

Aula 2

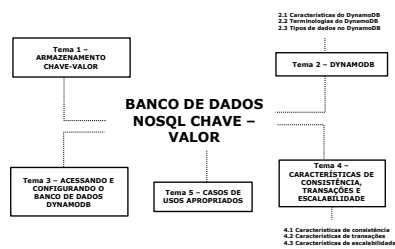
Banco de Dados NoSQL

Prof. Alex Mateus Porn

1

Conversa Inicial

2



3

Armazenamento chave-valor

4

- Utilização de pares de chave e valor para armazenar as informações
- Chaves do tipo String
- Valores
 - Qualquer tipo de dado
- Não possuem esquema predefinido

5

- Exemplo
 - Universidade: "Instituição de Ensino"
- Podem tanto persistir os dados em um banco quanto armazená-los temporariamente em memória

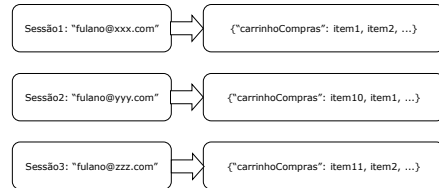
6

Exemplificando

- Carrinho de compras de um comércio eletrônico
- Os clientes acessam o catálogo de produtos
- Selecionam os itens que desejam
- Inserem-nos no carrinho
- A aplicação precisa guardar estes itens até que o cliente finalize suas compras



7



Fonte: Elaborado com base em Marquesone, 2017, p. 45

8

Simplificando



VOYSLA/SHUTTERSTOCK

9

Definindo

“A chave é um identificador único, associado a um item de dados e é usada para localizar esse item de dados rapidamente” (Elmasri, 2018, p. 807)

“O valor é o termo usado para se referir ao objeto de dado armazenado, o qual corresponde a uma estrutura de dados aleatória, o que possibilita armazenar desde cadeia de bytes a listas variadas de dados” (Elmasri, 2018, p. 807)

10

Limitações

- Como não há uma definição do tipo de dado, não é possível fazer uma indexação para cada campo
- Não é possível a geração de consultas mais complexas
- A única forma de realizar consultas é por meio da chave
- Processos de inclusão, edição e exclusão também ocorrem por meio da chave

11

Modelos de bancos de dados chave-valor

- DynamoDB
- CouchBase
- Riak
- Redis
- Memcached

DynamoDB	Redis
<ul style="list-style-type: none"> • Altamente disponível • Mantém a durabilidade dos dados • Totalmente gerenciado na nuvem por uma estrutura <i>serverless</i> • Encarece à medida que há alto processamento de dados 	<ul style="list-style-type: none"> • Mantém os pares chave-valor na memória • Permite acesso rápido aos dados e bom desempenho • Sacrifica a durabilidade dos dados • Necessita de gerenciamento e configuração de um servidor

12

DynamoDB

13

- Banco de dados NoSQL orientado a chave-valor e documentos
- De propriedade da Amazon Web Services (AWS)
- Totalmente gerenciado na nuvem (*serverless*)
- O custo é atribuído ao seu uso
- Disponível globalmente



14

Características do DynamoDB

- Capacidade de leitura e gravação adaptável, de acordo com as necessidades de uso
- Alta escalabilidade, o que permite escalar o banco de dados de forma horizontal, sem a preocupação com o provisionamento de servidores
- *Backup* e recuperação sob demanda
- Recurso TTL (*Time to Live*), que permite ao administrador do banco definir o tempo em que o dado permanecerá no banco de dados

15

Controle de acesso e segurança do DynamoDB

- Acesso via usuário ou grupos de usuários
- Acesso via chaves privadas
- Definição de quais tipos de dados cada usuário pode acessar
- Criptografia dos dados armazenados
- Definição de métricas de utilização e operacionais

16

Formas de acesso ao DynamoDB

- Acesso pelo AWS Console
- Acesso pelo AWS CLI (Command Line Interface)
- Acesso pelo AWS SDK
- Acesso às APIs do DynamoDB – Python, Ruby, Java etc.

