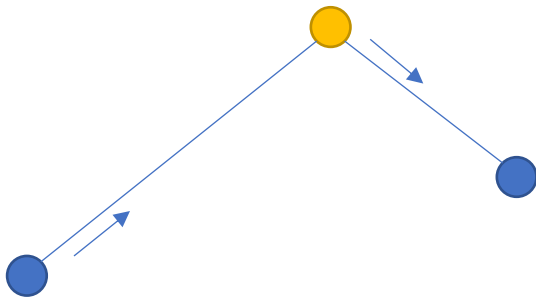


JANUSGRAPH

É um banco de dados distribuído para lidar com quantidades massivas de dados em formato de grafos.

Extremamente útil quando as conexões entre os dados são mais importantes do que os próprios dados.



Antes de se tornar JanusGraph, o Titan existiu (2012). A DataStax comprou a Aurelius em 2015 e continuou o projeto com outro nome.

características

Algumas características do Janus se destacam de outros bancos. Um exemplo é sua distribuição horizontal. Vejamos mais características:



ELASTIC SEARCH

Integração com motor de busca que permite consultas e indexação de dados



CONSISTÊNCIA

O Janus pode oferecer alta consistência (Cassandra) ou disponibilidade (Hbase) e manter integridade



TINKERPOP

A manipulação de grafos são facilitadas pela API de grafos Apache Tinker Pop, permitindo conjuntos atômicos

vantagens



ESCABILIDADE



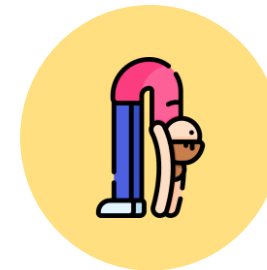
ALTA CONECTIVIDADE



CONSULTAS



SUPORTE



FLEXIBILIDADE

conclusão

O JanusGraph, antigo Titan, ofereceu uma solução inovadora para lidar com grandes volumes de dados distribuíveis e escaláveis. Embora o Titan tivesse sido descontinuado, ele foi transformado no Janus.



Seus pontos mais fortes são a escalabilidade e a eficiência das consultas baseadas em relações.



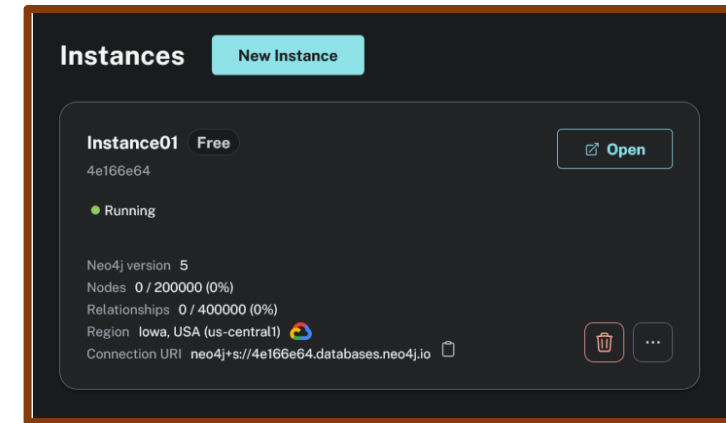
Seu pontos mais fracos são a complexidade do uso do banco.

Neo4J

Também baseado em **grafos**, o Neo original pode ser implementado em qualquer máquina com servidores **Docker**.

A AuraBD é oferecida como **SaaS** (*Software as a Service*).

No plano profissional é possível escolher um provedor **cloud**. Na versão gratuita, a instância do banco de dados é alocada no **GCP** (*Google Cloud Platform*) nos EUA.



Um grafo é um conjunto de vértices (nós, pontos, conexões) e arestas (conexões, linhas).

Podem ser interpretadas de várias maneiras para satisfazer os requisitos de um sistema.



navegando

O AuraDB possui o equivalente a um **SGDB**, o **Neo4JWorkspace** que possui uma **interface web**.

