

## Aula 1

### Banco de Dados NoSQL

Prof. Alex Mateus Porn

### Conversa Inicial

1

2

1.1 BANCO DE DADOS  
RELACIONAL OU NOSQL?

TEMA 1 – FUNDAMENTOS  
DE SISTEMA DE BANCO  
DE DADOS NOSQL

2.1 TEOREMA CAP

TEMA 2 – CARACTERÍSTICAS  
RELACIONADAS AOS BANCOS  
DE DADOS DISTRIBUÍDOS E  
SISTEMAS DISTRIBUÍDOS

FUNDAMENTOS DE  
BANCO DE DADOS  
NOSQL

TEMA 3 – MODELOS DE  
DADOS E LINGUAGENS  
DE CONSULTA

TEMA 5 – MODELOS  
DE DISTRIBUIÇÃO  
E CONSISTÊNCIA

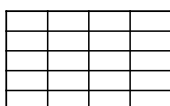
TEMA 4 – MODELOS DE  
DADOS AGREGADOS

### Fundamentos de sistemas de banco de dados NoSQL

3

4

### Modelo relacional

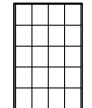


versus

### NoSQL



Chave-valor



Família de  
Colunas



Documentos



Grafos

### E agora?

### Banco de Dados Relacional ou NoSQL?

- Sadalage e Fowler (2019, p. 37) enfatizam que é preciso primeiro compreender que dados queremos armazenar e como queremos manipular
- NoSQL = desempenho de processamento
- Postagem de comentários no fórum
- Arquitetura híbrida

5

6

Banco de Dados Relacional	Banco de Dados NoSQL
<ul style="list-style-type: none"> <li>• Aplicações centralizadas, por exemplo, ERP, CRM.</li> <li>• Requerem alta disponibilidade, quando necessário.</li> <li>• Dados são gerados em velocidades moderadas.</li> <li>• Dados gerados a partir de poucas fontes.</li> <li>• Dados estruturados.</li> <li>• Transações complexas.</li> <li>• Moderado volume de dados.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicações descentralizadas, por exemplo, Web, Mobile, Big Data, Iot.</li> <li>• A disponibilidade precisa ser contínua, sem interrupção.</li> <li>• Dados gerados em alta velocidade, por exemplo, sensores.</li> <li>• Dados gerados a partir de múltiplas fontes, semi ou não estruturados.</li> <li>• Transações simples.</li> <li>• Alto volume de dados.</li> </ul>

7

## Características relacionadas aos bancos de dados

8

## Características relacionadas aos bancos de dados distribuídos e sistemas distribuídos

- Disponibilidade, para o compartilhamento dos dados e escalabilidade, em virtude do contínuo aumento do volume destas bases
- Escalabilidade horizontal
  - O sistema distribuído é expandido, adicionando nodos para armazenamento e processamento de dados à medida que o volume de dados aumenta



9

## Teorema CAP

- Manter consistência após a execução de uma operação
- Disponibilidade do sistema – mantê-lo em operação
- Sistema em funcionamento mesmo que por falhas ocorra a partição de nodos

10



11

## Modelos de dados e linguagens de consulta

12

### Modelos de dados

- Não utilizam esquema
- Vantagem
  - Dados semiestruturados, aceitando novos dados sem realizar alterações na estrutura do banco
- Desvantagem
  - Não há integridade de dados



Mariyani Sugianto/Shutterstock

### Linguagem de consulta

- Não podem realizar consultas com tantas condições e restrições

13

14

### Modelos de dados agregados

- Modelo em que o banco organiza os seus dados
- Categorias chave-valor, documento e família de colunas

15

16

#### Modelo relacional para um website de comércio eletrônico

Cliente		Pedido		
Id	Nome	Id	ClienteId	EnderecoEntregaId
1	Martin	99	1	77

Produto		EnderecoCobranca		
Id	Nome	Id	ClienteId	EnderecoCobranca
27	NoSQL Distilled	55	1	77

ItemPedido				Endereco	
Id	PedidoId	ProdutoId	Preco	Id	Cidade
100	99	27	32.45	77	Chicago

PagamentoPedido				
Id	PedidoId	NumeroCartao	EnderecoCobrancaId	txnId
33	99	1000-1000	55	abelif879rft

