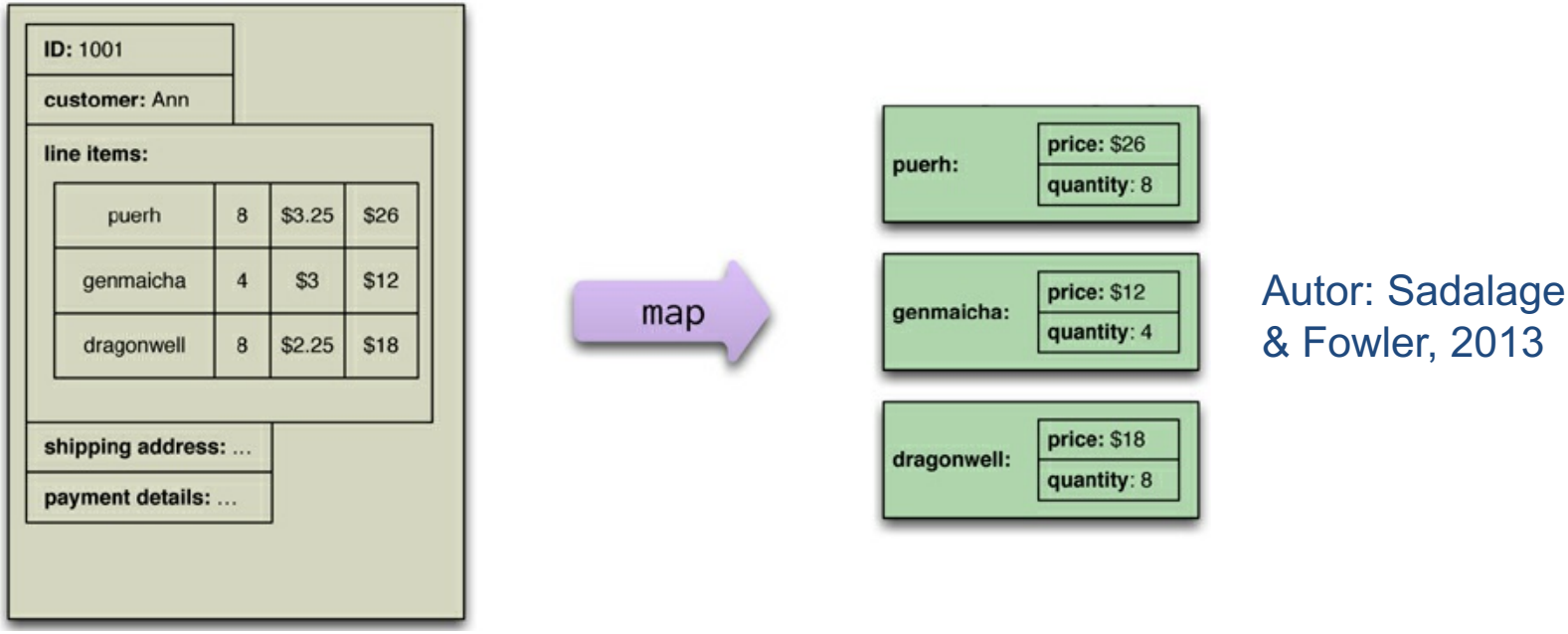
Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Introdução

- **Map-reduce (Mapear-reduzir)**: padrão de computação para organizar o processamento distribuído num *Cluster/Grid*.
- Exemplo: considere o Agregado de ”Pedidos” com dados do Cliente e dos Ítems de Produtos.
  - Considere que os Agregados de Pedidos podem estar distribuídos em vários nós (máquinas) do *Cluster*.
  - Esse cenário faz sentido quando temos aplicações que precisam acessar um pedido inteiro em um acesso.
  - Porem imagine cenário de acesso pelo Departamento de Marketing que precisa dados sobre consumo de cada produto (nova Visão Materializada).

