



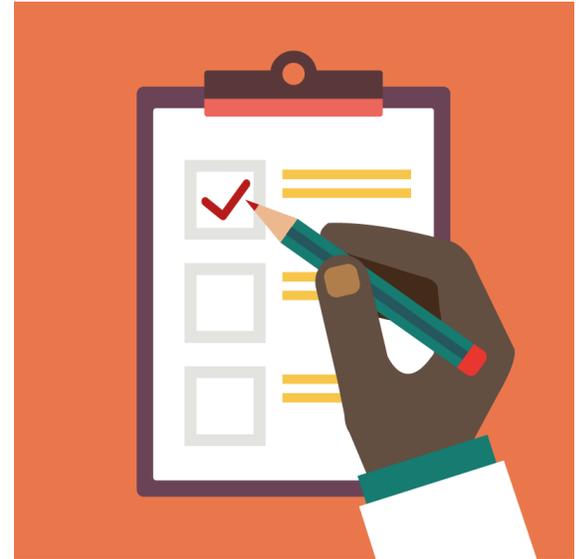
**PROVAS DE TI**  
TUDO PARA VOCÊ PASSAR

# NoSQL

Prof. Rodrigo Macedo

# Escopo do Curso

- Conceituação Geral: Paralelo BD relacional.
- Surgimento NoSQL.
- Modelo de dados Agregados.
- Consistência.
- Categorias BD NoSQL.
- Questões de concursos





Fonte: Disponível em:  
[http://nadafragil.com.br/cinco-coisas-para-jogar-fora-  
hoje-mesmo/](http://nadafragil.com.br/cinco-coisas-para-jogar-fora-hoje-mesmo/)



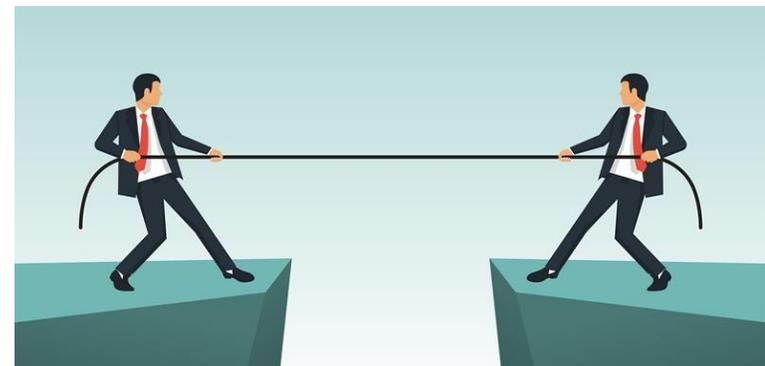
Fonte: Disponível em:  
[http://www.univasf.edu.br/  
~ricardo.aramos/disciplinas/  
/BD\\_AVANÇADO2013\\_2/BD  
A2013\\_2.htm](http://www.univasf.edu.br/~ricardo.aramos/disciplinas/BD_AVANÇADO2013_2/BD_A2013_2.htm)



Fonte: Disponível em:  
[http://infologico.com/2012/  
/10/como-tirar-pastas-do-  
modo-oculto-no-windows-  
7.html](http://infologico.com/2012/10/como-tirar-pastas-do-modo-oculto-no-windows-7.html)

# Concorrência

- Evitar situações em que, caso de concorrência, ocasione anomalias na aplicação, como: fazer reservas do mesmo quarto de hotel, para dois clientes simultaneamente.
- Geralmente, a concorrência é tratada por transações. Um recurso disponível em alguns SGBD's, como os de orientação relacional.
- Se baseiam no acrônimo ACID.



# Introdução

- Operações que utilizam banco de dados, na prática, funcionam por meio de transações que são abertas pelo banco de dados para troca de informações.
- As prerrogativas ao uso de transações, são **inerentes** ao tipo de sistema ou tecnologia que se utiliza (Deve haver o cuidado em qualquer uso de transações seja distribuído ou não).
- Toda transação tem um **início e fim**, que delimitam o tempo de vida de uma transação.
- Aplicada primordialmente em um grupo de **operações**, apesar de ser possível criar uma transação para cada operação.

# Transação

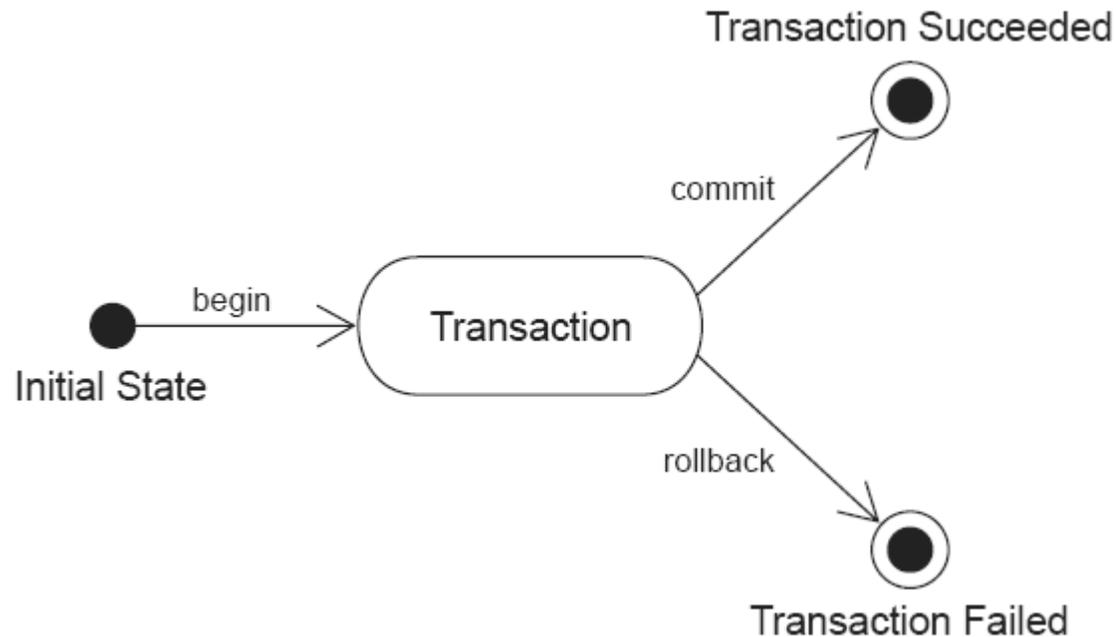
- “**Valida** uma sequência de ações que são executadas em um grupo de operações”.
- A transação só será efetivada, se a validação estiver completa.



**Transferência bancária**

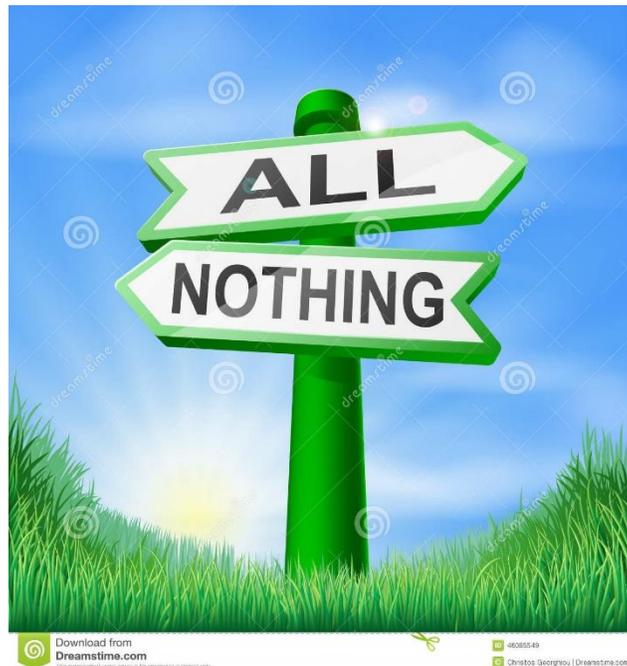
# Ciclo de Vida

- Para assegurar a integridade e consistências das transações no Banco de Dados, utiliza-se uma prerrogativa de características por nome de **ACID**.



# Atomicidade

- Essa prerrogativa informa que cada transação deve ser atômica. A transação deve executar de forma completa, ou então nenhuma operação dessa transação é executada.



# Consistência

- O estado das informações transacionais devem se manter consistentes, com a obrigação de não danificar o estado da estrutura transacional.



# Isolamento

- Cada transação deve ser isolada de outras transações. Para que, uma transação não interfira na outra.
- Transações paralelas = Transações em sequência.



# Durabilidade

- Garante que após a confirmação da transação (commit) serão duráveis. Os erros ou exceções que houverem na efetivação da transação não deve reverter as ações executadas pelas transações.

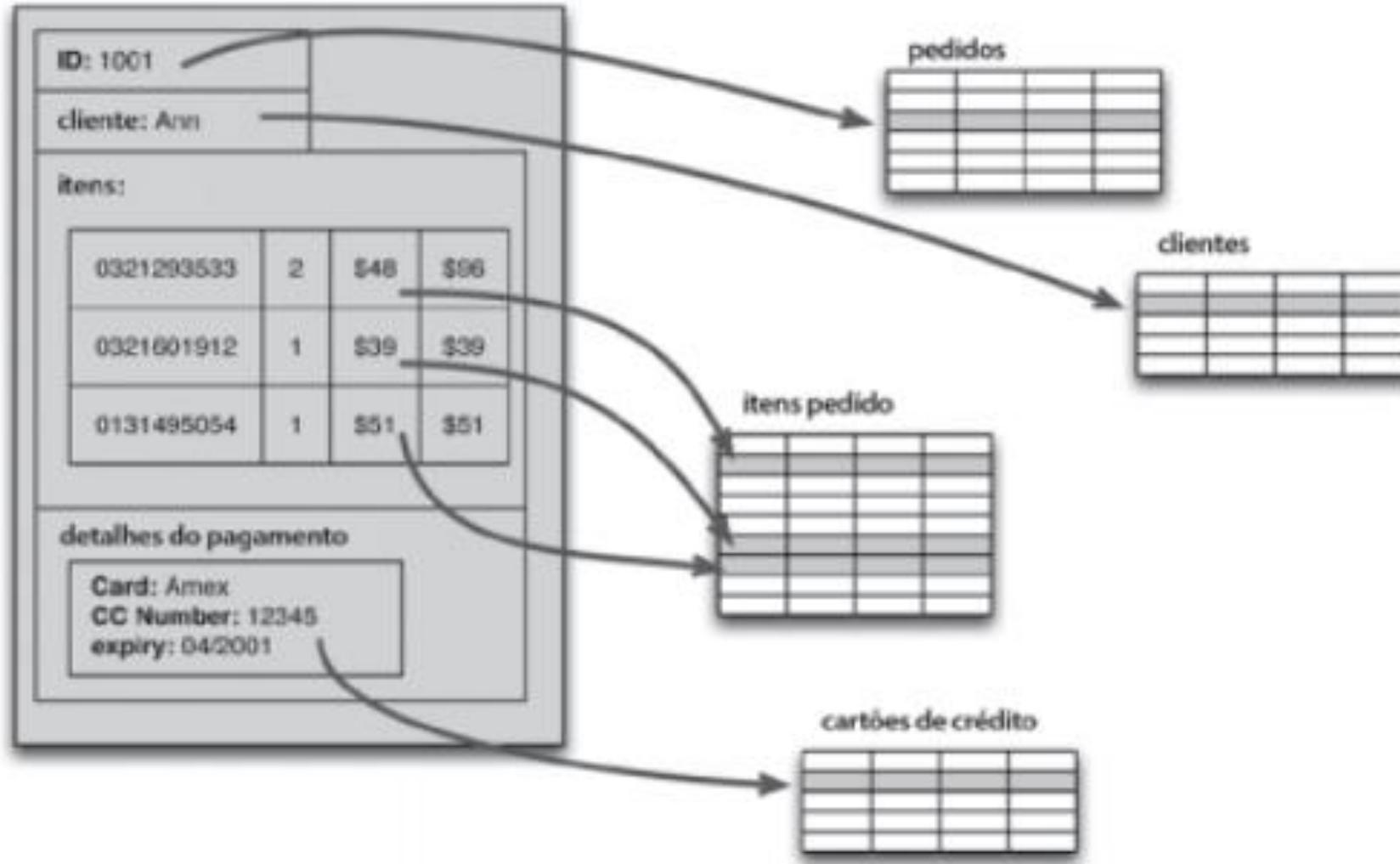


# Incompatibilidade de Impedância

- Diferença entre o modelo relacional e as estruturas de dados na memória.
- Na programação, por exemplo, um paradigma muito comum é a Programação Orientada a Objetos, que diferencia em muita da estrutura de banco de dados relacionais.
- Para resolver um pouco desse questionamento, muitos frameworks de mapeamento objeto-relacional (ORM) foram criados, como o Hibernate.



# Incompatibilidade de Impedância



Pedido que apresenta uma estrutura única na interface de usuário, porém, dividido em muitas linhas e tabelas num banco de dados relacional.

# O ataque dos clusters

- Houve um 'boom' no avanço da internet nos anos de 2000. Grandes conjuntos de dados estava surgindo. Links, redes sociais, atividades em log, dentre outros.
- Surgiu-se então a pergunta: Como armazenar e processar todos esses dados, se o volume aumenta de forma exponencial?



X



# Computação Distribuída

- Executa aplicações através de máquinas diferentes, como se estas fossem uma só.
- Tornou-se possível com a popularização das redes de computadores.
- As máquinas podem estar interligadas por redes intranets, internet, redes públicas e privadas



# O Surgimento do NoSQL

- O seu movimento surgiu pelos anos 2000, e foi cunhado a sua nomenclatura em um evento sobre Hadoop em 2009.
- Os bancos de dados NoSQL não utilizam SQL, pois possuem linguagem próprias de consulta, semelhante ao SQL, mas com todas as premissas para o ambiente distribuído.
- Orientado pela necessidade da execução em clusters (com exceção do modelo baseado em grafo).
- Seu objetivo inicial foi suprir as necessidades do Big Data.



