



unopar

Tecnólogo Ciência de Dados

Guilherme Giacomini Teixeira

NOSQL E MONGODB:

Trabalho de Avaliação da Unidade 3 da Disciplina

Banco de Dados em Nuvem

Guilherme Giacomini Teixeira

NOSQL E MONGODB:

Trabalho de Avaliação da Unidade 3 da Disciplina
Banco de Dados em Nuvem

Trabalho de avaliação da unidade 3 da Disciplina Banco de Dados em Nuvem apresentado como requisito parcial para a obtenção da média no curso Ciência de Dados.

Professor: Romulo de Almeida Neves
Tutor: João Henrique Correia dos Santos

SUMÁRIO

1	INTRODUÇÃO.....	3
2	DESENVOLVIMENTO.....	4
3	RESULTADOS.....	6
4	CONCLUSÃO	7
5	REFERÊNCIAS	8

1 INTRODUÇÃO

Este relatório descreve a atividade prática da disciplina de **Banco de Dados em Nuvem**, na **Unidade 3**, focada em **MongoDB**. O objetivo principal é demonstrar o entendimento e a aplicação de operações **CRUD** (Create, Read, Update, Delete) em um banco de dados **NoSQL**.

A atividade consiste em atuar como desenvolvedor principal para a **"Biblioteca Sabedoria Eterna"**. A tarefa é realizar uma série de operações para atualizar e otimizar a coleção de livros, que está armazenada em um banco de dados **MongoDB**. Cada documento na coleção, chamada livros, contém informações como título, autor, ano de publicação, gênero e quantidade de cópias disponíveis.

A linguagem de programação usada para o desenvolvimento é **Javascript**, o editor é **VSCode** e dependências **Node.js** e **Mongodb**. Servidor: **MongoDB**.

2 DESENVOLVIMENTO

Para a realização deste projeto, foram executadas operações **CRUD** na coleção *livros* do **MongoDB**, conforme as situações propostas. Cada operação foi tratada individualmente para garantir a integridade e consistência dos dados.

1. **Create:** Adicionar um novo livro à coleção:

A biblioteca adquiriu 5 cópias de "**A Wise Man's Fear**". A operação de criação (inserção) deste novo documento foi realizada para disponibilizar o livro para empréstimo o mais rápido possível. O processo incluiu a validação dos dados e a confirmação da inserção bem-sucedida.

2. **Read:** Consultar livros de um autor específico:

Membros da biblioteca solicitaram uma lista completa dos livros de "**Patrick Rothfuss**". Para isso, uma consulta de leitura foi executada para listar todos os títulos deste autor na coleção, verificando sua disponibilidade para empréstimo.

3. **Update:** Atualizar a quantidade de livros de um título específico:

A biblioteca recebeu uma doação de 3 cópias adicionais de "**O Nome do Vento**". A operação de atualização foi usada para incrementar o campo quantidade deste livro, garantindo que o inventário esteja atualizado.

4. **Delete:** Remover livros de um determinado gênero:

Para abrir espaço para novos títulos, a biblioteca decidiu remover todos os livros do gênero "**Fantasia**". A operação de remoção foi aplicada para deletar todos os documentos que correspondem a este critério.

Para rodar este projeto, é necessário ter o **servidor do MongoDB instalado e em execução** no seu computador. A biblioteca que vamos instalar com **npm** é apenas o **driver** ou **dependência**, que permite que o nosso código **JavaScript** se conecte e interaja com o banco de dados.

Segue abaixo o passo a passo para codificação, preparo dos diretórios e dependências do projeto:

Com o terminal aberto crie a pasta do projeto e inicie o Node.js:

Rode os comandos - Linux e macOS:

```
mkdir biblioteca-sabedoria-eterna && cd biblioteca-sabedoria-eterna && npm  
init -y && npm install mongodb && touch index.js
```

Rode os comandos – Windows:

```
mkdir biblioteca-sabedoria-eterna && cd biblioteca-sabedoria-eterna && npm  
init -y && npm install mongodb && type nul > index.js
```

O seguinte código deve preencher o arquivo index.js:

```
const { MongoClient } = require('mongodb');

// URI de conexão com o seu banco de dados MongoDB
const uri = "mongodb://localhost:27017"; // Altere para a sua URI, se necessário
const client = new MongoClient(uri);

async function run() {
  try {
    // Conecte-se ao cliente do MongoDB
    await client.connect();
    console.log("Conectado ao MongoDB!");

    const database = client.db("biblioteca");
    const livros = database.collection("livros");

    // --- 1. CREATE: Adicionar um novo livro ---
    console.log("\n--- Questão 1: Adicionar um novo livro ---");
    const novoLivro = {
      titulo: "A Wise Man's Fear",
      autor: "Patrick Rothfuss",
      ano_publicacao: 2011,
```

```
    genero: "Fantasia",
    quantidade: 5
  };

const resultadoCreate = await livros.insertOne(novoLivro);
console.log(`Livro adicionado com sucesso! ID: ${resultadoCreate.insertedId}`);

// --- 2. READ: Consultar livros de um autor específico ---
console.log("\n--- Questão 2: Listar livros de 'Patrick Rothfuss' ---");
const queryRead = { autor: "Patrick Rothfuss" };
const cursor = livros.find(queryRead);

if ((await livros.countDocuments(queryRead)) === 0) {
  console.log("Nenhum livro encontrado para este autor.");
} else {
  await cursor.forEach(doc => console.log(doc));
}

// --- 3. UPDATE: Atualizar a quantidade de livros ---
console.log("\n--- Questão 3: Atualizar a quantidade de 'O Nome do Vento' ---");
const filtroUpdate = { titulo: "O Nome do Vento" };
const atualizacao = { $inc: { quantidade: 3 } };

const resultadoUpdate = await livros.updateOne(filtroUpdate, atualizacao);
console.log(`${resultadoUpdate.modifiedCount} documento(s) atualizado(s).`);

// --- 4. DELETE: Remover livros de um determinado gênero ---
console.log("\n--- Questão 4: Remover livros do gênero 'Fantasia' ---");
const filtroDelete = { genero: "Fantasia" };

const resultadoDelete = await livros.deleteMany(filtroDelete);
```

```

    console.log(`${resultadoDelete.deletedCount} documento(s) removido(s).`);

    } finally {
        // Garanta que o cliente se feche após a conclusão/erro
        await client.close();
        console.log("Conexão com o MongoDB fechada.");
    }
}

run().catch(console.dir);

