

Distribuição e Consistência em NOSQL

Modelos sem Esquemas

- Diferentemente dos bancos de dados relacionais, os NOSQL não utilizam esquemas para a construção dos modelos de dados.
 - Chave-valor: É possível armazenar qualquer valor associado a uma chave. Pares chave-valor podem ter valores com estruturas completamente diferentes.
 - Documentos: A estrutura de um documento não é rígida, permitindo que documentos diferentes tenham estruturas diferentes.
 - Famílias de Colunas: É possível armazenar diferentes dados sob qualquer coluna.
 - Grafos: Novas arestas e propriedades podem ser adicionadas livremente.

Modelos sem Esquemas

Prós

- Permite alterar o armazenamento de dados a medida que o projeto evolui.
- Facilita a inclusão ou exclusão de novos dados, sem necessidade de manutenção.
- Garante maior flexibilidade para dados não uniformes.

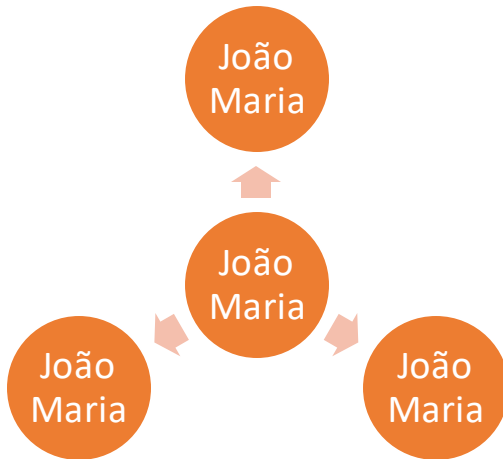
Contras

- Banco de dados desconhece o esquema e não tem como validar a consistência dos dados.
- A clareza do modelo fica mais dependente da forma como o código da aplicação está estruturado.

Modelos de Distribuição

Replicação

- Copia os dados completos em múltiplos nós.



Fragmentação (Sharding)

- Coloca dados diferentes em nós diferentes.



Modelos de Distribuição

- Servidor Único
 - A primeira e mais simples opção de distribuição: não distribuir.
 - Apesar do NOSQL ter um foco grande em clusters, pode fazer sentido adotar tecnologias NOSQL em instalação de servidor único.



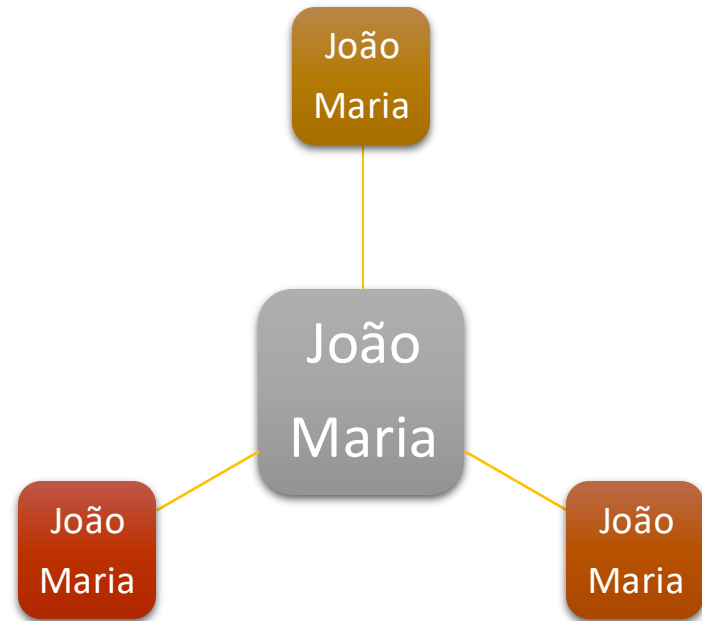
Modelos de Distribuição

- Fragmentação
 - Partes diferentes dos dados são colocadas em servidores (nós) diferentes.
 - Útil para escalar horizontalmente o acesso de múltiplos usuários a porções diferentes de dados.
 - Consegue escalar tanto às leituras quanto as gravações.
 - Se utilizada isoladamente, reduz a resiliência.



