FullCycle

Bancos de Dados

FullCycle

Bancos de Dados
Módulo 2

FullCycle

Aula 1
Quem é o seu professor

Lourenço Taborda

Arquiteto de Solução

https://linktr.ee/devtbd

FullCycle

Aula 50
Transações Distribuídas

Transações Distribuídas

- Bancos de dados distribuídos normalmente **dividem suas tabelas (ou coleções)** em **partições (ou nós ou shards) espalhadas** por **diferentes servidores** que são **acessados por muitos clientes**.

- **Transações** dos clientes geralmente **abrangem diferentes servidores**, pois as transações demanda leituras em várias partições.

- Uma **transação distribuída** é uma **transação de banco de dados** que **abrange vários servidores**.

"MIT 6.824: Lecture 12 - Distributed Transactions". Timilearning. Disponível em <https://timilearning.com/posts/mit-6.824/lecture-12-distributed-transactions/>. Acessado em 25 de Fevereiro de 2024 às 22:30.

Transações Distribuídas

- Ocorrem em sistemas distribuídos, onde **múltiplos nós ou servidores** participam da execução de uma transação.

- Podem abranger **vários recursos distribuídos** e, portanto, exigem **coordenação entre os nós** para garantir a **consistência** dos dados.

- **Protocolos de coordenação** como 2PC, 3PC, Paxos e Raft são usados para facilitar essa coordenação e garantir a integridade dos dados.

FullCycle

Transações Distribuídas

Coordenação

Várias operações de transações são executadas **em diferentes nós** ou servidores.

Um nó é designado como **coordenador da transação** para garantir a **consistência dos dados** e a **atomicidade das operações**.

Participantes

Nós ou servidores **envolvidos na transação**.

Durante a transação distribuída, os participantes executam operações de **leitura**, **escrita** ou **atualização** em seus respectivos recursos de dados.

Isolamento e Consistência

Operações de uma transação devem ser **invisíveis** para outras transações até que sejam concluídas com êxito (**isolamento**).

Operações devem manter a **integridade** e a **consistência** dos dados em todo o sistema (**consistência**).

Atomicidade

Todas as operações de uma transação sejam confirmadas ou revertidas de forma atômica em **todos os nós** participantes.

Ou **todas as operações são bem-sucedidas** e confirmadas, ou nenhuma delas é confirmada e **todas as alterações são revertidas.**

Protocolos de Coordenação

Garantem que **todas as operações de uma transação** sejam coordenadas e executadas de forma **consistente** e **segura** em um ambiente **distribuído**.

Protocolos comuns incluem o Two-Phase Commit (**2PC**), Three-Phase Commit (**3PC**), **Paxos** e **Raft**.

FullCycle

Aula 51
Transações Distribuídas: Confirmação Atômica (Atomic Commit)

Transações Distribuídas: Confirmação Atômica (Atomic Commit)

- Uma transação é **confirmada** ou **revertida**.
- Se é confirmada então as atualizações são **duráveis**.
- Se é revertida então não há efeitos visíveis.
- **Consistência ACID** (preservar estado válido) depende de **atomicidade**.

Se a transação atualiza **dados em múltiplos nós** então:

- **Todos** os nós **confirmam** ou todos **revertem** a transação.
- Se um nó falhar então todos os nós revertem a transação.

Portanto, este é um problema de **Confirmação Atômica** (Atomic Commit).

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Transações Distribuídas: Confirmação Atômica (Atomic Commit)

Confirmação Atômica	Consenso
Todos os nós votam para confirmar ou reverter.	Um ou mais nós **propõe um valor**.
Confirmação **somente se todos** os nós votam para confirmar.	**Qualquer valor** proposto pode ser escolhido.
Se **1 nó participante falha** então a transação é revertida.	**Nós com falha são tolerados** enquanto há quorum.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 52
Protocolos de Coordenação: Two-Phase Commit (2PC)

Protocolos de Coordenação: Two-Phase Commit (2PC)

- A confirmação em 2 fases busca garantir que **todas as operações de uma transação** sejam **confirmadas ou revertidas** de forma **atômica** em **todos os nós** participantes.

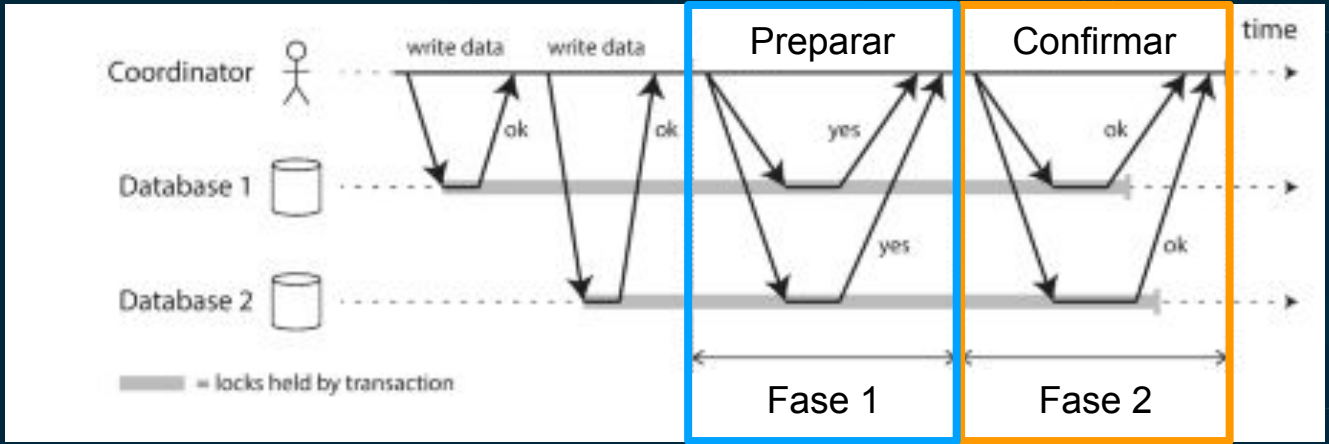
- Fase 1 de **preparação**: os nós **concordam em executar** a transação e preparam-se para confirmá-la.

- Fase 2 de **confirmação**: os nós **confirmam ou revertem** a transação com base na preparação.

"MIT 6.824: Lecture 12 - Distributed Transactions". Timilearning. Disponível em <https://timilearning.com/posts/mit-6.824/lecture-12-distributed-transactions/>. Acessado em 25 de Fevereiro de 2024 às 22:30.

FullCycle

Protocolos de Coordenação: Two-Phase Commit (2PC)



"MIT 6.824: Lecture 12 - Distributed Transactions". Timilearning. Disponível em <https://timilearning.com/posts/mit-6.824/lecture-12-distributed-transactions/>. Acessado em 25 de Fevereiro de 2024 às 22:30.

FullCycle

Protocolos de Coordenação: Two-Phase Commit (2PC)

Fase 1

Um dos nós é designado **coordenador da transação** e é responsável por **iniciar** e **coordenar** o protocolo.

Cada participante, inclusive coordenador, pode ser um **nó do banco de dados** que mantém os **recursos envolvidos na transação**.

Coordenador inicia a transação enviando uma **mensagem de preparação** (PREPARE) para todos os participantes.

Esta mensagem solicita que cada participante **prepare-se para executar a transação**, mas ainda não a execute.

Cada participante responde ao coordenador se está **pronto** (READY) **ou não** (ABORT) para realizar a transação **com sucesso**.

Se um participante encontrar um problema que impeça a execução da transação, ele responde com ABORT.

FullCycle

Protocolos de Coordenação: Two-Phase Commit (2PC)

Fase 2

Decisão do Coordenador → **Confirmação ou Reversão** → **Execução**

Após receber respostas de todos os participantes, o **coordenador decide se deve confirmar** a transação.

Se todos os participantes estiverem prontos para confirmar a transação, o **coordenador decide** pela confirmação (COMMIT).

Coordenador envia uma mensagem de confirmação (COMMIT) ou reversão (ABORT) para todos os participantes, dependendo da decisão tomada na etapa anterior.

Ao receber a mensagem de COMMIT, os **participantes executam a transação permanentemente e confirmam ao coordenador** que a transação foi concluída.

Ao receber a mensagem de ABORT, os **participantes revertem alterações temporárias e confirmam ao coordenador** que a transação foi revertida.

Two-Phase Commit é um protocolo síncrono, o que garante atomicidade da transação, porém, fica **bloqueado até que uma decisão seja alcançada**.

Isso **pode levar a problemas de desempenho e escalabilidade** em sistemas distribuídos com muitos participantes ou falhas de rede.

FullCycle

Aula 53
Protocolos de Coordenação:
Fault Tolerant Two-Phase Commit (2PC)

Protocolos de Coordenação: Fault Tolerant Two-Phase Commit (2PC)

- Confirmação em 2 Fases Tolerante à Falha.
- Fornece maior robustez e **resiliência à falha**.
- Garante ao mesmo tempo a **atomicidade das transações**.
- Vulnerabilidade do 2PC é que o coordenador é um **ponto único de falha**.
- Durante a falha do coordenador logo após a preparação não há a confirmação e, logo, **não há progresso da transação nos nós**.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Protocolos de Coordenação: Fault Tolerant Two-Phase Commit (2PC)

Detecção de Falhas

Mecanismos para **detectar** e lidar com falhas de nós participantes e do coordenador.

Timeouts para detectar falhas de comunicação.

Coordenação Assíncrona

Participantes podem continuar com as operações de transação, mesmo na **ausência temporária do coordenador**.

Coordenação retomada posteriormente quando o coordenador se recuperar.

Eleição e Failover

Seleção de um novo coordenador em caso de falha do nó coordenador atualmente ativo.

Implementação de **algoritmos de eleição distribuída**, como Paxos.

Recuperação de Falhas

Lógica robusta para lidar com diferentes falhas, como coordenador falho na **preparação** ou participante falho durante **confirmação**.

Retransmissão de mensagens, **revisão** do estado da transação e execução de procedimentos de **recuperação**.

Resiliência a Partições de Rede

Estratégias de **reconfiguração** e **ressincronização** para garantir a consistência das transações após a resolução da partição de rede.

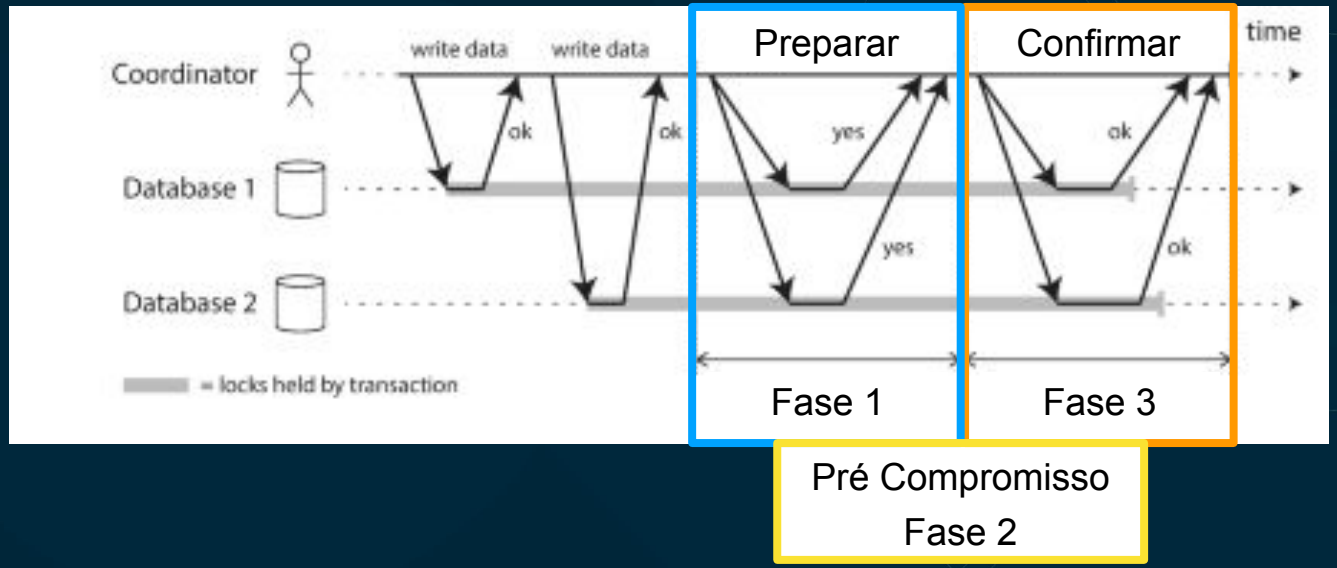
Nós podem perder a conectividade com o coordenador ou entre si.

FullCycle

Aula 54
Protocolos de Coordenação: Three-Phase Commit (3PC)

FullCycle

Protocolos de Coordenação: Three-Phase Commit (3PC)



"MIT 6.824: Lecture 12 - Distributed Transactions". Timilearning. Disponível em <https://timilearning.com/posts/mit-6.824/lecture-12-distributed-transactions/>. Acessado em 25 de Fevereiro de 2024 às 22:30.

FullCycle

Protocolos de Coordenação: Three-Phase Commit (3PC)

Extensão do protocolo Two-Phase Commit (2PC) para mitigar bloqueio indefinido

Preparação

Coordenador envia uma mensagem de preparação para todos os participantes.

Participantes respondem indicando sua prontidão para a transação.

Avança somente se todos confirmam.

Pré Compromisso

Coordenador envia uma mensagem de pré compromisso para todos os participantes, indicando sua **intenção de confirmar a transação**.

Participantes podem confirmar ou negar o pré compromisso com base em sua prontidão para confirmar a transação.

Confirmação

Se **todos confirmaram o pré compromisso** então coordenador envia uma mensagem de COMMIT para todos os participantes.

Participantes executam e confirmam a transação.

Caso algum participante não confirme transação, há a reversão e envio de mensagem de ROLLBACK para os demais participantes.

Three-Phase Commit frequentemente considerado irrealista em ambientes distribuídos devido à sua **dependência de sincronia perfeita** entre os nós participantes.

Ainda pode entrar em um estado de **bloqueio indefinido** em caso de falha do coordenador.

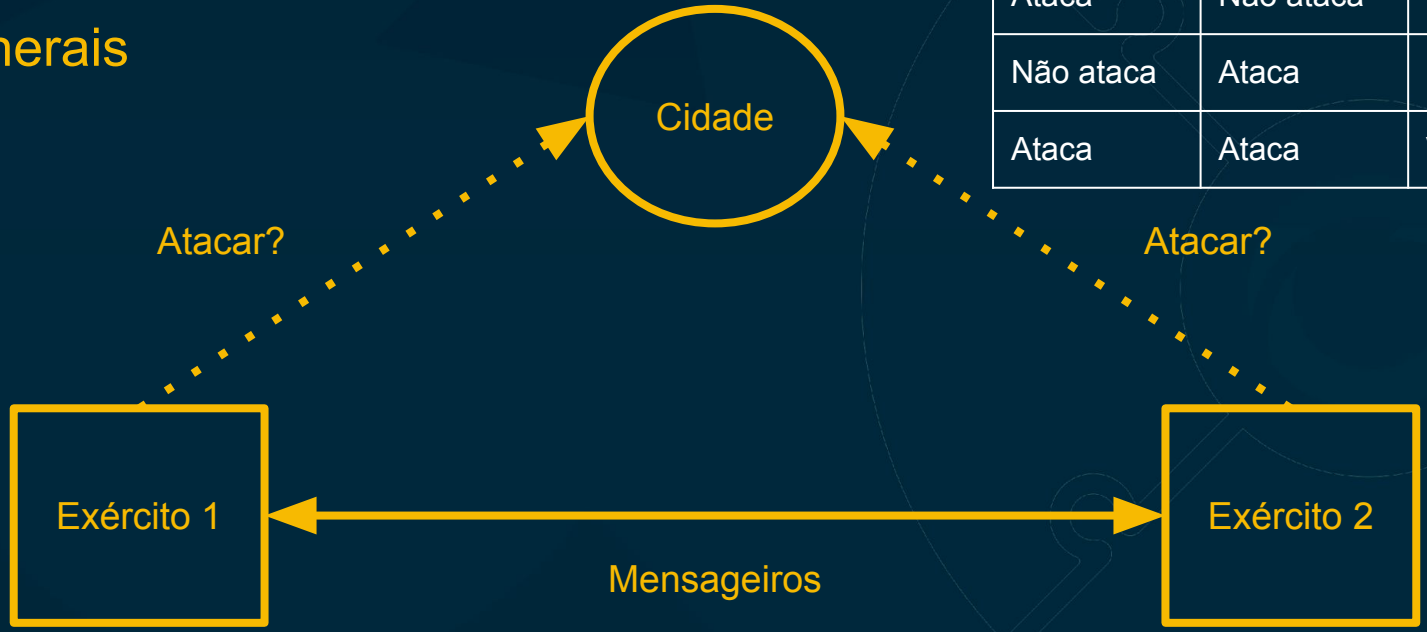
FullCycle

Aula 55
Modelos de Sistemas Distribuídos: Problema dos 2 Generais

Modelos de Sistemas Distribuídos

Problema dos 2 Generais

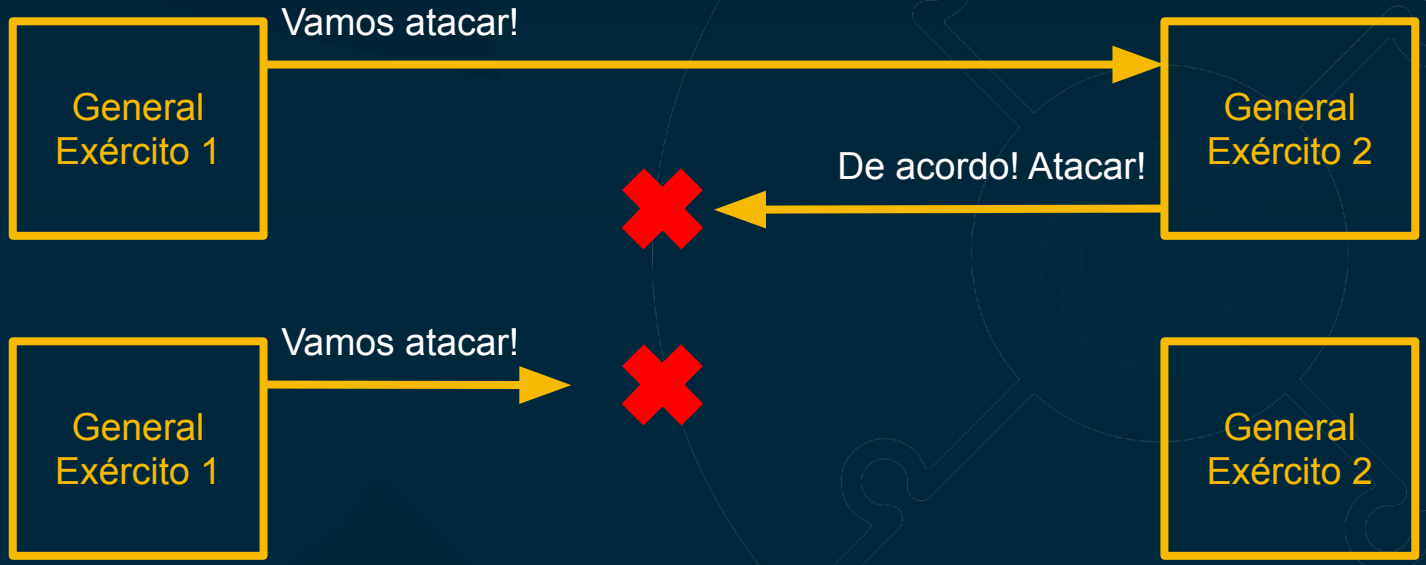
Exército 1	Exército 2	Resultado
Não ataca	Não ataca	Nada
Ataca	Não ataca	Derrota
Não ataca	Ataca	Derrota
Ataca	Ataca	**Vitória**



Desejado: Exército 1 ataca se e somente se Exército 2 ataca.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Problema dos 2 Generais



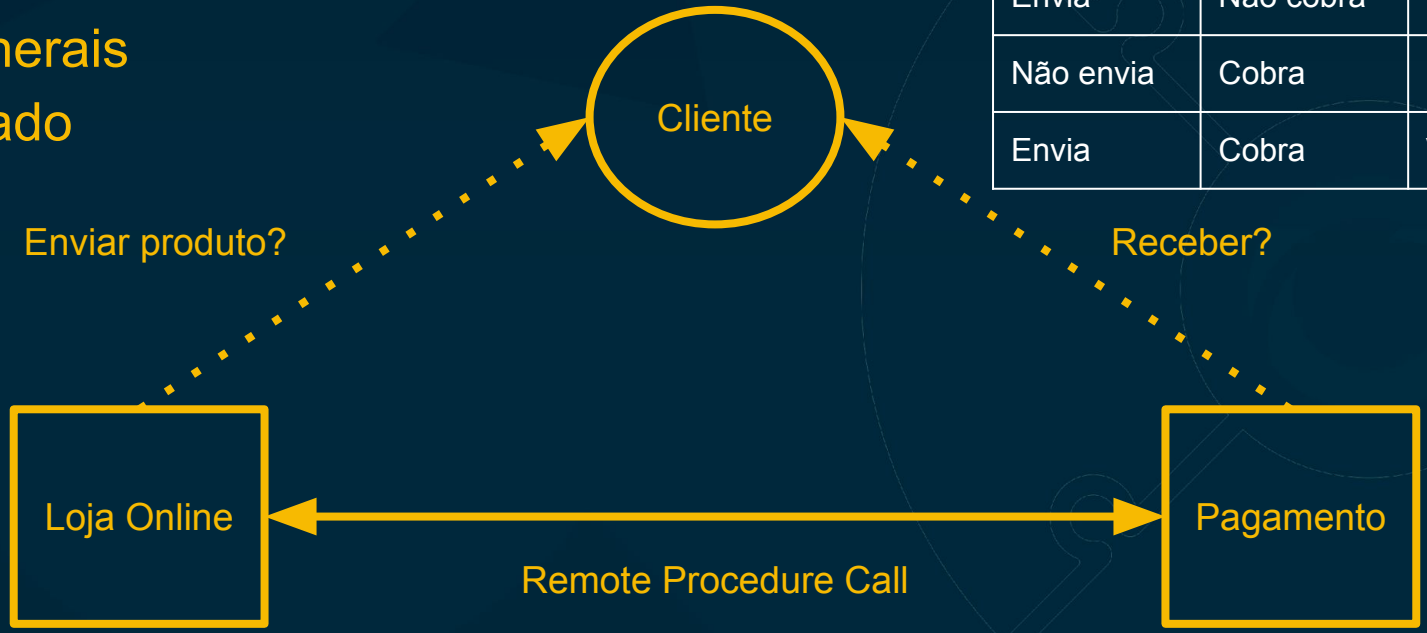
Para o General 1 estas situações são indistinguíveis!
Para decidir é necessário comunicar!