# O Surgimento do NoSQL

- Diferenciam-se também dos SGBD's relacional, pela flexibilidade de atuar **sem um esquema pré-definido**. Permitindo que sejam adicionados livremente, campos aos registros do banco de dados, sem ter de definir primeiro quaisquer mudanças na estrutura.
- Não resolvem todo tipo de problema. E pode atuar perfeitamente em conjunto, com outra abordagem de banco de dados, como a relacional (**Persistência Poliglota**).
- Os Banco de Dados NoSQL se enquadram em tecnologias open source.



# Classificação BD NoSQL

- Os banco de dados NoSQL se classificam em:
  1. Orientado a chave-valor.
  2. Orientado a família de colunas.
  3. Orientado a documentos.
  4. Orientado a grafos.

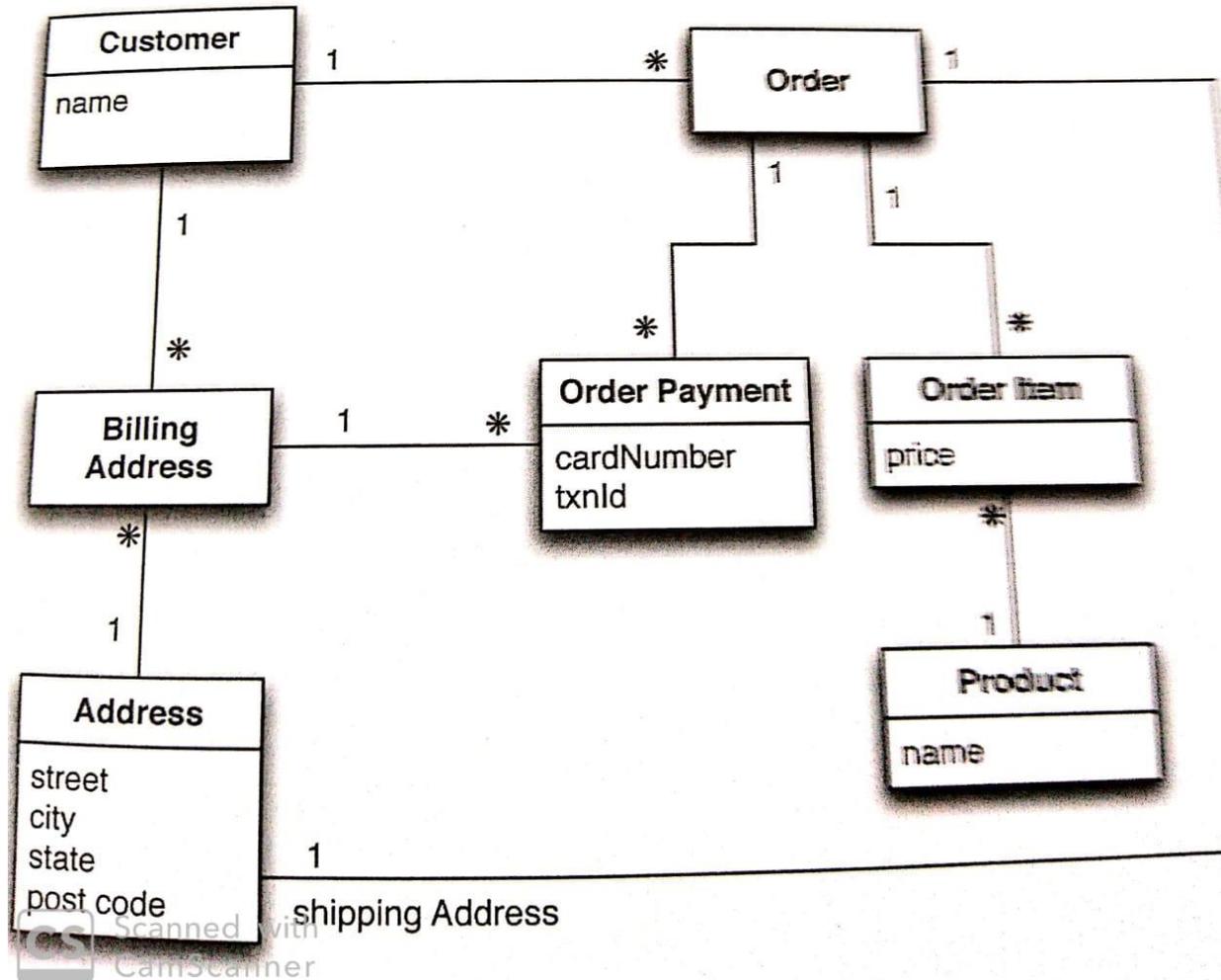


# Modelo de dados agregados

- Em detrimento do modelo relacional, que temos a necessidade de realizar operações de junção, para formar uma determinada informação a ser apresentado num relatório, um agregado, é visto em termos de um registro complexo, que permita que listas e outras estruturas de dados sejam aninhados dentro dele.
- **Um agregado** é um conjunto de objetos relacionados que desejamos tratar como uma unidade.
- Dados estruturados x não estruturados.

# Caso de uso 1 – Gerenciamento de Vendas

Diagrama de classe  
UML



# Caso de uso 1 – Gerenciamento de Vendas

Modelagem de dados oriundas de um SGBD 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

# Caso de uso 1 – Gerenciamento de Vendas

**Modelagem de dados oriundas de dois agregados: cliente e pedido.**

```
// em clientes
{
  "id":1,
  "name":"Martin",
  "billingAddress":[{"city":"Chicago"}]
}
// em pedidos
{
  "id":99,
  "customerId":1,
  "orderItems":[
    {
      "productId":27,
      "price": 32.45,
      "productName": "NoSQL Essencial"
    }
  ],
  "shippingAddress":[{"city":"Chicago"}]
  "orderPayment":[
    {
      "ccinfo":"1000-1000-1000-1000",
      "txnId":"abelif879rft",
      "billingAddress": {"city": "Chicago"}
    }
  ],
}
```

CS Scanned with CamScanner

CS Scanned with CamScanner

# Consequência da orientação a agregados

- O motivo mais importante para a orientação a agregados é que ela realmente auxilia a execução em um cluster, que é um dos fatores propulsores do NoSQL.
- Nem todas as categorias de NoSQL, podem trabalhar com agregados, como é o caso, dos banco de dados orientado a grafos.
- A maioria dos banco de dados, trabalham com a linguagem JSON para construção de seus agregados.

# Agregados em BD chave-valor x documento

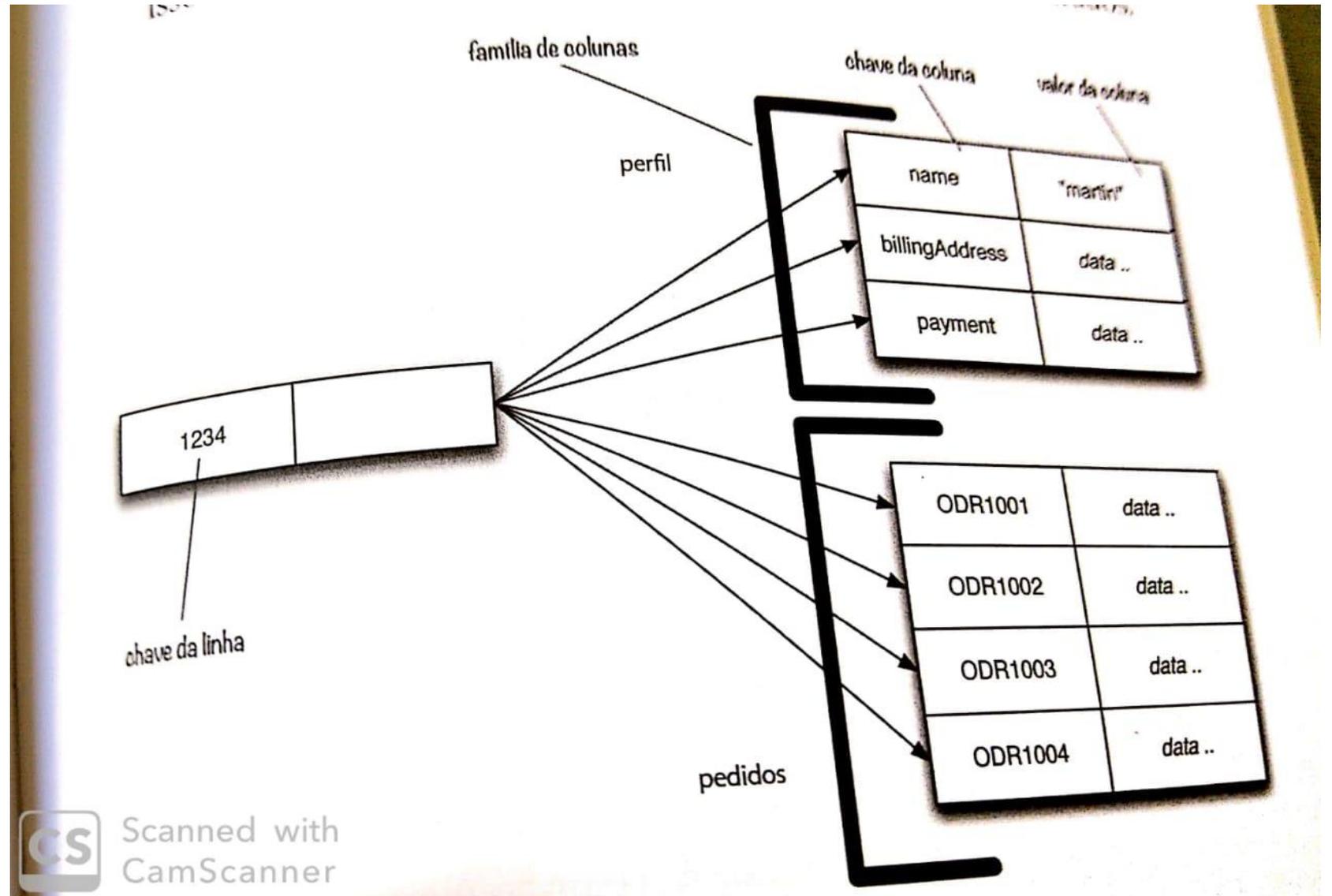
- Nos bancos de dados chave-valor, não é possível fazer uma pesquisa parcial em um agregado. Em toda a pesquisa por chave, o resultado será o agregado como um todo.
- Com um banco de dados orientado a documentos, podemos submeter consultas no agregado, com base em algum de seus campos. Além de viabilizar também, consultas e recuperações parciais.

# Agregados em BD família de coluna

- A melhor maneira de pensar nesse modelo é em uma estrutura agregada de dois níveis. No primeiro, teremos uma chave que vai descrever a identificação da linha, e o segundo, que são chamados de colunas, que vai especificar o campo e o valor de uma determinada coluna.
- Nesse caso, para realizar uma busca em um determinado campo, deve ser feito assim:
- `Get('1234', 'name')`.

# Agregados em BD família de coluna

Se executarmos o comando abaixo de consulta, o que será retornado?  
`Get('1234', 'name')`

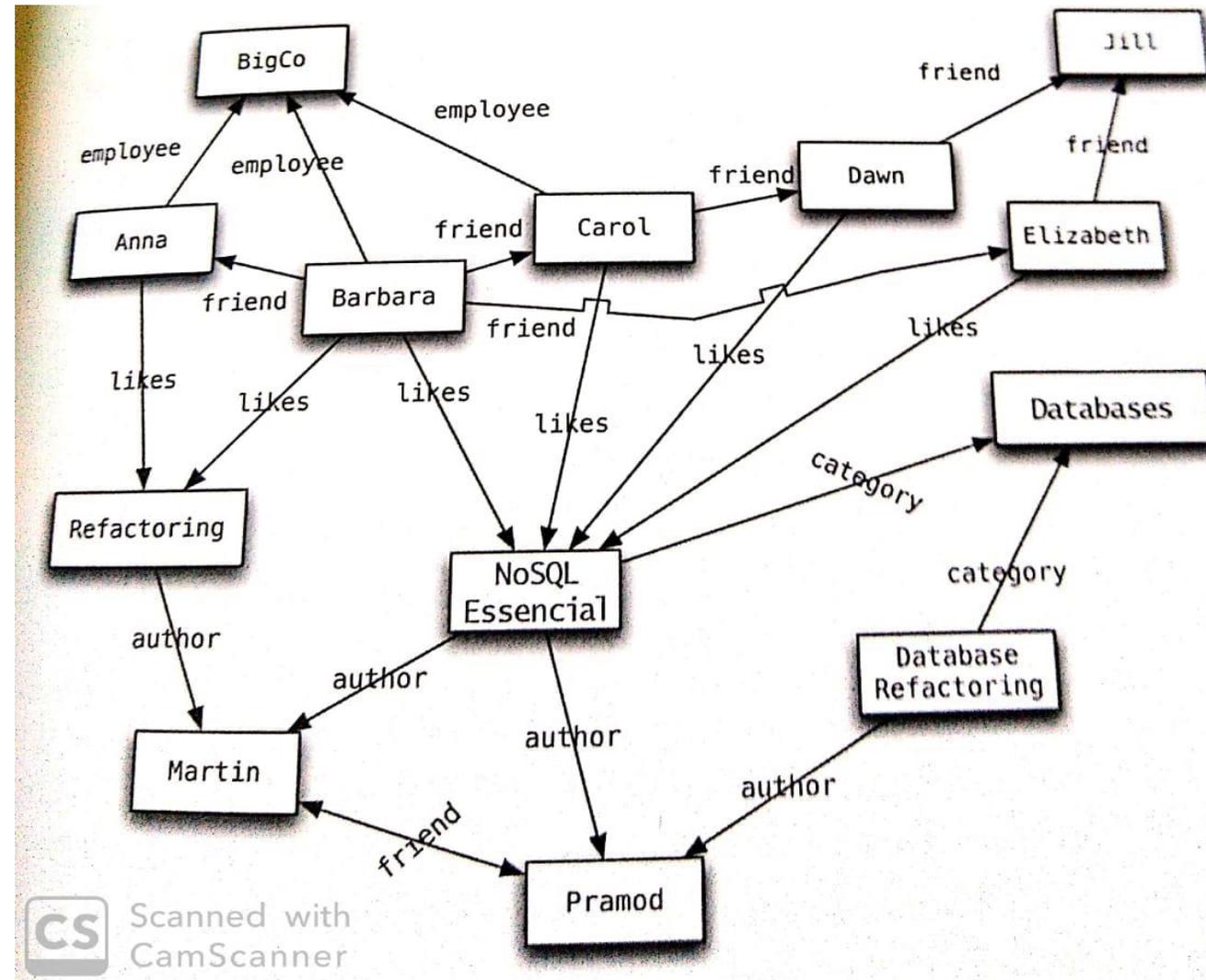


# Modelo de Dados sem Agregados

- Em NoSQL, o BD que não faz uso de agregados, é o orientado a grafos.
- Seu funcionamento deriva da estrutura de dados grafos, onde se possui nós e interconexões entre nós, que é chamado de aresta. Logo, essa abordagem, é específica para tratar de pequenos nós e grande relacionamentos entre os nós.

