

Modelagem em Banco de Dados NoSQL



Figura 1: Logo do MongoDB.

<https://www.mongodb.com/company/newsroom/brand-resources> Acesso em: 06 mai. 2024.

Banco de Dados Não Relacional
Desenvolvimento de Software Multiplataforma
Prof. Ms. Ricardo Leme

Sumário

Introdução.....	2
Relacionamento 1:1:.....	3
Campo de referência:.....	3
Aninhamento:.....	5
Escolhendo a abordagem ideal.....	6
Relacionamento 1:N:.....	7
Exemplo de Relacionamento 1:N no MongoDB.....	7
Relacionamento N:N:.....	9
Array de referências:.....	9
Coleção de junção:.....	11
Escolhendo a abordagem ideal para N:N:.....	13
Conclusão.....	14
Recursos Adicionais.....	15
Referências.....	15

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

Introdução

No MongoDB, a modelagem de relacionamentos 1:1, 1:N e N:N se diferencia um pouco da abordagem tradicional em bancos relacionais, devido à sua natureza não relacional. Vamos analisar cada caso em detalhes:

Relacionamento 1:1:

No contexto 1:1, um documento em uma coleção se relaciona com no máximo um documento em outra coleção. Para implementá-lo no MongoDB, existem duas abordagens principais:

Campo de referência:

- O documento na coleção "principal" contém um campo que armazena o `_id` do documento relacionado na coleção "detalhada".
- Exemplo:
 - Coleção "clientes":

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

```
{
  "_id": ObjectId("1234567890"),
  "nome": "João Silva",
  "endereço": {
    "_id": ObjectId("9876543210"), // Referência ao documento
  }
}
```

- Coleção "endereços":

```
{
  "_id": ObjectId("9876543210"),
  "logradouro": "Rua das Flores",
  "numero": 123,
  "cidade": "Itu",
  "estado": "SP"
}
```

Vantagens:

- Simplicidade na estrutura do documento.
- Desempenho eficiente para consultas que acessam ambos os documentos.

Desvantagens:

- Requer junções em código para recuperar o documento completo.

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

- Dificulta a atualização de dados relacionados em diferentes operações.

Aninhamento:

- O documento na coleção "principal" contém o documento "detalhado" aninhado em seu próprio campo.
- Exemplo:

```
{
  "_id": ObjectId("1234567890"),
  "nome": "João Silva",
  "endereço": {
    "logradouro": "Rua das Flores",
    "numero": 123,
    "cidade": "Itu",
    "estado": "SP"
  }
}
```

Vantagens:

- Retorna todo o documento relacionado em uma única consulta.
- Simplifica a atualização de dados relacionados em uma única operação.

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

Desvantagens:

- Aumenta o tamanho do documento principal, podendo afetar o desempenho.
- Dificulta a normalização de dados, caso precise ser acessado por outras coleções.

Escolhendo a abordagem ideal

A escolha entre campo de referência e aninhamento depende de diversos fatores, como:

- **Frequência de acesso aos dados relacionados:** Se o acesso frequente exigir a recuperação de ambos os documentos juntos, o aninhamento pode ser vantajoso.
- **Padrão de atualização dos dados:** Se as atualizações precisarem modificar tanto o documento principal quanto o relacionado em uma única operação, o aninhamento facilita essa tarefa.
- **Tamanho e normalização dos dados:** Se a coleção principal precisa manter um tamanho reduzido ou os dados relacionados precisam ser acessados por outras coleções, o campo de referência pode ser mais adequado.

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

Relacionamento 1:N:

Neste cenário, um documento em uma coleção se relaciona com vários documentos em outra.

A implementação no MongoDB segue as mesmas abordagens do 1:1, utilizando campo de referência ou aninhamento.

A principal diferença reside na estrutura do array que armazena os documentos relacionados no campo aninhado ou na quantidade de documentos que podem referenciar o mesmo documento principal no campo de referência.

Exemplo de Relacionamento 1:N no MongoDB

Cenário:

Imagine uma loja online que vende produtos e cada produto pode ter várias avaliações de clientes. Nesse caso, há um relacionamento 1:N entre produtos e avaliações: um produto pode ter várias avaliações, mas uma avaliação pertence a apenas um produto.

Modelagem no MongoDB:

- **Coleção produtos:**
 - `_id`: ObjectId (identificador único do produto)

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

- **nome**: String (nome do produto)
- **descricao**: String (descrição do produto)
- **preco**: Number (preço do produto)
- **avaliacoes**: Array (opcional)
 - **_id**: ObjectId (identificador único da avaliação)
 - **cliente**: String (nome do cliente que fez a avaliação)
 - **texto**: String (texto da avaliação)
 - **avaliacao**: Number (avaliação do produto em estrelas)
- **Coleção **avaliacoes** (opcional):**
 - **_id**: ObjectId (identificador único da avaliação)
 - **produto_id**: ObjectId (identificador do produto relacionado à avaliação)
 - **cliente**: String (nome do cliente que fez a avaliação)
 - **texto**: String (texto da avaliação)
 - **avaliacao**: Number (avaliação do produto em estrelas)

Explicação:

- A coleção **produtos** pode armazenar o array **avaliacoes** dentro do próprio documento do produto, aninhando as

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

avaliações. Essa abordagem é útil se as avaliações forem frequentemente acessadas junto com o produto e se o tamanho do array não for muito grande (até 16MB).