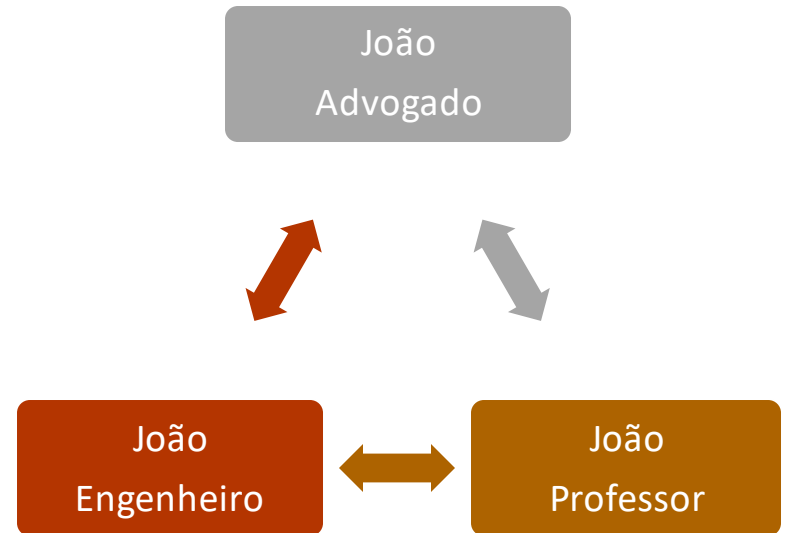
Modelos de Distribuição

- Replicação Mestre-Escravo
 - Um nó é designado como mestre (ou primário), os demais são considerados escravos (ou secundários).
 - As operações de escrita são realizadas somente no nó primário e as de leitura em qualquer nó.
 - Esse modelo é mais escalável em ambientes de leitura intensa e pouca gravação.
 - Pode ser utilizada como estratégia de contingência.



Modelos de Distribuição

- Replicação Ponto-a-Ponto
 - Não existe um mestre e todas as réplicas têm peso igual.
 - Todas podem receber gravações e a perda de uma não impede o acesso aos dados.
 - Conflitos de gravação podem ocorrer e precisam ser tratados.



Consistência

- Abordagens otimista e pessimista:
 - Pessimista
 - Evita que os conflitos de concorrência ocorram
 - Estratégias comuns: serialização e bloqueios
 - Otimista
 - Permite que a ocorrência de conflitos de concorrência, mas detecta e atua para resolver os conflitos
 - Estratégias comuns: atualização condicional, controle de versões
- Programação concorrente envolve a busca de um equilíbrio entre consistência e performance.

Consistência

- Consistência Lógica
 - Garantir que diferentes itens de dados façam sentido juntos.
 - Nos bancos de dados relacionais a consistência lógica é reforçada através de transações (propriedades ACID)
 - No mundo NOSQL
 - Agregados garantem atualização atômica dentro de um único agregado. A consistência lógica entre agregados não é garantida
 - Bancos de dados orientados a grafos asseguram as propriedades ACID, da mesma forma que os bancos relacionais

Consistência

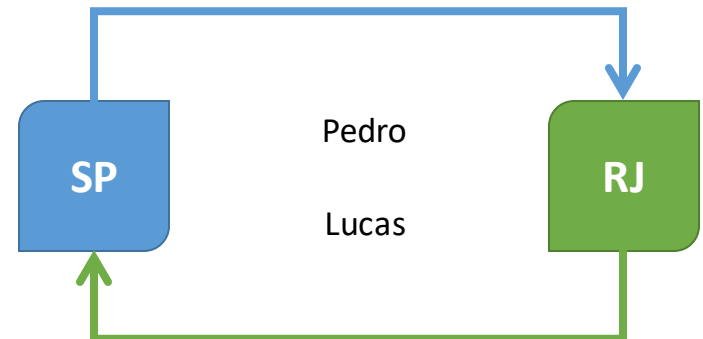
- Consistência de Replicação
 - Garantir que um item de dados tenha o mesmo valor quando lido a partir de diferentes réplicas
 - Ambientes replicados são considerados eventualmente consistentes, o que significa que quando uma atualização é realizada, mas não foi ainda totalmente propagada, os dados são considerados inconsistentes, mas quando as atualizações encerrarem, os dados tendem a alcançar um resultado consistente.