# Modelo de Dados sem Agregados

Típica consulta:  
“Encontre os livros  
na categoria Banco  
de Dados que  
tenham sido escritos  
por um autor de  
quem meu amigo  
goste”.



# Modelos de Distribuição

- Há dois caminhos a serem seguidos na distribuição dos dados: a replicação e a fragmentação.
- A replicação obtém os mesmos dados e os copia em múltiplos nós. Já a fragmentação obtém dados diferentes e os coloca em nós diferentes.
- A replicação pode incluir:
  1. Mestre-escravo.
  2. Ponto a ponto.



# Modelos de Distribuição – Um único servidor

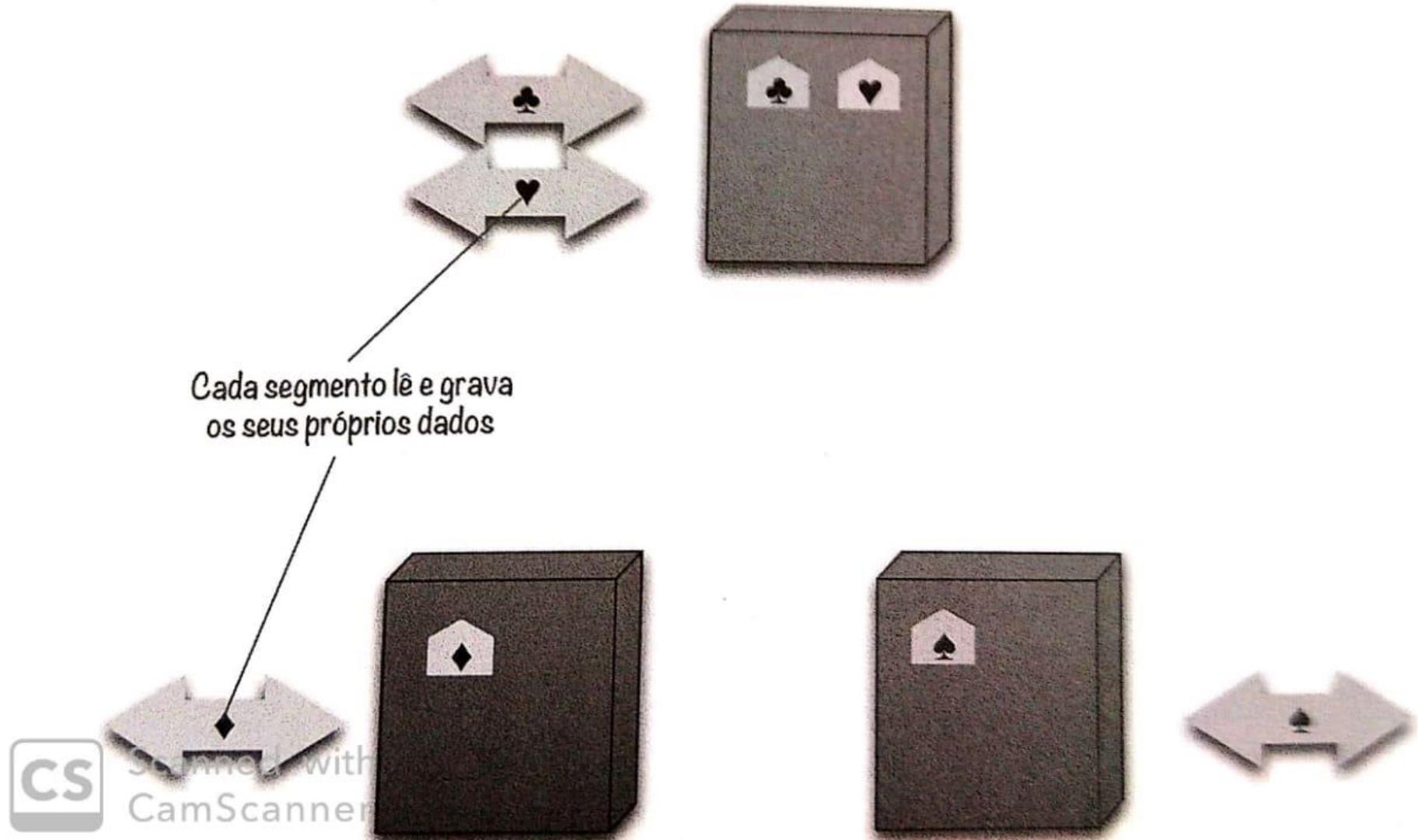
- Essa é a primeira e mais simples distribuição de todas. O banco de dados é armazenado em um único servidor para lidar com armazenamento e leitura de dados.
- Embora os banco de dados NoSQL, trabalhem em sua essência, melhor de forma distribuída (com exceção do banco de dados orientado a grafos), eles podem trabalhar de forma standalone também.



# Modelos de Distribuição – Fragmentação

- Coloca-se partes diferentes dos dados em servidores diferentes. Admite o uso da computação distribuída.
- Otimizada para performance tanto na leitura quanto na gravação dos dados.
- Permite ampliar horizontalmente as gravações nos dados.
- Muitas das vezes pode gerar um overhead muito grande. Apenas utilizar esse modelo de distribuição quando os recursos do cluster foram suficientes para executá-lo.

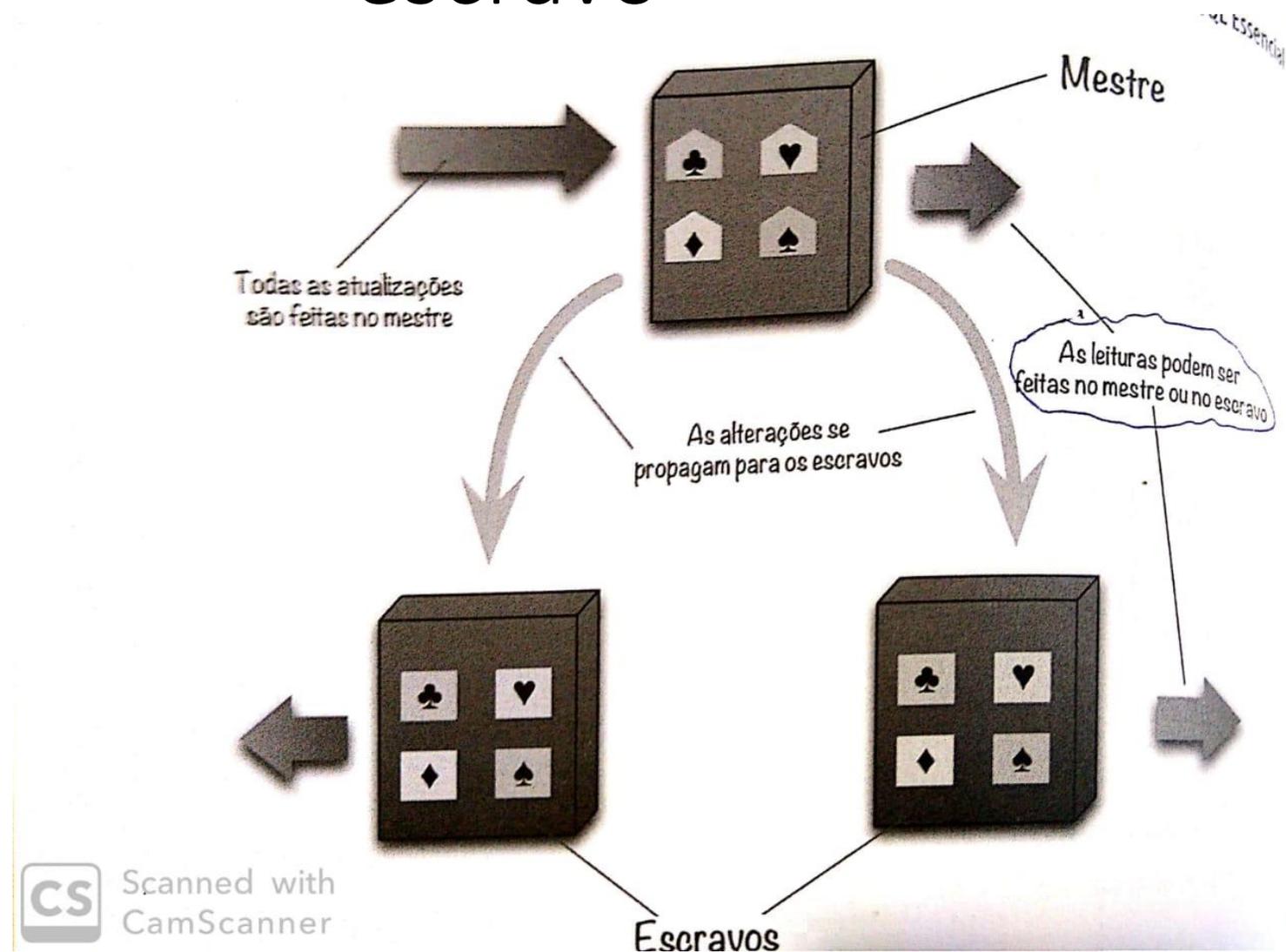
# Modelos de Distribuição – Fragmentação



# Modelos de Distribuição – Replicação mestre - escravo

- Nesse modelo, o nó mestre é responsável pela fonte oficial dos dados, bem como, pelo processamento de atualização nesses dados.
- Um processo de sincronização ocorre entre os nós escravos com o nó mestre. Caso o mestre falhe, os escravos podem lidar com as solicitações de leitura.
- Esse modelo é específico para trabalhar com a escalabilidade quando há um conjunto de dados com muitas leituras.

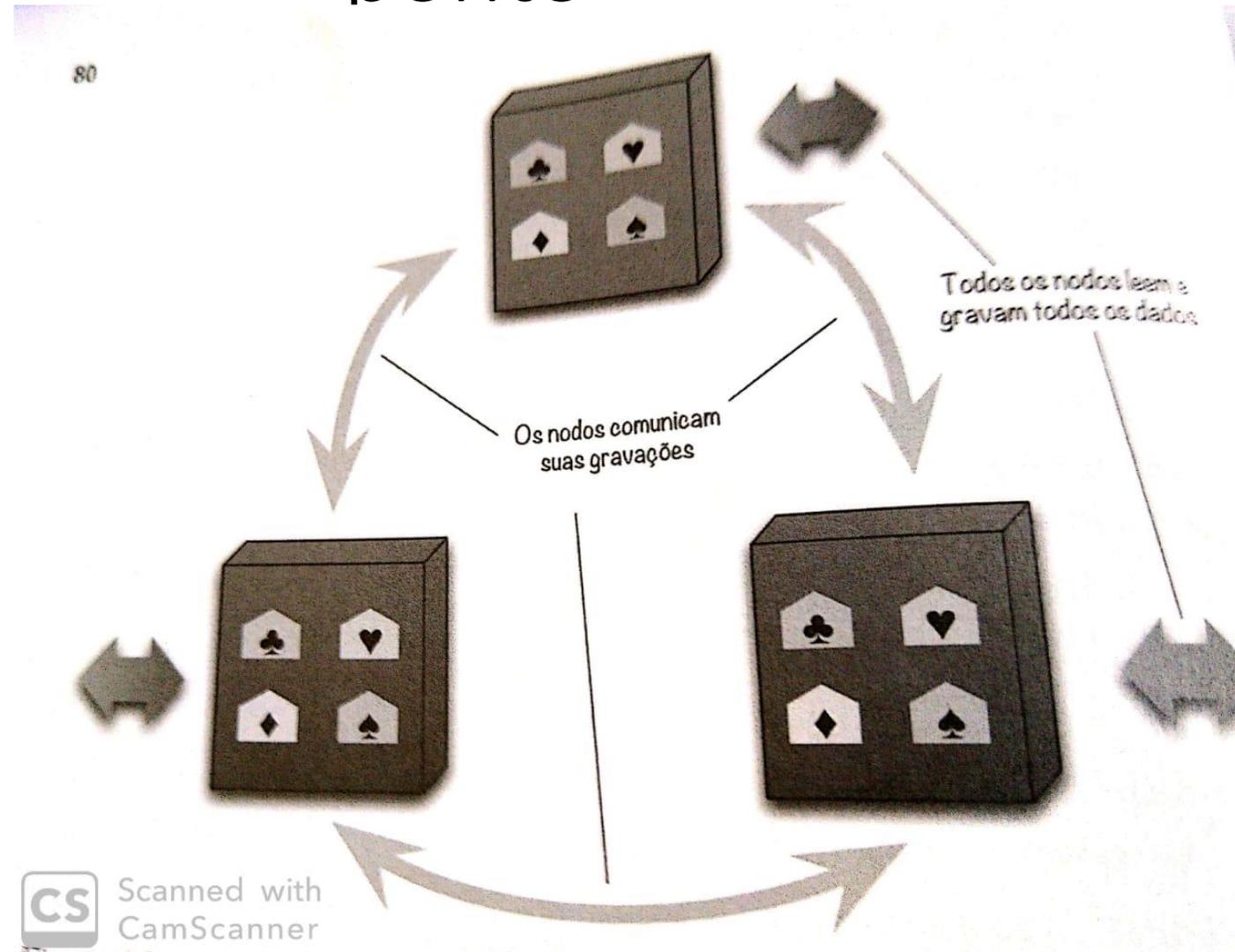
# Modelos de Distribuição – Replicação mestre - escravo



# Modelos de Distribuição – Replicação ponto a ponto

- Na replicação mestre – escravo, não há escalabilidade com a gravação dos dados. Ela fornece resiliência contra a falha de um escravo, mas não de um mestre.
- Enquanto que a replicação ponto a ponto, todos os nós tem peso igual, todos podem receber gravações e a perda de alguma delas não impede o acesso ao armazenamento de dados.
- Esse modelo permite que todos os nós façam leituras e gravações para todos os lados.