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Modelos de Sistemas Distribuídos

Problema dos 2 Generais Aplicado

Loja	Pagamento	Resultado
Não envia	Não cobra	Nada
Envia	Não cobra	Derrota
Não envia	Cobra	Derrota
Envia	Cobra	**Vitória**



Desejado: Loja Online entrega se e somente se Pagamento é recebido

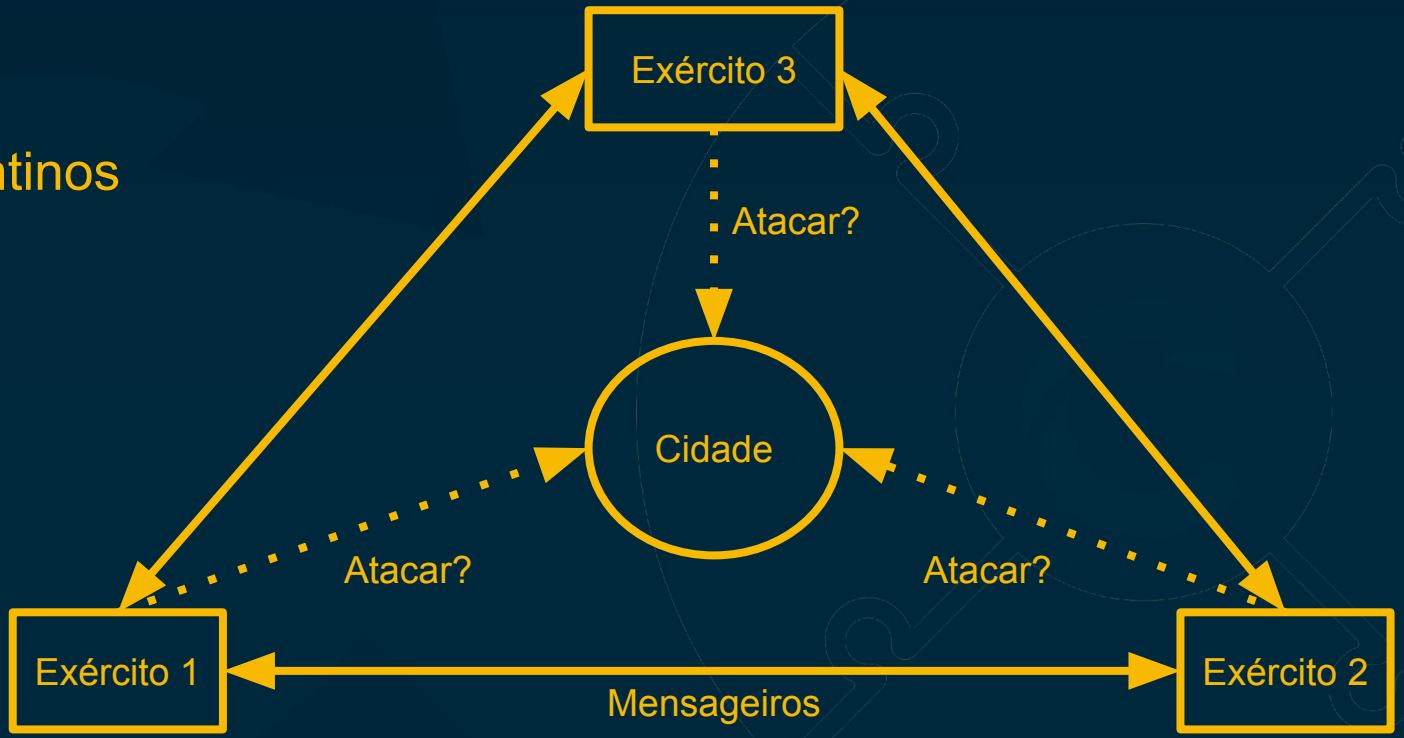
"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 56
Modelos de Sistemas Distribuídos: Problema dos Generais Bizantinos

Modelos de Sistemas Distribuídos

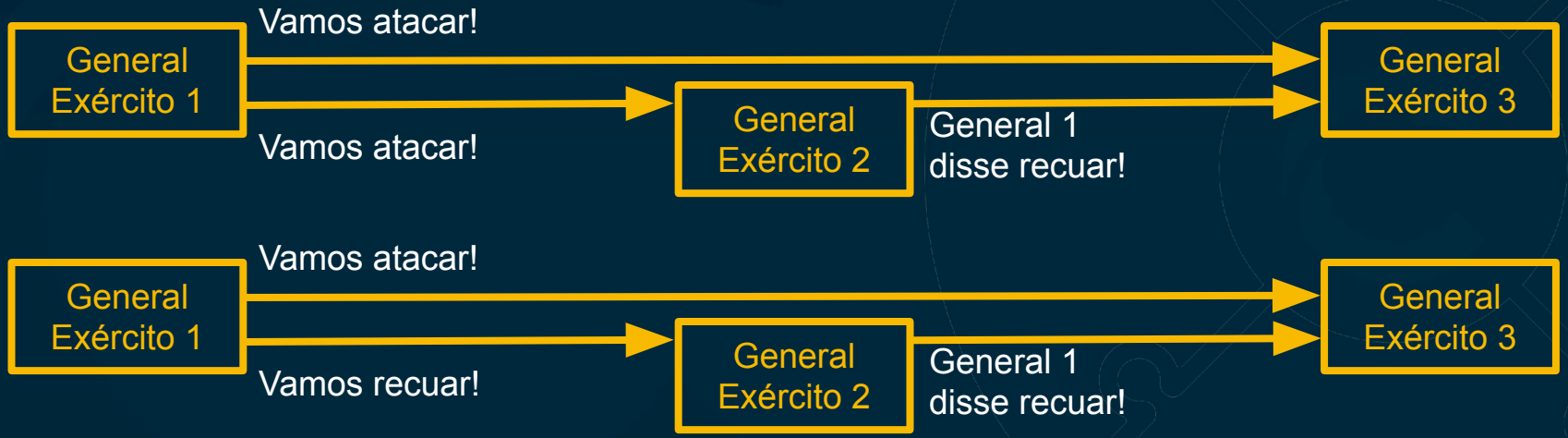
Problema dos Generais Bizantinos



Problema: alguns generais podem ser traidores!

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Problema dos Generais Bizantinos

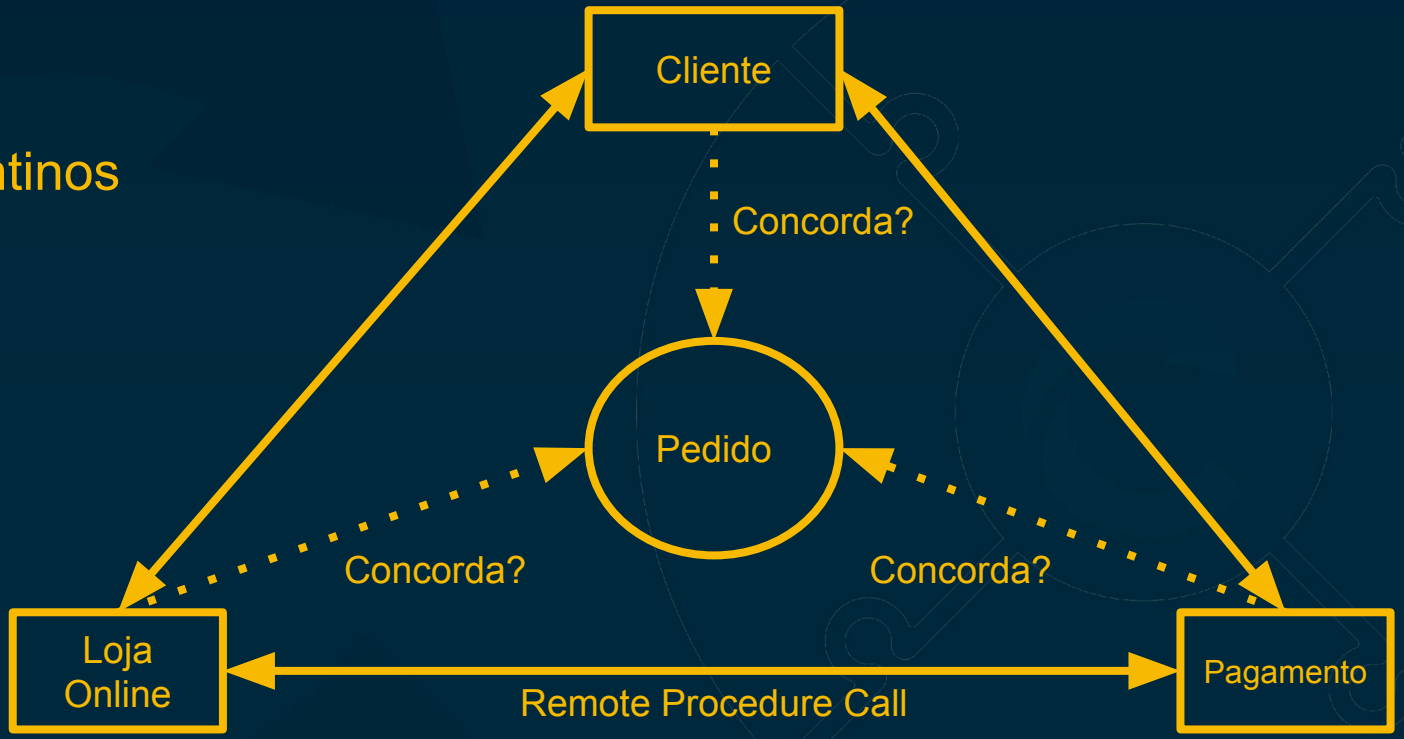


Para o General 3 estas situações são indistinguíveis!
Generais honestos precisam concordar com o plano de ação.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Modelos de Sistemas Distribuídos

Problema dos Generais Bizantinos Aplicado



Em quem podemos confiar?

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Problema dos Generais Bizantinos

Cada general pode ser honesto ou malicioso.

Teorema: $3f + 1$ generais honestos para tolerar f generais maliciosos.

Generais honestos não sabem quem é malicioso.

Generais maliciosos podem conspirar juntos.

Criptografia com assinaturas digitais ajuda.

Portanto, generais honestos precisam estar de acordo com o plano de ação.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 57
Modelos de Sistemas Distribuídos: Conclusão

Modelos de Sistemas Distribuídos

Problema dos 2 Generais	=	**Modelo de Redes**
Problema dos Generais Bizantinos	=	**Modelo de Comportamento do Nó**

Em sistemas reais ambos nós e redes podem falhar.

Modelos de Sistemas Distribuídos Modelo consistem em premissas para:

REDE
Perda de Mensagens

NÓ
Falha

SINCRONIA
Latência

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Modelos de Sistemas Distribuídos

REDE	NÓ	SINCRONIA
Links confiáveis	Se falha então parada	Síncrono
Links com alguma perda	Se falha então recuperação	Parcialmente síncrono
Links arbitrários adversos	Se viola algoritmo então falho (Bizantino)	Assíncrono

Base para algoritmos distribuídos:
Para cada 1 dessas 3 partes escolha 1 desses comportamentos.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 58
Replicação

Fundamentos para Replicação: Relógios e Tempo nos Sistemas Distribuídos

Medir o tempo é necessários nos sistemas distribuídos:

- Agendadores (schedulers),
- Temporizadores (timeouts & retries),
- Detectores de falha,
- Medição de desempenho,
- Rastreabilidade (log),
- Dados com validade temporal (cache)
- **Determinar ordem de ocorrência de eventos em múltiplos nós.**

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Tipos de relógio

- Relógio **Físico**: contagem do número de segundos decorridos.

  não é o mesmo que

- Relógio **Lógico**: contagem de eventos/mensagens enviados.
  - Relógio Lógico **Lamport** → **ordem causal**
  - Relógio Lógico **Vetorial** → **detecta eventos concorrentes**

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Fundamentos para Replicação: Protocolos de Transmissão

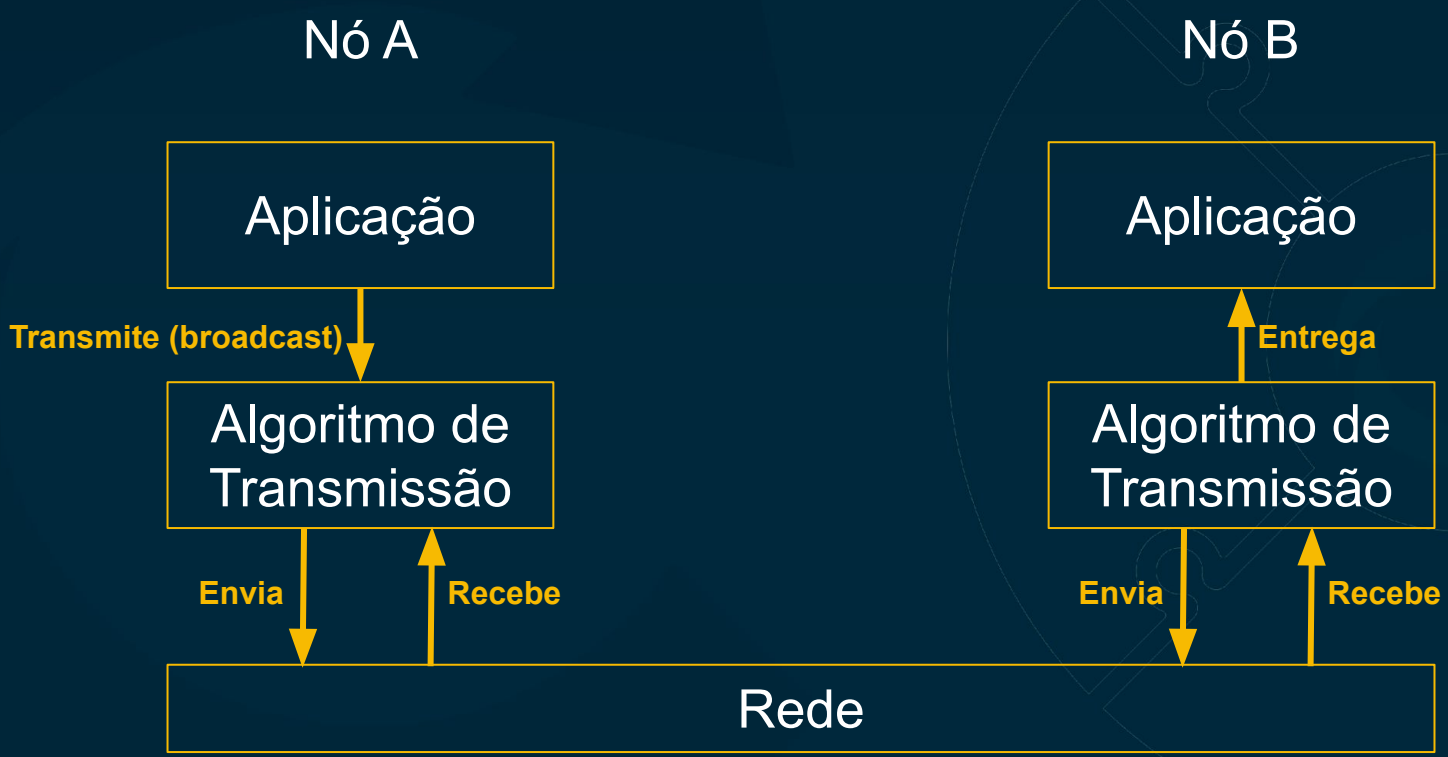
Transmissão ou multitransmissão é uma **comunicação em grupo**:

- Um nó envia mensagens e todos os nós entregam estas mensagens;
- Membros do grupo podem ser estáticos ou dinâmicos;
- Pode ser:
  - **Melhor esforço → perde algumas mensagens;**
  - **Confiável → todas as mensagens são transmitidas.**

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Fundamentos para Replicação: Protocolos de Transmissão



"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Formas de **transmissão confiável** (reliable broadcast):

- **FIFO broadcast**
  Se m1 e m2 são transmitidas pelo mesmo nó e a transmissão de m1 precede a transmissão de m2 então m1 tem que ser entregue antes de m2.

- **Causal broadcast**
  Se a transmissão de m1 precede a transmissão de m2 então m1 tem que ser entregue antes de m2.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Formas de **transmissão confiável** (reliable broadcast):

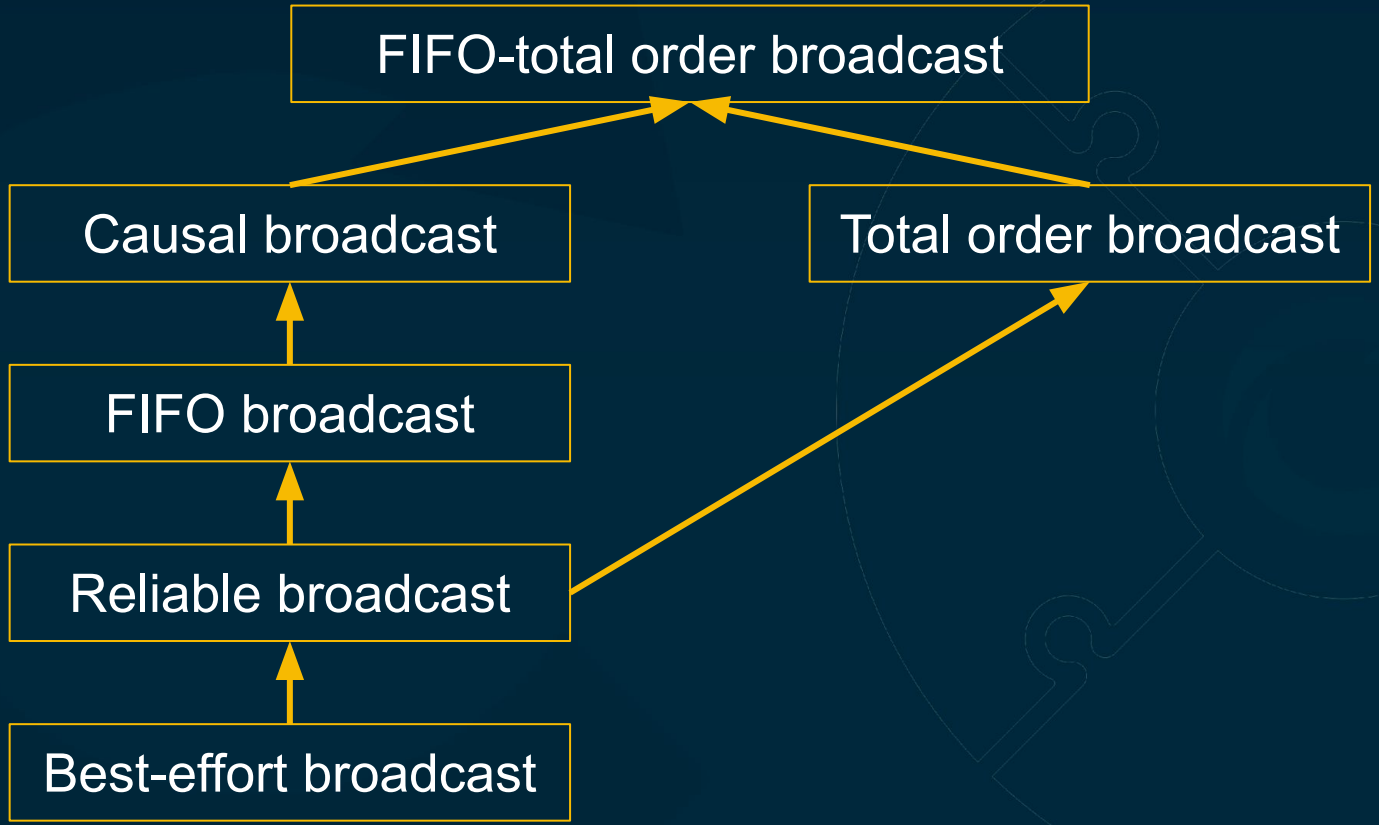
- **Total order broadcast**
  Se m1 é entregue antes de m2 em um nó então m1 tem que ser entregue antes de m2 em todos os nós.

