

Assingment Week 3

Database 2

Sumario

MongoDB	2
Refêrencias	9

MongoDB

MongoDB é um banco de dados NoSQL de código aberto, popular, que oferece alto desempenho, alta disponibilidade e fácil escalabilidade. Ele é projetado para armazenar e gerenciar grandes volumes de dados não estruturados ou semiestruturados.

Por que escolher MongoDB?

1. **Schema Flexível:** MongoDB usa um modelo baseado em documentos, permitindo schemas dinâmicos e flexíveis. Isso é particularmente útil para aplicações com requisitos de dados em evolução.
2. **Escalabilidade:** Suporta escalabilidade horizontal através de sharding, facilitando o gerenciamento de grandes quantidades de dados e altas cargas de tráfego.
3. **Desempenho:** MongoDB oferece persistência de dados de alto desempenho, especialmente para aplicações que requerem processamento de dados em tempo real.
4. **Linguagem de Consulta Rica:** Apesar de ser um banco de dados NoSQL, MongoDB fornece uma poderosa linguagem de consulta que suporta consultas complexas e agregações.
5. **Amigável para Desenvolvedores:** Com drivers para a maioria das linguagens de programação e uma API intuitiva, MongoDB é fácil para os desenvolvedores trabalharem.

Arquitetura do MongoDB

O MongoDB é construído em uma arquitetura distribuída, composta por vários componentes-chave:

1. **Documentos:** A unidade básica de dados no MongoDB. São estruturas semelhantes a JSON chamadas BSON (Binary JSON).
2. **Coleções:** Grupos de documentos, análogos às tabelas em bancos de dados relacionais.

3. **Bancos de dados:** Contêineres para coleções.
4. **Shards:** Partições de dados distribuídas em vários servidores para suportar implantações de big data.
5. **Réplicas:** Cópias redundantes de dados para garantir alta disponibilidade e recuperação de desastres.

Tipos de Dados no MongoDB

MongoDB suporta uma variedade de tipos de dados, incluindo:

- Strings
- Números (inteiros, floats, doubles)
- Booleanos
- Arrays
- Objetos (documentos aninhados)
- Null
- Data
- ObjectId (identificador único)
- Binary data
- Code (JavaScript)

Instalando com Docker

Baixando a imagem do MongoDB do Docker Hub:

```
> docker pull mongodb/mongodb-community-server:latest
latest: Pulling from mongodb/mongodb-community-server
9cb31e2e37ea: Pull complete
cca84104d156: Pull complete
508c2d2cf56e: Pull complete
1d6a5cea7eeb: Pull complete
26c5a58dd9e0: Pull complete
bb7d47df74d3: Pull complete
24d05ace52a2: Pull complete
243eca24b2fa: Pull complete
43ac0e3d73a1: Pull complete
4f4fb700ef54: Pull complete
a0031db3ed09: Pull complete
Digest: sha256:158505d6ce19b6f849dff4f7f0e61efe5d168c72e1952011a3b1d406fefe17c0
Status: Downloaded newer image for mongodb/mongodb-community-server:latest
docker.io/mongodb/mongodb-community-server:latest
```

Comando Docker para baixar a imagem do MongoDB

Executando o container do MongoDB:

```
> docker run --name mongodb -p 27017:27017 -d mongodb/mongodb-community-server:latest
7b020b2fcd5b5c3562d3b36e0727efd1f11fa8905c58347851ad313acdca2c6
```

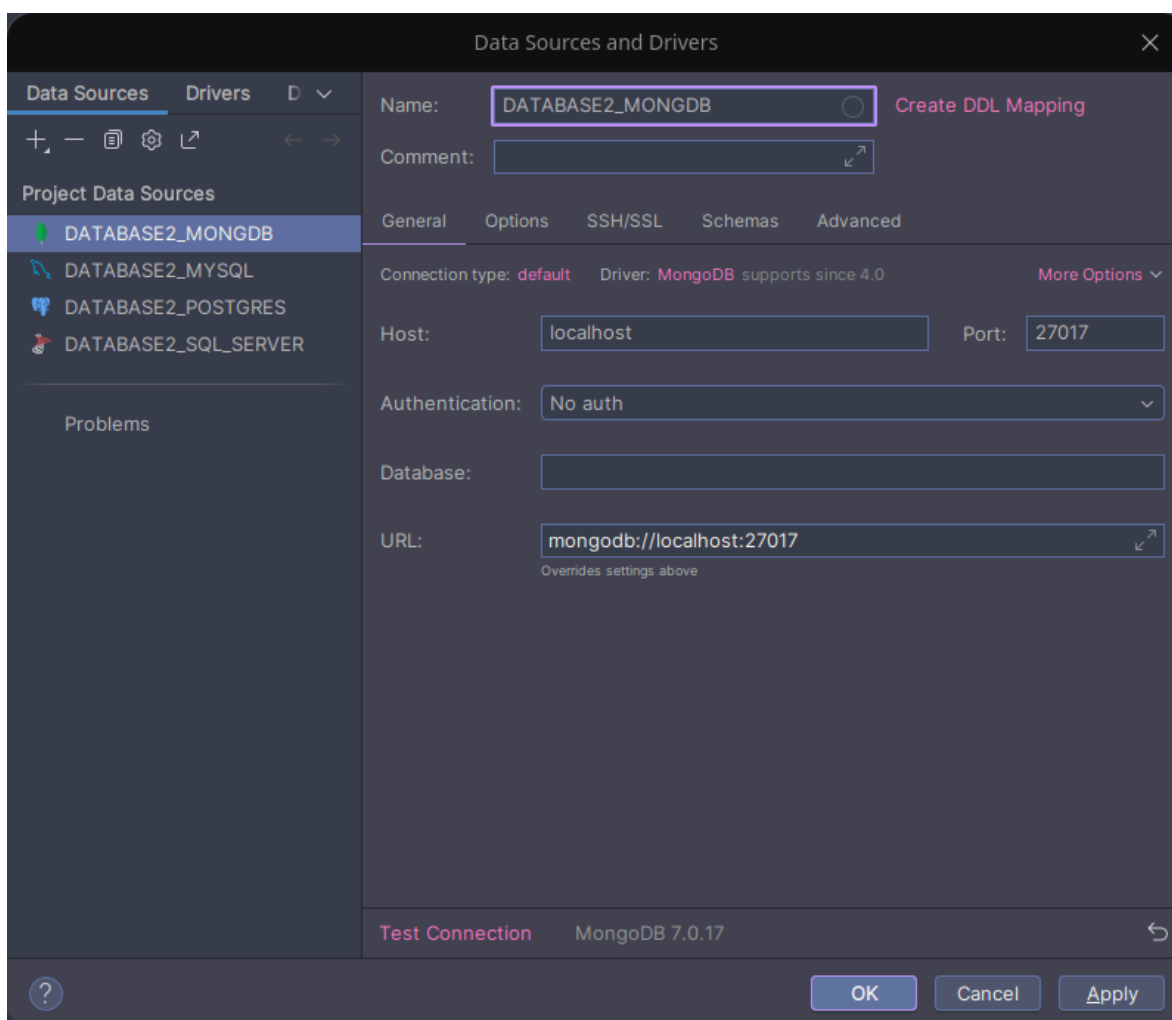
Comando Docker para executar o container do MongoDB