# Modelos de Distribuição – Replicação ponto a ponto



# Consistência

- Uma das maiores mudanças em um banco de dados relacional centralizado, para um banco de dados NoSQL distribuído, está em como se pensa na questão da consistência.
- Há pelo menos três problemas de consistência:
  1. Consistência de Atualização.
  2. Consistência de Leitura.
  3. Consistência de Replicação.



# Consistência de Atualização

- Dois usuários, Ana e Gabriel, que administram um portal de finanças, ao examinarem o website percebem que o nome de um ativo de uma empresa está incorreto. Assim que percebem o equívoco, ambos resolvem atualizar o nome do ativo ao mesmo tempo.
- Esse tipo de conflito é chamado escrita-escrita: quando duas pessoas resolvem atualizar o mesmo item de dados ao mesmo tempo.
- Assim que as gravações chegarem ao servidor, este, decidirá aplicar uma e depois a outra. Nesse caso, pelo menos uma alteração será perdida.



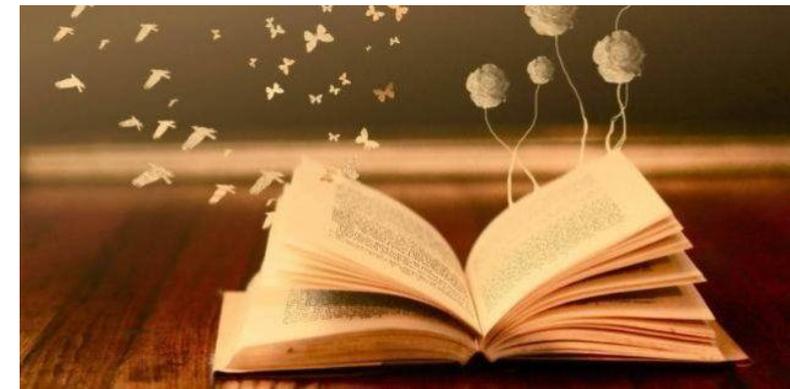
# Consistência de Atualização

- Esse tipo de consistência deu subsídios para a implementação dos sistemas de versionamento, sobretudo aqueles que funcionam de forma distribuída como o Git.
- Quando dois usuários tentam atualizar o mesmo item, o servidor devolve a notificação para ambos os usuários (quando há conflitos no mesmo arquivo), e cada usuário precisará resolver manualmente o conflito em seu workspace.

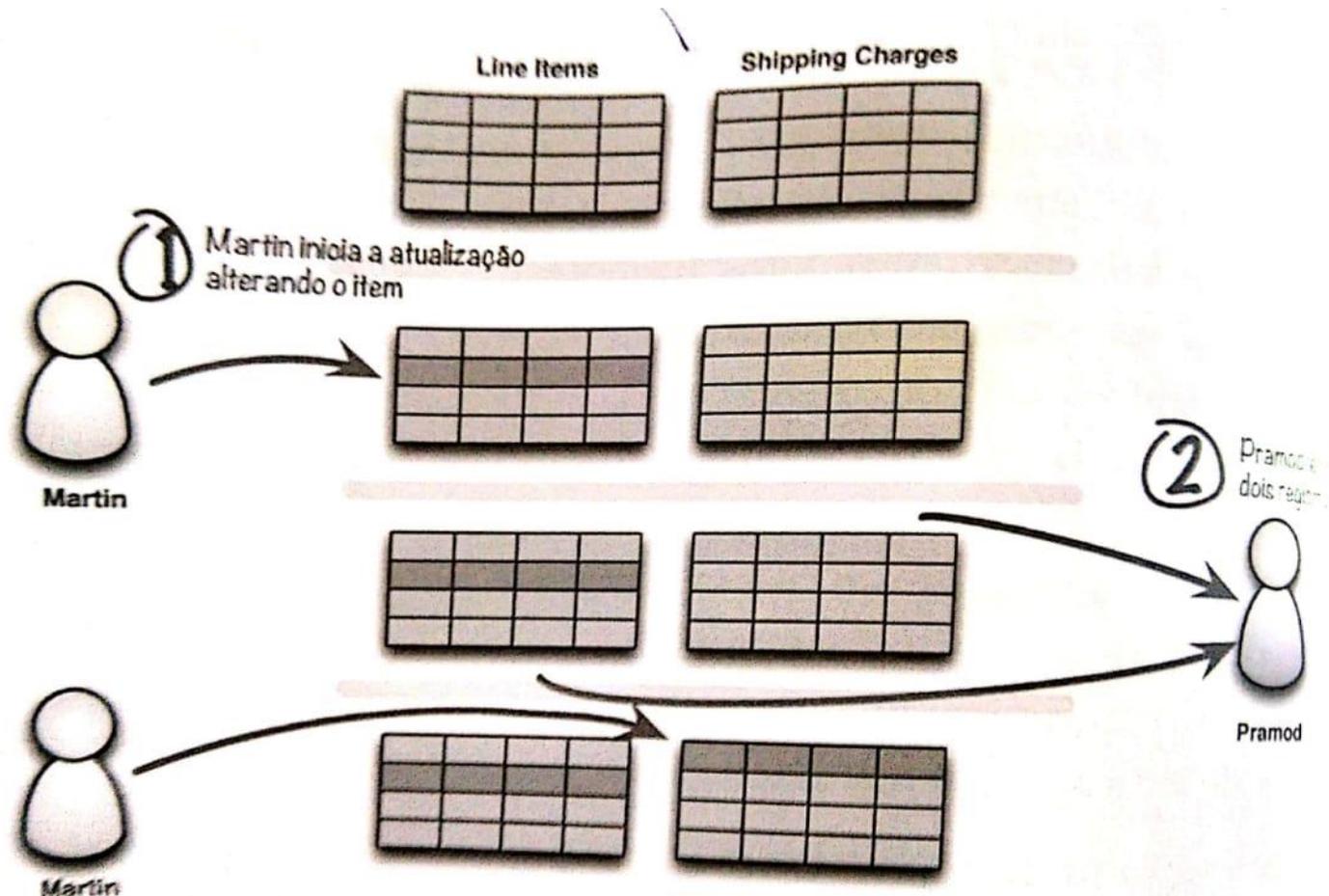


# Consistência de Leitura

- Imagine um pedido que tenha itens e uma taxa de frete. A taxa é calculada com base nos itens de pedido. A medida que novos itens seja adicionado, a taxa de frete precisa ser recalculada e atualizada.
- O perigo é justamente que o novo usuário realize a leitura antes de que haja essa nova atualização da taxa de frete.
- Esse exemplo é conhecido como conflito de leitura – gravação.



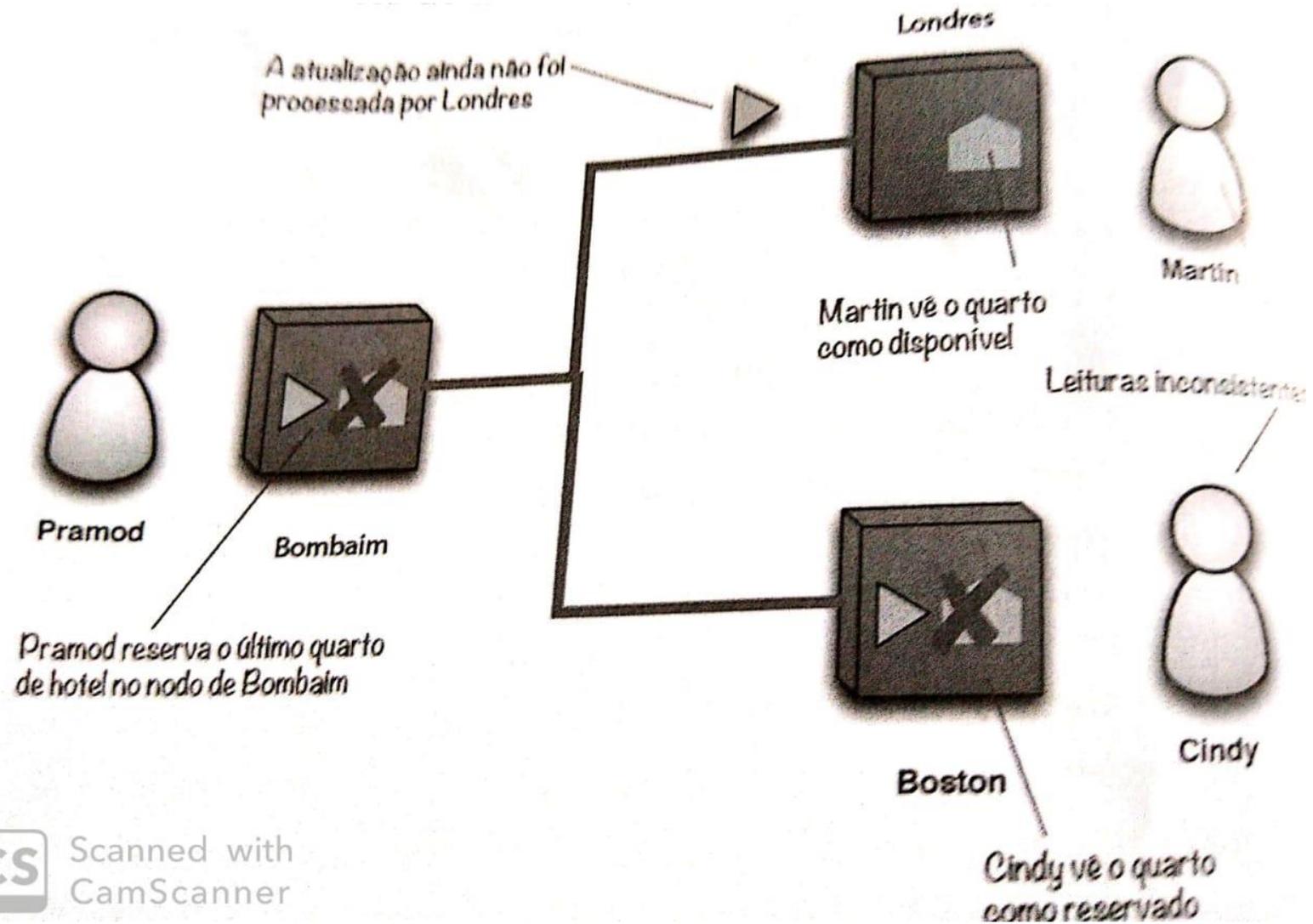
# Consistência de Leitura



# Consistência de Replicação

- Imagine que haja um último quarto de hotel disponível para um evento a qual você queira ir.
- Considerando que o sistema de reservas do hotel é executado em muitos nós, o casal Martin e Cindy está pensando em reservar esse quarto, porém ambos, estão discutindo pelo telefone. Martin está em Londres e Cindy em Boston.
- Enquanto isso, Pramod, que também vai participar do evento, reserva o último quarto do hotel. Considerando que a replicação demore um pouco a ser persistida em todos os nós, Cindy vê o quarto reservado no hotel, porém Martin, que está mais longe geograficamente, o vê disponível.

# Consistência de Replicação



# Teorema CAP

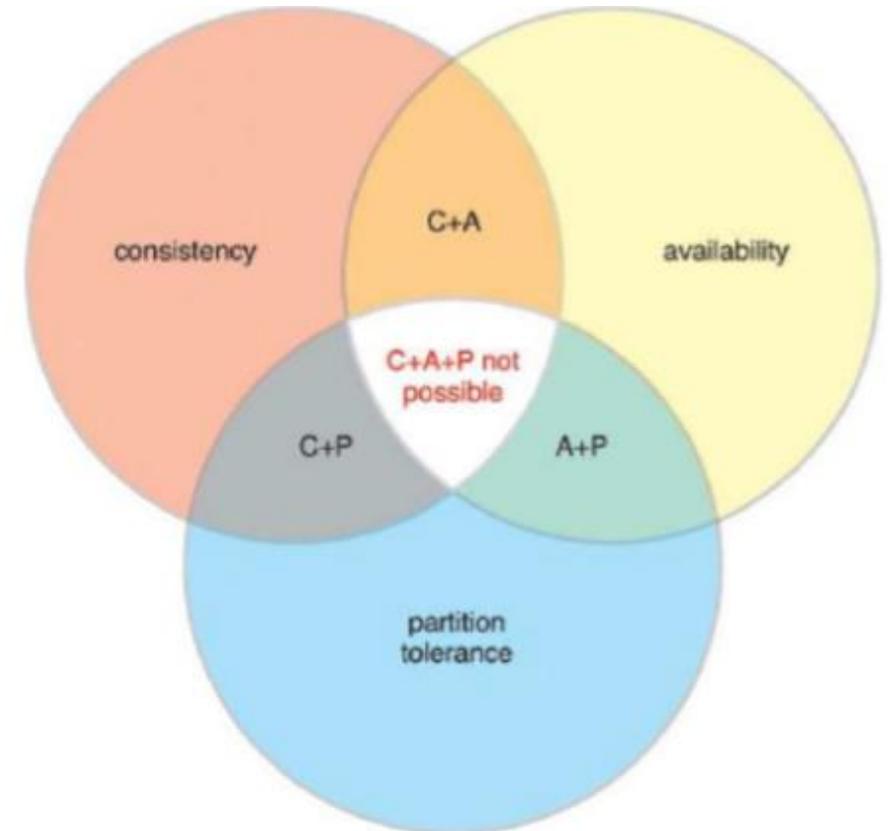
- O Teorema CAP se divide em três propriedades:

1. Consistência

2. Disponibilidade

3. Tolerância a partições.

É aceitável que obtenha-se pelo menos duas dessas características.