- **FIFO-total order broadcast**
  Combinação de FIFO broadcast + total order broadcast.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Fundamentos para Replicação: Protocolos de Transmissão



"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Replicação

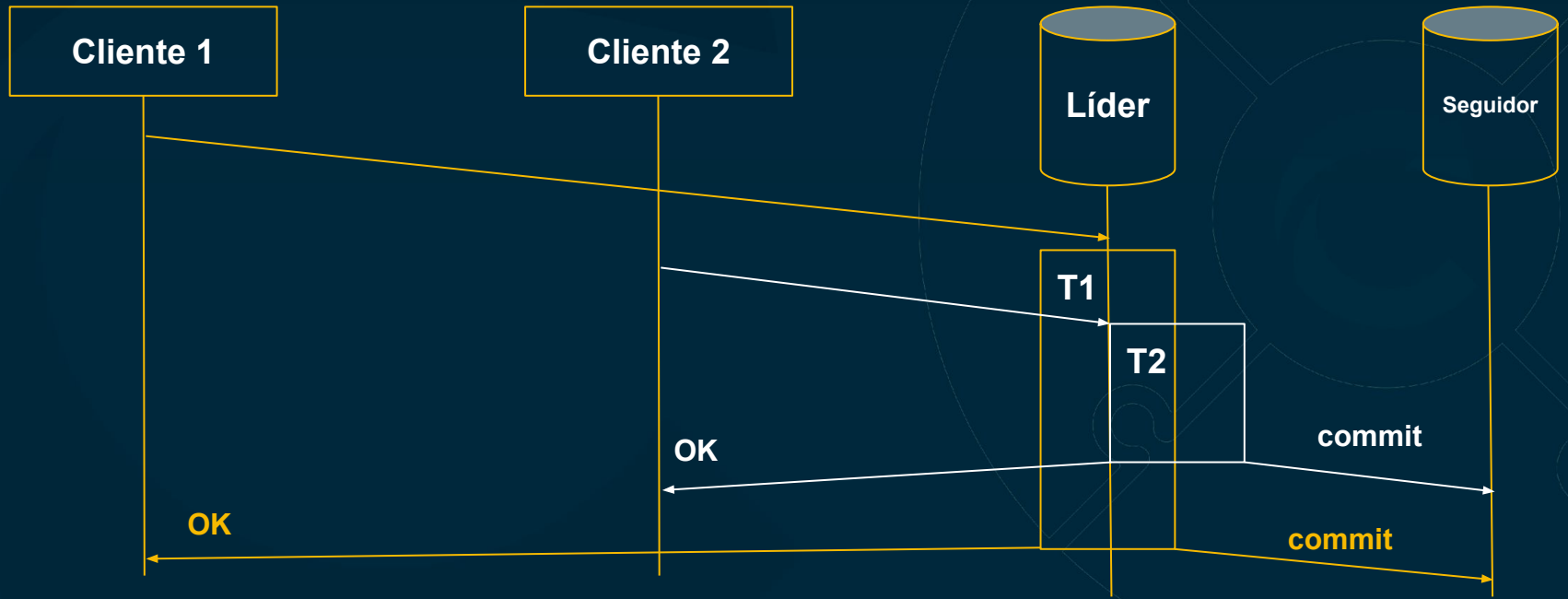
Replicação é manter uma cópia dos mesmos dados em múltiplos nós.

- Um nó com uma cópia dos dados é uma réplica.
- Se algumas réplicas estão indisponíveis, outras estarão acessíveis.
- Replicação permite a distribuição de carga sobre muitas réplicas.
- Replicação é fácil com dados estáticos: é apenas uma cópia.
- Requer foco em mudança dos dados.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Replicação

Replicação baseada em **total order broadcast** é amplamente utilizada.



"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 59
Quorum

Quorum (1/9)

A **replicação** é útil porque **melhora a confiabilidade** de um sistema. **Com tolerância à falha a confiabilidade aumenta**.

Enquanto **uma réplica estiver indisponível**, as demais réplicas restantes continuam o processamento das solicitações.

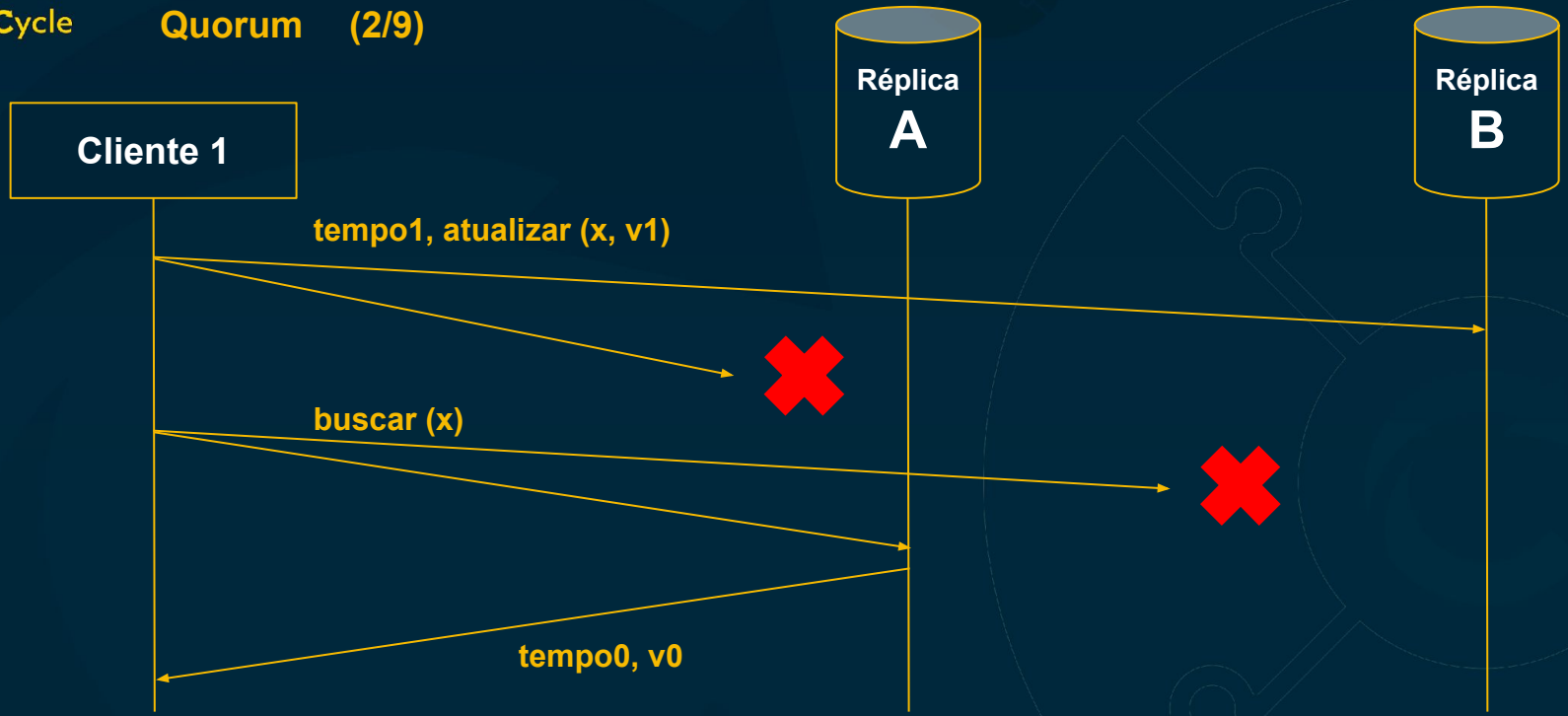
Indisponibilidade pode ocorrer devido à:

1. **falha no nó** (travamento, falha de hardware etc);
2. **partição de rede** (incapacidade de comunicação com outros nós);
3. **manutenção planejada** (reiniciar um nó com atualizações).

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Quorum (2/9)



Escrever na Réplica B e ler da Réplica A = Cliente **não recebe o último valor.**

Exigir escritas e leituras em ambas as réplicas **não é tolerante à falha**.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Quorum (3/9)

Imagine…

1) postar na sua rede social favorita;
2) atualizar o feed;
3) não ver o seu post recém postado!

Este comportamento é confuso para os usuários.

Por esta razão, sistemas exigem **consistência "ler após escrita"**.

→ read-after-write consistency

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Quorum (4/9)

Consistência "ler após escrita" pode não devolver para o cliente original o valor escrito por ele mesmo pois outro cliente concorrente pode ter atualizado o valor.

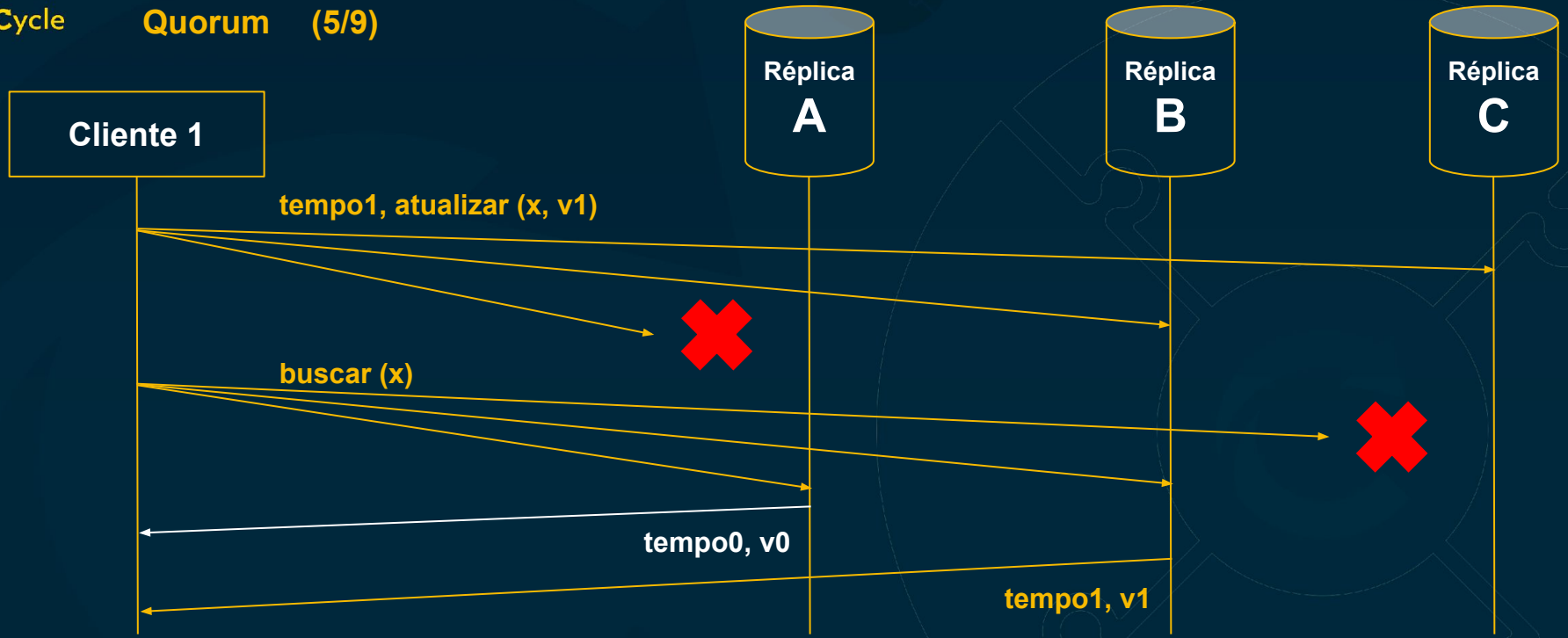
Consistência "ler após escrita" exige ler o último valor escrito ou o valor mais recente.

Com 3 réplicas o problema da tolerância à falha é resolvido.
Todas as requisições de leitura escrita são enviadas às **3 réplicas**.
A requisição é um sucesso com **2 ou mais** respostas recebidas.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Quorum (5/9)



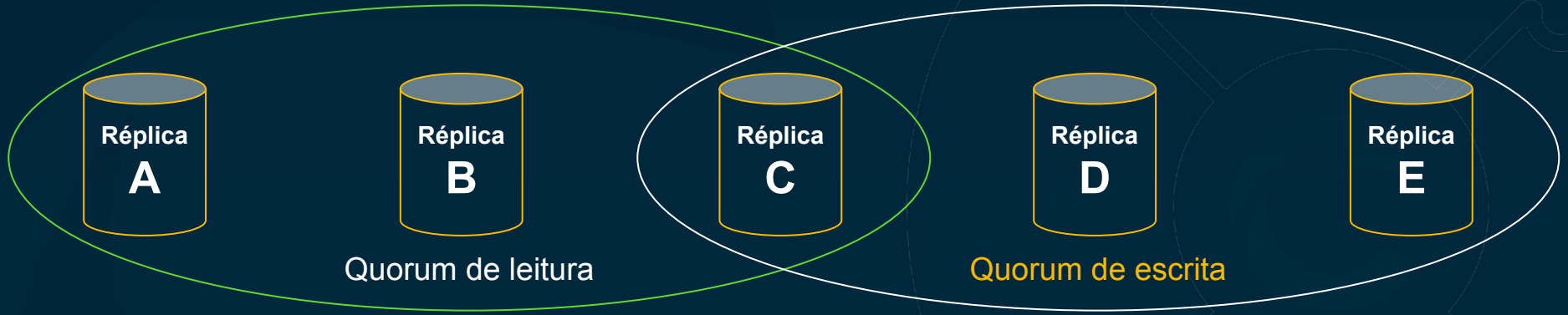
Escrita com sucesso em B e C.

Leitura com sucesso em A e B.

A escolha entre (t0, v0) e (t1, v1) é baseada no tempo.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Quorum de maioria é a solução comum em sistemas distribuídos.



Conceito: quorum é o **número mínimo de membros cuja presença é imprescindível** para dar validade às deliberações e votos de um determinado órgão colegiado.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Quorum (7/9)

Em um sistema com ***n*** réplicas:

1) Se uma escrita é reconhecida por ***w*** réplicas (quorum de escrita);
2) e subsequentemente é lida a partir de ***r*** réplicas (quorum de leitura);
3) e ***r + w > n***

Então, a leitura verá o valor escrito ou o valor mais recente.

Quoruns de leitura e escrita **compartilham 1 ou mais réplicas**.

Maioria → ***r = w = (n+1) / 2*** para ***n = 3, 5, 7***
Leituras toleram ***n-r*** réplicas indisponíveis
Escritas toleram ***n-w*** réplicas indisponíveis

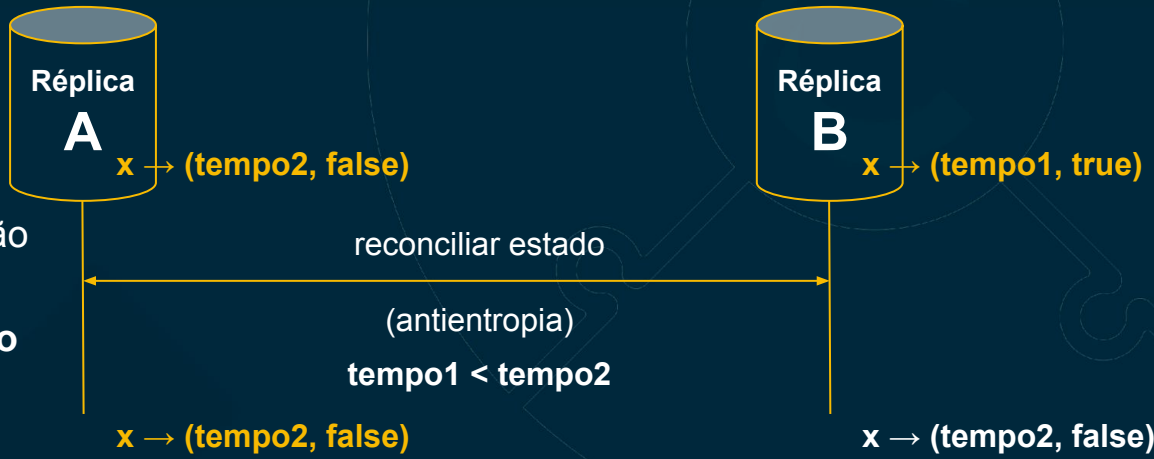
"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Quorum (8/9)

Nesta abordagem de quorum para replicação, **algumas atualizações** podem estar **pendentes para algumas réplicas** em **algum dado momento**.

Uma solução para **ressincronizar réplicas** novamente entre si é aplicar um **processo de antientropia**.

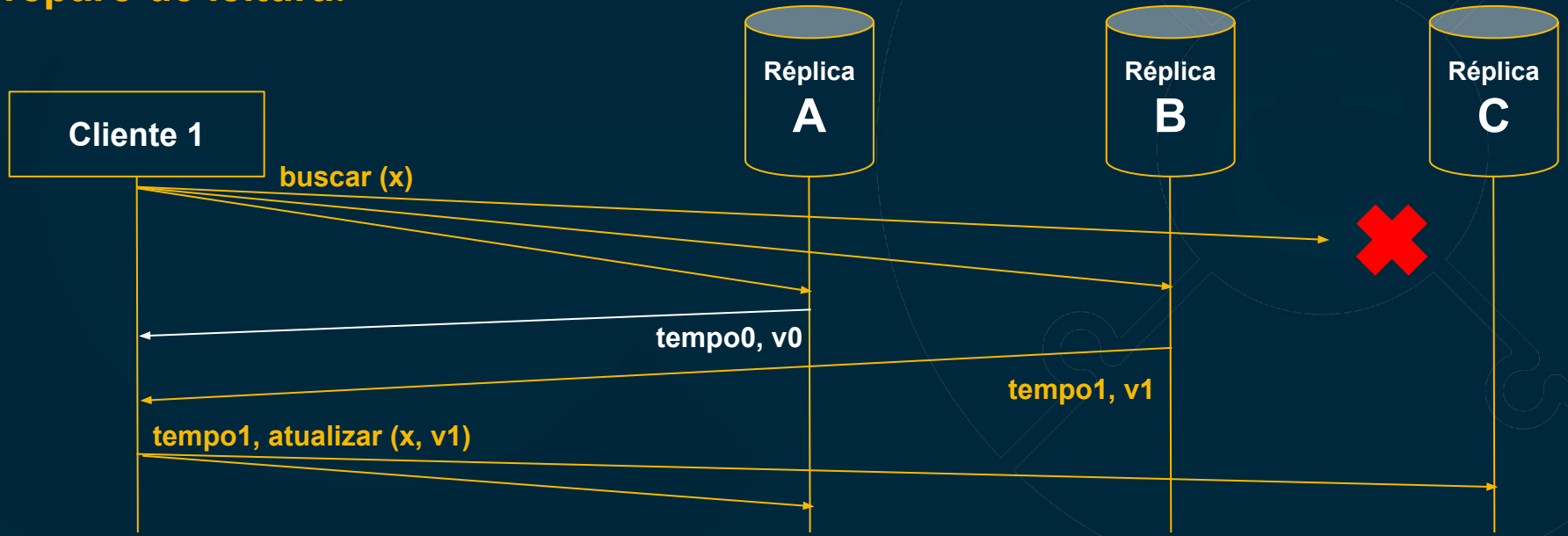
- Réplicas **periodicamente** comunicam-se entre si para **verificar inconsistências**.
- Registros com o **último tempo** são **propagados**.
- Registros com **tempos mais cedo** são **descartados**.



"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Quorum (9/9)

A solução alternativa para **ressincronizar réplicas** é utilizar os clientes para apoiar no processo de propagar as atualizações dos dados. Este processo é o **reparo de leitura**.



"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 60
Consenso

Consenso

Replicação baseada em **total order broadcast** é amplamente utilizada e é feita com um **nó líder que roteia as mensagens**. Este nó líder distribui as mensagens via **FIFO broadcast** (First In First Out). Assim, **todos os nós entregam a mesma sequência de mensagens na mesma ordem**.

Problema: nó líder é um ponto único de falha.

Soluções:

1) **Intervenção humana** - operador humano seleciona novo líder e reconfigura cada nó seguidor para seguir o novo líder.
2) **Algoritmo de Consenso** - transferência automática de liderança de um nó para outro.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Consenso

Consenso: é um **acordo** que **muitos nós alcançam** sobre um **valor**.

