

Aula 3

Banco de Dados NoSQL

Prof. Alex Mateus Porn

Conversa Inicial

1.1-MODELAGEM DE DADOS
ORIENTADA A DOCUMENTOS

2.1-DOCUMENTOS E TIPOS DE DADOS
SUPPORTADOS
2.2-INSTALANDO E ACESSANDO O MONGODB
2.3-CONFIGURANDO O MONGODB

TEMA 1 - SISTEMAS
NOSQL BASEADOS EM
DOCUMENTOS

TEMA 2 - MONGODB

BANCO DE DADOS NOSQL ORIENTADO A DOCUMENTOS

3.1 - OPERAÇÃO INSERT
3.2 - OPERAÇÃO FIND
3.3 - OPERAÇÃO UPDATE
3.4 - OPERAÇÃO DELETE

TEMA 3 - OPERAÇÕES
CRUD NO MONGODB

TEMA 4 - CARACTERÍSTICAS DE
CONSISTÊNCIA, TRANSAÇÕES E
DISPONIBILIDADE

4.1 - CARACTERÍSTICAS DE
TRANSAÇÕES
4.2 - CARACTERÍSTICAS DE
DISPONIBILIDADE
4.3 - CARACTERÍSTICAS DE
CONSISTÊNCIA

TEMA 5 - CASOS DE
USOS APROPRIADOS

Sistemas NoSQL baseados em documentos

Armazenamento baseado em documentos

- Considerado como uma extensão dos bancos de dados orientados a chave-valor
- Não se aplicam critérios de normalização de dados
- A criação de joins e esquemas é inexistente
- Os documentos são estruturas flexíveis, como por exemplo, XML, JSON e BSON

Exemplo

```
1- {  
2-   "clientes": [  
3-     {  
4-       "nome": "Cliente X",  
5-       "dataNascimento": "25/03/1985",  
6-       "endereco": "Rua das Avenidas, 290",  
7-       "telefone": "(42) 3542-9898",  
8-       "celular": "(42)-99999-9898"  
9-     },  
10-    {  
11-      "nome": "Cliente Y",  
12-      "nascimento": {  
13-        "dia": 23,  
14-        "mes": 08,  
15-        "ano": 1985  
16-      },  
17-      "endereco": "Avenida das Velas, 275",  
18-      "telefone": "(42) 3542-9999",  
19-      "contato": {  
20-        "celular": "(42) 98888-8989",  
21-        "email": "clientey@xyxy.com"  
22-      }  
23-    }  
24-  ]  
25- }
```

Exemplificando

- Imaginemos um documento como sendo uma linha da tabela e um conjunto de documentos como sendo a tabela com todos os registros



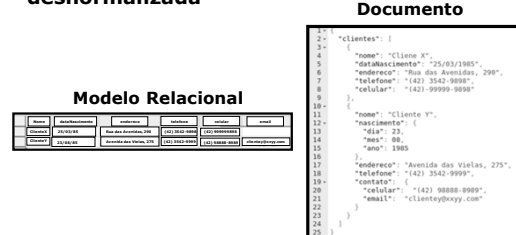
Características

- Permitem a criação de consultas e filtros sobre qualquer valor armazenado
- Os dados são armazenados de forma desnormalizada, proporcionando escalabilidade e velocidade de leitura
- Não fornecem relacionamentos entre os documentos
- Mantém os dados integrados ao próprio documento, ao invés de áreas de armazenamento separadas

Exemplificando

| Modelo Relacional | Orientado a Documentos |
|-------------------|------------------------|
| Esquema | Não Existe |
| Tabelas | Documentos |
| Chave Estrangeira | Não Existe |
| Relacionamentos | Não Existe |
| Linhas | Registros |
| Colunas | Atributos |

- Os dados são armazenados de forma desnormalizada



Exemplificando

Documento de posts e comentários

```

1 {
2   "Post": {
3     "nome": "Postagem X",
4     "comentarios": ["Comentário 1", "Comentário 2"]
5   },
6   "nome": "Postagem Y",
7   "comentarios": ["Comentário 1", "Comentário 2"]
8 }

```

Tabelas de posts e comentários



- Não tem relacionamentos entre os documentos
- Mantém os dados integrados ao próprio documento, ao invés de áreas de armazenamento separadas