# Teorema CAP

- Um sistema com um único servidor, é o exemplo típico de CA (Consistência + Disponibilidade, mas não tolerância a partições).
- Considerando o exemplo de Pramod e Martin, sobre a reserva de hotel, podemos ter a tolerância a partições, pois o sistema é distribuído, bem como a consistência se o problema da replicação foi devidamente resolvido. Porém, se em algum momento, a conexão da rede falhar, nenhum dos sistemas poderá emitir reserva no hotel, logo não teríamos a disponibilidade nesse caso.

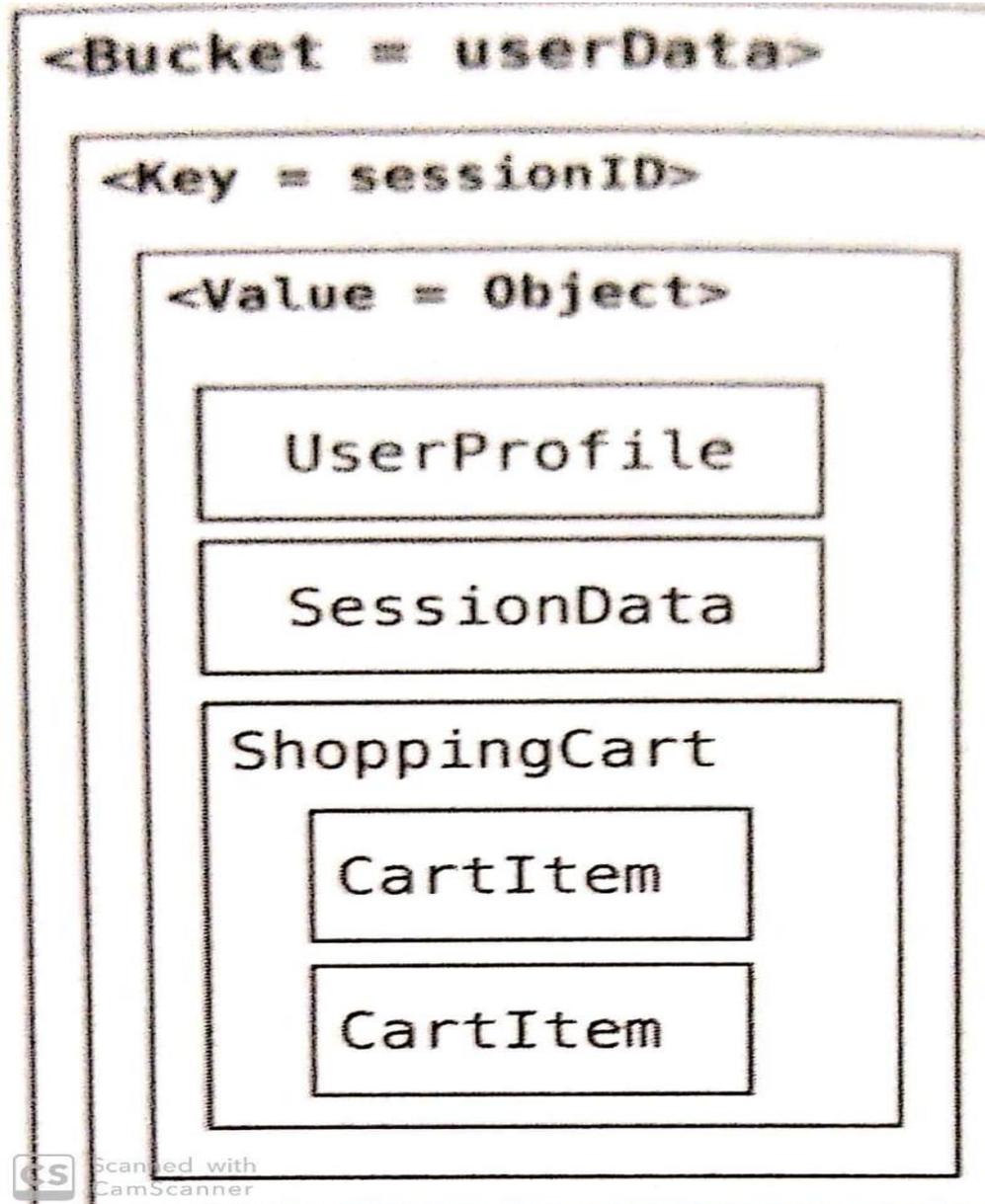
# Banco de Dados Orientado a Chave-Valor

- No banco de dados de chave-valor, o usuário pode obter o valor por meio de uma chave, inserir um valor para uma determinada chave ou apagar uma chave do depósito de dados.
- Como depósitos de chave-valor sempre fazem o acesso pela chave primária, geralmente possui um ótimo desempenho e podem ser escalados facilmente.
- Exemplo:

Key: 1245 Value: "Fernando"; Key: 2343 Value: "Name=Rodrigo,age=25".

Oracle	Riak
Instância de banco de dados	Cluster Riak
Tabela	Bucket
Linha	chave-valor
Rowid	Chave

# Banco de Dados Orientado a Chave-Valor



O valor da chave não precisa ser necessariamente algo tão simples como o exemplo anterior, o valor pode ser uma estrutura de dados um pouco mais complexa, como uma lista, conjunto ou hash

# Casos de Uso

- **Armazenamento de informações de sessão:** Geralmente toda a sessão web é única e recebe um valor único de sessionId. Armazenar essa informação num BD chave – valor pode conferir mais performance na consulta de informações de sessão.
- **Perfis de usuário, preferências:** Dados de usuários em aplicações, como: username, algum atributo único, bem como preferências no site, como: cor, fuso horário, dentre outras, podem ser armazenadas com BD chave – valor.
- **Dados de carrinho de compras:** Manter a persistência do carrinho de compras por usuários nos sites e sessões, também podem ser armazenadas com o BD chave – valor, colocando um agregado de valores para uma determinada chave por exemplo.

# Quando não usar

- **Relacionamento entre dados:** Se você precisar de relacionamentos entre diferentes conjunto de dados, ou a correlação entre os dados, esse banco de dados não é uma boa opção.
- **Consulta por dados parciais:** Se você pesquisar as chaves baseando-se em algo que esteja na parte do valor dos pares chave-valor, então esse banco de dados não poderá ajudar muito.

# Banco de Dados Orientado a Documentos

- Documentos são o conceito principal em banco de dados de documentos. Esses documentos podem ser XML, JSON, BSON, entre outros.
- Geralmente roda na porta 27017.

O campo `_id` é um campo encontrado em todos os documentos no MongoDB, assim como o Rowid no Oracle.

Oracle	MongoDB
Instância de banco de dados	Instância MongoDB
Esquema	Banco de dados
Tabela	Coleção
Linha	Documento
Rowid	<code>_id</code>
Junção	DBRef

# Banco de Dados Orientado a Documentos

Exemplo de um documento no MongoDB.

```
tradicionalmente, vamos  
{  
  "firstname": "Pramod",  
  "citiesvisited": [ "Chicago", "London", "Pune", "Bangalore" ],  
  "addresses": [  
    { "state": "AK",  
      "city": "DILLINGHAM",  
      "type": "R"  
    },  
    { "state": "MH",  
      "city": "PUNE",  
      "type": "R" }  
  ],  
  "lastcity": "Chicago"  
}
```



# Recursos de Consulta

SQL	MONGO
SELECT * FROM ORDER	Db.order.find()
SELECT * FROM ORDER where id="54312"	Db.order.find({"id":"54312"})

# Casos de Uso

- **Sistema de Gerenciamento de Conteúdo:** Uma vez que os banco de dados de documentos não tem esquemas predefinidos e entendem documentos JSON, eles funcionam bem em sistemas de gerenciamento de conteúdo.
- **Análises Web ou em tempo real (analytics):** Uma vez que partes do documento podem ser atualizadas, é muito fácil armazenar visualizações de páginas ou visitantes únicos.
- **Aplicativos de comércio eletrônico:** Aplicativos de e-commerce, precisam ter esquemas flexíveis para produtos e pedidos, assim como precisam ter a capacidade de alterar seus modelos de dados sem refatorações custosas de banco de dados ou migração de dados.

# Quando não usar

- **Transações complexas que abranjam diferentes operações:** Se você precisar de operações atômicas em múltiplos documentos, os bancos de dados de documentos podem não ser o ideal. Entretanto, há alguns bancos de dados de documentos que suportam esses tipos de operações, como o RavenDB.

# Banco de Dados Orientado a Família de Colunas

- Armazenam dados em família de colunas como linhas que tenham muitas colunas associadas, fazendo uso de uma chave de linha.
- São grupos de dados relacionados que, frequentemente, são acessados juntos. Por exemplo, muitas vezes acessamos as informações de perfil de um cliente ao mesmo tempo, mas não seus pedidos.

SGBDR	Cassandra
Instância de bancos de dados	Cluster
Banco de dados	Keyspace
Tabela	Família de colunas
Linha	Linha
Coluna (a mesma para todas as linhas)	Coluna (podem ser diferentes por linha)

# Casos de Uso

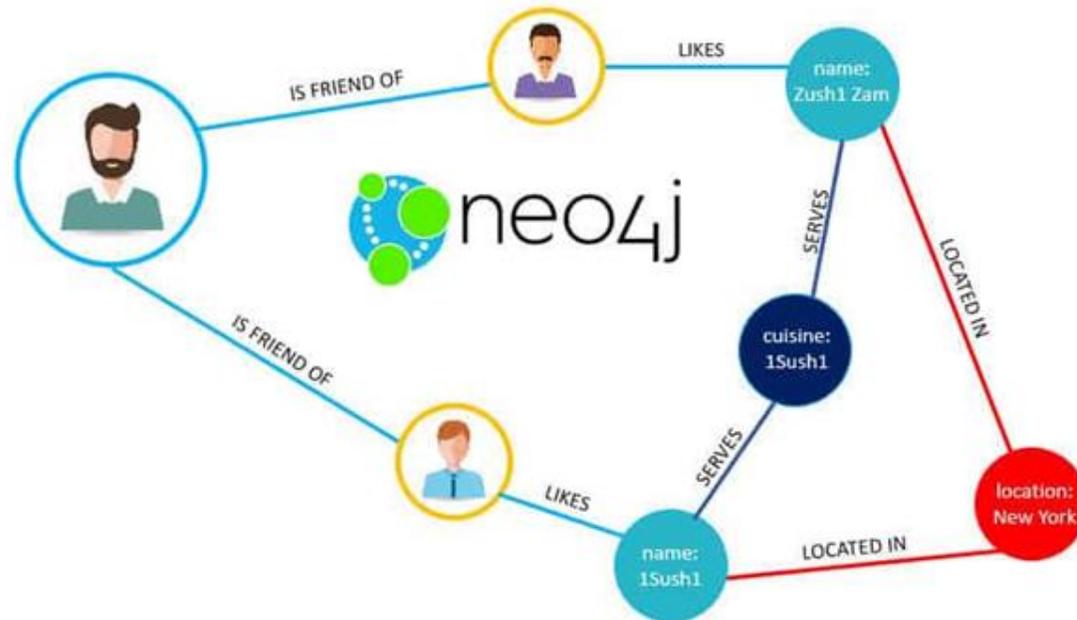
- **Registro de eventos (log):** Ótima escolha para armazenar informações sobre eventos, como por exemplo, estados do aplicativo ou erros encontrados por ele.
- **Contadores:** Muitas vezes, em aplicativos web, você precisa contar e categorizar visitantes de uma página para fazer análises.

# Quando não usar

- O Cassandra não é apropriado para os primeiros protótipos ou para adoções iniciais de tecnologia: durante as primeiras etapas, não temos certeza sobre como os padrões de consulta poderão mudar e, quando mudarem, teremos de alterar o formato das família de colunas.

# Banco de Dados Orientado a Grafos

- Baseado na estrutura de dados grafos, em que se tem nós, que representam entidades, e arestas, que representam relacionamentos.



# Casos de Uso

- **Dados conectados:** É nas redes sociais que os banco de dados orientado a grafos podem ser instalados e utilizados de maneira muito eficaz. Esses grafos não são apenas para relacionar amigos. Pode também representar funcionários, seu conhecimento, e onde trabalharam com outros funcionários em outros projetos.
- **Mecanismos de recomendação:** À medida que os nós e relacionamentos são criados no sistema, eles podem ser utilizados para fazer recomendações do tipo “seus amigos também compraram este produto” ou “ao faturar este item, estes outros geralmente também o são”.