Este **valor** é a **próxima mensagem** a ser entregue (Total Order Broadcast).

Assim que um nó decide sobre uma certa ordem da mensagem, os demais nós disponíveis decidirão a mesma ordem.

Consenso e Total Order Broadcast são formalmente equivalentes.

Um ou mais **nós propõem um valor** e o algoritmo de **consenso decide** entre os valores propostos.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Algoritmos de Consenso

Paxos

1998

Garante que **um conjunto de nós** em um sistema distribuído **concorde sobre um único valor**, mesmo em face de **falhas** e **particionamentos** de rede.

Multi Paxos

Extensão do algoritmo Paxos que **simplifica** sua implementação e **melhora** seu desempenho.

Introduz um **líder único** responsável por **coordenar** as operações de **consenso**.

Raft

2014

Divide o problema de consenso em **termos de líderes e seguidores**.

O líder **coordena** o **consenso**.

Seguidores **replicam** o estado do líder.

Viewstamped Replication

VSR

Baseado em **estados de réplica**, onde as réplicas mantêm um registro do estado de replicação atual.

Também usa um **líder** para coordenar as operações de replicação.

ZooKeeper Atomic Broadcast

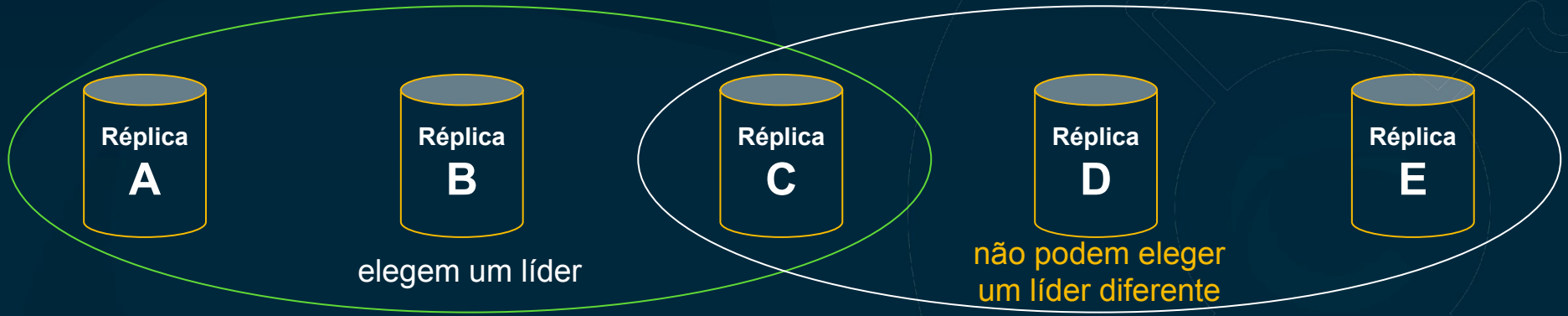
Zab

Semelhante ao Paxos, mas com algumas otimizações e ajustes para as necessidades do **ZooKeeper**.

Todas as réplicas processam as operações **na mesma ordem**.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

Eleição do nó líder



Multi Paxos, Raft, VSR e Zab usam líderes para sequenciar mensagens.

Detector de falha no líder é timeout.

Garante **1 único líder por termo** (split-brain).

Precisa de quorum para eleger um líder no termo.

"Concurrent and Distributed Systems". University of Cambridge. Disponível em <https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/materials.html>. Acessado em 03 de Março de 2024 às 13:30.

FullCycle

Aula 61
Raft

Algoritmo de Consenso Raft

https://thesecretlivesofdata.com/raft/

FullCycle

Aula 62
Fundamentos de Indexação

Fundamentos de Indexação

Principais **gargalos** em banco de dados:

1) falta de índices e 2) índices mal projetados.

Encontrar **maneiras eficientes de recuperar e manipular dados** torna-se cada vez mais importante à medida que os **bancos de dados crescem em tamanho**.

Uma **estratégia de indexação bem concebida é fundamental** para alcançar esta eficiência.

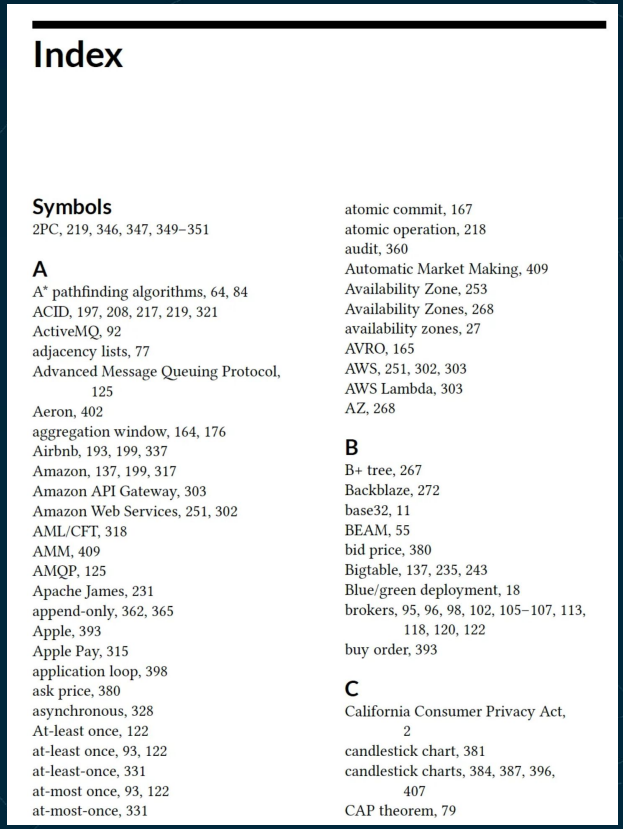
"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Fundamentos de Indexação

Índice é uma estrutura de dados que acelera a recuperação de dados.

O **índice de banco de dados** acelera a recuperação de dados sem a necessidade de verificar cada registro (seja este uma linha de uma tabela ou um documento em uma coleção).

Semelhante ao índice de livro, o qual lista **palavras-chave junto aos números de páginas** para a localizar informações rapidamente.



Index

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Fundamentos de Indexação

Índice de banco de dados:

- composto por **uma lista ordenada de valores**, com **cada valor conectado a ponteiros** que levam às **páginas de dados** onde esses valores estão persistidos.

- constituído pelas **chaves feitas a partir de 1+ colunas da tabela ou campos do documento**.

- é armazenado em uma estrutura de **Árvore B ou B+**, normalmente em **disco**, para a maioria dos bancos de dados relacionais e não-relacionais.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Fundamentos de Indexação

Índice de banco de dados:

- outras estruturas de dados, como **bitmap** e **hashmap**, podem ser usadas para implementar índices.
- embora todas essas estruturas ofereçam acesso eficiente aos dados, os detalhes de sua **implementação são diferentes**.
- para bancos de dados **relacionais**, os índices são frequentemente implementados usando uma **árvore B+**, que é uma variante da árvore B.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Fundamentos de Indexação

Índice de banco de dados:

Determinar índices certos é um equilíbrio entre

- **respostas rápidas às consultas** e
- **custos de atualização**.

Índices estreitos:

- com **menos atributos**;
- **economizam espaço** em disco;
- cobrem **consultas específicas**.

Índices amplos:

- com **mais atributos**;
- ocupam **mais espaço** em disco;
- atendem a uma gama mais ampla de consultas.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

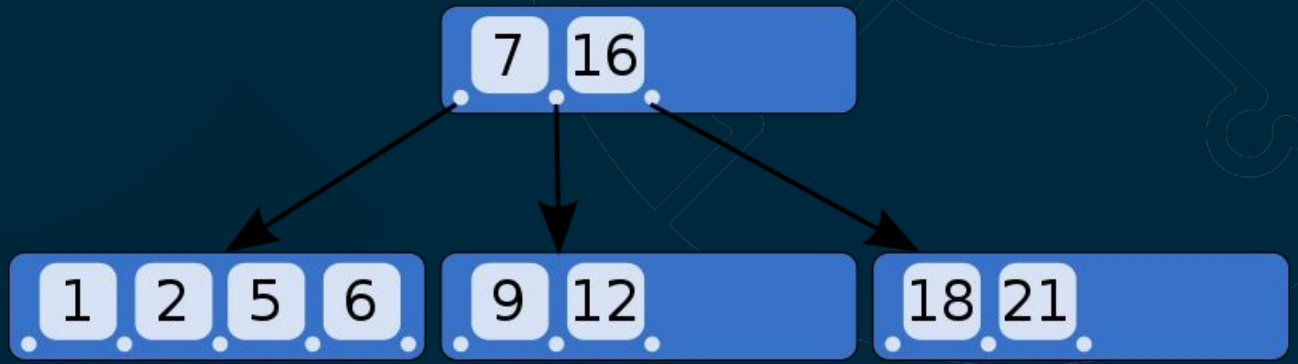
FullCycle

Aula 63
Árvores B e B+

Árvore B:

- inventado por Rudolf **Bayer** e Edward M. **McCreight** enquanto trabalhavam no **Boeing** Research Labs, com o propósito de gerenciar com eficiência páginas de índice para **grandes arquivos de acesso aleatório**.

- Adequada para sistemas de armazenamento que **leem e escrevem** blocos de dados relativamente **grandes**, como **bancos de dados** e sistemas de arquivos.



"B-tree". Wikipedia. Disponível em <https://en.wikipedia.org/wiki/B-tree/>. Acessado em 01 de Abril de 2024 às 15:35.

Árvore B:

- é uma estrutura de dados em **árvore balanceada (com auto equilíbrio)** que mantém **dados ordenados** e permite pesquisas, acesso sequencial, inserções e exclusões em **tempo logarítmico** → T(n) = O (log n).
    - a razão entre o **número de operações** e o **tamanho da entrada** diminui e **tende a zero** quando n aumenta.
    - Um algoritmo que deve acessar todos os elementos de sua entrada não pode levar tempo logarítmico, pois o **tempo necessário para ler uma entrada de tamanho n é da ordem de n**.

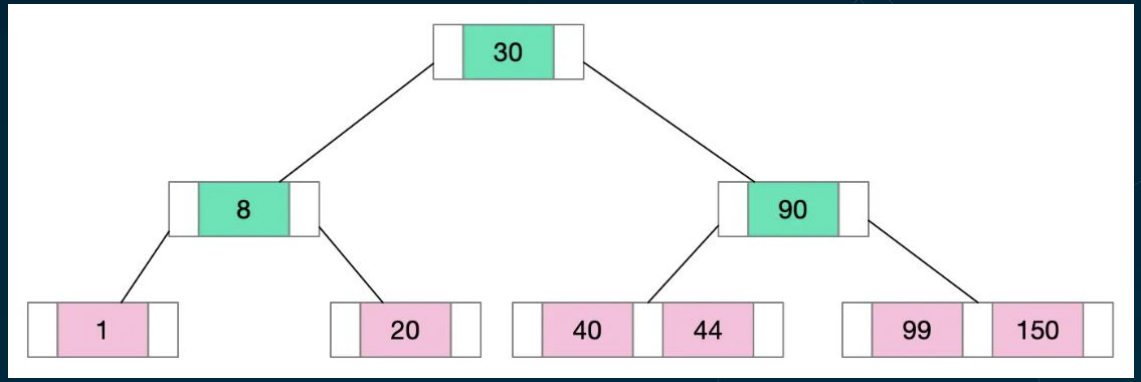
"B-tree". Wikipedia. Disponível em <https://en.wikipedia.org/wiki/B-tree/>. Acessado em 01 de Abril de 2024 às 15:35.

Estrutura da Árvore B:

- Todas as **folhas estão no mesmo nível** - o que torna a árvore balanceada.
- A raiz tem pelo menos dois filhos.
- Todos os nós internos (exceto a raiz) possuem um **número de filhos que varia de d** (grau mínimo da árvore) a **2d**.
- Um nó não-folha com **'k' filhos contém k-1 chaves**.
  Isso significa que se um nó tiver três filhos (k=3), ele conterá 2 chaves (k-1) que segmentam os dados em 3 partes correspondentes a cada nó filho.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Estrutura da Árvore B:



- excelente estrutura de dados para armazenar dados que não cabem na memória principal porque seu design **minimiza o número de acessos ao disco**.
- E como a árvore é balanceada, com todos os nós folhas na mesma profundidade, os **tempos de pesquisa permanecem consistentes e previsíveis**.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Estrutura da Árvore B+:

- **Árvore B+ é uma variante da Árvore B**.
  amplamente utilizada em sistemas de armazenamento baseados em disco, especialmente para **índices de bancos de dados**.

- Ponteiros de dados são **armazenados apenas nos nós folha**.
  Os nós internos contêm apenas chaves e **ponteiros para outros nós**. Isso significa que muito mais chaves podem ser armazenadas em nós internos, **reduzindo a altura total** da árvore. Isso **diminui o número de acessos** ao disco necessários para muitas operações.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

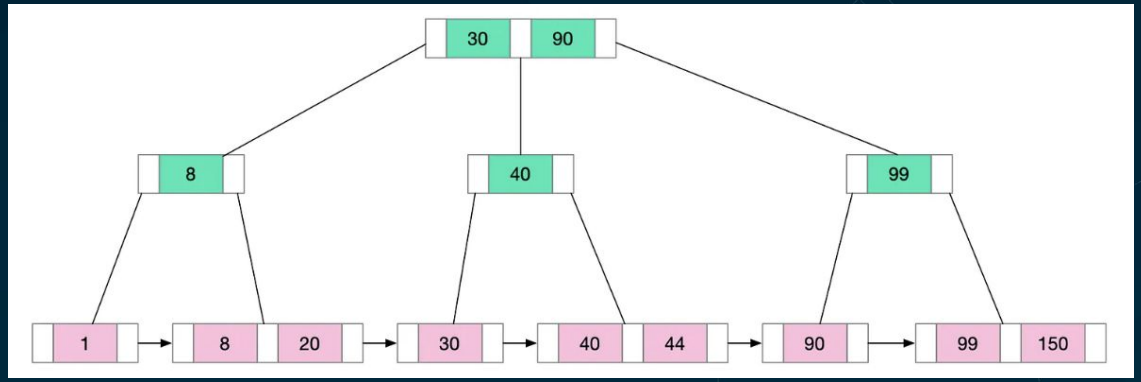
Estrutura da Árvore B+:

- Todos os nós folha estão vinculados em uma lista vinculada (**linked list**). Isso torna as consultas de intervalo eficientes. Podemos acessar o primeiro nó do intervalo e simplesmente seguir a lista vinculada para recuperar o restante.

- **Cada chave aparece duas vezes**, uma vez nos nós internos e uma vez nos nós folhas. A chave nos nós internos atua como um ponto de divisão para decidir em qual subárvore o valor desejado poderia estar.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Estrutura da Árvore B+:



- Particularmente adequadas para sistemas com grandes quantidades de dados que não cabem na memória principal.
- Como os dados só podem ser acessados a partir dos nós folha, cada pesquisa requer **um percurso de caminho da raiz até uma folha**. Todas as operações de acesso a dados levam um tempo consistente.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

FullCycle

Aula 64
Índice Agrupado e Não Agrupado
Clustered Index e Non-clustered Index

Índice Agrupado e Não Agrupado 1/7

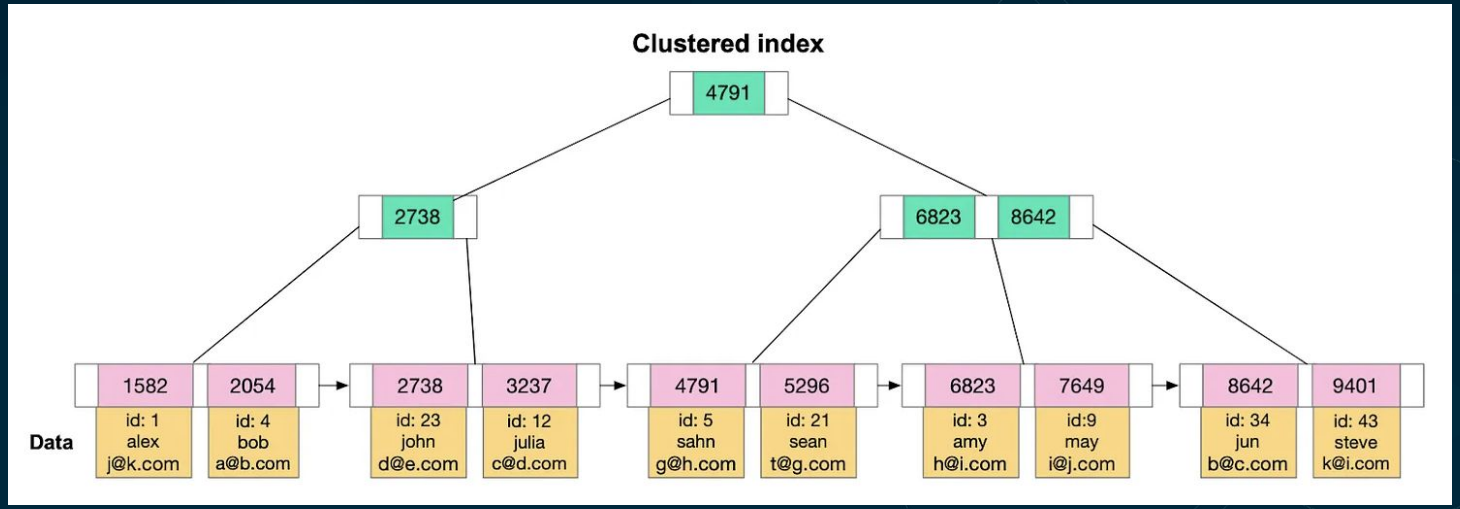
Índice Agrupado (Clustered Index):

- **reordena a maneira** como os registros são armazenados **fisicamente**.
- não armazena registros aleatoriamente ou mesmo na ordem em que foram inseridas.
- Em vez disso, organiza-os para se **alinharem com a ordem do índice**, daí o termo “agrupado”.
- Os atributos específicos usados para organizar essa ordem são chamados de **chave clusterizada**.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Índice Agrupado e Não Agrupado 2/7

Índice Agrupado (Clustered Index):



- Arranjo dos dados determina a **ordem física dos dados no disco**.
- **Lógica da lista telefônica**: classificada por sobrenome e depois nome. Números de telefone e endereço são armazenados com o índice ordenado.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Índice Agrupado e Não Agrupado 3/7

Índice Agrupado (Clustered Index):

- **Dados físicos podem ser classificadas em apenas uma ordem.** Adicionar ou alterar o índice clusterizado pode ser **demorado**, pois requer a **reordenação física** dos registros.

- É importante selecionar a chave clusterizada com cuidado. Normalmente, é benéfico escolher uma **chave sequencial exclusiva** para **evitar entradas duplicadas** e **minimizar divisões de página** ao inserir novos dados.

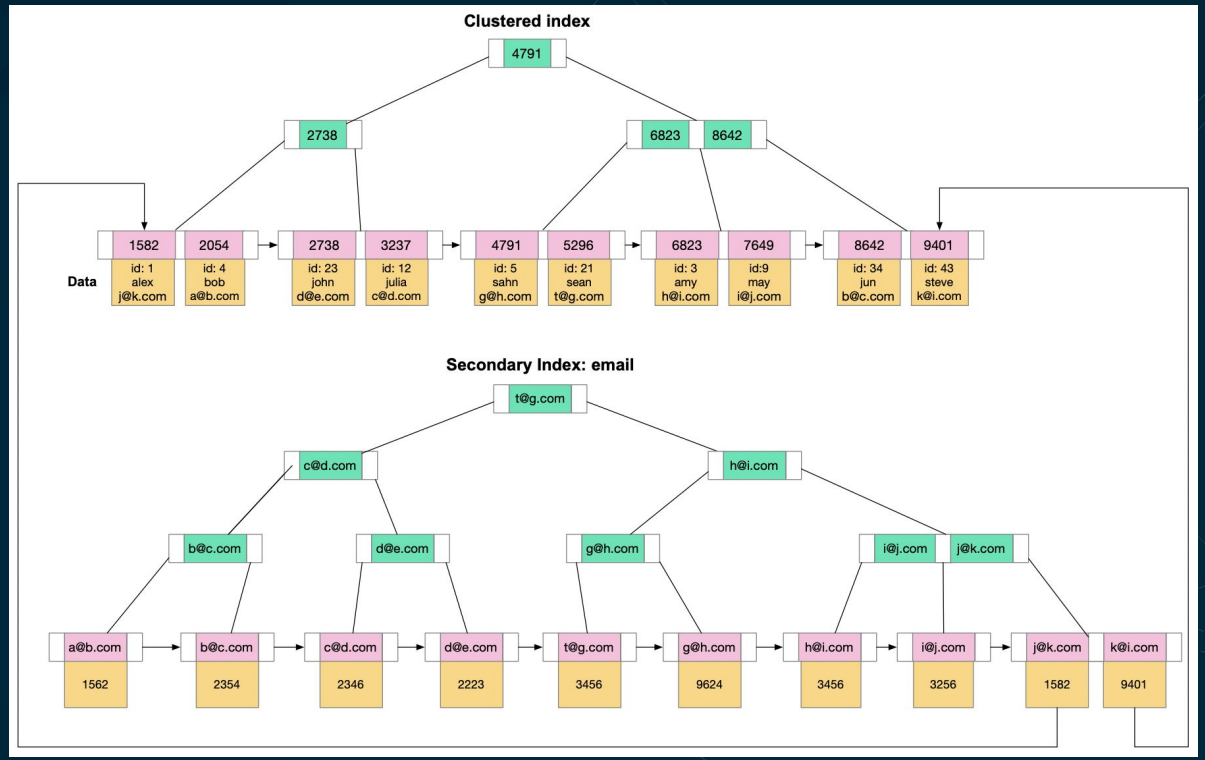
- Em muitos bancos de dados, a **restrição de chave primária** cria automaticamente um índice clusterizado nessa coluna.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Índice Agrupado e Não Agrupado 4/7

Índice Não Agrupado (Non-clustered Index):

- É como o índice encontrado no final de um livro.
- Vinculam as entradas do índice às páginas de dados.