- Se as avaliações precisarem ser acessadas e gerenciadas de forma independente, ou se o array de avaliações for muito grande, pode ser interessante criar uma coleção separada `avaliacoes` com um campo `produto_id` para referenciar o produto relacionado. Essa abordagem permite normalizar os dados e facilita consultas e agregações complexas nas avaliações.

Relacionamento N:N:

Na relação N:N, diversos documentos em uma coleção se relacionam com diversos documentos em outra. No MongoDB, essa relação é implementada através de:

Array de referências:

- Cada documento em ambas as coleções contém um array que armazena os `_ids` dos documentos relacionados na outra coleção.
- Exemplo:
 - Coleção "produtos":

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

```
{
  "_id": ObjectId("1234567890"),
  "nome": "Camisa",
  "categorias": [
    ObjectId("9876543210"), // Categoria "Vestuário"
    ObjectId("0123456789") // Categoria "Promoções"
  ]
}
```

- Coleção "categorias":

```
{
  "_id": ObjectId("9876543210"),
  "nome": "Vestuário"
}
{
  "_id": ObjectId("0123456789"),
  "nome": "Promoções"
}
```

Vantagens:

- Flexibilidade para relacionar diversos documentos entre as coleções.
- Permite consultas eficientes para recuperar documentos relacionados.

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

Desvantagens:

- Requer junções em código para obter a lista completa de documentos relacionados em cada lado.
- Pode aumentar a complexidade da consulta e do código aplicativo.

Coleção de junção:

- Cria-se uma nova coleção que armazena os relacionamentos entre os documentos das coleções originais.
- Essa coleção de junção contém os `_ids` dos documentos relacionados de ambas as coleções.
- Exemplo:
 - Coleção "produtos":

```
{  
  "_id": ObjectId("1234567890"),  
  "nome": "Camisa"  
}
```

- Coleção "categorias"

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

```
{
  "_id": ObjectId("9876543210"),
  "nome": "Vestuário"
}
{
  "_id": ObjectId("0123456789"),
  "nome": "Promoções"
}
```

- Coleção de junção "produto_categoria":

```
{
  "_id": ObjectId("ABCDEF1234567890"),
  "produto_id": ObjectId("1234567890"),
  "categoria_id": ObjectId("9876543210")
}
{
  "_id": ObjectId("0123456789ABCDEF"),
  "produto_id": ObjectId("1234567890"),
  "categoria_id": ObjectId("0123456789")
}
```

Vantagens:

- Normalização dos dados, facilitando consultas e agregações complexas.
- Elimina a necessidade de junções em código.

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

Desvantagens:

- Cria uma nova coleção para armazenar os relacionamentos, aumentando a complexidade do modelo de dados.
- Pode afetar o desempenho em casos de alta frequência de inserção e remoção de relacionamentos.

Escolhendo a abordagem ideal para N:N:

A escolha entre array de referências e coleção de junção depende de diversos fatores, como:

- **Complexidade das consultas:** Se as consultas precisarem recuperar a lista completa de documentos relacionados com frequência, a coleção de junção pode ser mais eficiente.
- **Frequência de atualização dos relacionamentos:** Se as relações entre documentos forem alteradas com frequência, o array de referências pode ser mais simples de gerenciar.
- **Padrão de acesso aos dados:** Se o padrão de acesso envolve recuperar apenas o `_id` do documento relacionado, o array de referências pode ser suficiente.
- **Normalização dos dados:** Se a normalização dos dados for crucial para consultas complexas ou agregações, a coleção de junção é mais adequada.

Banco de Dados Não Relacional

Desenvolvimento de Software Multiplataforma

Prof. Ms. Ricardo Leme

Conclusão

A modelagem no MongoDB oferece um paradigma flexível e poderoso para gerenciar dados não relacionais. Ao escolher a abordagem adequada, utilizar ferramentas auxiliares e considerar os aspectos importantes mencionados, você estará apto a criar modelos de dados eficientes e escaláveis para o seu aplicativo MongoDB.


Lembre-se: A modelagem ideal depende dos requisitos específicos do seu projeto, dos padrões de acesso aos dados e da sua experiência com o MongoDB. Explore as diferentes opções, experimente ferramentas e busque conhecimento para dominar a arte da modelagem no MongoDB!

Banco de Dados Não Relacional
Desenvolvimento de Software Multiplataforma
Prof. Ms. Ricardo Leme

Recursos Adicionais

O vídeo a seguir, apresentado por Daniel Kupalo na MongoDB Dev Day 2022, discute os princípios de modelagem de dados para o MongoDB.

Kupalo, um desenvolvedor sênior de equipe e defensor do MongoDB, compartilha sua vasta experiência com o banco de dados, oferecendo insights valiosos sobre como modelar dados de forma eficaz para aplicativos MongoDB.

 The Principles of Data Modeling for MongoDB

Referências

MongoDB. Modelagem de dados. Disponível em: <https://www.mongodb.com/docs/manual/data-modeling/>. Acesso em: 06 de maio de 2024.