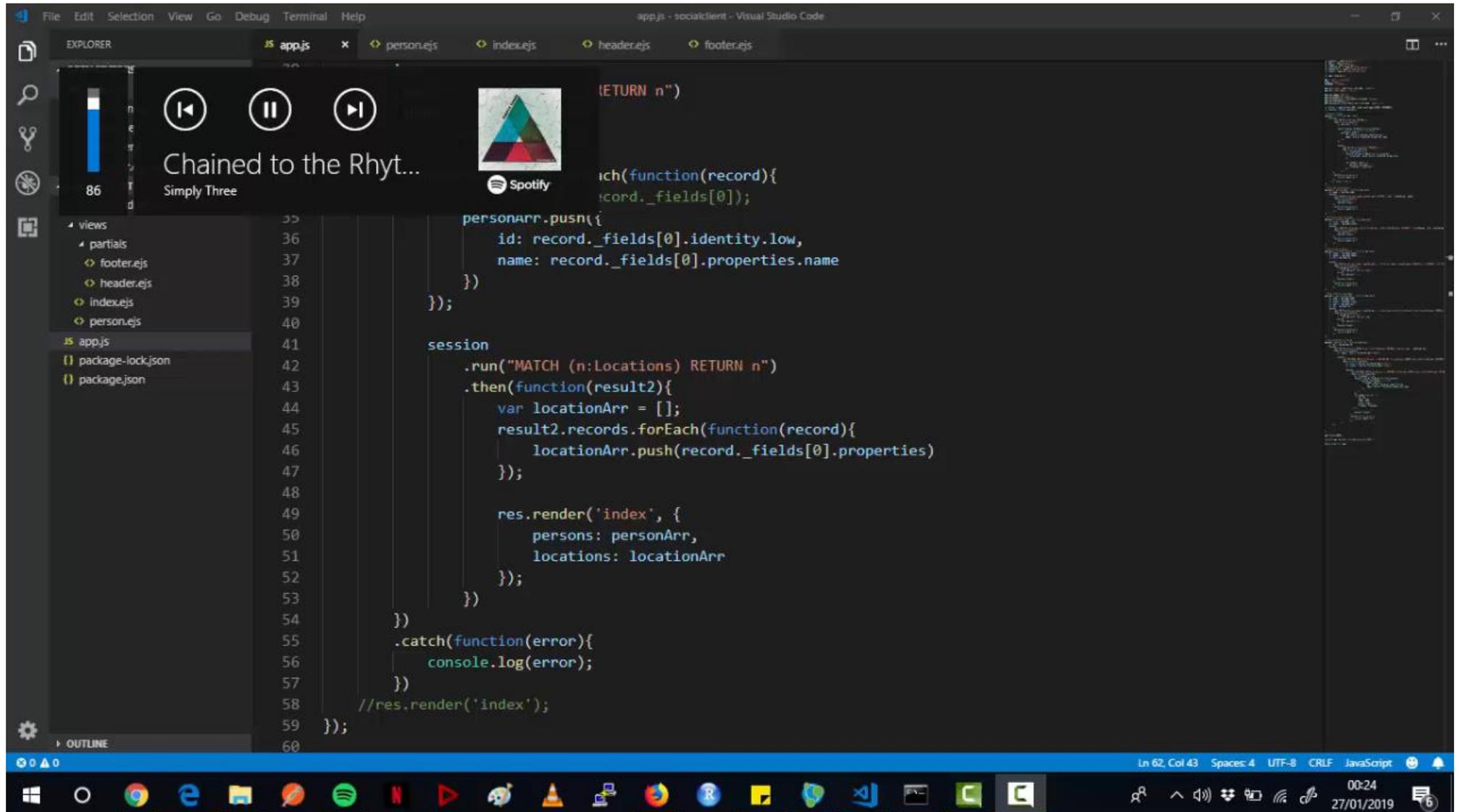
# Quando não usar

- Quando você quiser atualizar todas as entidades ou um subconjunto delas – por exemplo, em um aplicativo de análise no qual todas as entidades talvez precisem ser atualizadas com uma propriedade alterada – bancos de dados de grafos podem não ser ideais, uma vez que alterar uma propriedade em todos os nós não é uma operação direta.

# Quais linguagens suportam NoSQL?

	Amazon Dynamo	Neo4j	Cassandra	MongoDB
C				X
C#				X
C++			X	X
Go			X	X
Java	X	X	X	X
Javascript	X			X
Node.js	X	X	X	X
Perl			X	X
PHP	X	X	X	X
Python		X	X	X
Ruby	X	X	X	X
Scala		X	X	X

# Caso de uso – Social media



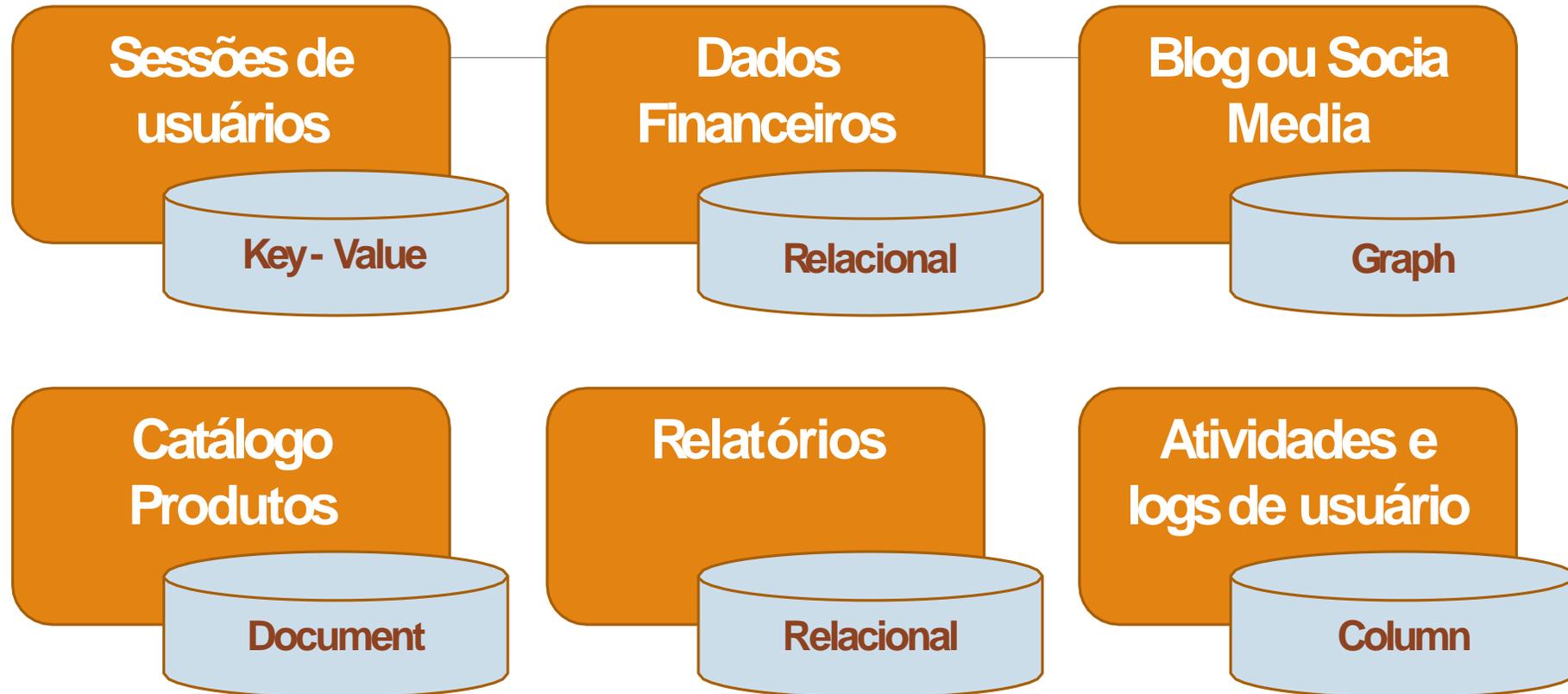
The screenshot displays the Visual Studio Code interface for a project named 'socialclient'. The Explorer sidebar on the left shows a file structure with folders 'views' and 'partials', and files 'app.js', 'package-lock.json', and 'package.json'. The main editor area shows the content of 'app.js', which includes a Spotify player widget and a REST client. The Spotify player is titled 'Chained to the Rhythm...' by 'Simply Three' and is currently playing track 86. The REST client is configured with a 'MATCH (n:Locations) RETURN n' query and is rendering the results in the 'index' view, passing 'personArr' and 'locationArr' to the template. The code in 'app.js' uses the REST client to fetch location data and push it into 'locationArr'.

```
personArr.push({
  id: record._fields[0].identity.low,
  name: record._fields[0].properties.name
});
});

session
  .run("MATCH (n:Locations) RETURN n")
  .then(function(result2){
    var locationArr = [];
    result2.records.forEach(function(record){
      locationArr.push(record._fields[0].properties)
    });

    res.render('index', {
      persons: personArr,
      locations: locationArr
    });
  });
})
.catch(function(error){
  console.log(error);
})
//res.render('index');
});
```

# Quando e qual utilizar?

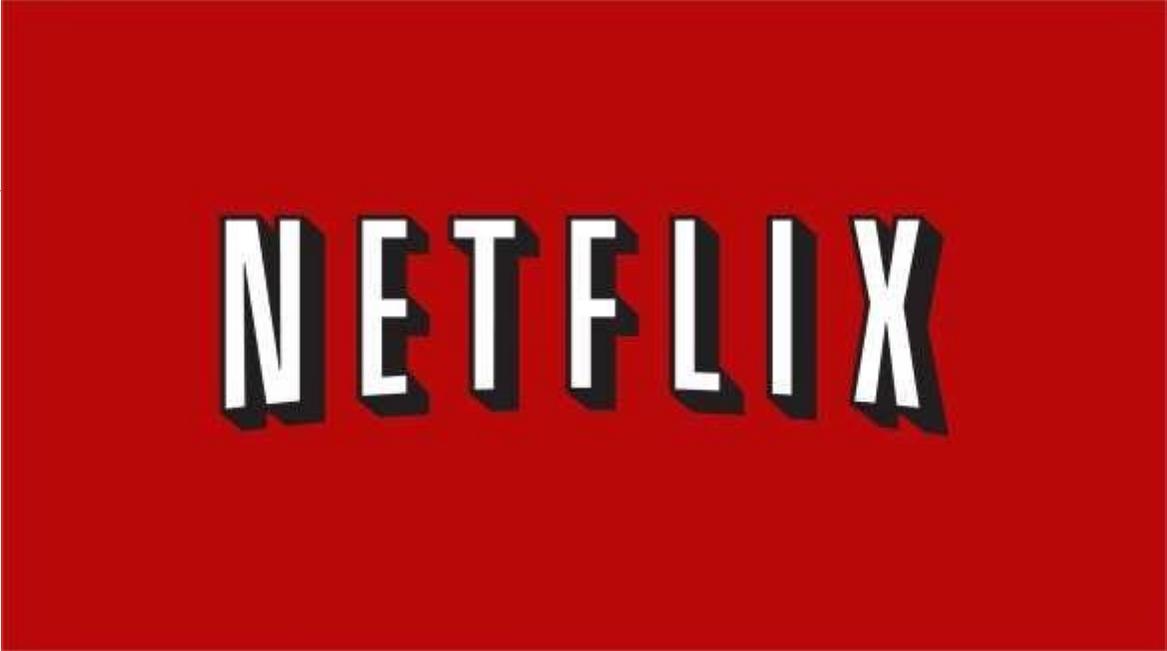


Fonte: Martin Fowler

Fonte: <http://www.martinfowler.com/bliki/PolyglotPersistence.html>

---

# Cases

The Netflix logo is displayed in white, bold, sans-serif capital letters with a slight 3D effect, centered on a solid red rectangular background. The letters are slightly shadowed to the right and bottom, giving them a floating appearance. The red background is framed by thin white horizontal lines on the left and right sides.

NETFLIX

## **SGBD:**

sistema de processamento de faturas mensais

## **NOSQL:**

Sistema focado em recomendações de melhores filmes.

---

# Cases



## **SGBD:**

Sistemas de processamento de ordem de venda

## **NOSQL:**

Sistema de pesquisa, recomendações e adaptações de preços em tempo real

---

# Cases



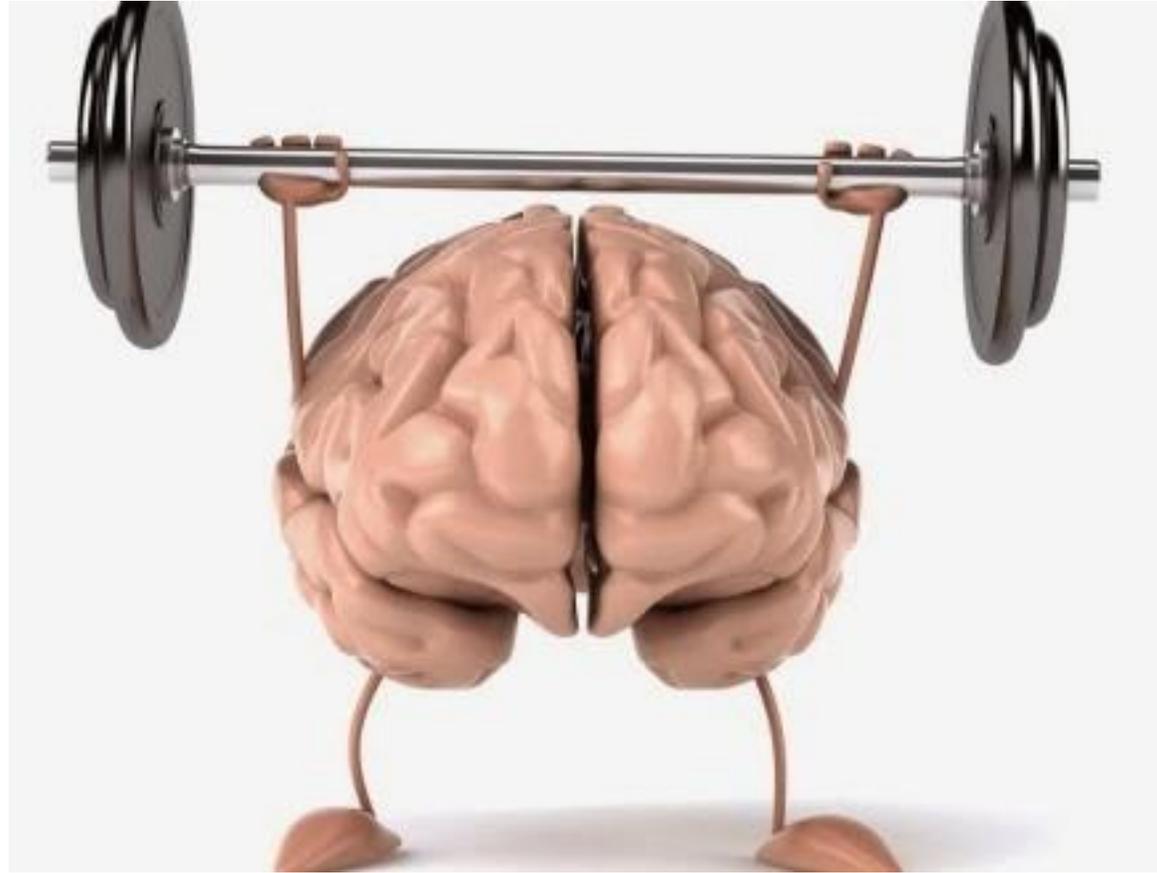
## **SGBD:**

Dados de clientes, produtos e RH

## **NOSQL:**

Explorar, analisar e virtualização de dados

---



**Q1) [CCV-UFC UFC 2019]** Sobre os banco de dados NoSQL, assinale a afirmativa correta.

- A) Bancos de dados NoSQL não podem ser indexados.
- B) Bancos de dados NoSQL são considerados banco de dados relacionais.
- C) Nos bancos de dados NoSQL devem ser definidos um esquema de dados fixo antes de qualquer operação.
- D) São exemplos de bancos de dados NoSQL: MongoDB, Firebird, DynamoDB, SQLite, Microsoft Access e Azure Table Storage.
- E) Os bancos de dados NoSQL usam diversos modelos para acessar e gerenciar dados, como documento, gráfico, chave-valor, em memória e, pesquisa.