Verificando se o container do MongoDB está em execução:

```
> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
7b020b2fcd5   mongodb/mongodb-community-server:latest  "python3 /usr/local/..."  29 seconds ago  Up 29 seconds  0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp  mongodb
```

Comando Docker para verificar containers em execução

Conectando-se ao MongoDB usando o DataGrip:



Configurações de conexão do DataGrip para MongoDB

Trabalhando com MongoDB

Uma vez que você tenha o MongoDB configurado, você pode começar a trabalhar com coleções e documentos. Aqui estão algumas operações básicas:

1. **Criar um banco de dados:** Use o comando `use` para criar ou mudar para um banco de dados.
2. **Criar uma coleção:** As coleções são criadas automaticamente quando você insere documentos.
3. **Inserir documentos:** Use `insertOne()` ou `insertMany()` para adicionar documentos a uma coleção.

4. **Consultar documentos:** Use o método `find()` para recuperar documentos de uma coleção.
5. **Atualizar documentos:** Use `updateOne()` ou `updateMany()` para modificar documentos existentes.
6. **Excluir documentos:** Use `deleteOne()` ou `deleteMany()` para remover documentos de uma coleção.

Exemplos Práticos

Aqui estão alguns exemplos de operações comuns no MongoDB:

```
// Inserir um documento
db.usuarios.insertOne({
  nome: "João Silva",
  idade: 30,
  email: "joao@exemplo.com",
  interesses: ["esportes", "música", "tecnologia"]
})

// Consultar documentos
db.usuarios.find({ idade: { $gt: 25 } })

// Atualizar um documento
db.usuarios.updateOne(
  { nome: "João Silva" },
  { $set: { idade: 31 } }
)

// Excluir um documento
db.usuarios.deleteOne({ email: "joao@exemplo.com" })
```

Índices no MongoDB

Os índices são cruciais para melhorar o desempenho das consultas. MongoDB suporta vários tipos de índices:

1. Índices de campo único
2. Índices compostos
3. Índices multikey (para arrays)
4. Índices geoespaciais
5. Índices de texto

Para criar um índice:

```
db.usuarios.createIndex({ email: 1 })
```

Agregações

O framework de agregação do MongoDB permite operações avançadas de processamento de dados:

```
db.vendas.aggregate([
  { $match: { data: { $gte: new Date("2023-01-01") } } },
  { $group: { _id: "$produto", total: { $sum: "$quantidade" } } },
  { $sort: { total: -1 } }
])
```

Conclusão

MongoDB é uma excelente escolha para projetos que requerem flexibilidade, escalabilidade e desempenho. Seu modelo baseado em documentos o torna particularmente adequado para aplicações com estruturas de dados complexas e hierárquicas. Ao usar o Docker para instalação, você pode configurar rapidamente um ambiente MongoDB para desenvolvimento ou testes.

À medida que você continua explorando o MongoDB, descobrirá seu rico conjunto de recursos, incluindo suporte para transações ACID, consultas geoespaciais, e integrações com big data e ferramentas de análise. Com sua capacidade de lidar com grandes volumes de dados e consultas complexas, o MongoDB se estabeleceu como uma

solução robusta para uma ampla gama de casos de uso, desde aplicações web e mobile até análise de big data e IoT.

Refêrencias

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-community-with-docker/>

<https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-20-04>

<https://www.geeksforgeeks.org/how-to-install-and-configure-mongodb-in-ubuntu/>

<https://www.geeksforgeeks.org/mongodb-tutorial/>