É possível se conectar à base de métodos de **baixo nível**:

- Console virtual e CLI
- Drivers para linguagens de programação
- API REST
- API GraphQL

Uma seleção de dados com **MATCH**:

```
MATCH (movie:Movie)
WHERE movie.rating > 7
RETURN movie.title
```



Relacionamentos são armazenados fisicamente na base de dados com as arestas

```
MATCH (actor:Actor)-[:ACTED_IN]->(movie:Movie {title: 'Unbrea
RETURN actor.name
```

implementando

Existem **drivers** para diversas linguagens de **programação**. Um exemplo simples de como acessar uma base de dados através da linguagem **Go**:

Lembrando: estamos implementando uma **conexão** simples com a BD local. Usamos **Python** para fazer a conexão.

```
from neo4j import GraphDatabase

if __name__ == "__main__":
    driver = GraphDatabase.driver("bolt://localhost:7687", au
    with driver.session() as session:
        session.execute_write(
            "CREATE (a:Person {name: $name, from: $country, r
            name="Floriano",
            country="Brazil",
            role="Teacher",
        ).single()[0]

        res = session.execute_read(
            "MATCH (a:Person) WHERE role = $role RETURN a LIM
            role="student",
        )

        print(res)

    driver.close()
```

tudo sobre **MONGODB**

O que é?

Nota

"É um sistema de gerenciamento de banco de dados não relacional, baseado em software livre, que utiliza documentos flexíveis em vez de tabelas e linhas para processar e armazenar várias formas de dados."
(IBM)

Usando o sistema de **documentos** e **coleções** de documentos que imitam os **dados** e **tabelas**.



```
Connecting to: mongodb+srv://cluster0.ab123.mongodb.net
Using MongoDB: 7.0

> use test
switched to db test

> const nameRegex = /max/i
nameRegex: /max/i

> db.users.find({name: nameRegex}, {_id: 0, name: 1})
{
  "_id": ObjectId("5553a99ae4b02cf7151206f9"),
  "name": "Maximo Heathcote"
},
{
  "_id": ObjectId("5553a99ae4b02cf7151206f9"),
  "name": "Maximus Borer"
},
{
  "_id": ObjectId("5553a99ae4b02cf7151206f9"),
  "name": "Maximo Heathcote"
},
{
  "_id": ObjectId("5553a99ae4b02cf7151206f9"),
  "name": "Maximus Borer"
}

> db.users.find()
TypeError: db.users.find is not a function

> db.reports.find(
  { "callLetters": "GTOT" },
  { "airTemperature": 1, "dewPoint": 1 }
);
{
  "_id": ObjectId("5553a99ae4b02cf7151206f9"),
  "airTemperature": {
    "value": 4.4,
    "quality": "1"
  },
  "dewPoint": {
    "value": 4.2,
    "quality": "1"
  }
}
```

Imagens de mongodb.com

características do BANCO



CONSULTAS

Documentos armazenados como BSON com estruturas complexas sem um esquema fixo



ESCABILIDADE

Altamente escalável, o Mongo pode ser distribuído em vários servidores



PERFORMANCE

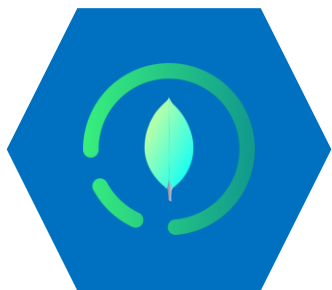
Tendo indexação para acelerar consultas e tendo uma arquitetura interna suficiente para grandes massas de dados



CONSULTAS

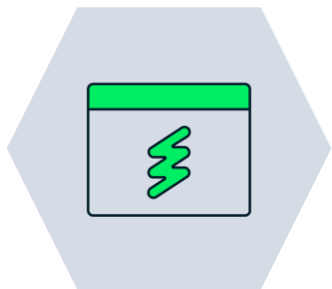
O Mongo oferece uma linguagem flexível para consultas, podendo filtrar, projetar, etc...

inúmeras **FERRAMENTAS**



MongoDB Compass

Uma **GUI** visual que **interage** com os bancos MongoDB e permite que o usuário **explore**, **visualize** e **analise** esses dados