Consistência

- O Teorema CAP
 - Dadas as 3 propriedades, Consistência (C), Disponibilidade (A) e Tolerância a Partições (P), somente é possível obter 2 delas. (BREWER, 2000)

Consistência

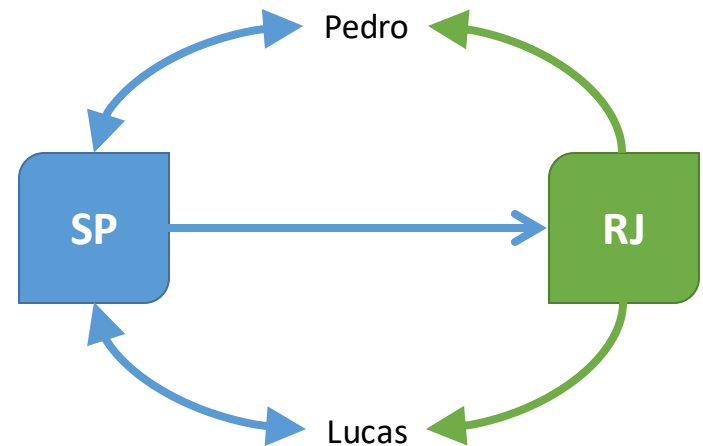
- O Teorema CAP
 - Cenário 1:
 - Reserva de quarto de hotel
 - 2 nós, São Paulo e Rio de Janeiro
 - Replicação ponto-a-ponto
 - Priorizando a consistência: um hotel só realiza a reserva se conseguir confirmar com o outro a disponibilidade de quarto



Distribuição e Consistência

Consistência

- O Teorema CAP
 - Cenário 2:
 - Reserva de quarto de hotel
 - 2 nós, São Paulo e Rio de Janeiro
 - Replicação mestre-escravo
 - Priorizando a disponibilidade: ambos os hotéis realizam as reservas no mestre.



Consistência

ACID

- Atomicity
- Consistency
- Isolation
- Durability

BASE

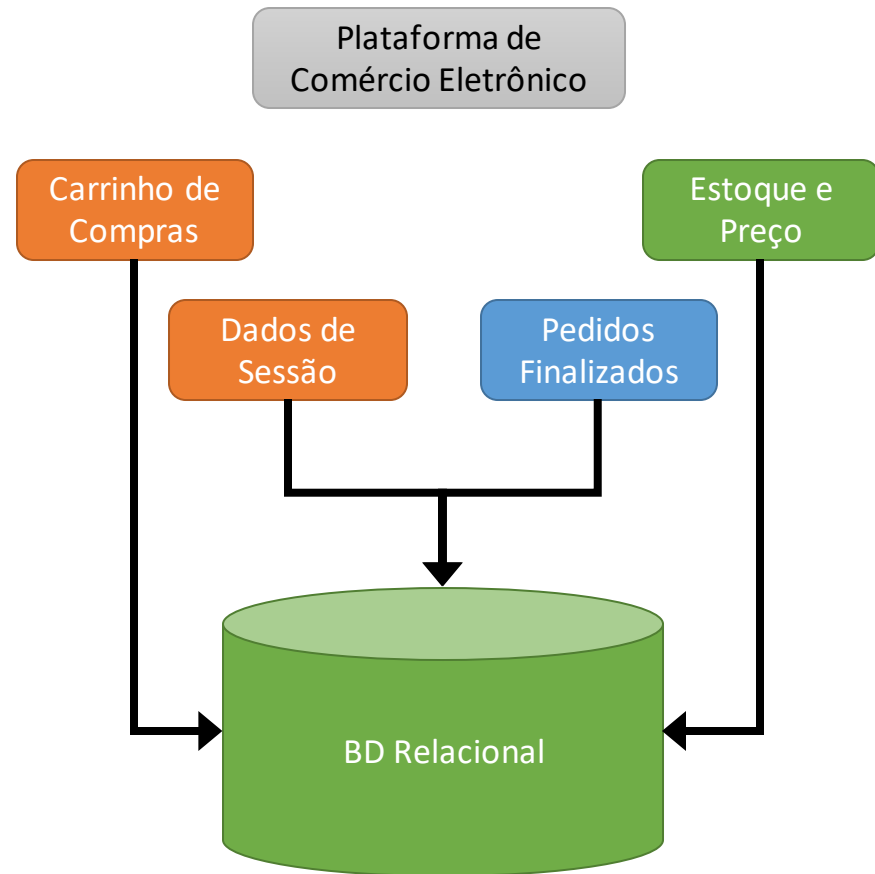
- Basically Available
- Soft state
- Eventual consistency