17

#### Agregação de dados para um website de comércio eletrônico

```
//em Clientes
{
  "id": 1,
  "nome": "Martin",
  "enderecoCobranca": [{"Cidade": "Chicago"}]}

//em Pedidos
{
  "id": 99,
  "clienteId": 1,
  "itemPedido": [
    {
      "produtoId": 27,
      "preco": 32.45,
      "produtoNome": "NoSQL Distilled"
    }
  ],
  "enderecoEntrega": [{"Cidade": "Chicago"}],
  "pagamentoPedido": [
    {
      "numeroCartao": "1000-1000",
      "txnId": "abelif879rft",
      "enderecoCobranca": {"Cidade": "Chicago"}
    }
  ]
}
```

Fonte: Sadalage e Fowler, 2019, p. 45

18

- Não há uma regra universal para determinar as agregações, depende de como os dados serão manipulados
- Uma única agregação do exemplo anterior indica que todos os pedidos do cliente serão acessados ao mesmo tempo
- A agregação em duas partes possibilita acessar um pedido por vez

19

- Composto por um conjunto de objetos relacionados que são tratados como uma unidade, facilitando a execução no cluster
- Quanto mais agregados estiverem os dados, menor será o número de nodos, melhorando a execução de dados pelo cluster

20

### Modelos de distribuição e consistência

21

### Modelo de distribuição

- Não ~~use~~ distribuição!
- Um único servidor é mais facilmente gerenciado e menos complexo para os desenvolvedores
- Usar quando:
  - Escalabilidade
  - Melhoria de desempenho

22

- Quanto mais agregado estiverem os dados, menor será o número de nodos, melhorando a execução de dados pelo *cluster*

23

- Distribuição
  - Fragmentação  
Copia os mesmos dados em múltiplos nodos
  - Replicação  
Coloca dados diferentes em nodos diferentes

24

- Ambas as técnicas podem ser usadas de forma isolada ou em conjunto

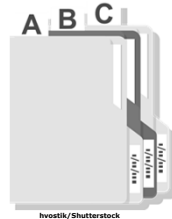
25

## Fragmentação

- Escalabilidade horizontal, direcionando o acesso a diferentes partes de dados em diferentes servidores
- Cada servidor desempenha o papel de gerenciar um subconjunto de dados, que representa um fragmento, o qual lê e grava seu próprio dado

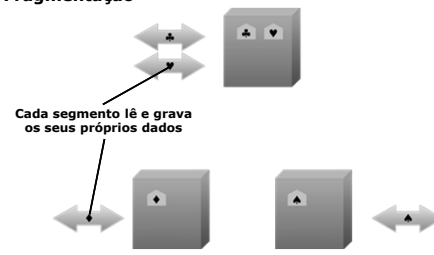
26

- O modelo de dados deve ser projetado com a agregação mais adequada possível, possibilitando ao servidor encontrar a maior parte dos dados em um nodo



27

## Fragmentação



Fonte: Elaborado com base em Sadalage e Fowler, 2019, p. 75

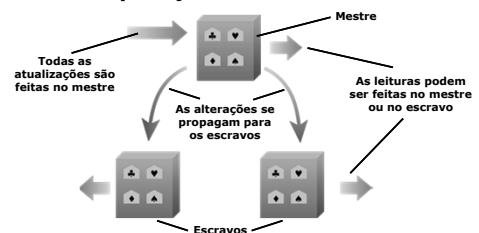
28

## Replicação

- Copia os dados para múltiplos servidores, permitindo acesso aos dados em múltiplos lugares
- Diminui o tempo de recuperação de informações e propicia escalabilidade
- Replicação
  - Mestre-escravo e ponto a ponto

29

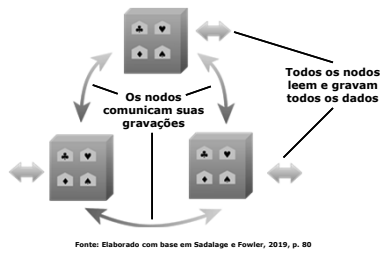
## Replicação mestre-escravo



Fonte: Elaborado com base em Sadalage e Fowler, 2019, p. 78

30

### Replicação ponto a ponto



### Consistência

- Teorema CAP
  - Distribuído = tolerância a partição
- Consistência eventual
  - Criar recursos programáveis para estabelecer a consistência dos dados
- Conflitos de gravação e de leitura-gravação
- Controle de concorrência pessimista e otimista

### Referências

- SADALAGE, P. J. ; FOWLER, M. NoSQL Essencial: Um guia conciso para o Mundo emergente da persistência poliglota. 1. ed. São Paulo: Novatec, 2019.