17

Terminologias do DynamoDB

Modelo relacional	DynamoDB
• Database	• Não existe
• Tabelas	• Tabelas
• Chave Primária	• Chave Primária
• Chave Estrangeira	• Não existe
• Linhas	• Itens
• Colunas	• Atributos

18

Exemplificando

DynamoDB chave-valor

Modelo relacional

Pedido	Clienteld	Produtoid	Total
0	12345	568	100,00
1	54321	548	75,00
4	32145	345	10,00
3	43523	214	36,00

```
1- {
2-   "Pedido": "0",
3-   "Clienteld": "12345",
4-   "Produtoid": "568",
5-   "Total": 100.00
6- }
7-
8- {
9-   "Pedido": "1",
10-  "Clienteld": "54321",
11-  "Produtoid": "548",
12-  "Total": 75.00
13- }
14-
15- {
16-   "Pedido": "2",
17-   "Clienteld": "32145",
18-   "Produtoid": "345",
19-   "Total": 10.00
20- }
21-
22- {
23-   "Pedido": "3",
24-   "Clienteld": "43523",
25-   "Produtoid": "214",
26-   "Total": 36.00
27- }
```

Definindo a chave primária

DynamoDB chave-valor

Chave primária

Pedido	Clienteld	Produtoid	Total
0	12345	568	100,00
1	54321	548	75,00
4	32145	345	10,00
3	43523	214	36,00

Modelo relacional

```
1- {
2-   "Pedido": "0",
3-   "Clienteld": "12345",
4-   "Produtoid": "568",
5-   "Total": 100.00
6- }
7-
8- {
9-   "Pedido": "1",
10-  "Clienteld": "54321",
11-  "Produtoid": "548",
12-  "Total": 75.00
13- }
14-
15- {
16-   "Pedido": "2",
17-   "Clienteld": "32145",
18-   "Produtoid": "345",
19-   "Total": 10.00
20- }
21-
22- {
23-   "Pedido": "3",
24-   "Clienteld": "43523",
25-   "Produtoid": "214",
26-   "Total": 36.00
27- }
```

Tipos de dados suportados

- Scalar Type
- Set Type
- Document Type

Tipo de dados Scalar Type

Scalar Type	String	Aceita texto no formato UTF8 Não aceita valores vazios Exemplo: "Instituição de Ensino", "EaD", "Graduação"
	Número	Aceita números positivos ou negativos Exemplo: 127, 12.56, -18.50, 0, 298
	Boolean	Aceita somente os valores verdadeiro ou falso Como o String, não aceita valores vazios
	Binário	Dados como imagens, vídeos, documentos (dados binários BLOB)
	Null	Tipo de dados indefinido ou desconhecido

Tipo de dados Set Type

- Conjunto de valores do tipo Scalar Type
 - Conjuntos de dados suportados pelo tipo Set Type
 - Strings: ["Instituição de Ensino", "Educação a distância", "Curitiba", "Paraná"]
 - Números: [2020, 109.20, -45, -29, 75, 0, 192]
 - Boolean: [true, true, false, true, false, false, true]
- (...)

(...)

- Conjunto de dados não suportado pelo tipo Set Type
 - ["Instituição de Ensino", 2020, true, "Curitiba", 192, false]

Tipo de dados Document Type

Conjunto de valores do tipo Document Type

- Dados suportados pelo tipo Document Type
 - ✓ ["Instituição de Ensino", 2020, true, "Curitiba", 192, false]
 - ✓ []
 - ✓ {"nome": "Instituição de Ensino"}
- Dados não suportados pelo tipo Document Type
 - ✓ [" ", " "]
 - ✓ {"nome": " " }

Exemplificando

```
1 "produtoId": "568",
2 "imagem": "dGhpicyB0ZXh0IGlzIGJhc2U2NC1lbmNvZGVk",
3 "nome": "Feijão",
4 "marca": "Xyz",
5 "tipo": ["preto", "branco"],
6 "fornecedor": {
7   "nome": "Abdce",
8   "endereco": ["Av. João XXIII", 135]
9 }
```