Persistência Poliglota

- Abordagem que parte do princípio de que cada bancos de dados é projetado para resolver um conjunto específico de problemas e que a adoção de um único banco de dados como solução para todos os problemas tende a alcançar um resultado limitado.

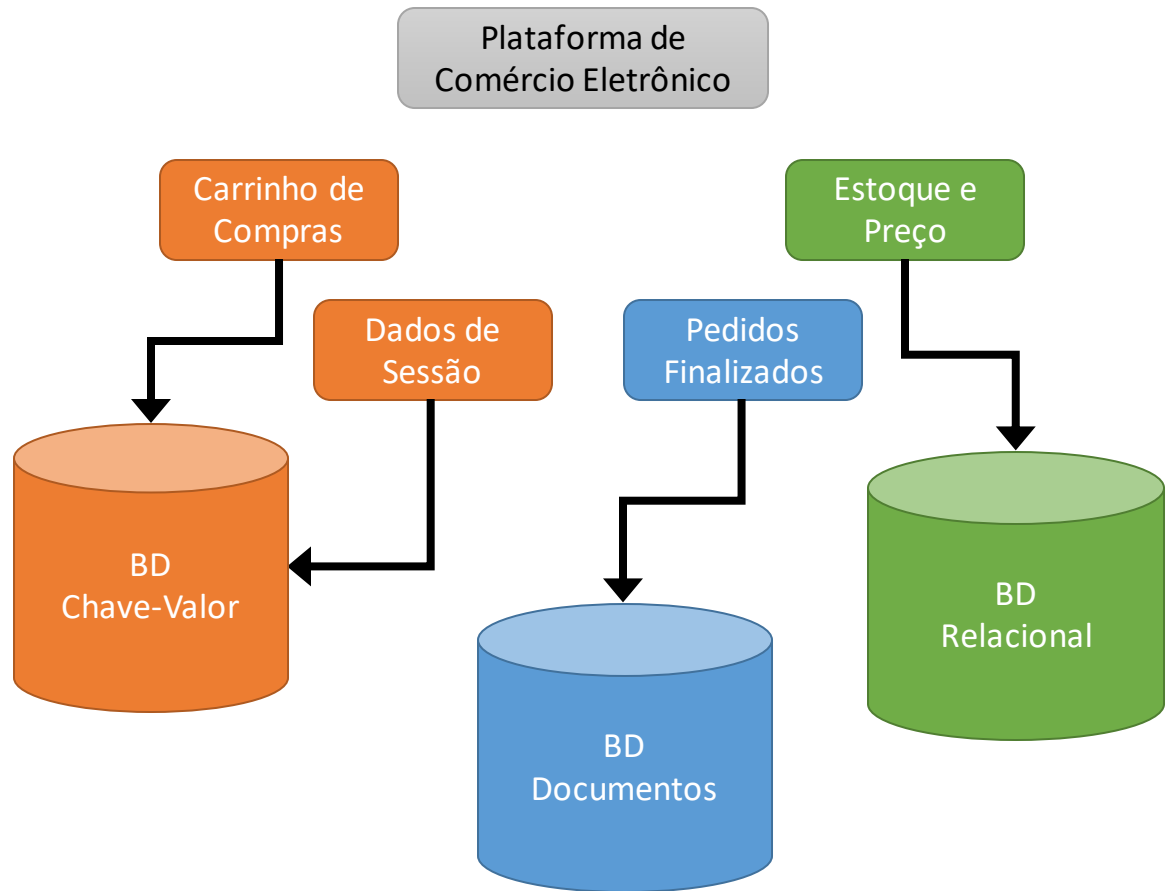
Cenário 1:

BD Relacional como solução única



Cenário 2:

Diversificação do armazenamento de dados



Cenário 3:

Ampliação do cenário para incluir perfil social