Exemplo1: Map-reduce de Pedidos (operação Map)



Autor: Sadalage & Fowler, 2013

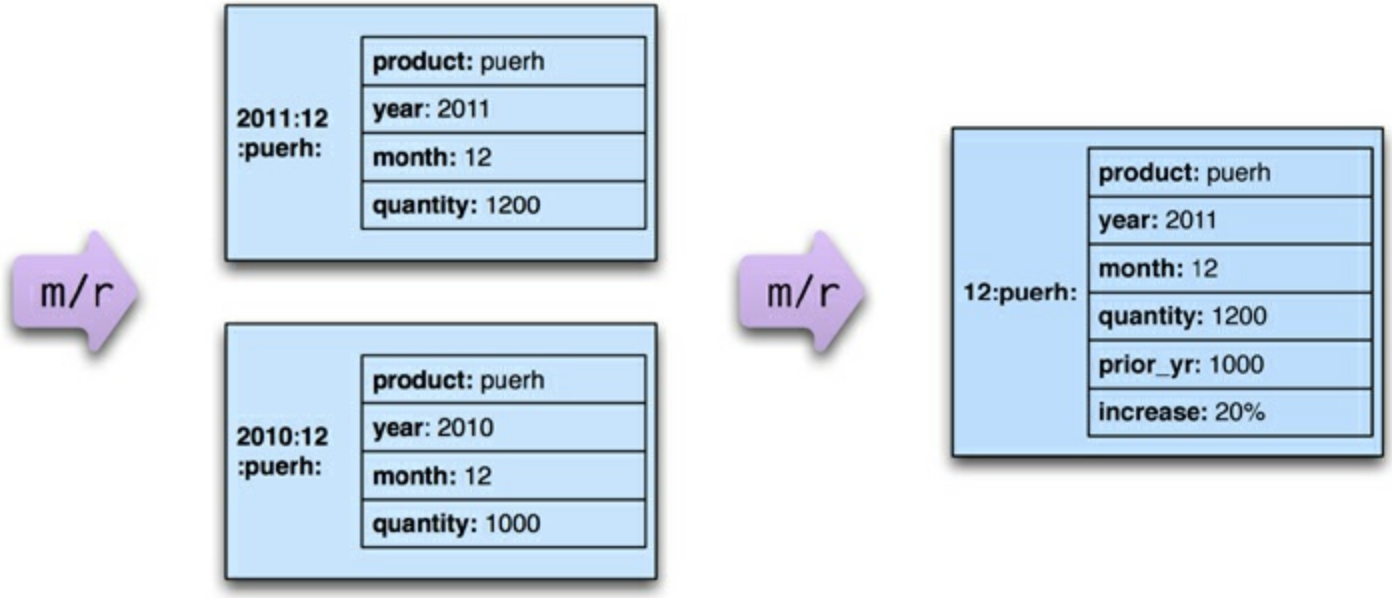
- Cenário Principal: dados agregados a partir do pedido
- Cenário alternativo: acesso a por produto.
- Operação map recebe como entrada um Pedido e gera na saída vários pares de chave/valor (nome do produto/dados do produto).

Exemplo1: Map-reduce de Pedidos (operação Map)

Autor: Sadalage & Fowler, 2013

- Operação *reduce*: função que recebe como entrada múltiplos mapeamentos com a mesma chave (Nome do Produto) e combina seus valores .
- Para desenvolver uma aplicação Map-reduce, basta fornecer as duas funções (Map e Reduce).
- Vantagens do Map/Reduce: são operações independentes e podem ser distribuídos entre os nós de um *Cluster/Grid*

Exemplo2: Map-reduces compostos



Autor: Sadalage & Fowler, 2013

- Cenário: Departamento de Marketing quer estudar o desempenho de vendas dos produtos entre os anos de 2011 e 2010 .
- A primeira fase do Map-reduce calcula o desempenho dos produtos no mês e Ano
- A segunda fase do Map-reduce calcula o incremento do produto do ano 2011 em relação ao ano de 2010.
- A primeira fase do Map-reduce, poderia ser uma Visão Materializada.

Map-reduce

- Map-Reduce são operações independentes e podem ser distribuídos entre os nós de um *Cluster/Grid.*
- Se o resultado de uma computação for amplamente utilizado pode ser armazenado como uma Visão Materializada;
- Visões Materializadas podem ser implementatas por meio de Map-Reduce, de forma a fatorar operações que são comuns em cenários de acessos distintos.