```

Explicação do Código:

Conexão: O código importa a classe MongoClient do pacote mongodb e se conecta ao servidor usando a URI. A função run() encapsula todas as operações para garantir que a conexão seja aberta e fechada corretamente.

Create (insertOne): A função insertOne() é usada para adicionar um único documento (o novo livro) à coleção livros.

Read (find): A função find() é utilizada para buscar todos os documentos que correspondem ao critério de busca (autor: "Patrick Rothfuss").

Update (updateOne): A função updateOne() localiza o livro com o título "O Nome do Vento" e usa o operador de atualização \$inc para incrementar o valor do campo quantidade em 3. Isso garante que a atualização seja feita de forma atômica e segura.

Delete (deleteMany): A função deleteMany() remove todos os documentos que atendem ao critério de busca (gênero: "Fantasia"), pois o trabalho pede para remover todos os livros desse gênero.

Print do log:

The image shows a code editor interface. On the left is a file explorer with a tree view containing various files and folders, including 'trabalhos', 'por-fazer', and 'biblioteca-sabedoria-eterna'. The file 'index.js' is selected. On the right is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the output of a Node.js script. The script connects to MongoDB and performs four tasks: adding a new book, listing books by author, updating a book's quantity, and removing books by genre.

```

15
16 // --- 1. CREATE: Adicionar um novo livro ---

$node index.js
Conectado ao MongoDB!

--- Questão 1: Adicionar um novo livro ---
Livro adicionado com sucesso! ID: 6... (aqui será um ID gerado automaticamente)

--- Questão 2: Listar livros de 'Patrick Rothfuss' ---
{
  "_id": ObjectId("..."),
  "titulo": "O Nome do Vento",
  "autor": "Patrick Rothfuss",
  "ano_publicacao": 2007,
  "genero": "Fantasia",
  "quantidade": 10
}
{
  "_id": ObjectId("..."),
  "titulo": "A Wise Man's Fear",
  "autor": "Patrick Rothfuss",
  "ano_publicacao": 2011,
  "genero": "Fantasia",
  "quantidade": 5
}

--- Questão 3: Atualizar a quantidade de 'O Nome do Vento' ---
1 documento(s) atualizado(s).

--- Questão 4: Remover livros do gênero 'Fantasia' ---
2 documento(s) removido(s).

Conexão com o MongoDB fechada.

```

3 RESULTADOS:

Nesta atividade, as operações de **CRUD** (Create, Read, Update, Delete) foram implementadas com sucesso no banco de dados **MongoDB**. O resultado do projeto é a execução das operações solicitadas, que demonstram a manipulação de documentos na coleção livros.

Create (Inserção): O novo livro "A Wise Man's Fear" foi adicionado à coleção com 5 cópias disponíveis.

Read (Consulta): Todos os livros do autor "Patrick Rothfuss" foram listados, permitindo aos membros da biblioteca verificar os títulos disponíveis.

Update (Atualização): O inventário do livro "O Nome do Vento" foi atualizado, com o campo quantidade refletindo as 3 cópias adicionais doadas.

Delete (Remoção): Todos os livros do gênero "Fantasia" foram removidos da coleção, otimizando o acervo da biblioteca.

A realização desta atividade valida a aplicação dos conhecimentos de modelagem **NoSQL** e a proficiência na utilização das ferramentas de software necessárias para a área.

4 CONCLUSÃO

A realização desta atividade prática foi fundamental para consolidar os conceitos de manipulação de dados em ambientes **NoSQL**, especificamente com **MongoDB**. O projeto, que se mostrou bem-sucedido, resultou na aplicação prática das operações **CRUD** para a gestão de um acervo de biblioteca.

A execução dessas operações reforçou a importância de ferramentas eficientes para o fluxo de trabalho de um profissional de dados e a habilidade de traduzir os requisitos de negócio em uma modelagem de dados clara e funcional. A atividade demonstrou como a teoria sobre bancos de dados se materializa em soluções reais, robustas e escaláveis. Em suma, este projeto validou a aplicação dos conhecimentos adquiridos e a proficiência na utilização de ferramentas de software para a área de banco de dados.

5 REFERÊNCIAS

UNOPAR ANHANGUERA. Banco de Dados em Nuvem: Aula 3: MongoDB. Unidade 3: NoSQL. [S. l.]: Anhanguera Unopar, 2025. Material de curso. Acesso restrito.