

Relatório de Projeto: Banco de Dados Não Relacional (NoSQL)

Estudo de Caso Utilizando o MongoDB

Dados do Aluno / Disciplina

Aluno: Luiz Gustavo Alvino da Silva

Disciplina: Banco de Dados II

Matrícula: CJ3025896

Curso: ADS

Instituição de Ensino: IFSP Campos do Jordão

Resumo

Este trabalho tem como objetivo explorar a utilização de bancos de dados não relacionais, com foco no MongoDB, um dos Sistemas de Gerenciamento de Banco de Dados (SGBD) NoSQL mais populares. A pesquisa aborda o conceito de bancos de dados NoSQL, sua importância no cenário atual de grandes volumes de dados, e as principais características que os diferenciam dos bancos de dados relacionais. O projeto desenvolvido utiliza o MongoDB para modelar um sistema de gerenciamento de dados, demonstrando sua flexibilidade e escalabilidade. A conclusão aponta as vantagens do MongoDB, bem como algumas áreas para melhorias.

Introdução

O crescente volume de dados gerados pelas aplicações modernas exige novas abordagens para o armazenamento e recuperação de informações. Os bancos de dados não relacionais (NoSQL) surgiram como uma alternativa aos bancos de dados tradicionais, oferecendo escalabilidade, flexibilidade e alta performance. Este trabalho tem como objetivo apresentar uma análise detalhada dos bancos de dados NoSQL, com foco no MongoDB, um dos mais utilizados no mercado. A introdução abordará as principais características desse tipo de banco de dados, sua evolução e os tipos de modelos de dados que existem. Além disso, será apresentado um estudo de caso utilizando o MongoDB, com a implementação de um sistema simples para demonstrar suas funcionalidades.

Metodologia

A metodologia adotada neste trabalho consiste na pesquisa bibliográfica sobre bancos de dados NoSQL, com ênfase nos modelos de dados e na análise do SGBD MongoDB. O trabalho será dividido em duas etapas principais: primeiro, uma explicação teórica sobre bancos de dados NoSQL e seus modelos de dados, seguida da escolha e implementação do MongoDB como SGBD para o desenvolvimento de um projeto prático.

- **Modelos de Banco de Dados NoSQL**

Os bancos de dados NoSQL podem ser classificados em quatro tipos principais de modelos de dados:

1. **Documentos:** Utilizam documentos como a estrutura de dados principal, sendo o MongoDB um exemplo típico desse modelo.
2. **Chave-valor:** Armazenam dados como pares de chave e valor, sendo simples e eficientes para grandes volumes de dados.
3. **Grafos:** Utilizam grafos para representar relações entre dados, sendo adequados para modelos que envolvem muitas conexões.

4. **Colunas:** Organizam os dados em colunas em vez de linhas, como no Cassandra.

- **Escolha do SGBD MongoDB**

O MongoDB foi escolhido devido à sua popularidade, flexibilidade e alto desempenho em situações de grandes volumes de dados. Ele adota o modelo de banco de dados orientado a documentos, armazenando dados no formato BSON (uma forma binária do JSON). Sua estrutura permite a criação de coleções de documentos, onde cada documento pode ter um formato diferente, tornando o sistema altamente flexível e escalável.



Resultados Obtidos

O projeto desenvolvido consistiu em um sistema de gerenciamento de dados para uma biblioteca, utilizando o MongoDB. O sistema foi projetado para armazenar informações sobre livros, autores e usuários, aproveitando a flexibilidade do modelo orientado a documentos. A seguir, detalho a estrutura do banco de dados, algumas operações realizadas e exemplos de consultas.

Estrutura do Banco de Dados

A estrutura do banco de dados foi modelada utilizando coleções para cada entidade principal:

1. Coleção de Livros

A coleção `livros` armazena os documentos com informações sobre os livros da biblioteca. Cada documento pode ter um título, autor, ano de publicação, gênero e disponibilidade.

Exemplo de documento:

```
json
{
  "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99b7"),
  "titulo": "Introdução ao MongoDB",
  "autor": "John Doe",
  "ano": 2020,
  "genero": "Tecnologia",
  "disponivel": true
}
```

2. Coleção de Autores

A coleção `autores` armazena informações sobre os autores dos livros. Um documento de autor pode conter o nome, data de nascimento, e uma lista de livros escritos. Exemplo de documento:

```
json
{
  "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99c8"),
  "nome": "John Doe",
  "data_nascimento": "1980-04-25",
  "livros": [
    ObjectId("5f8d0c8f0b4c3d3e9c2a99b7")
  ]
}
```

3. Coleção de Usuários

A coleção `usuarios` armazena dados dos usuários da biblioteca, incluindo nome, e-mail e livros que o usuário emprestou. Exemplo de documento:

```
json
{
  "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99d9"),
  "nome": "Maria Silva",
  "email": "maria.silva@email.com",
  "livros_emprestados": [
    ObjectId("5f8d0c8f0b4c3d3e9c2a99b7")
  ]
}
```