"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Índice Agrupado e Não Agrupado 5/7

Índice Não Agrupado (Non-clustered Index):

- armazenados **separadamente** dos dados;
- **ordem física** dos dados **não é igual** à **ordem lógica** estabelecida pelo índice.
- O acesso aos dados usando um índice não agrupado **envolve pelo menos duas leituras de disco**, uma para acessar o índice e outra para acessar os dados. Isso contrasta com um **índice clusterizado, onde o índice e os dados são a mesma coisa**.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Índice Agrupado e Não Agrupado 6/7

Índice Não Agrupado (Non-clustered Index):

- É possível usar vários índices não clusterizados, cada um sendo útil para diferentes tipos de consultas.
- Benéficos para consultas que envolvem **colunas não incluídas no índice clusterizado**.
- Melhoram o desempenho de consultas que **não envolvem a chave clusterizada** ou que **não exigem a verificação de um intervalo de dados**.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

Índice Agrupado e Não Agrupado 7/7

Índice Não Agrupado (Non-clustered Index):

- Embora possam **acelerar as operações de leitura**, eles **podem retardar as operações de gravação**, pois cada índice deve ser atualizado sempre que os dados são modificados.

- É crucial **encontrar um equilíbrio** ao decidir o número e o tipo de índices não clusterizados.

"Database Indexing Strategies". ByteByteGo. Disponível em <https://blog.bytebytego.com/p/database-indexing-strategies/>. Acessado em 01 de Abril de 2024 às 15:35.

FullCycle

Aula 65
Tipos de Índices

FullCycle

Tipos de Índices 1/2

Índice Primário
Primary Index

Organiza os dados com base nas **chaves primárias**.

Garante **unicidade** das chaves primárias.

Acelera a recuperação de dados com base nas chaves primárias, geralmente **sequenciais**.

Denso ou **Esparso**.

Índice Secundário
Secondary Index

Organiza os dados com base em colunas que **não são chaves primárias**.

Permite **buscas eficientes** nessas colunas.

Não garante unicidade como o índice primário.

Denso ou **Esparso**.

Índice Agrupado
Clustered Index

Registros são **armazenados fisicamente na ordem** especificada pelo índice.

Frequentemente, a **chave primária** serve como índice agrupado.

Pode ser baseado em qualquer atributo do registro.

Índice de Texto Completo
Full Text Search

Usados para **pesquisas de texto** livre em **grandes blocos de texto**, como documentos, artigos ou registros de blog.

Apresenta recursos como pesquisa de **palavras-chave**, **frases**, **proximidade** e **relevância**.

FullCycle

Tipos de Índices 2/2

Índice de Hash
Hash Index

Ideal para **igualdades exatas**.

Não são eficazes para buscas de intervalos.

Busca registros de forma eficiente usando uma **função de hash**, a qual transforma uma entrada (**chave**) em um valor de **tamanho fixo** (o valor de hash).

Índice de Bitmap
Bitmap Index

Otimiza consultas que envolvem atributos com um **número limitado de valores distintos**.

Realiza operações de **conjunto** (interseção, união e diferença) de forma muito eficiente usando **operações bitwise** (bit a bit).

Índice Geoespacial
Spatial Index

Armazena e consulta dados com informações geográficas, como **coordenadas de latitude e longitude**.

Suporta consultas espaciais, como **busca de pontos** em uma área específica ou cálculo de **distância** entre dois pontos.

Índice de Vetor
Vector Search

Organiza os dados de forma a permitir consultas eficientes em **várias dimensões**, como busca por **similaridade** ou **vizinhança** em um espaço vetorial.

FullCycle

Aula 66
Conjectura RUM

Conjectura RUM

RUM = **R**ead, **U**pdate & **M**emory

- **Métodos de acesso** aos dados se adaptam ao **hardware** e **carga** de trabalho
- **Pequenas mudanças** na carga de trabalho ou no hardware levam a uma **reformulação dos métodos de acesso**.
- **Desafios fundamentais** ao projetar um novo método de acesso são minimizar:
  - **tempos de leitura (R),**
  - **custo de atualização (U) e**
  - **sobrecarga de memória / armazenamento (M).**

"Designing Access Methods: The RUM Conjecture". Harvard University. Disponível em <https://stratos.seas.harvard.edu/files/stratos/files/rum.pdf>. Acessado em 03 de Abril de 2024 às 19:25.

Conjectura RUM

RUM = **R**ead, **U**pdate & **M**emory

- **Conjectura RUM: Ler, atualizar, armazenar – otimizar dois às custas do terceiro.**

- Uma solução **ideal** é um método de acesso que sempre fornece o **menor custo de leitura**, o **menor custo de atualização**, e **não requer memória extra** ou espaço de armazenamento sobre os dados básicos.

- Na prática, as estruturas de dados são projetadas para **comprometer entre as 3 sobrecargas de RUM**, enquanto o design ideal depende de vários fatores, como **hardware**, **carga** de trabalho e **expectativas** do usuário.

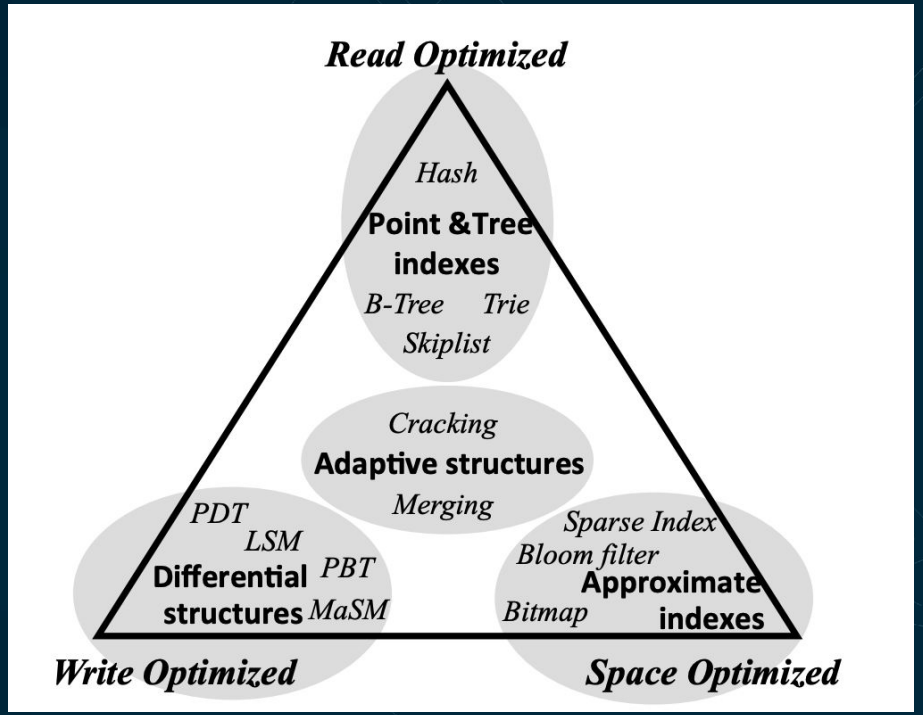
"Designing Access Methods: The RUM Conjecture". Harvard University. Disponível em <https://stratos.seas.harvard.edu/files/stratos/files/rum.pdf>. Acessado em 03 de Abril de 2024 às 19:25.

Conjectura RUM

RUM = **R**ead, **U**pdate & **M**emory

Leitura Otimizada

- Hash, B-Tree, Trie, Skiplist
- Tempo de acesso constante
- Usa mais espaço
- Escrita mais lenta



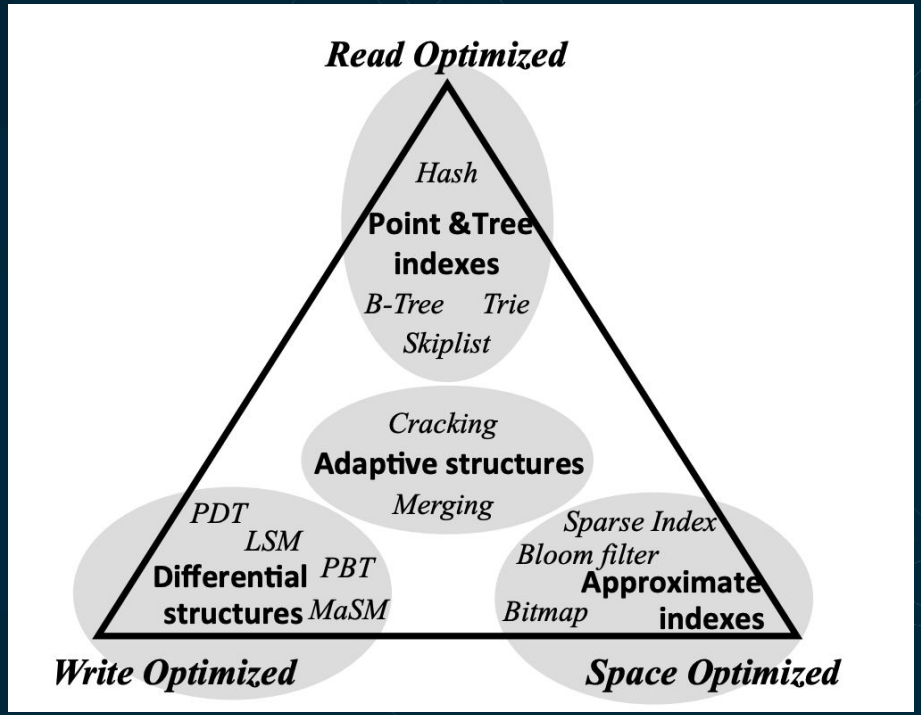
"Designing Access Methods: The RUM Conjecture". Harvard University. Disponível em <https://stratos.seas.harvard.edu/files/stratos/files/rum.pdf>. Acessado em 03 de Abril de 2024 às 19:25.

Conjectura RUM

RUM = **R**ead, **U**pdate & **M**emory

Escrita Otimizada

- Position Differential Tree (PDT), Log-structured Merge Tree (LSM), Materialized Sort-Merge (MaSM), Partitioned B-tree (PBT)
- Tempo reduzido para escrita com o uso de estruturas secundárias diferenciais
- Consolida updates e aplica em bulk



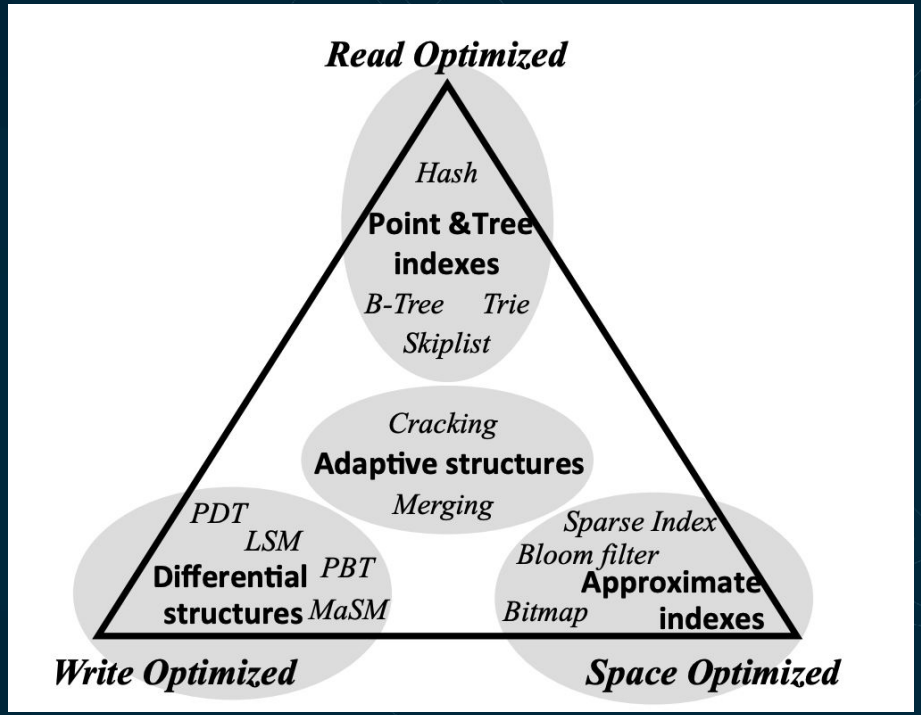
"Designing Access Methods: The RUM Conjecture". Harvard University. Disponível em <https://stratos.seas.harvard.edu/files/stratos/files/rum.pdf>. Acessado em 03 de Abril de 2024 às 19:25.

Conjectura RUM

RUM = **R**ead, **U**pdate & **M**emory

Espaço Otimizado

- Índices esparsos (Sparse Index), Bloom filter, Bitmap
- Técnicas de compressão
- Hash e índice com perdas (lossy)



"Designing Access Methods: The RUM Conjecture". Harvard University. Disponível em <https://stratos.seas.harvard.edu/files/stratos/files/rum.pdf>. Acessado em 03 de Abril de 2024 às 19:25.

FullCycle

Aula 67
Índices na Prática no MongoDB

https://www.mongodb.com/pt-br/docs/manual/indexes/

FullCycle

Aula 68
Formato GeoJSON

GeoJSON

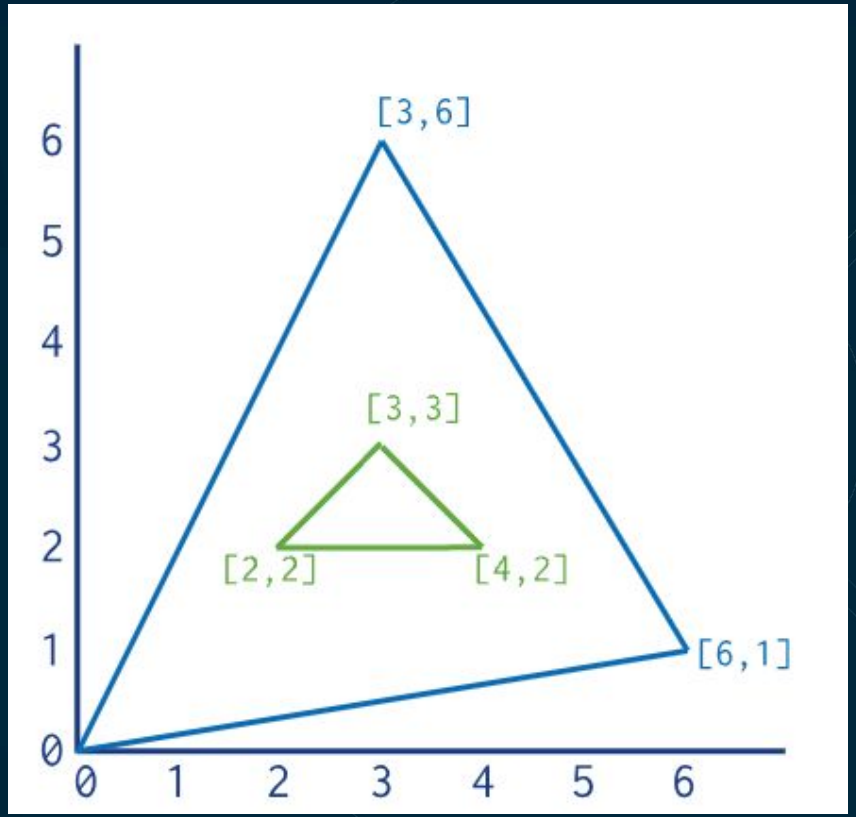
- Formato de **intercâmbio de dados geoespaciais** baseado em notação de objeto JavaScript (**JSON**).
- Define **vários tipos de objetos JSON** e a maneira como eles são **combinados** para **representar dados sobre características geográficas**, suas propriedades e suas extensões espaciais.
- Usa o sistema de referência de coordenadas geográficas **World Geodetic System 1984** e unidades de graus decimais.

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

Formato GeoJSON 2/8

Polígono GeoJSON

```
{
  type : "Polygon",
  coordinates : [
    [ [ 0 , 0 ] , [ 3 , 6 ] , [ 6 , 1 ] , [ 0 , 0 ] ],
    [ [ 2 , 2 ] , [ 3 , 3 ] , [ 4 , 2 ] , [ 2 , 2 ] ]
  ]
}
```



"Objetos GeoJSON". MongoDB. Disponível em <https://www.mongodb.com/pt-br/docs/manual/reference/geojson/#std-label-geospatial-indexes-store-geojson>. Acessado em 03 de Abril de 2024 às 20:55.

Ponto GeoJSON

As coordenadas do ponto estão na **ordem x, y** (leste, norte para coordenadas projetadas, **longitude e latitude para coordenadas geográficas**):

```
{
    "type": "Point",
    "coordinates": [100.0, 0.0]
}
```

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

Reta GeoJSON

A reta é um conjunto (array) de posições:

```
{
    "type": "LineString",
    "coordinates": [
        [100.0, 0.0],
        [101.0, 1.0]
    ]
}
```

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

Formato GeoJSON 5/8

Polígono GeoJSON

O polígono é uma matriz (array) de anel linear. O primeiro elemento da matriz representa o anel externo. Quaisquer elementos subsequentes representam anéis internos (ou orifícios).

```
{
    "type": "Polygon",
    "coordinates": [
        [
            [100.0, 0.0],
            [101.0, 0.0],
            [101.0, 1.0],
            [100.0, 1.0],
            [100.0, 0.0]
        ]
    ]
}
```

```
{
    "type": "Polygon",
    "coordinates": [
        [
            [100.0, 0.0],
            [101.0, 0.0],
            [101.0, 1.0],
            [100.0, 1.0],
            [100.0, 0.0]
        ],
        [
            [100.8, 0.8],
            [100.8, 0.2],
            [100.2, 0.2],
            [100.2, 0.8],
            [100.8, 0.8]
        ]
    ]
}
```

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

Formato GeoJSON 6/8

Multi Pontos GeoJSON

É um conjunto (array) de posições:

```
{
    "type": "MultiPoint",
    "coordinates": [
        [100.0, 0.0],
        [101.0, 1.0]
    ]
}
```

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

Multi Retas GeoJSON

É um conjunto (array) de retas:

```
{
    "type": "MultiLineString",
    "coordinates": [
        [
            [100.0, 0.0],
            [101.0, 1.0]
        ],
        [
            [102.0, 2.0],
            [103.0, 3.0]
        ]
    ]
}
```

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

Formato GeoJSON 8/8

Multi Polígonos GeoJSON

É um conjunto (array) de polígonos:

```
{
    "type": "MultiPolygon",
    "coordinates": [
        [
            [
                [102.0, 2.0],
                [103.0, 2.0],
                [103.0, 3.0],
                [102.0, 3.0],
                [102.0, 2.0]
            ]
        ],
        [
            [
                [100.0, 0.0],
                [101.0, 0.0],
                [101.0, 1.0],
                [100.0, 1.0],
                [100.0, 0.0]
            ],
            [
                [100.2, 0.2],
                [100.2, 0.8],
                [100.8, 0.8],
                [100.8, 0.2],
                [100.2, 0.2]
            ]
        ]
    ]
}
```

"The GeoJSON Format". Internet Engineering Task Force. Disponível em <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1.2>. Acessado em 03 de Abril de 2024 às 20:55.

FullCycle

Aula 69
Visão Geral sobre Análise de Texto

Pesquisa de Texto Completo (Full Text Search)

- A análise de texto habilita a **Pesquisa de Texto Completo (Full Text Search)**.
- Transforma registros (documentos) em **estruturas de dados para pesquisas**.
- A busca retorna **todos os resultados relevantes** em vez de apenas correspondências exatas.
- Correlaciona buscas do usuário com dados representados no **índice textual**.

"Text analysis overview". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-overview.html>. Acessado em 03 de Abril de 2024 às 21:55.

Tokenização (Tokenization)

- Divide um texto em partes menores, chamados **tokens**.
  Na maioria dos casos, esses tokens são **palavras individuais**.