**Q1) [CCV-UFC UFC 2019]** Sobre os banco de dados NoSQL, assinale a afirmativa correta.

- A) Bancos de dados NoSQL não podem ser indexados.
- B) Bancos de dados NoSQL são considerados banco de dados relacionais.
- C) Nos bancos de dados NoSQL devem ser definidos um esquema de dados fixo antes de qualquer operação.
- D) São exemplos de bancos de dados NoSQL: MongoDB, Firebird, DynamoDB, SQLite, Microsoft Access e Azure Table Storage.
- E) Os bancos de dados NoSQL usam diversos modelos para acessar e gerenciar dados, como documento, gráfico, chave-valor, em memória e, pesquisa.**

**Q2) [CESPE POLICIA FEDERAL 2018]** Acerca de banco de dados, julgue o item seguinte.

NoSQL são bancos de dados que não aceitam expressões SQL e devem ser armazenados na nuvem.

**Q3) [CESPE CNJ 2013]** No que se refere ao desenvolvimento *web* de alto desempenho,  
julgue o item subsequente  
A escalabilidade dos bancos de dados NoSQL é garantida pela ausência de um esquema (*scheme free*).

**Q2) [CESPE POLICIA FEDERAL 2018]** Acerca de banco de dados, julgue o item seguinte.

NoSQL são bancos de dados que não aceitam expressões SQL e devem ser armazenados na nuvem. **ERRADO**

**Q3) [CESPE CNJ 2013]** No que se refere ao desenvolvimento *web* de alto desempenho, julgue o item subsequente

A escalabilidade dos bancos de dados NoSQL é garantida pela ausência de um esquema (*scheme free*). **ERRADO**

**Q4) [CESGRANRIO BANCO DO BRASIL 2018]** Uma das características inerentes ao modelo chave-valor de bancos de dados NoSQL é a(o)

- A) suporte à compreensão da semântica do valor associado à chave
- B) favorecimento à evolução de esquemas conceituais
- C) dependência de linguagem de consulta específica
- D) estrutura de armazenamento interna complexa
- E) falta de suporte à concorrência

**Q5) [CESPE TJ-SE 2014]** Acerca de bancos de dados semiestruturados e bancos de dados NOSQL, julgue os itens subsecutivos.

Devido à escalabilidade esperada para os bancos de dados NOSQL, a implementação desses bancos utiliza modelos de armazenamento de dados totalmente distintos dos utilizados em sistemas relacionais.

**Q4) [CESGRANRIO BANCO DO BRASIL 2018]** Uma das características inerentes ao modelo chave-valor de bancos de dados NoSQL é a(o)

A) suporte à compreensão da semântica do valor associado à chave

**B) favorecimento à evolução de esquemas conceituais**

C) dependência de linguagem de consulta específica

D) estrutura de armazenamento interna complexa

E) falta de suporte à concorrência

**Q5) [CESPE TJ-SE 2014]** Acerca de bancos de dados semiestruturados e bancos de dados NOSQL, julgue os itens subsecutivos.

**Devido à escalabilidade esperada para os bancos de dados NOSQL, a implementação desses bancos utiliza modelos de armazenamento de dados totalmente distintos dos utilizados em sistemas relacionais. ERRADO**

**Q6) [ESAF ESAF 2015]** Em relação a Big Data e NoSQL, é correto afirmar que

- A) os “3 Vs” principais do Big Data referem-se a Volume, Velocidade e Versatilidade de dados.
- B) na era do Big Data, as únicas estratégias eficientes para garantir a privacidade são consentimento individual, opção de exclusão e anonimização.
- C) o Hadoop, o mais conhecido e popular sistema para gestão de Big Data, foi criado pela IBM, a partir de sua ferramenta de Data Mining WEKA.
- D) o NoSQL é um sistema relacional, distribuído, em larga escala, muito eficaz na organização e análise de grande quantidade de dados.
- E) o Cassandra é um sistema de banco de dados baseado na abordagem NoSQL, originalmente criado pelo Facebook, no qual os dados são identificados por meio de uma chave.

**Q6) [ESAF ESAF 2015]** Em relação a Big Data e NoSQL, é correto afirmar que

- A) os “3 Vs” principais do Big Data referem-se a Volume, Velocidade e Versatilidade de dados.
- B) na era do Big Data, as únicas estratégias eficientes para garantir a privacidade são consentimento individual, opção de exclusão e anonimização.
- C) o Hadoop, o mais conhecido e popular sistema para gestão de Big Data, foi criado pela IBM, a partir de sua ferramenta de Data Mining WEKA.
- D) o NoSQL é um sistema relacional, distribuído, em larga escala, muito eficaz na organização e análise de grande quantidade de dados.
- E) o Cassandra é um sistema de banco de dados baseado na abordagem NoSQL, originalmente criado pelo Facebook, no qual os dados são identificados por meio de uma chave.**

**Q7) [INSTITUTO AOCP CASAN 2016]** Em relação à Big Data e NoSQL, é correto afirmar que:

- A) são conceitos concorrentes, portanto não podem ser implementados juntos.
- B) são conceitos que se complementam e com características eficientes para trabalhar com pequenas quantidades de informações.
- C) são duas ferramentas de empresas concorrentes.
- D) são conceitos que se complementam.
- E) os SGBDs Oracle e MySQL são implementações desses conceitos.

**Q8) [CESPE CNJ 2013]** No que se refere ao desenvolvimento *web* de alto desempenho,  
julgue os itens subsequentes

Uma característica de bancos de dados NoSQL é o suporte à replicação de dados. Entre as abordagens utilizadas para replicação, inclui-se a mestre-escravo.

**Q7) [INSTITUTO AOCP CASAN 2016]** Em relação à Big Data e NoSQL, é correto afirmar que:

- A) são conceitos concorrentes, portanto não podem ser implementados juntos.
- B) são conceitos que se complementam e com características eficientes para trabalhar com pequenas quantidades de informações.
- C) são duas ferramentas de empresas concorrentes.
- D) são conceitos que se complementam.**
- E) os SGBDs Oracle e MySQL são implementações desses conceitos.

**Q8) [CESPE CNJ 2013]** No que se refere ao desenvolvimento *web* de alto desempenho,  
julgue os itens subsequentes

Uma característica de bancos de dados NoSQL é o suporte à replicação de dados. Entre as abordagens utilizadas para replicação, inclui-se a mestre-escravo. **CERTO**

**Q9) [CESPE TJ-SE 2014]** Acerca de bancos de dados semiestruturados e bancos de dados NOSQL, julgue os itens subsecutivos.

Bancos de dados NOSQL orientados a documentos são apropriados para o armazenamento de dados semiestruturados.

**Q10) [FGV IBGE 2016]** Considere as seguintes características de um projeto de banco de dados.

- I. O modelo de dados é conhecido a priori e é estável;
- II. A integridade dos dados deve ser rigorosamente mantida;
- III. Velocidade e escalabilidade são preponderantes.

Dessas características, o emprego de bancos de dados NoSQL é favorecido somente por:

- A) I;    B) I e II;    C) II;    D) II e III;    E) III.

**Q9) [CESPE TJ-SE 2014]** Acerca de bancos de dados semiestruturados e bancos de dados NOSQL, julgue os itens subsecutivos.

Bancos de dados NOSQL orientados a documentos são apropriados para o armazenamento de dados semiestruturados. **CERTO**

**Q10) [FGV IBGE 2016]** Considere as seguintes características de um projeto de banco de dados.

- I. O modelo de dados é conhecido a priori e é estável;
- II. A integridade dos dados deve ser rigorosamente mantida;
- III. Velocidade e escalabilidade são preponderantes.

Dessas características, o emprego de bancos de dados NoSQL é favorecido somente por:

- A) I;    B) I e II;    C) II;    D) II e III;    **E) III.**

**Q11) [FGV IBG 2017]** Bancos de Dados NoSQL podem armazenar dados em diversos formatos não relacionais, como documentos compostos por pares de campo-e-valor (field-and-value), conforme a estrutura exemplificada a seguir.

```
{  
  field1: value1,  
  field2: value2,  
  field3: value3,  
  ...  
  fieldN: valueN  
}
```

O Banco de Dados NoSQL utilizado para armazenar documentos compostos por pares campo-e-valor, no formato BSON (JSON-like), é o:

- A) OpenLink Virtuoso;
- B) Neo4j;
- C) Apache HBase;
- D) MongoDB;
- E) Titan.

**Q11) [FGV IBG 2017]** Bancos de Dados NoSQL podem armazenar dados em diversos formatos não relacionais, como documentos compostos por pares de campo-e-valor (field-and-value), conforme a estrutura exemplificada a seguir.

```
{  
  field1: value1,  
  field2: value2,  
  field3: value3,  
  ...  
  fieldN: valueN  
}
```

O Banco de Dados NoSQL utilizado para armazenar documentos compostos por pares campo-e-valor, no formato BSON (JSON-like), é o:

- A) OpenLink Virtuoso;
- B) Neo4j;
- C) Apache HBase;
- D) MongoDB;**
- E) Titan.

**Q12) [CESGRARIO BANCO DO BRASIL 2018]** O termo NoSQL refere-se

- A) a uma abordagem teórica que segue o princípio de não utilização da linguagem SQL em bancos de dados heterogêneos.
- B) à renúncia às propriedades BASE (Basically Available, Soft state, Eventual consistency), potencializando seu espectro de uso.
- C) ao aumento da escalabilidade das bases de dados neles armazenados, aliado a um desempenho mais satisfatório no seu acesso.
- D) à facilidade de implementação de bases de dados normalizadas, com vistas a minimização de redundâncias no conjunto de dados.
- E) à implementação simultânea das três componentes do modelo CAP: consistência, disponibilidade e tolerância ao particionamento.

**Q12) [CESGRARIO BANCO DO BRASIL 2018]** O termo NoSQL refere-se

- A) a uma abordagem teórica que segue o princípio de não utilização da linguagem SQL em bancos de dados heterogêneos.
- B) à renúncia às propriedades BASE (Basically Available, Soft state, Eventual consistency), potencializando seu espectro de uso.
- C) ao aumento da escalabilidade das bases de dados neles armazenados, aliado a um desempenho mais satisfatório no seu acesso.
- D) à facilidade de implementação de bases de dados normalizadas, com vistas a minimização de redundâncias no conjunto de dados.
- E) à implementação simultânea das três componentes do modelo CAP: consistência, disponibilidade e tolerância ao particionamento.**

**Q13) [CESPE TCE-PA 2016]** Acerca de segurança de banco de dados e de desenvolvimento de *software*, julgue o item subsecutivo.

Os bancos de dados NoSQL são imunes a ataques de injeção SQL.

**Q14) [CESPE CNJ 2013]** No que se refere ao desenvolvimento *web* de alto desempenho,

julgue os itens subsequentes

Apesar de implementarem tecnologias distintas, todos os bancos de dados NoSQL apresentam em comum a implementação da tecnologia chave-valor.

**Q13) [CESPE TCE-PA 2016]** Acerca de segurança de banco de dados e de desenvolvimento de *software*, julgue o item subsecutivo.

Os bancos de dados NoSQL são imunes a ataques de injeção SQL. **ERRADO**

**Q14) [CESPE CNJ 2013]** No que se refere ao desenvolvimento *web* de alto desempenho,  
julgue os itens subsequentes

Apesar de implementarem tecnologias distintas, todos os bancos de dados NoSQL apresentam em comum a implementação da tecnologia chave-valor.  
**ERRADO**

**Q15) [FGV PROCEMPA 2014]** O teorema CAP (CAP Theorem) é um importante resultado teórico na Ciência da Computação, e frequentemente é referenciado na comparação entre sistemas de bancos de dados “tradicionais” e aqueles que são conhecidos pela sigla NoSQL. Esse teorema aborda as propriedades (ou garantias) que um sistema de banco de dados deve prover, e é central na discussão das conveniências de utilização de um ou outro modelo.

Assinale a opção que descreve, corretamente, o significado das letras na sigla CAP.

- a) Concurrency, Access, Time.
- b) Consistency, Atomicity, Parallel processing.
- c) Consistency, Availability, Partition tolerance.
- d) Concurrency, Availability, Pear processing.
- e) Control, Atomicity, Partition network.