MongoDB Shell

Interface de linha de comando para **interagir** com o banco. Permite: **consultas**, scripts e comandos **administrativos**



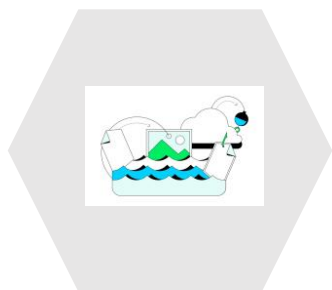
MongoDB Atlas

Uma plataforma que permite **implantar**, **gerenciar** e **escalar** clusters na **nuvem** de maneira simples e eficiente



MongoDB Ops Manager

Plataforma de **gerenciamento** e **monitoramento**, permitido **backup**, automação e segurança



MongoDB Atlas Data Lake

Ferramenta que permite **consultar** e **analisar** dados em nuvem, como Amazon S3, ou Azure Data Lake Storage, usando o MongoDB Query Language (**MQL**)



MongoDB BI Connector

Uma ferramenta que faz **ponto** entre MongoDB e ferramentas de **Business Inteligente** (BI) como Power BI ou Tableau



Drivers MongoDB

Bibliotecas de cliente disponíveis para várias **linguagens de programação** que permitem que aplicativos se **conectem** e **interajam** com o banco de dados MongoDB

instalação da BELEZA

Podemos instalar o MongoDB de várias maneiras diferentes, sendo local, por executáveis, por Docker ou pelo MongoDB Atlas

1. Baixe o MongoDB ZIP: acessando o [link](#) e clique em *download*

2. Extrair o ZIP: após o download do ZIP, extraia o conteúdo para a pasta de sua escolha. Um exemplo: *C:\mongodb*. Opcional: criar uma pasta para ser um lugar para armazenar os dados do MongoDB. Exemplo: *C:\mongodb\data*

3. Iniciar o MongoDB: abra o prompt de comando e navegue até o diretório que criamos. Exemplo:

```
cd C:\mongodb\bin
```

4. Executar: dentro do diretório *bin*, execute o arquivo *mongod.exe*. Você pode especificar o diretório de dados. Exemplo:

```
mongod.exe --dbpath C:\mongodb\data
```

5. Iniciar o cliente: vá até *bin* novamente e execute mongo.exe. Exemplo:

```
cd C:\mongodb\bin  
Mongo.exe
```

comandos **BASICS**

Alguns comandos no MongoDB são essenciais para começar a criar e mexer no seu próprio banco de dados.

1. Criar uma coleção:

```
db.createCollection(nome_da_colecao)
```

```
db.createCollection("family")
```

2. Adicionar pessoas à família:

```
db.collection.insert(document)
```

```
db.family.insert({nome: Maria, idade: 50, familyrole: grandma})  
db.family.insert({nome: Joseph, idade: 80, familyrole: grandpa})
```

3. Filtrando o vovô:

```
db.collection.find(filter)  
db.family.find({familyrole: grandpa})
```

4. Na verdade era mãe:

```
db.colletion.update(filter, update)  
db.family.update(  
    {nome: Maria},  
    {$set: {familyrole: mother}}  
)
```

5. Ish... Bateu as botas:

```
db.collection.remove(filter)  
db.family.remove({nome: Joseph})
```

- `show collections` → listar todas coleções no banco
- `db.collection.drop()` → remove a coleção especificada do banco

enfim, **INCRÍVEL**

O MongoDB se difere dos outros por ser uma solução:

- Flexível
- Acessível
- Única
- fácil
- incrível
- bonita

E com suas ferramentas, você pode ter um monitoramento e uma administração completa de forma fácil, sendo melhor para uma grande quantidade de dados.

Links úteis para um estudo abrangente do MongoDB:

[Documentações Gerais](#)

[Guia para Iniciantes](#)

[Operações de Consulta e Projeção](#)

[Operações de Atualização](#)

[Estágios de Aggregations](#)

[Operações de Agregação](#)

acabou...

OBRIQADO(A)