- Ao indexar a frase **a raposa marrom rápida salta** como **string** e a pessoa usuária pesquisar por **raposa rápida**, **não haverá uma correspondência**.

- No entanto, ao **tokenizar a frase e indexar cada palavra** separadamente, os termos na string de consulta poderão ser **pesquisados individualmente**.

- Assim, buscas por **raposa rápida**, **raposa marrom** terão correspondências.

"Text analysis overview". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-overview.html>.
Acessado em 03 de Abril de 2024 às 21:55.

Tokenização (Tokenization)

- **Texto**: A raposa marrom rápida salta.

  Tokens: ["A", "raposa", "marrom", "rápida", "salta"]

- **Texto**: Minha terra tem palmeiras onde canta o sabiá; as aves, que aqui gorjeiam, não gorjeiam como lá.

  Tokens: ["Minha", "terra", "tem", "palmeiras", "onde", "canta", "o", "sabiá", "as", "aves", "que", "aqui", "gorjeiam", "não", "gorjeiam", "como", "lá"]

"Text analysis overview". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-overview.html>. Acessado em 03 de Abril de 2024 às 21:55.

Normalização (Normalization)

- **Tokenização** permite a **correspondência em termos individuais**, mas cada token ainda é **correspondido literalmente**.
- Análise de Texto pode **normalizar tokens** em um formato **padrão**.
- Isso permite **combinar tokens que não são exatamente iguais** aos termos de pesquisa, mas **semelhantes o suficiente** para ainda serem relevantes.

"Text analysis overview". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-overview.html>. Acessado em 03 de Abril de 2024 às 21:55.

Analisador (Analyzer)

- Análise de texto é realizada por um **analisador**, **conjunto de regras** que regem todo o processo.

- Etapas do processo de análise de texto incluem:
  - Mudanças no texto antes da tokenização;
  - Como o texto é convertido em tokens;
  - Alterações de normalização feitas em tokens antes da indexação ou pesquisa.

"Text analysis overview". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-overview.html>. Acessado em 03 de Abril de 2024 às 21:55.

FullCycle

Aula 70
Análise de Índice e Pesquisa

Análise de Índice e Pesquisa 1/6

Analisador (Analyzer) contém 3 blocos:

- filtros de **caracteres**
  - recebe o texto original como um **fluxo de caracteres** e transforma o fluxo **adicionando, removendo ou alterando caracteres**.
  - **0 ou muito filtros de caracteres** podem ser aplicados **em ordem**.

- **tokenizadores**
  - recebe um fluxo de caracteres, **divide-o em tokens individuais** (geralmente palavras individuais) e gera um **fluxo de tokens**.
  - **1 tokenizador** por analisador.

- filtros de **token**
  - recebe o fluxo de tokens e pode **adicionar, remover ou alterar tokens**.
  - **0 ou muito filtros de token** podem ser aplicados **em ordem**.

"Anatomy of an analyzer". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analyzer-anatomy.html>. Acessado em 03 de Abril de 2024 às 22:25.

Análise de Índice e Pesquisa 2/6

Análise de Texto acontece em 2 momentos:

- **Em Tempo de Indexação**
  - Quando o **documento é indexado**, quaisquer valores de campo de texto são analisados.
  - O analisador (conjunto de regras de análise) usado neste momento é chamado de **analisador de índice (index analyzer)**.

- **Em Tempo de Pesquisa (Search / Query Time)**
  - **A string de consulta** (o texto que o usuário está procurando) é analisada ao executar uma pesquisa de texto completo em um campo de texto.
  - O analisador usado neste momento é chamado de **analisador de pesquisa (search analyzer)**.

"Index and search analysis". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-index-search-time.html>. Acessado em 24 de Abril de 2024 às 20:00.

Análise de Índice e Pesquisa 3/6

O mesmo analisador deve ser usado nos momentos de indexação e pesquisa, na maioria dos casos.

Assim, os valores dos campos e as strings de consulta são alterados pelo analisador no **mesmo formato de tokens**. Os **tokens corresponderão** conforme o esperado em pesquisa.

Exemplo Clássico:

```
{
      "text" : "The QUICK brown foxes jumped over the dog!"
}
```

Index Analyzer converte, normaliza e retorna os tokens: **[quick, brown, fox, jump, over, dog]**

"Index and search analysis". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-index-search-time.html>. Acessado em 24 de Abril de 2024 às 20:00.

Análise de Índice e Pesquisa 4/6

Exemplo Clássico:

```
{
      "text" : "The QUICK brown foxes jumped over the dog!"
}
```

Os tokens **[quick, brown, fox, jump, over, dog]** são indexados e a pessoa usuária busca por "**Quick fox**". Sem Query Analyzer não haverá correspondência entre

Quick ≠ QUICK fox ≠ foxes

Quando a string de consulta é analisada usando o mesmo analisador: **[quick, fox]**

"Index and search analysis". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-index-search-time.html>. Acessado em 24 de Abril de 2024 às 20:00.

Análise de Índice e Pesquisa 5/6

Exemplo Clássico:

{

"text" : "**The QUICK brown foxes jumped over the dog!**"

}

text tokens: **[quick, brown, fox, jump, over, dog]**

query tokens: **[quick, fox]**

Como o valor do campo e a string de consulta foram analisados da mesma maneira, os mesmos tokens foram gerados.

Quick e fox são correspondências exatas.

A pesquisa corresponde ao documento.

Token	Query string	`text` field
`quick`	X	X
`brown`		X
`fox`	X	X
`jump`		X
`over`		X
`dog`		X

"Index and search analysis". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-index-search-time.html>. Acessado em 24 de Abril de 2024 às 20:00.

Análise de Índice e Pesquisa 6/6

Quando usar um analisador de pesquisa diferente

{ "text" : "Apple" } → Index Analyzer → [a, ap, app, appl, apple]

{ "search" : "appli" } → Search Analyzer (o mesmo) → [a, ap, app, appl, appli]

Token	appli	apple
a	X	X
ap	X	X
app	X	X
appl	X	X
appli		X

Problema: correspondência errônea entre "Apple" e "appli".

Solução: usar Search Analyzer que produz um único token em vez de um conjunto de prefixos.

[appli]

"Index and search analysis". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-index-search-time.html>. Acessado em 24 de Abril de 2024 às 20:00.

FullCycle

Aula 71
Derivação e Grafos de Tokens
Stemming e Token Graphs

Derivação e Grafos de Tokens 1/4

Derivação (Stemming):

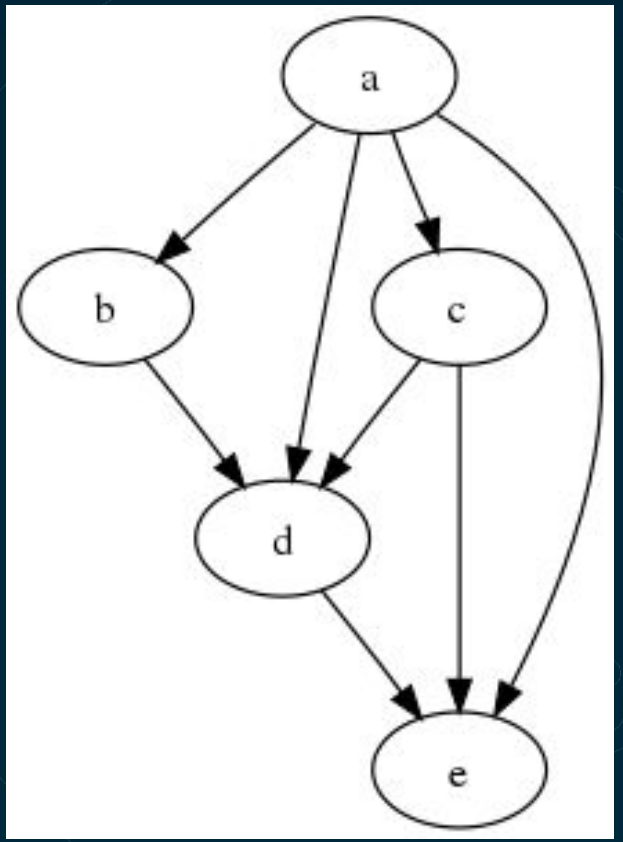
- **É o processo de redução de uma palavra à sua forma raiz.**
  Variações de uma palavra terão uma correspondência na busca.
  - Exemplos: "**walk**ing" e "**walk**" são derivadas para "**walk**"
    "**caminha**ndo" e "**caminha**da" são derivadas para "**caminha**"
- **Depende do idioma; geralmente envolve remoção de prefixos e sufixos das palavras.**

"Stemming". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/stemming.html>. Acessado em 24 de Abril de 2024 às 20:00.

Derivação e Grafos de Tokens 2/4

Grafos de Token (Token Graphs):

- Um grafo é formado por vértices (nós) e por arestas as quais conectam pares de vértices.
- Vértices podem ser qualquer tipo de objeto que esteja conectado em pares por arestas.
- Um grafo acíclico direcionado (DAG) apresenta cada aresta direcionada de um vértice para outro, de modo que ao seguir essas direções nunca será formado um loop fechado.



"Token graphs". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/token-graphs.html>. Acessado em 24 de Abril de 2024 às 20:30.

Derivação e Grafos de Tokens 3/4

Grafos de Token (Token Graphs):

- Quando um tokenizador converte um texto em um fluxo de tokens, é registrado:
  - Posição (position) de cada token no fluxo e
  - Número de posições (positionLength) que um token abrange.

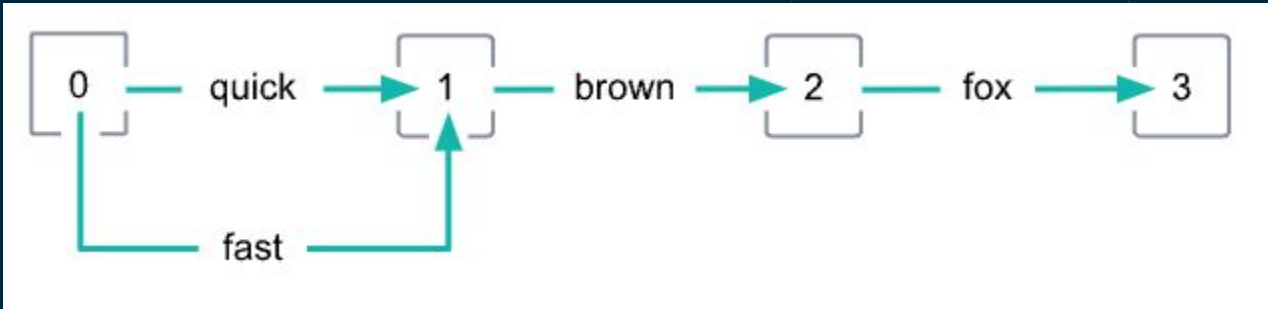
- Com estas informações, temos um grafo acíclico direcionado (DAG) chamado de Token Graph. Cada posição é um nó. Cada token é uma aresta.

"Token graphs". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/token-graphs.html>. Acessado em 24 de Abril de 2024 às 20:30.

Derivação e Grafos de Tokens 4/4

Grafos de Token (Token Graphs):

- **Sinônimos:**



"Token graphs". Elastic. Disponível em <https://www.elastic.co/guide/en/elasticsearch/reference/current/token-graphs.html>. Acessado em 24 de Abril de 2024 às 20:30.

FullCycle

Aula 72
Dependência Funcional e Redundância

Dependência Funcional e Redundância 1/2

Dependência Funcional:

- A pessoa modeladora de dados decide **como representar relacionamentos** entre entidades que são objeto da modelagem em banco de dados relacional.
- É fundamental identificar **atributos funcionalmente dependentes** de outros.

FNO	**F**NOME	**F**SAL	**D**NO	**D**ORCAM
E1	Adams	50K	D3	800K
E2	Boyd	60K	D2	900K
E3	Cope	45K	D1	800K
E4	Davis	75K	D2	900K
E5	Eliot	75K	D3	800K

FNO	Número do **Funcionário**
FNOME	Nome do **Funcionário**
FSAL	Salário do **Funcionário**
DNO	Número do **Departamento**
DORCAM	Orçamento do **Departamento**

FNO → FNOME
FNO → FSAL
FNO → DNO
DNO → DORCAM

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

Dependência Funcional e Redundância 2/2

Redundância:

- O relacionamento entre Funcionários e Departamentos sofre de redundância porque **Orçamento do Departamento depende funcionalmente do Número do Departamento**.

- A redundância é substituída por **projeções do relacionamento**.

FNO	FNOME	FSAL	DNO
E1	Adams	50K	D3
E2	Boyd	60K	D2
E3	Cope	45K	D1
E4	Davis	75K	D2
E5	Eliot	75K	D3

DNO	DORCAM
D1	800K
D2	900K
D3	800K

- A idéia geral é que o relacionamento original está em alguma forma normal.
- A projeção é alguma forma normal mais elevada.

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Aula 73
Primeira Forma Normal (1NF)

Primeira Forma Normal (1NF) 1/2

- Um relacionamento está na Primeira Forma Normal (1NF) se possui a propriedade de que **nenhum de seus domínios possui elementos que são eles próprios conjuntos**.
- Um **relacionamento não normalizado** é aquele que não está na Primeira Forma Normal.

RELACIONAMENTO NÃO NORMALIZADO

FNO	FNOME	FSAL	FTELEFONES	DNO	DORCAM
E1	Adams	50K	(84) 98571-5336, (68) 97958-0596	D3	800K
E2	Boyd	60K	(95) 98177-9440, (44) 96813-8522	D2	900K
E3	Cope	45K	(54) 99154-3328, (63) 97648-8599	D1	800K
E4	Davis	75K	(85) 98937-6685, (66) 96983-0110	D2	900K
E5	Eliot	75K	(14) 98547-2252, (69) 99354-1327	D3	800K

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Primeira Forma Normal (1NF) 2/2

- Um relacionamento está na Primeira Forma Normal (1NF) se possui a propriedade de que **nenhum de seus domínios possui elementos que são eles próprios conjuntos**.
- Um **relacionamento não normalizado** é aquele que não está na Primeira Forma Normal.

RELACIONAMENTO 1FN

FNO	FNOME	FSAL	DNO	DORCAM
E1	Adams	50K	D3	800K
E2	Boyd	60K	D2	900K
E3	Cope	45K	D1	800K
E4	Davis	75K	D2	900K
E5	Eliot	75K	D3	800K

FNO	FTELEFONE
E1	(84) 98571-5336
E1	(68) 97958-0596
E2	(95) 98177-9440
E2	(44) 96813-8522
…	…

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Aula 74
Segunda Forma Normal (2NF)

FullCycle

Segunda Forma Normal (2NF)

- Um relacionamento está na Segunda Forma Normal (2NF) se estiver na Primeira Forma Normal (1NF) e
- **Todo atributo normal, que não é chave do relacionamento, é totalmente dependente da chave candidata deste relacionamento.**

FNO	**F**NOME	**F**SAL	**D**NO
E1	Adams	50K	D3
E2	Boyd	60K	D2
E3	Cope	45K	D1
E4	Davis	75K	D2
E5	Eliot	75K	D3

DNO	**D**ORCAM
D1	800K
D2	900K
D3	800K

TNO	**T**NUMERO	**F**NO
T1	(84) 98571-5336	E1
T2	(68) 97958-0596	E1
T3	(95) 98177-9440	E2
T4	(44) 96813-8522	E2
…	…	…

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Aula 75
Terceira Forma Normal (3NF)

FullCycle

Terceira Forma Normal (3NF)

- Um relacionamento está na Terceira Forma Normal (3NF) se estiver na Segunda Forma Normal (2NF) e
- **Atributos normais, que não são chave do relacionamento, não devem possuir dependência transitiva entre si, e são dependentes exclusivamente da chave do relacionamento.**

SEGUNDA FORMA NORMAL

FNO	FNOME	FSAL	FIRPF	DNO
E1	Adams	50K	13.7K	D3
E2	Boyd	60K	16.5K	D2
E3	Cope	45K	12.4K	D1
E4	Davis	75K	20.6K	D2
E5	Eliot	75K	20.6K	D3

TERCEIRA FORMA NORMAL

FNO	FNOME	FSAL	DNO
E1	Adams	50K	D3
E2	Boyd	60K	D2
E3	Cope	45K	D1
E4	Davis	75K	D2
E5	Eliot	75K	D3

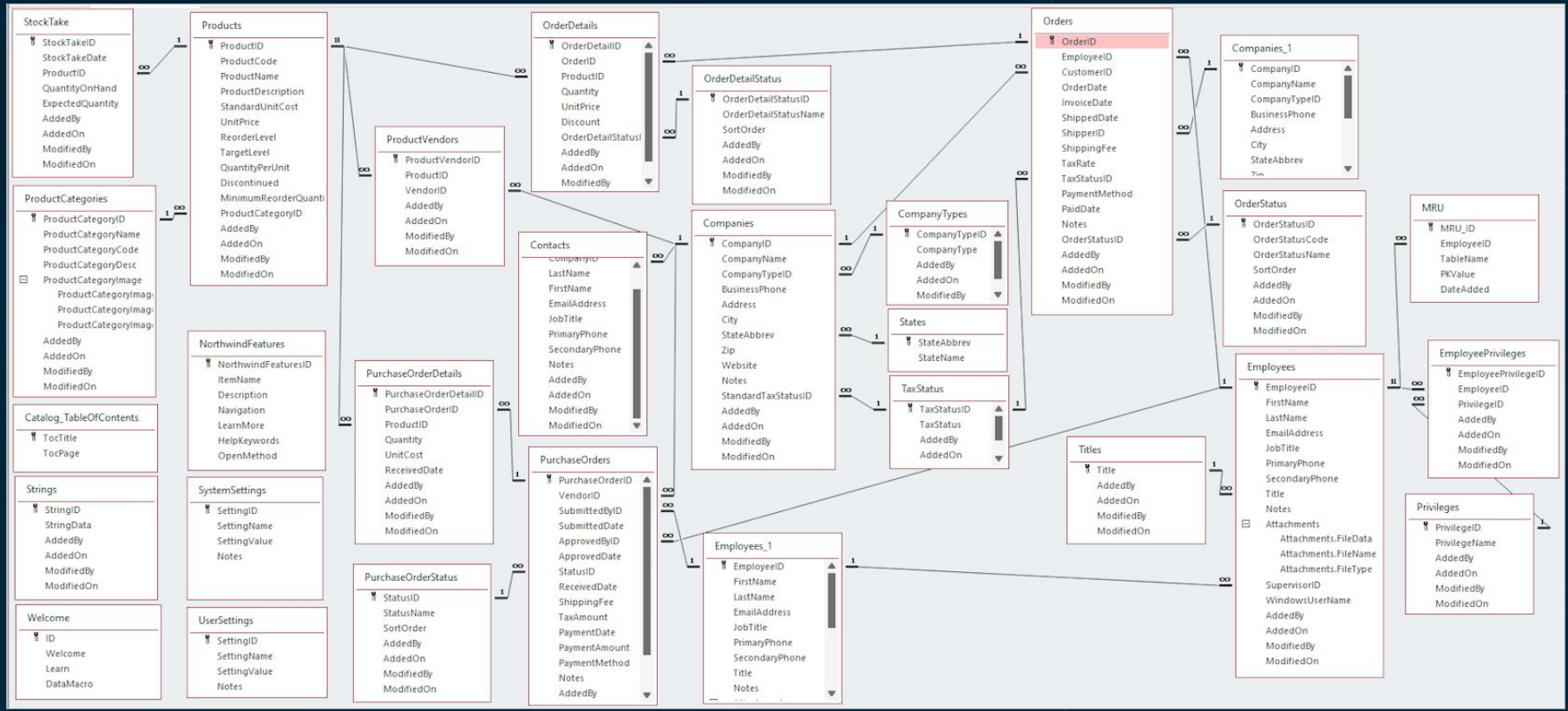
"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Aula 76
Modelo de Dados Relacional