**Q15) [FGV PROCEMPA 2014]** O teorema CAP (CAP Theorem) é um importante resultado teórico na Ciência da Computação, e frequentemente é referenciado na comparação entre sistemas de bancos de dados “tradicionais” e aqueles que são conhecidos pela sigla NoSQL. Esse teorema aborda as propriedades (ou garantias) que um sistema de banco de dados deve prover, e é central na discussão das conveniências de utilização de um ou outro modelo.

Assinale a opção que descreve, corretamente, o significado das letras na sigla CAP.

- a) Concurrency, Access, Time.
- b) Consistency, Atomicity, Parallel processing.
- c) Consistency, Availability, Partition tolerance.**
- d) Concurrency, Availability, Pear processing.
- e) Control, Atomicity, Partition network.

**Q16) [FGV PROCEMPA 2014]** Bancos de dados conhecidos como NoSQL podem ser particionados em diferentes servidores, o que introduz o problema de processar consultas que envolvem múltiplos nós de processamento. Um modelo usualmente empregado nessas circunstâncias é conhecido como:

- a) CAP Theorem.
- b) Map/Reduce;
- c) Hash tables;
- d) Clustered columns;
- e) Data Thread.

**Q16) [FGV PROCEMPA 2014]** Bancos de dados conhecidos como NoSQL podem ser particionados em diferentes servidores, o que introduz o problema de processar consultas que envolvem múltiplos nós de processamento. Um modelo usualmente empregado nessas circunstâncias é conhecido como:

- a) CAP Theorem.
- b) Map/Reduce;**
- c) Hash tables;
- d) Clustered columns;
- e) Data Thread.

**Q17) [FGV TJBA 2015]** Analise as afirmativas a respeito da classe de gerenciadores de bancos de dados, surgida em anos recentes, conhecida como NoSQL:

- I. Mesmo sem suportar tabelas relacionais, baseiam-se em esquemas de dados previamente definidos;
- II. Suas estruturas não permitem o uso de linguagens do tipo do SQL para recuperação de dados;
- III. Garantem operações com as propriedades conhecidas pela sigla ACID;
- IV. Privilegiam a rapidez de acesso e a disponibilidade dos dados em detrimento das regras de consistência das transações.

O número de afirmativas corretas é:

- a) uma.
- b) duas;
- c) três;
- d) quatro;
- e) cinco.

**Q17) [FGV TJBA 2015]** Analise as afirmativas a respeito da classe de gerenciadores de bancos de dados, surgida em anos recentes, conhecida como NoSQL:

- I. Mesmo sem suportar tabelas relacionais, baseiam-se em esquemas de dados previamente definidos;
- II. Suas estruturas não permitem o uso de linguagens do tipo do SQL para recuperação de dados;
- III. Garantem operações com as propriedades conhecidas pela sigla ACID;
- IV. Privilegiam a rapidez de acesso e a disponibilidade dos dados em detrimento das regras de consistência das transações.

O número de afirmativas corretas é:

- a) uma.**
- b) duas;
- c) três;
- d) quatro;
- e) cinco.

**Q18) [CESPE TJDFT 2015]** A respeito de tipos de bancos de dados, julgue o item que se segue.

Sistemas de bancos de dados classificados como NoSQL permitem a inserção de dados sem que haja um esquema predefinido.

**Q19) [CESPE TJDFT 2015]** Julgue o item subsecutivo, referente às tecnologias de bancos de dados.

Em um banco de dados NoSQL orientado a documentos, a inexistência de um esquema impossibilita a definição de índices.

**Q20) [CESPE 2016 FUNPRESPE]** A respeito de banco de dados, julgue o próximo item.

Os bancos de dados NoSQL não permitem a atualização de seus dados, por serem orientados a documentos e(ou) coleções.

**Q18) [CESPE TJDFT 2015]** A respeito de tipos de bancos de dados, julgue o item que se segue.

**Sistemas de bancos de dados classificados como NoSQL permitem a inserção de dados sem que haja um esquema predefinido. CERTO.**

**Q19) [CESPE TJDFT 2015]** Julgue o item subsecutivo, referente às tecnologias de bancos de dados.

**Em um banco de dados NoSQL orientado a documentos, a inexistência de um esquema impossibilita a definição de índices. ERRADO.**

**Q20) [CESPE 2016 FUNPRESPE]** A respeito de banco de dados, julgue o próximo item.

**Os bancos de dados NoSQL não permitem a atualização de seus dados, por serem orientados a documentos e(ou) coleções. ERRADO.**

**Q21) [UFMT IF-MT 2015]** A coluna da esquerda apresenta tipos de modelos de dados relacionados com Banco de Dados NoSQL e a da direita, suas propriedades. Numere a coluna da direita de acordo com a da esquerda.

1 - Chave-Valor

2 - Documento

3 - Grafo

4 - Coluna

O conteúdo do dado é organizado em codificações semelhantes à YAML.

Tem enfoque principal no relacionamento entre os dados, com valores sendo armazenados em nós e suas ligações.

Para cada identificador único tem-se associado um valor que pode ser de qualquer tipo.

Organizado em tabelas com foco nos campos das tuplas.

Assinale a sequência correta.

a) 2, 3, 1, 4

b) 3, 1, 4, 2

c) 1, 4, 2, 3

d) 4, 2, 3, 1

**Q21) [UFMT IF-MT 2015]** A coluna da esquerda apresenta tipos de modelos de dados relacionados com Banco de Dados NoSQL e a da direita, suas propriedades. Numere a coluna da direita de acordo com a da esquerda.

1 - Chave-Valor

2 - Documento

3 - Grafo

4 - Coluna

O conteúdo do dado é organizado em codificações semelhantes à YAML.

Tem enfoque principal no relacionamento entre os dados, com valores sendo armazenados em nós e suas ligações.

Para cada identificador único tem-se associado um valor que pode ser de qualquer tipo.

Organizado em tabelas com foco nos campos das tuplas.

Assinale a sequência correta.

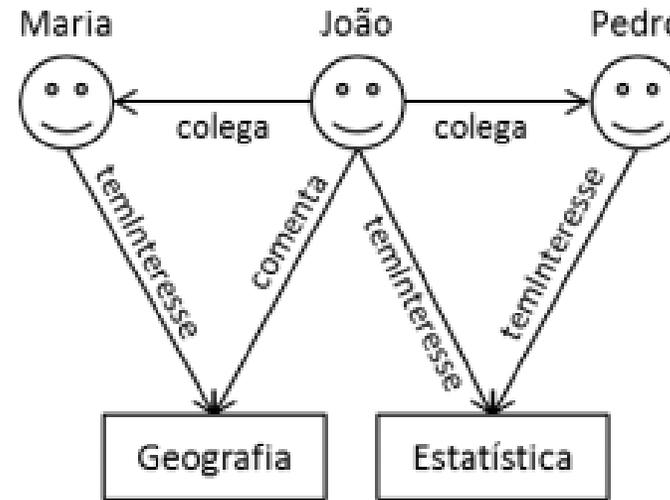
**a) 2, 3, 1, 4**

b) 3, 1, 4, 2

c) 1, 4, 2, 3

d) 4, 2, 3, 1

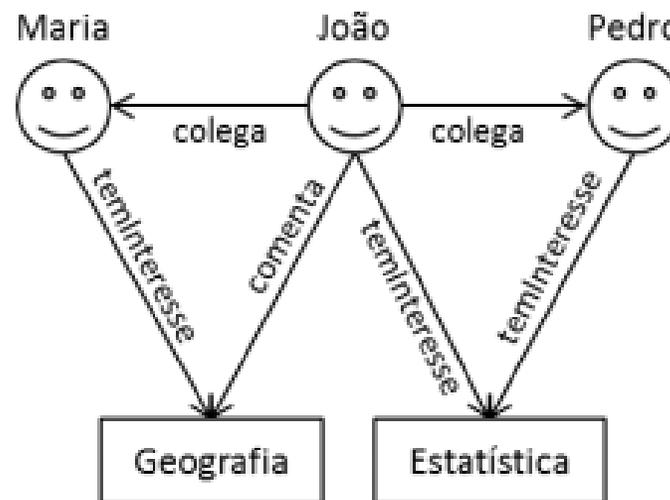
**Q22) [FGV IBGE 2017]** Observe a figura a seguir que ilustra relações entre colegas e seus interesses.



O tipo de Banco de Dados NoSQL, não relacional, que armazena tais informações, utilizando estruturas de vértices e arestas, com propriedades associadas, é o:

- a) Colunar;
- b) Documento;
- c) Grafo;
- d) Chave-valor;
- e) Tabular.

**Q22) [FGV IBGE 2017]** Observe a figura a seguir que ilustra relações entre colegas e seus interesses.



O tipo de Banco de Dados NoSQL, não relacional, que armazena tais informações, utilizando estruturas de vértices e arestas, com propriedades associadas, é o:

- a) Colunar;
- b) Documento;
- c) Grafo;**
- d) Chave-valor;
- e) Tabular.

**Q23) [CESPE TCU 2015]** Julgue o item subsecutivo, a respeito de sistemas de bancos de dados.

Como forma de permitir as buscas em documentos semiestruturados, um banco de dados NoSQL do tipo orientado a documentos armazena objetos indexados por chaves utilizando tabelas de hash distribuídas.

**Q24) [CESPE FUNPRESP 2016]** Com relação à forma como os dados são armazenados e manipulados no desenvolvimento de aplicações, julgue o item a seguir.

Em um banco de dados NoSQL do tipo grafo, cada arco é definido por um identificador único e expresso como um par chave/valor.

**Q23) [CESPE TCU 2015]** Julgue o item subsecutivo, a respeito de sistemas de bancos de dados.

**Como forma de permitir as buscas em documentos semiestruturados, um banco de dados NoSQL do tipo orientado a documentos armazena objetos indexados por chaves utilizando tabelas de hash distribuídas. ERRADO.**

**Q24) [CESPE FUNPRES 2016]** Com relação à forma como os dados são armazenados e manipulados no desenvolvimento de aplicações, julgue o item a seguir.

**Em um banco de dados NoSQL do tipo grafo, cada arco é definido por um identificador único e expresso como um par chave/valor. ERRADO.**

**Q25) [FCC SEFAZ-SC 2018]** Como não existe um modelo de armazenamento de dados único adequado para atender a todos os cenários e necessidades das aplicações de Big Data, em complemento ao modelo relacional, foram criados quatro novos tipos de armazenamento NoSQL (Not only SQL). Dentre esses, aqueles que modelam os dados usando o formato de linhas e colunas são APENAS os tipos de bancos de dados orientados a colunas,

- a) chave-valor e grafos.
- b) chave-valor e documentos.
- c) documentos e grafos.
- d) chave-valor.
- e) grafos.

**Q25) [FCC SEFAZ-SC 2018]** Como não existe um modelo de armazenamento de dados único adequado para atender a todos os cenários e necessidades das aplicações de Big Data, em complemento ao modelo relacional, foram criados quatro novos tipos de armazenamento NoSQL (Not only SQL). Dentre esses, aqueles que modelam os dados usando o formato de linhas e colunas são APENAS os tipos de bancos de dados orientados a colunas,

a) chave-valor e grafos.

**b) chave-valor e documentos.**

c) documentos e grafos.

d) chave-valor.

e) grafos.

**Q26) [IF SUL Rio-Grandense 2019]** Sobre o banco de dados MongoDB, é INCORRETO afirmar:

- a) MongoDB é um banco de dados orientado a documentos com estrutura semelhante a objetos do tipo JSON.
- b) As operações no MongoDB são feitas através da execução de funções.
- c) No MongoDB, os schemas são dinâmicos, sendo possível em um mesmo documento ter dados com diferentes campos.
- d) No MongoDB, pode ser utilizada a linguagem SQL para fazer consultas de dados.

**Q26) [IF SUL Rio-Grandense 2019]** Sobre o banco de dados MongoDB, é INCORRETO afirmar:

- a) MongoDB é um banco de dados orientado a documentos com estrutura semelhante a objetos do tipo JSON.
- b) As operações no MongoDB são feitas através da execução de funções.
- c) No MongoDB, os schemas são dinâmicos, sendo possível em um mesmo documento ter dados com diferentes campos.
- d) No MongoDB, pode ser utilizada a linguagem SQL para fazer consultas de dados.**

**Q27) [FCC MPE-PE 2018]** No gerenciador de bancos de dados MongoDB, os comandos para exibir a lista de bancos de dados presentes no servidor e para gerar um backup são, respectivamente:

- a) show dbs e mongodump.
- b) list dbs e mongodrop.
- c) copy dbs e mongouse.
- d) use dbs e mongoname.
- e) serve dbs e mongofind.

**Q27) [FCC MPE-PE 2018]** No gerenciador de bancos de dados MongoDB, os comandos para exibir a lista de bancos de dados presentes no servidor e para gerar um backup são, respectivamente:

**a) show dbs e mongodump.**

b) list dbs e mongodrop.

c) copy dbs e mongouse.

d) use dbs e mongoname.

e) serve dbs e mongofind.

**Q1 – LETRA E.  
Q2 - ERRADO.  
Q3 – ERRADO.  
Q4 - LETRA B.  
Q5 - ERRADO.  
Q6 - LETRA E.  
Q7 – LETRA D.  
Q8 – CERTO.  
Q9 - CERTO.  
Q10 - LETRA E.  
Q11 – LETRA D.  
Q12 - LETRA E.  
Q13 - ERRADO.  
Q27 - LETRA A.**

# **GABARITO**

**Q14 – ERRADO.  
Q15 – LETRA C.  
Q16 - LETRA B.  
Q17 – LETRA A.  
Q18 – CERTO.  
Q19 - ERRADO.  
Q20 - ERRADO.  
Q21 - LETRA A.  
Q22 - LETRA C.  
Q23 - ERRADO.  
Q24 - ERRADO.  
Q25 - LETRA B.  
Q26 - LETRA D.**