Modelagem de dados

- Modelagem incorporada (embedded)
 - Estrutura não normalizada, onde os dados normalmente são acessados juntos, como se fosse um documento dentro do outro (Monteiro, 2019)
- Modelagem referenciada
 - Os documentos são armazenados separadamente, mas um dos documentos tem a referência para o outro, de modo a minimizar a quantidade de dados duplicados, aumentar o desempenho nas operações de escrita, porém diminuir o desempenho nas consultas (Monteiro, 2019)

Exemplificando

- Modelagem incorporada (embedded)

```
1 {
2   "Codigo": 1,
3   "nome": "Usuário X",
4   "contato": {
5     "fone": "(42) 98889-8989",
6     "email": "usuario@yyxx.com"
7   },
8   "acesso": {
9     "nivel": 5,
10    "grupo": "desenvolvimento"
11  }
12 }
```

Subdocumento incorporado

Subdocumento incorporado

13

Modelagem referenciada

```
1 {
2   "Codigo": 1,
3   "nome": "Usuário X"
4 }
```

```
1 {
2   "Codigo": 2,
3   "idNome": 1,
4   "fone": "(42) 98889-8989",
5   "email": "usuario@yyxx.com"
6 }
```

```
1 {
2   "Codigo": 3,
3   "idNome": 1,
4   "nivel": 5,
5   "grupo": "desenvolvimento"
6 }
```

14

MongoDB

15

- Banco de dados NoSQL orientado a documentos
- Com base na definição de coleções e índices
- Diferentes tipos de documentos podem coexistir em uma mesma coleção

```
1 {
2   "tipo": "Livro",
3   "titulo": "Introdução ao MongoDB",
4   "isbn": "978-85-7522-422-9",
5   "editora": "Novatec",
6   "autor": [
7     "Howe, David",
8     "Plugging, Belcar",
9     "Membrey, Peter"
10  ]
11 }
```

```
1 {
2   "tipo": "CD",
3   "artista": "Beverly",
4   "titulo": "Beverly",
5   "genero": "Pop",
6   "dataLancamento": "04/06/2002",
7   "listasMusicas": [
8     "musica": "1",
9     "titulo": "Smells Like Teen Spirit",
10    "duracao": "3:30"
11  ]
12 }
```

```
1 {
2   "tipo": "CD",
3   "artista": "Beverly",
4   "titulo": "Smells Like Teen Spirit",
5   "genero": "Pop",
6   "dataLancamento": "04/06/2002",
7   "listasMusicas": [
8     "musica": "1",
9     "titulo": "Smells Like Teen Spirit",
10    "duracao": "3:30"
11  ]
12 }
```

16

Modelo de um banco de dados MongoDB

Fonte: elaborado com base em Howe, Membrey, Pluggo, 2015.

17

Tipos de dados suportados

| Tipos de dados | Finalidade |
|----------------|------------------------------------------------------------------------------------------------|
| String | Utilizado principalmente para armazenar valores textuais. Exemplo: {"universidade": "XYZ"} |
| Integer | Utilizado para armazenar valores numéricos. Exemplo: {"idade": 42} |
| Boolean | Utilizado para armazenar os valores "verdadeiro" ou "falso". Exemplo: {"praticaEsporte": true} |
| Double | Utilizado para armazenar valores de ponto flutuante. Exemplo: {"peso": 73.5} |
| Arrays | Utilizado para armazenar arrays. Exemplo: {"animais": ["gato", "cachorro", "cavalo"]} |
| Null | Utilizado para armazenar um valor nulo. Exemplo: {"idade": null} |

18

Exemplificando

```

1 {
2   "codigo": 1475,
3   "nome": "Alex",
4   "peso": 65.8,
5   "casado": true,
6   "pets": [
7     "cachorro",
8     "gato"
9   ],
10  "habilitacao": null
11 }

```

Diagram illustrating a JSON document structure with corresponding data types:

- "codigo": 1475 → Integer
- "nome": "Alex" → String
- "peso": 65.8 → Double
- "casado": true → Boolean
- "pets": ["cachorro", "gato"] → Array
- "habilitacao": null → Null

19

Instalação e acesso ao MongoDB

- Possui tanto uma versão online quanto offline
- Download
- www.mongodb.com/try/download/community

Available Downloads

Version: 4.4.0 (current)

Platform: Windows

Package: msi

Download

sudo apt install mongodb

20

Configurando o MongoDB

www.mongodb.com

The database for modern applications

MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era.

No database makes you more productive.

By MongoDB Inc. in the cloud

Start free

21

Criando sua conta para acesso ao MongoDB

Get started free

No credit card required.