FullCycle

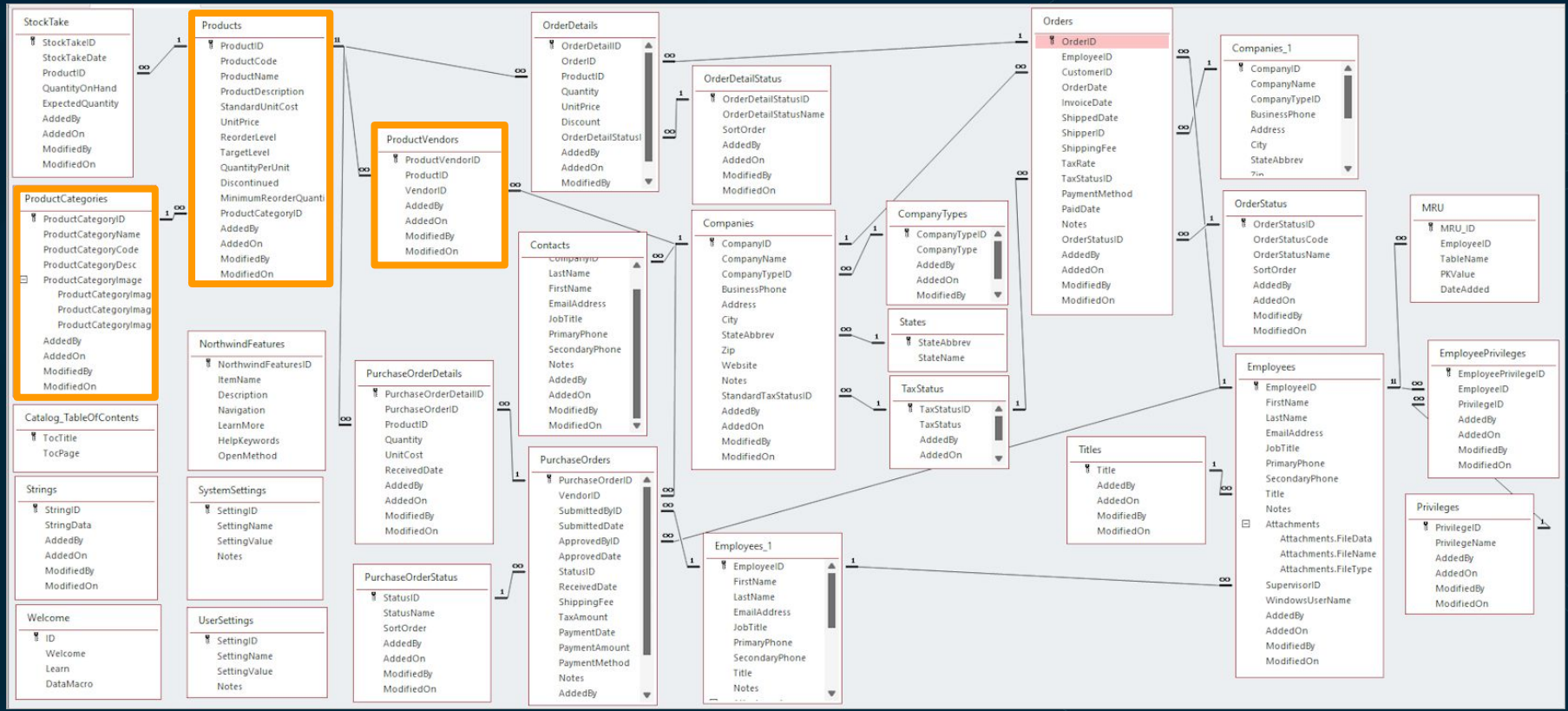
Modelo de Dados Relacional



"Northwind 2.0 Developer Edition: Template Tutorial". Microsoft. Disponível em <https://support.content.office.net/en-us/media/559a04f2-11b2-44b8-ae4a-92284d1576bd.png>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

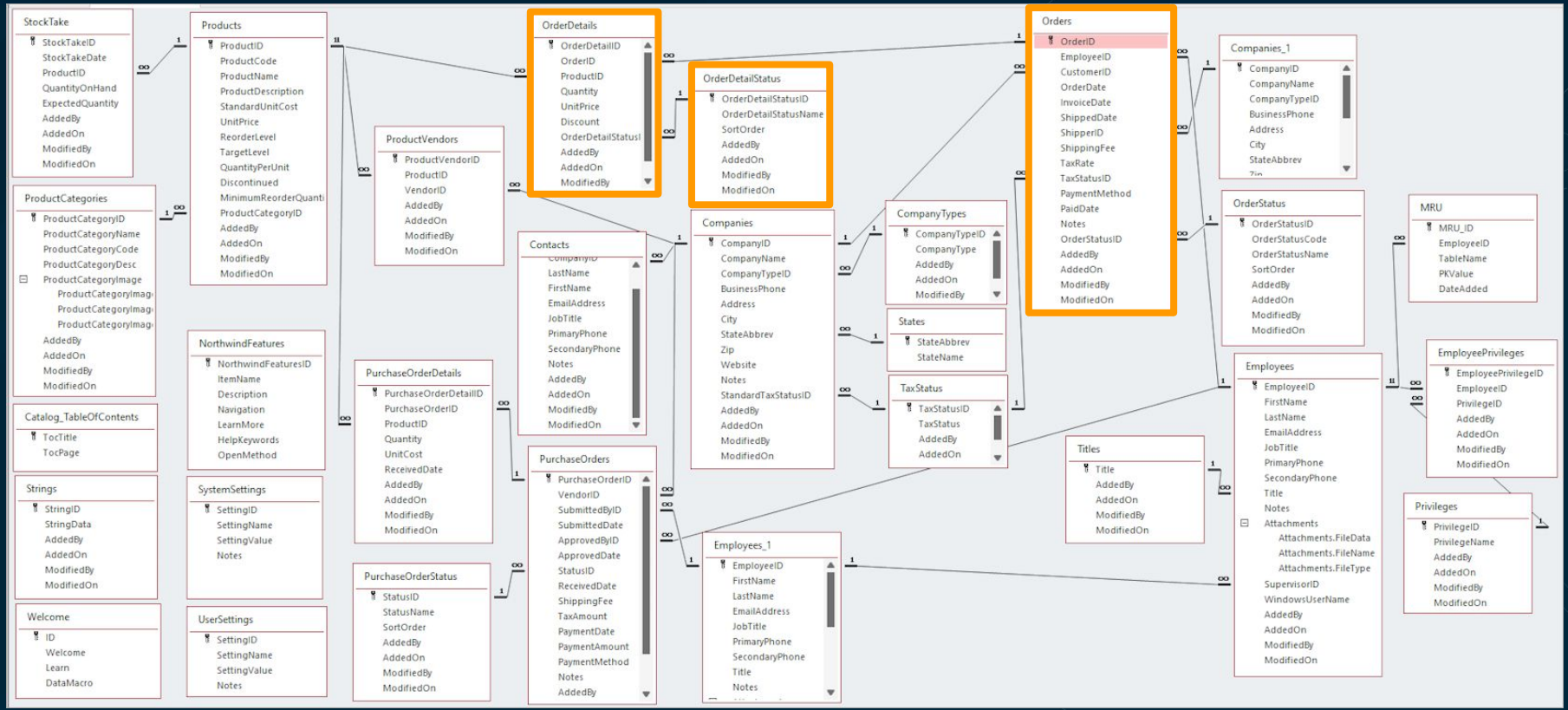
Modelo de Dados Relacional



"Northwind 2.0 Developer Edition: Template Tutorial". Microsoft. Disponível em <https://support.content.office.net/en-us/media/559a04f2-11b2-44b8-ae4a-92284d1576bd.png>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

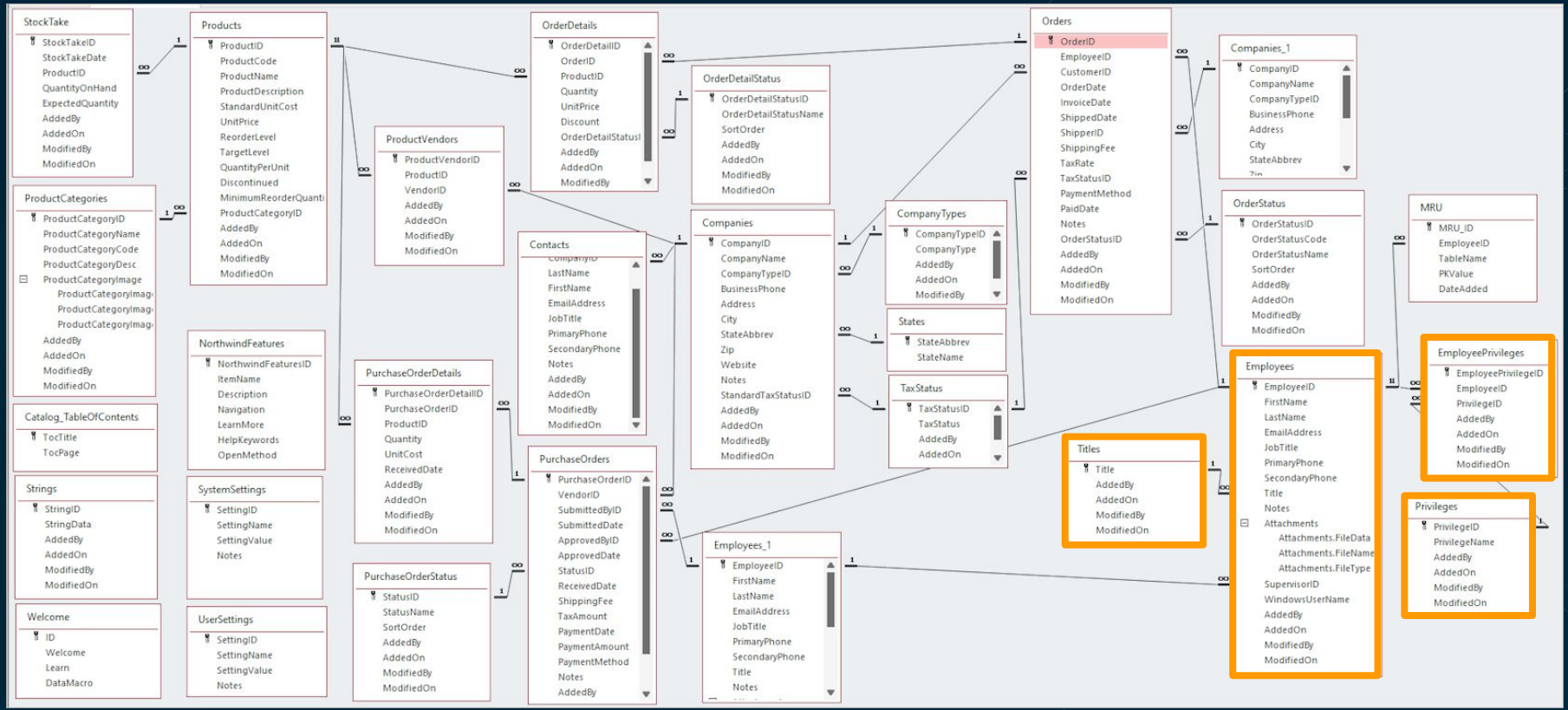
Modelo de Dados Relacional



"Northwind 2.0 Developer Edition: Template Tutorial". Microsoft. Disponível em <https://support.content.office.net/en-us/media/559a04f2-11b2-44b8-ae4a-92284d1576bd.png>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Modelo de Dados Relacional



"Northwind 2.0 Developer Edition: Template Tutorial". Microsoft. Disponível em <https://support.content.office.net/en-us/media/559a04f2-11b2-44b8-ae4a-92284d1576bd.png>. Acessado em 11 de Maio de 2024 às 19:40.

FullCycle

Aula 77
Outras Formas Normais e Desnormalização

Outras Formas Normais e Desnormalização 1/3

- Terceira Forma Normal (3FN) gera representações lógicas finais na maioria das vezes.

- **Forma Normal Boyce-Codd (BCNF)**
  - Refinamento da 3NF
  - Um relacionamento 3NF não é BCNF se:
    - houver múltiplas chaves e/ou
    - chaves compostas por múltiplos atributos e/ou
    - atributos comuns entre as chaves.

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

Outras Formas Normais e Desnormalização 2/3

- **Quarta Forma Normal (4NF)**
  - **Apresenta 3NF e**
  - **Não há dependências multivaloradas**
    - **Relacionamentos Muito-para-Muitos (M:N) resolvidos independentemente.**

Música → Artista
Música → Álbum
Artista e Álbum podem não estar relacionados
Uma mesma música pode estar em vários álbuns diferentes e ser cantada por artistas diferentes

MUSIC_ID	ALBUM_ID
MU1	AL1
MU1	AL2
MU2	AL1

MUSIC_ID	ARTIST_ID
MU1	AR1
MU1	AR2

"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

Outras Formas Normais e Desnormalização 3/3

- **Desnormalização**
  - **É uma técnica de otimização por meio da adição de dados redundantes a uma ou mais tabelas.**
  - **Evita junções dispendiosas em um banco de dados relacional.**
  - **Não significa “reverter a normalização” ou “não normalizar”.**
  - **Aplicada após a normalização.**

FNO	FNOME	FSAL	FTELEFONE1	FTELEFONE2	DNO	DORCAM
E1	Adams	50K	(84) 98571-5336	(68) 97958-0596	D3	800K
E2	Boyd	60K	(95) 98177-9440	(44) 96813-8522	D2	900K
E3	Cope	45K	(54) 99154-3328	(63) 97648-8599	D1	800K
E4	Davis	75K	(85) 98937-6685	(66) 96983-0110	D2	900K
E5	Eliot	75K	(14) 98547-2252	(69) 99354-1327	D3	800K

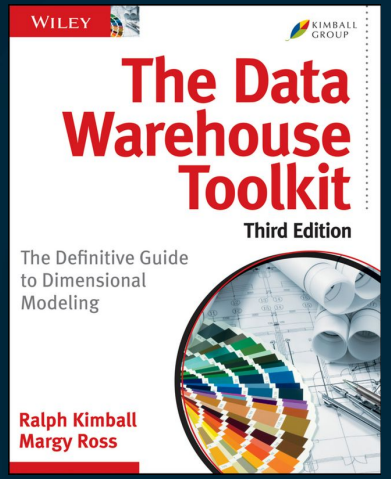
"E. F. Codd and Relational Theory, Revised Edition". C. J. Date. Technics Publications. Disponível em <https://www.amazon.com/F-Codd-Relational-Theory-Revised/dp/1634629280>. Acessado em 11 de Maio de 2024 às 19:40.

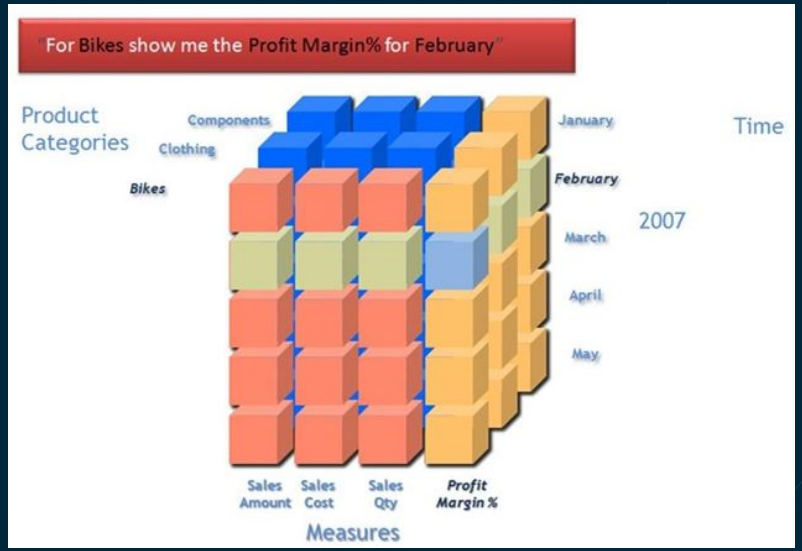
FullCycle

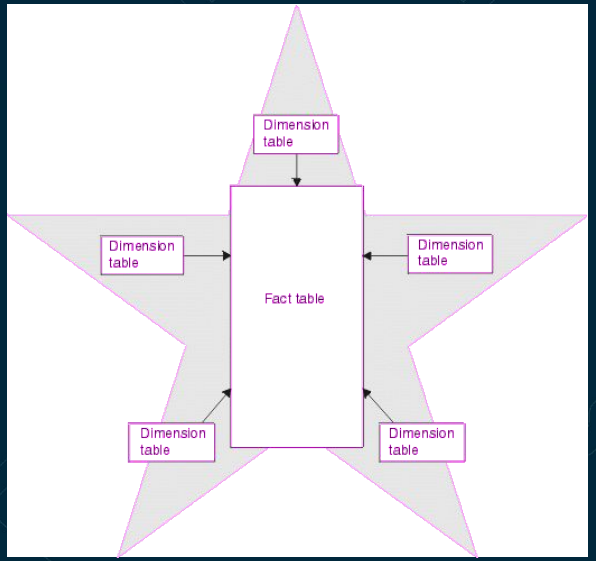
Aula 78
Modelo Dimensional de Dados (DNF)

Modelo Dimensional de Dados 1/7

- Técnica de modelagem difundida por Ralph Kimball para Data Warehouses.
- Também conhecido por **Modelo Estrela**, **Star Schema** e **Dimensional Normal Form**.
- Faz uso da desnormalização.



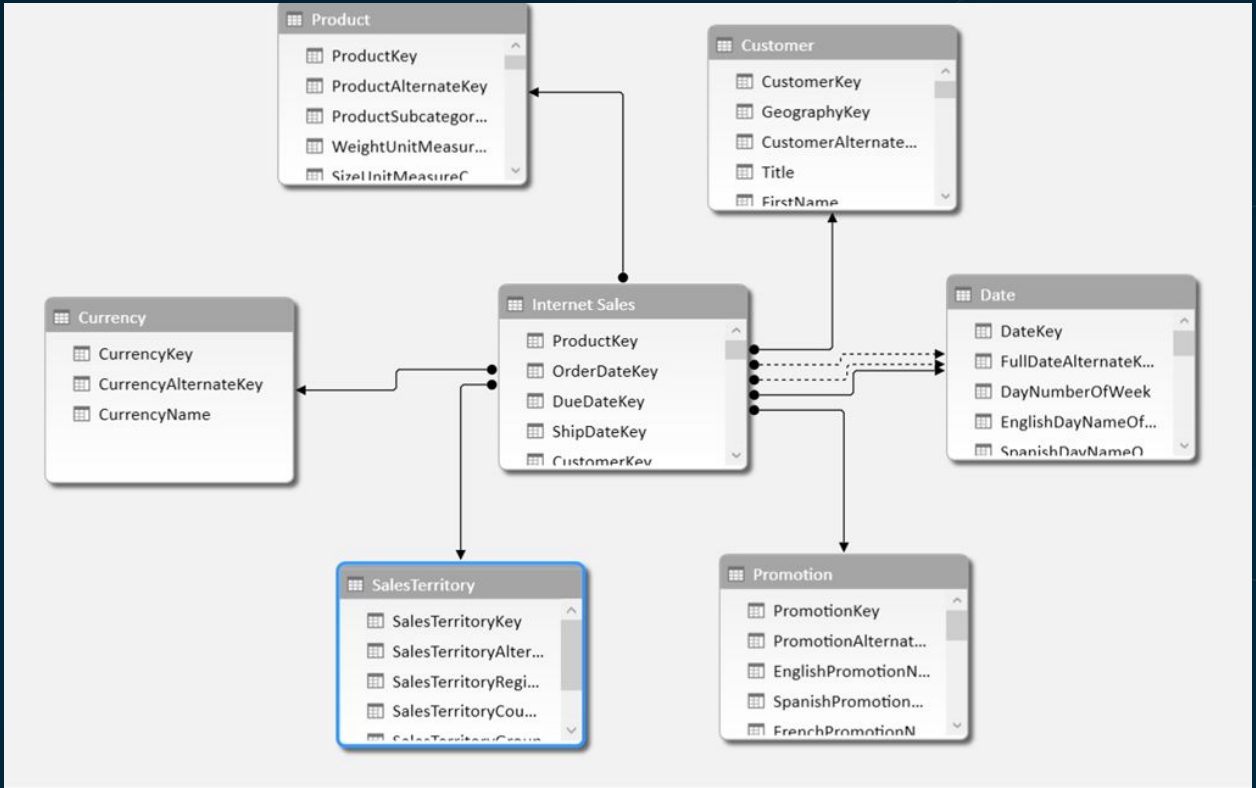




"The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modeling". Kimball, Ralph. Ross, Margy. Wiley.

FullCycle

Modelo Dimensional de Dados 2/7



"The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modeling". Kimball, Ralph. Ross, Margy. Wiley.

Modelo Dimensional de Dados 3/7

A Modelagem de Dados Dimensional requer 4 decisões:

1) Selecionar o processo de negócio.
2) Declarar o grão.
3) Identificar as dimensões.
4) Identificar os fatos.

Processo de Negócio: atividades operacionais executadas pela organização.
Grão: determina o que cada registro (linha na tabela) representa.
Dimensão: atributos descritivos (o que, quem, onde, quando, por que e como).
Fato: medições resultantes da execução do processo de negócio.