Set

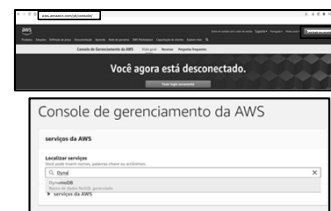
Scalar

Document

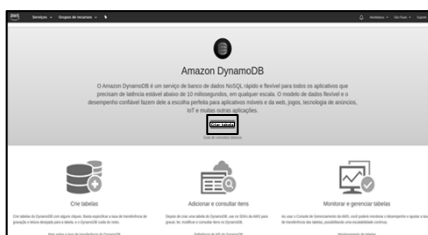
Acessando e configurando o banco de dados DynamoDB

Acessando o DynamoDB

Acesse: <<https://aws.amazon.com>>



Criação de tabelas no DynamoDB





31

Características de consistência, transações e escalabilidade

32

Características de consistência

- **Eventualmente Consistente (EC)**
 - Leitura dos dados realizada nos dados armazenados em cache
- **Fortemente Consistente (FC)**
 - Leitura dos dados realizada diretamente nas tabelas do DynamoDB

33

Características de transações

- **Capacidade de leitura**
 - Definida em RCU (Read Capacity Unit)
- **Capacidade de gravação**
 - Definida em WCU (Write Capacity Unit)

34

- 1 RCU FC = 1 leitura por segundo em uma tabela
- 1 RCU EC = 2 leituras por segundo na tabela
 - Pelo fato de a leitura ser realizada no cache e não diretamente na tabela
- 1 RCU = 1 bloco de 4 Kb
 - Em cada leitura por segundo, é lido um bloco de 4 Kb de dados
- 1 WCU = 1 gravação em uma tabela
- 1 WCU = 1 bloco de 1 Kb
 - Em cada gravação por segundo, é gravado um bloco de 1 Kb de dados

35

Exemplificando

- **Tabela Filmes**
 - 5 RCU

$$\left[\begin{array}{l} \text{Capacidade de Leitura (RCU) da tabela Filmes} \\ \text{Eventualmente Consistente} \\ 4\text{Kb (bloco de leitura)} \times 5 \text{ RCU} = 20 \text{ Kb/seg} \\ \\ 5 \text{ RCU} \times 2 \text{ tabelas} = 40 \text{ Kb/seg} \\ \text{Fortemente Consistente} \\ 4\text{Kb} \times 5 \text{ RCU} = 20 \text{ Kb/seg} \end{array} \right.$$

36

Capacidade de Gravação (WCU) da tabela Filmes

$$1 \text{ Kb (bloco de gravação)} \times 5 \text{ WCU} = 5 \text{ Kb/seg}$$

37

Análise da capacidade de provisionamento

Provisionamento necessário para gravação do registro

$$1,2 \text{ Kb} / 1 \text{ Kb (bloco de gravação)} = 1,2 \text{ RCU} = 2 \text{ RCU}$$

38

Provisionamento necessário para leitura do registro Eventualmente Consistente

$$1,2 \text{ Kb} / 4 \text{ Kb} = 0,3$$

$$0,3 / 2 \text{ tabelas} = 0,15 \text{ RCU} = 1 \text{ RCU}$$

Fortemente Consistente

$$1,2 \text{ Kb} / 4 \text{ Kb (bloco de leitura)} = 0,3 \text{ RCU} = 1 \text{ RCU}$$

39

Características de escalabilidade

- Escalabilidade para cima sempre que necessário
- Escalabilidade para baixo somente até quatro vezes ao dia

40

Casos de uso apropriados

41

- Aplicações com grandes quantidades de dados e volumes extremamente altos de mudanças de estado
- Carrinhos de compras em sites de comércio eletrônico
- Aplicativos orientados por sessão

42

Referências

- AWS. Serverless Application Repository: Descubra, implante e publique aplicativos sem servidor. AWS, c2020. Disponível em: <<https://aws.amazon.com/pt/serverless/serverlessrepo>>. Acesso em: 27 jan. 2021.
- ELMASRI, N. Sistemas de banco de dados. 7. ed. São Paulo: Pearson, 2018.
- MARQUESONE, R. Big Data: Técnicas e tecnologias para extração de valor dos dados. São Paulo: Casa do Código, 2017.