Your Company (optional)

How are you using MongoDB? (For learning MongoDB)

Your Work Email

Username

Last Name

First Name

Phone

Address

Agree to the terms of service and privacy policy

Get started free

22

Criando o banco de dados no MongoDB

Create a cluster

Choose your cloud provider, region, and specs.

Build a Cluster

Cloud Provider & Region: GCP: Sao Paulo (southamerica-west1)

Recommended spec (D)

Cluster Tier: M0 Sandbox (Shared RAM, 512 MB Storage)

FREE

23

Visualizando e acessando o banco de dados

Clusters

Cluster0

Version: 4.3.0

CONNECT | METRICS | COLLECTIONS

Cluster Tier: M0 Sandbox (General)

Region: GCP: Sao Paulo (southamerica-west1)

Type: Replica Set - 3 nodes

Linked Realm App: None Linked

Connect to Cluster0

Setup connection security

Whitelist a connection IP address

Create a Database User

Username: root

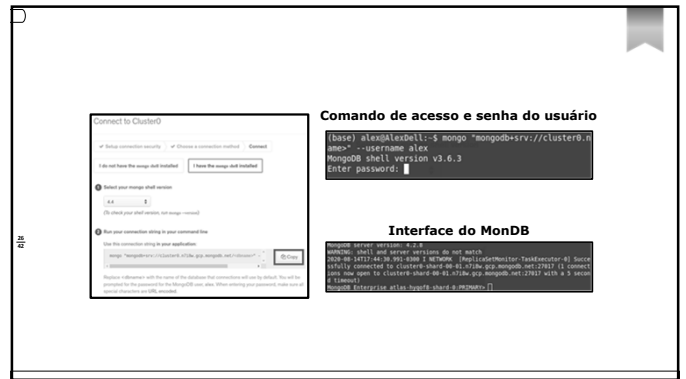
Password: root

Create Database User

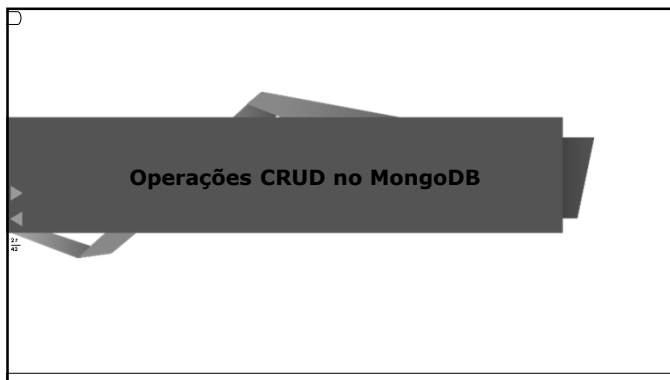
24



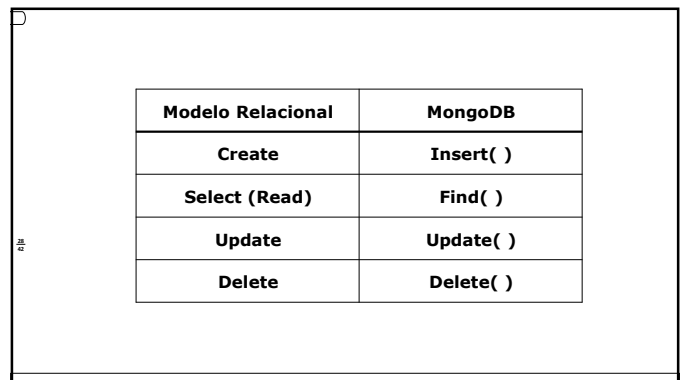
25



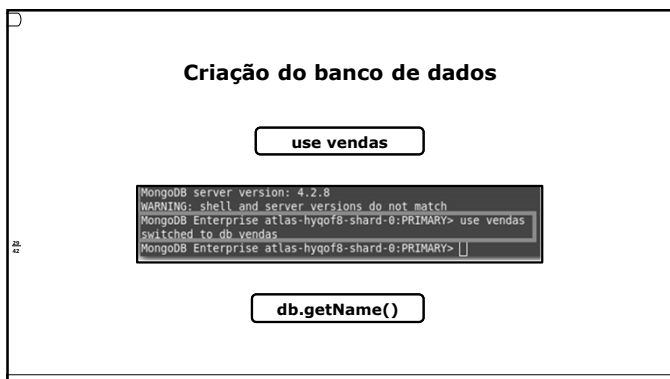
26



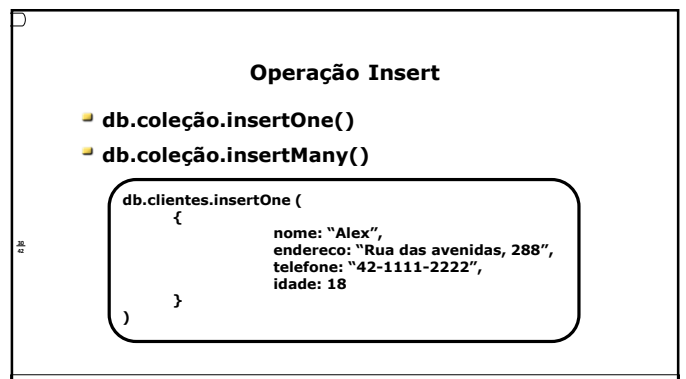
27



28



29



30

Operação Find

- db.coleção.find()

db.clientes.find()

db.clientes.find({idade: 18})

gt (greater than) – maior que;
gte (greater than or equals) – maior ou igual à;
lt (less than) – menor que;
lte (less than or equals) – menor ou igual à.

db.clientes.find({idade: {\$gt: 18}})

31

Operação Update

- db.coleção.updateOne()
- db.coleção.updateMany()

db.clientes.updateOne({nome: "Alex"}, {\$set {idade: 25}})

32

Operação Delete

- db.coleção.deleteOne()
- db.coleção.deleteMany()

db.clientes.deleteMany({idade: 25})

33

Características de consistência, transações e disponibilidade

34

Características de transações

- Operações atômicas para um único documento
- Para atomicidade em vários documentos, o MongoDB oferece suporte a transações de vários documentos