Agenda

- Introdução
- Agregados
- Exemplo de relações e agregados
- Modelo de dados de chave-valor e de documentos
- Armazenamento de Famílias de Colunas
- Banco de Dados de Grafos
- Map-reduce
- Evolução de esquemas não relacionais

Evolução de Esquemas não relacionais

Alterações no esquema de dados são necessárias para acompanhar a evolução de requisitos das aplicações demandas pelo negócio que estão relacionadas.

- Aplicações de Big Data (no SQL) não necessitam de esquemas de dados a *priori*;
- Alterações podem gerar inclusão/remoção/alteração na estrutura do banco de dados ou agregados;
- As soluções de Banco de Dados não-relacionais permitem as alterações a qualquer momento;
- As aplicações que acessam o Banco devem estar preparadas para acessar dados em diferentes estruturas.

Evolução de Esquemas não relacionais

Exemplo: considere a necessidade de alterar um atributo e incluir um novo atributo para um ítem de produto:

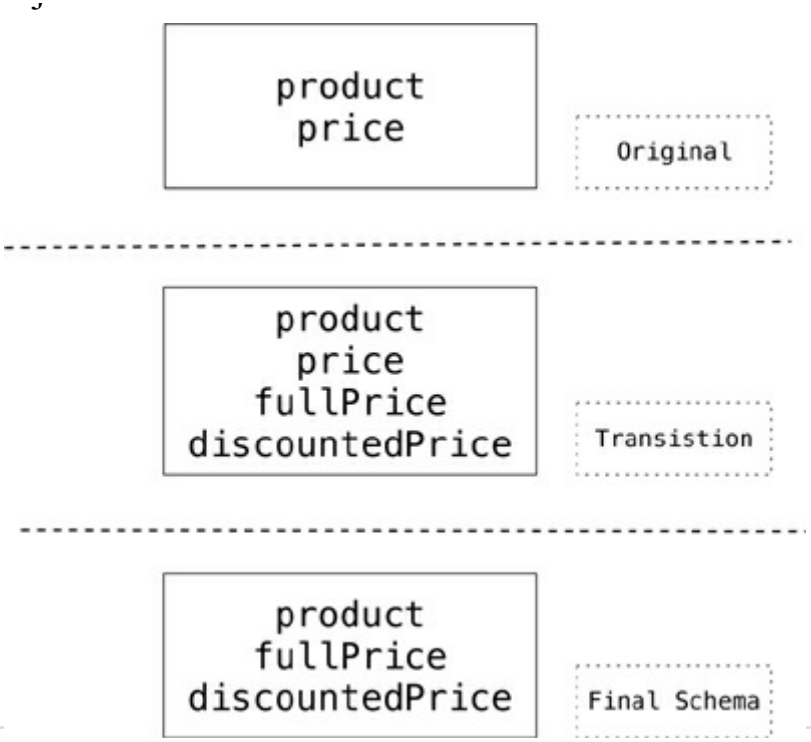
```
{
"_id": "4BD8AE97C47016442AF4A580",
"customerid": 99999,
"name": "Foo Sushi Inc",
"since": "12/12/2012",
"order": {
    "orderid": "4821-UXWE-122012","orderdate": "12/12/2001",
    "orderItems": [{"product": "Fortune Cookies",
                    "price": 19.99}]
    }
}
```

```
{
"_id": "5BD8AE97C47016442AF4A580",
"customerid": 66778,
"name": "India House",
"since": "12/12/2012",
"order": {
  "orderid": "4821-UXWE-222012",
    "orderdate": "12/12/2001",
    "orderItems": [{"product": "Chair Covers",
                    "fullPrice": 29.99,
                    "discountedPrice":26.99}]
    }
}
```

Assim que ocorrer essas alterações para os novos pedidos e podem ser gravados e lidos, porem os antigos devem ter problemas, pois irão procurar *fullprice* e o documento tem somente *price*.

Estratégia: Migração incremental

Migração Incremental: Na medida que forem sendo feitas leituras, serão feitas migrações para o novo esquema:



Bibliografia

- SADALAGE, P.J; FOWLER, M. NoSQL Essencial. Novatec. 2013.

Conceitos de Big Data

PCS5787 Ciência dos Dados - Modelagem de Dados não Relacionais
Prof. Dr. Pedro Luiz Pizzigatti Corrêa
pedro.correa@usp.br
segundo semestre 2020