Operações Realizadas

Com a estrutura do banco de dados pronta, o projeto inclui operações básicas de inserção, atualização e consulta de dados. A seguir, alguns exemplos de operações realizadas:

1. **Inserção de um Novo Livro** Para inserir um novo livro na coleção `livros`, utilizamos a operação `insertOne` do MongoDB:

```
javascript
db.livros.insertOne({
  "titulo": "Avançando no MongoDB",
  "autor": "Jane Doe",
  "ano": 2022,
  "genero": "Tecnologia",
  "disponivel": true
});
```

2. **Consulta de Livros Disponíveis** Para consultar todos os livros que estão disponíveis na biblioteca, podemos utilizar o comando `find` com uma condição de filtro:

```
javascript
db.livros.find({ "disponivel": true });
```

Exemplo de resultado de consulta:

json

```
{
  "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99b7"),
  "titulo": "Introdução ao MongoDB",
  "autor": "John Doe",
  "ano": 2020,
  "genero": "Tecnologia",
  "disponivel": true
}
```

3. **Consulta de Livros por Autor** Para encontrar todos os livros de um autor específico, podemos usar o comando `find` para buscar por autor. Exemplo de consulta para buscar todos os livros do autor "John Doe":

javascript

```
db.livros.find({ "autor": "John Doe" });
```

Exemplo de resultado de consulta:

json

```
[
  {
    "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99b7"),
    "titulo": "Introdução ao MongoDB",
    "autor": "John Doe",
    "ano": 2020,
    "genero": "Tecnologia",
    "disponivel": true
  }
]
```

4. **Emprestar um Livro para um Usuário** Para registrar o empréstimo de um livro por um usuário, podemos usar a operação `updateOne` para adicionar o livro à lista de livros emprestados do usuário. Exemplo de operação:

javascript

```
db.usuarios.updateOne(
  { "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99d9") },
  { $push: { "livros_emprestados": ObjectId("5f8d0c8f0b4c3d3e9c2a99b7") } }
);
```

Exemplos de Consultas Mais Complexas

1. **Buscar Todos os Livros de Um Gênero Específico** Se quisermos buscar todos os livros de um gênero específico, como "Tecnologia", utilizamos a seguinte consulta:

javascript

```
db.livros.find({ "genero": "Tecnologia" });
```

2. **Listar Usuários e os Livros Emprestados** Podemos listar todos os usuários e os livros que eles emprestaram com a seguinte consulta:

javascript

```
db.usuarios.aggregate([
  {
    $lookup: {
      from: "livros",
      localField: "livros_emprestados",
      foreignField: "_id",
      as: "livros_emprestados"
    }
  }
]);
```

Exemplo de resultado de consulta:

json

```
[
  {
    "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99d9"),
    "nome": "Maria Silva",
    "email": "maria.silva@email.com",
    "livros_emprestados": [
      {
        "_id": ObjectId("5f8d0c8f0b4c3d3e9c2a99b7"),
        "titulo": "Introdução ao MongoDB",
        "autor": "John Doe",
        "ano": 2020,
        "genero": "Tecnologia",
        "disponivel": true
      }
    ]
  }
]
```

Conclusão do Projeto

O MongoDB foi eficaz na implementação do sistema de gerenciamento de dados da biblioteca, oferecendo flexibilidade na modelagem de dados e performance nas consultas. O sistema foi capaz de realizar operações essenciais como inserção, atualização, e consulta de dados de maneira rápida e escalável. A utilização de coleções e documentos permitiu representar de forma simples e eficiente os dados da biblioteca, adaptando-se facilmente à necessidade de mudanças e expansão no futuro.

Conclusão

O MongoDB se mostrou uma excelente escolha para o gerenciamento de dados em cenários que exigem flexibilidade e escalabilidade. A capacidade de armazenar dados em formato JSON/BSON permitiu que a estrutura do banco de dados fosse ajustada conforme a necessidade, sem a rigidez dos bancos relacionais. O sistema desenvolvido atendeu às necessidades do projeto, provando que bancos de dados NoSQL, como o MongoDB, são ideais para ambientes com grandes volumes de

dados e para aplicações que exigem agilidade no processamento de informações. Para melhorias, seria interessante a implementação de uma interface de usuário mais sofisticada e a adição de funcionalidades de pesquisa avançada.

Referências Bibliográficas

- CHERRY, S. *MongoDB: The Definitive Guide*. 3. ed. O'Reilly Media, 2020.
 - ESTRIN, M.; FITZGERALD, M. *Database Internals: A Deep Dive into How Distributed Data Systems Work*. O'Reilly Media, 2017.
 - HAWKINS, D. *NoSQL for Dummies*. Wiley Publishing, 2015.
 - STONEBRAKER, M.; HAMILTON, J. *The Evolution of NoSQL Databases*. Communications of the ACM, 2017.
-