35

Características de disponibilidade

Conforme Boaglio (2015, p. 149-150), alta disponibilidade é a possibilidade de ter os dados replicados em diferentes lugares, denominados como nodos, e se um nodo cair, outro assume o seu lugar, evitando assim que a aplicação deixe de funcionar

- Arquitetura conhecida como replica set
- Novos nodos secundários podem ser adicionados a qualquer instante, sem interromper o cluster inteiro
- Recurso denominado heartbeat para verificar a disponibilidade de cada nodo, onde a cada dois segundos os nodos "conversam" para verificarem se estão ativos

36

Características de consistência

- Garante conformidade com as propriedades ACID a nível de documento
- Se os dados estiverem espalhados em vários documentos, não há garantia dessas propriedades entre os dados

37

Casos de usos apropriados

38

- Para Marquesone (2017, p. 49)
 - Armazenamento de conteúdos de páginas Web
 - Catalogação de documentos de uma empresa
 - Gerenciamento de inventário de um e-commerce
- Conforme AWS (2020)
 - Blogs e plataformas de vídeo

39

Referências

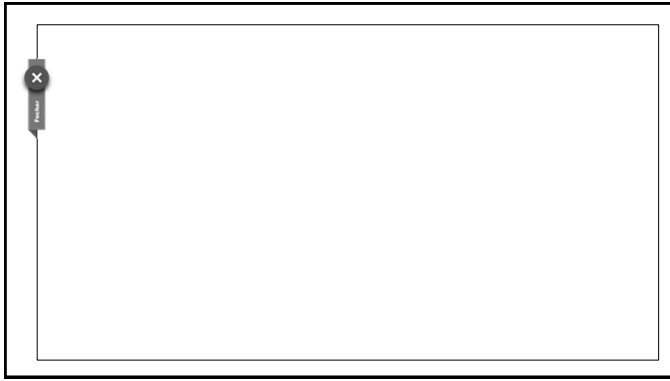
40

- AWS. O que é um banco de dados de documentos? AWS, c2020. Disponível em: <<https://aws.amazon.com/pt/nosql/document>>. Acesso em: 12 ago. 2020.
- BOAGLIO, F. MongoDB: Construa novas aplicações com novas tecnologias. São Paulo: Casa do Código, 2015.
- HOWS, D.; MEMBREY P.; PLUGGE, Eelco. Introdução ao MongoDB. São Paulo: Novatec, 2015.

41

- MARQUESONE, R. Big Data: Técnicas e tecnologias para extração de valor dos dados. São Paulo: Casa do Código, 2017.
- MONTEIRO, D. Introdução para modelagem de dados para banco orientado a documentos. 18 abr. 2019. Disponível em: <<https://imasters.com.br/banco-de-dados/introducao-para-modelagem-de-dados-para-banco-orientado-documentos>>. Acesso em: 12 ago. 2020.

42



43