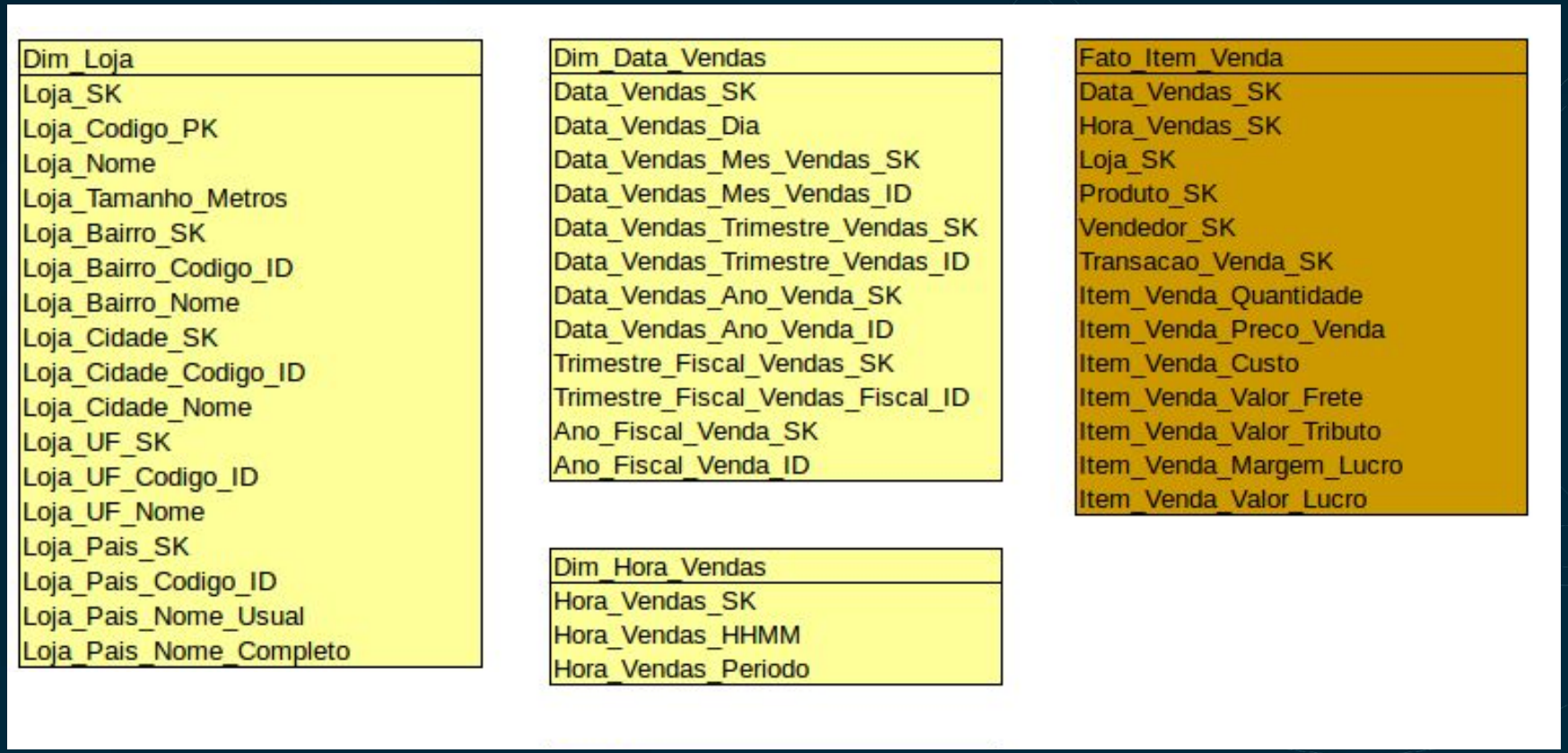
"The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modeling". Kimball, Ralph. Ross, Margy. Wiley.

Modelo Dimensional de Dados 4/7

Exemplo de Prática de Modelagem Dimensional

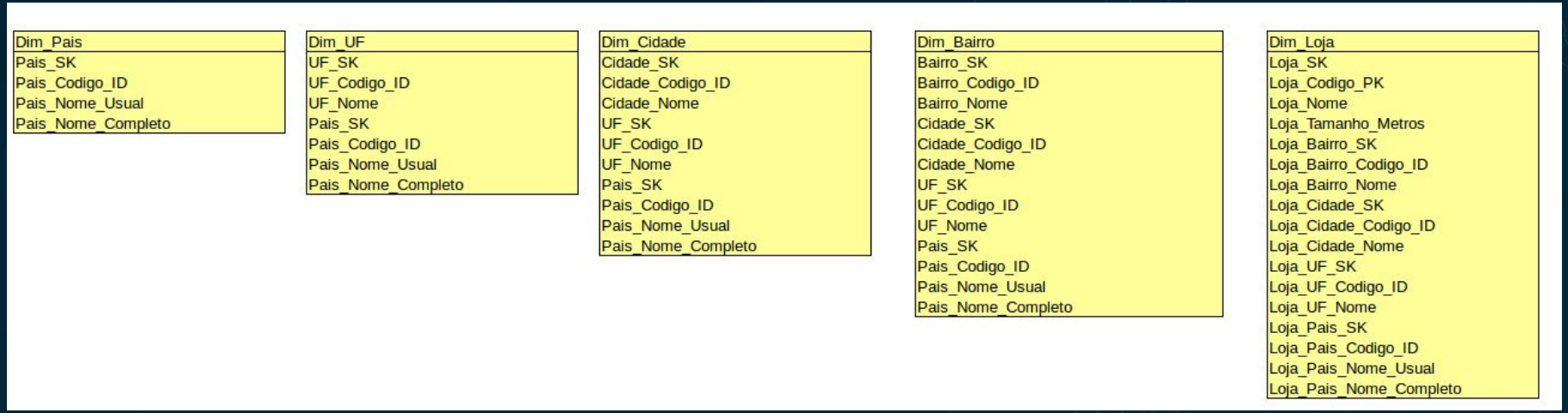
Modelo Dimensional de Dados 5/7

Exemplo de Prática de Modelagem Dimensional



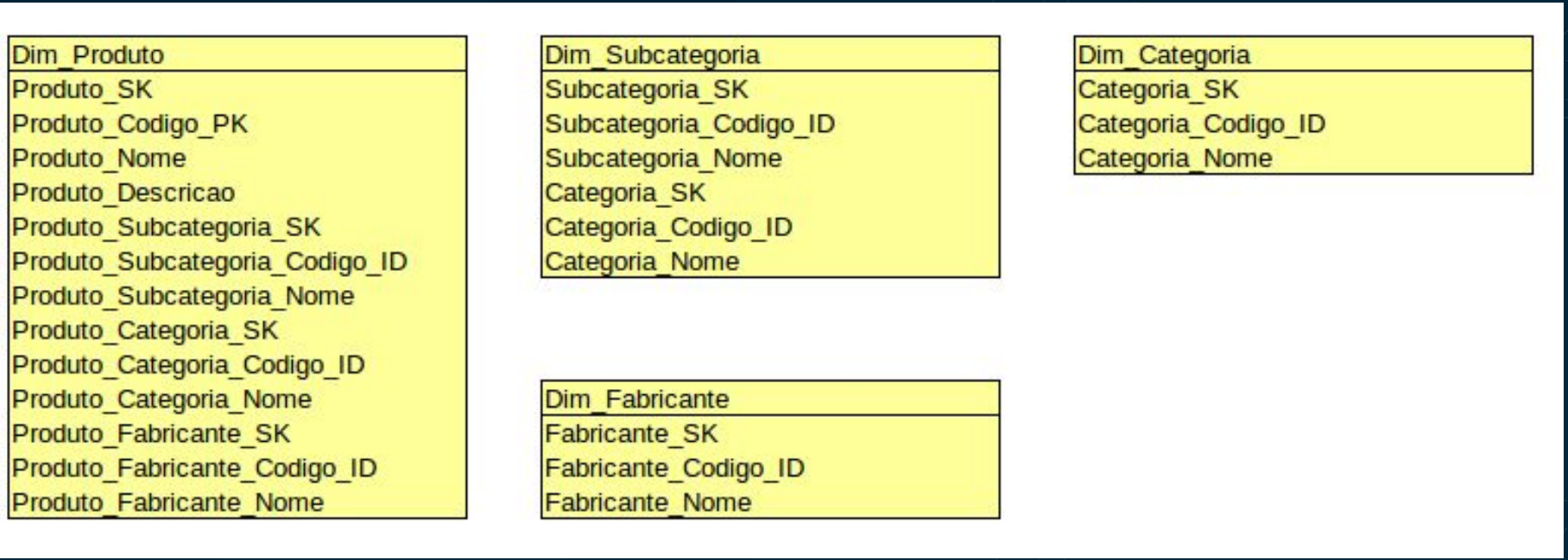
Modelo Dimensional de Dados 6/7

Exemplo de Prática de Modelagem Dimensional



Modelo Dimensional de Dados 7/7

Exemplo de Prática de Modelagem Dimensional



FullCycle

Aula 79
Esquema JSON

FullCycle

Esquema JSON 1/6

- JSON = JavaScript Object Notation ← Notação de Objeto JavaScript
- 2 estruturas fundamentais:
  - Coleção de pares chave-valor ← objeto, registro, struct
  - Lista ordenada de valores ← array, vetor, lista, sequência

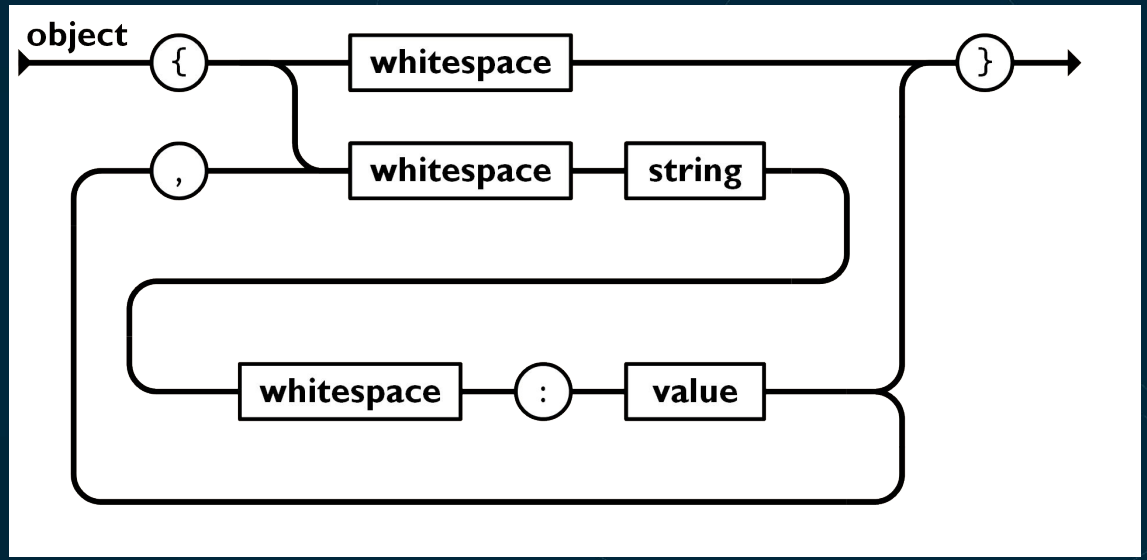
"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>

Esquema JSON 2/6

- Objeto: conjunto desordenado de pares chave-valor.

{ } ← objeto vazio

```
{
        "nome" : "Lourenço",
        "sobrenome" : "Taborda"
}
```



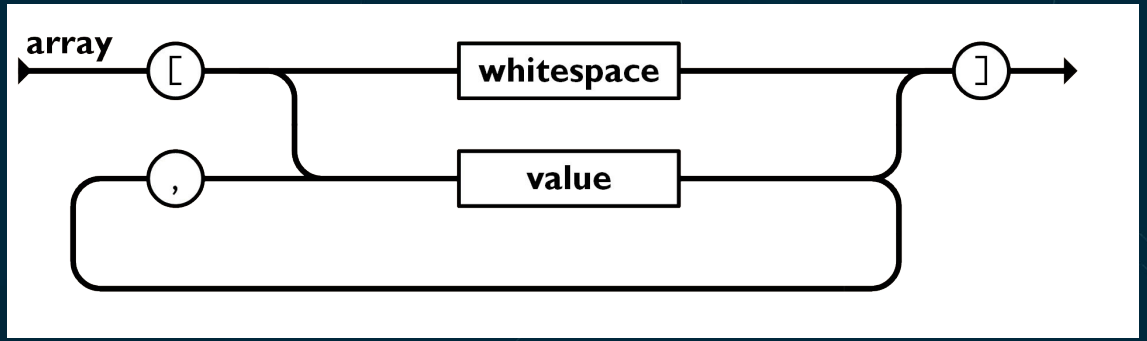
"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>

Esquema JSON 3/6

- Array: coleção ordenada de valores.

  [ ] ← array vazio

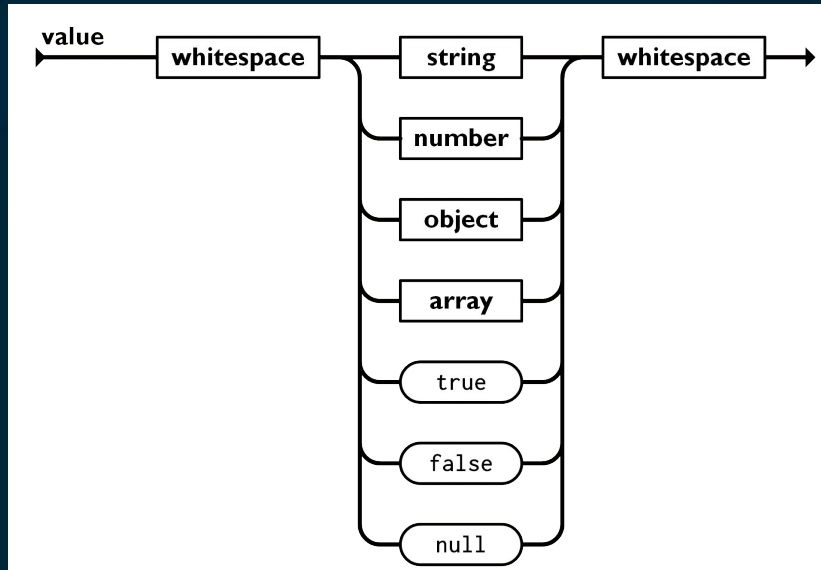
  [quick, brown, fox, jump, over, dog]

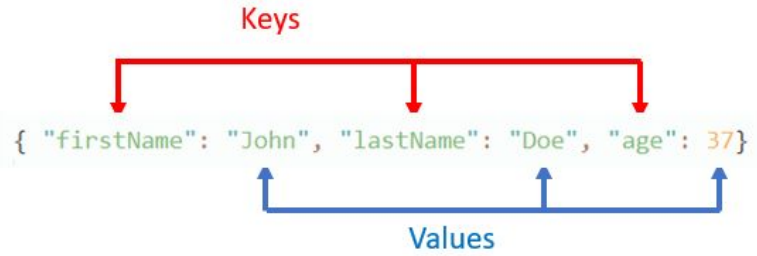


"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>

Esquema JSON 4/6

- Valor: string entre aspas, número, true/false, null, objeto ou array.





"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>

Esquema JSON 5/6

- Infinitas maneiras de organizar um objeto JSON!

"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>

FullCycle

Esquema JSON 6/6

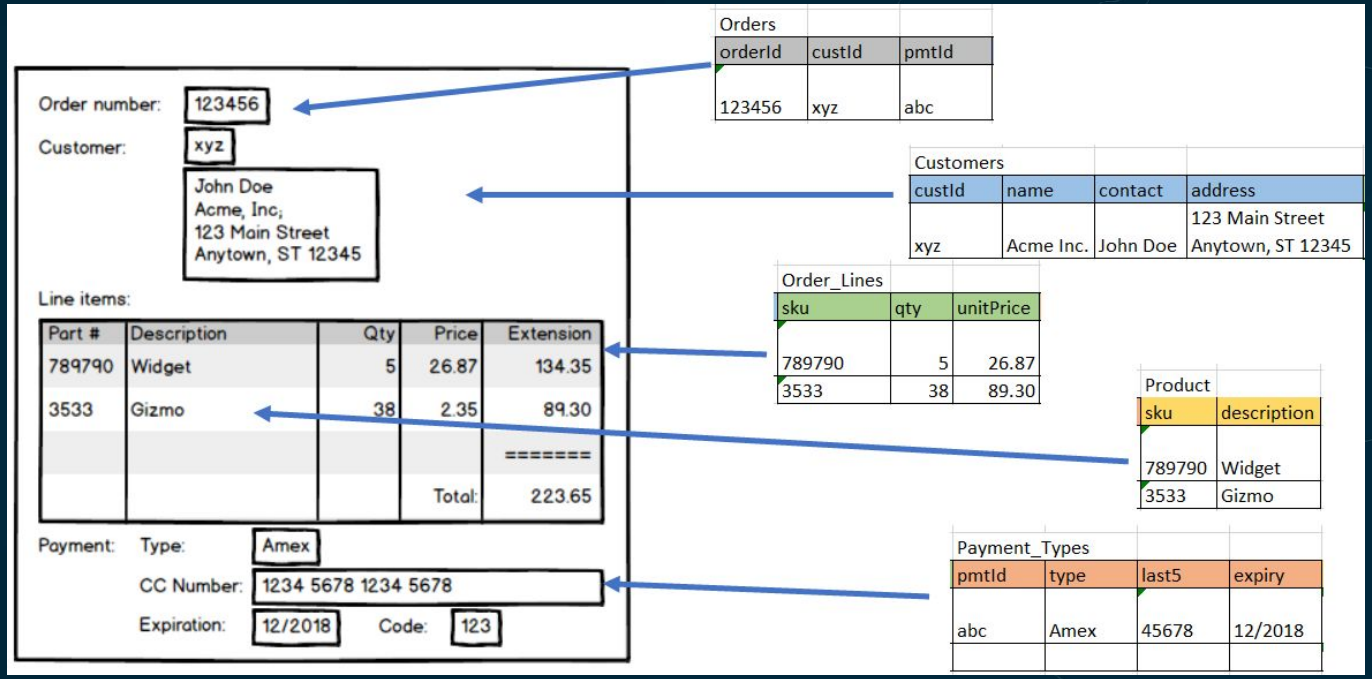
```
db.createCollection("students", {
   validator: {
      $jsonSchema: {
         bsonType: "object",
         title: "Student Object Validation",
         required: [ "address", "major", "name", "year" ],
         properties: {
            name: {
               bsonType: "string",
               description: "'name' must be a string and is required"
            },
            year: {
               bsonType: "int",
               minimum: 2017,
               maximum: 3017,
               description: "'year' must be an integer in [ 2017, 3017 ] and is required"
            },
            gpa: {
               bsonType: [ "double" ],
               description: "'gpa' must be a double if the field exists"
            }
         }
      }
   }
} )
```

FullCycle

Aula 80
JSON em Bancos de Dados

JSON em Bancos de Dados 1/2

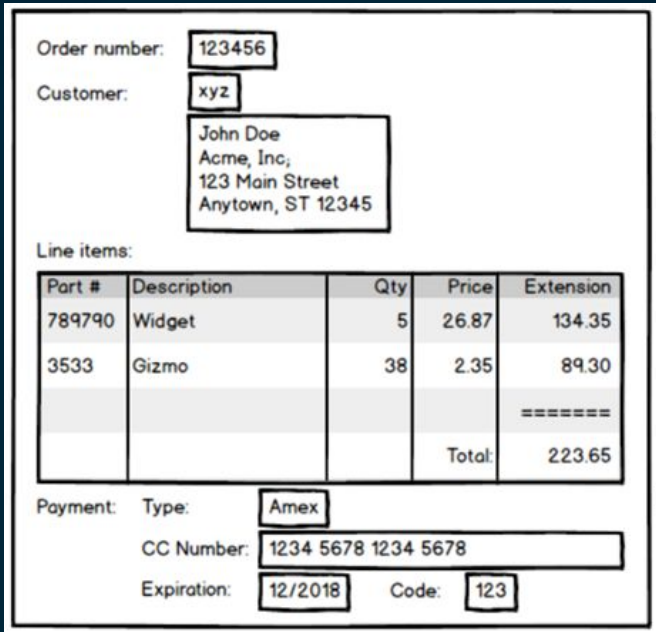
Escalabilidade horizontal distribuída, flexibilidade e evolução facilitada do modelo de dados



"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>

JSON em Bancos de Dados 2/2

Escalabilidade horizontal distribuída, flexibilidade e evolução facilitada do modelo de dados



Order number: 123456

Customer: xyz

John Doe
Acme, Inc;
123 Main Street
Anytown, ST 12345

Line items:

Part #	Description	Qty	Price	Extension
789790	Widget	5	26.87	134.35
3533	Gizmo	38	2.35	89.30
				=======
			Total:	223.65

Payment: Type: Amex

CC Number: 1234 5678 1234 5678

Expiration: 12/2018 Code: 123

```
{
    "orderId": 123456,
    "customer": {
        "custId": "xyz",
        "name": "Acme Inc.",
        "contact": "John Doe",
        "address": "123 Main Street, Anytown, ST 12345"
    },
    "line_items": [
        { "sku": "789790", "description": "Widget", "qty": 5, "unitPrice": 26.87 },
        { "sku": "3533", "description": "Gizmo", "qty": 38, "unitPrice": 89.30 }
    ],
    "payment": { "type": "Amex", "last5": "45678", "expiry": "12/2018" }
}
```

"Hackolade Studio Tutorial 1: Beginner". Hackolade. Disponível em: <https://community.hackolade.com/slides/slide/part-2-overview-of-json-and-json-schema-4> e <https://hackolade.com/help/OverviewofJSONandJSONSchema.html>