



NEO4j

Francine Fusão
Matheus H. Mafra
Lucas Hellmann

O que é?

Neo4j é um sistema de gerenciamento de banco de dados gráfico (SGBD) que armazena dados na forma de nós e relacionamentos. Ele é projetado para gerenciar, analisar e visualizar dados fortemente conectados. Ele usa um modelo de dados de grafo, onde os dados são representados como nós (entidades) conectados por relacionamentos. Esta estrutura é ideal para representar e consultar dados altamente interconectados, diferentemente dos bancos de dados relacionais tradicionais que se concentram em tabelas e linhas.

Modelos de licenciamento

01

Community Edition

Gratuito e de código aberto, perfeito para aprendizagem e projetos menores.

02

Enterprise Edition

Opção paga com recursos avançados, como alta disponibilidade, melhorias de segurança e muito mais.

03

AuraDB

Serviço totalmente gerenciado baseado em nuvem para implantação e escalabilidade perfeitas.

Características principais

01

Armazenamento gráfico nativo

Os dados são armazenados em uma estrutura semelhante a um gráfico, otimizada para atravessar relacionamentos.

02

Cypher Query Language

Uma linguagem de consulta declarativa projetada especificamente para bancos de dados gráficos.

03

Altamente escalável

Pode lidar com conjuntos de dados massivos e consultas complexas.

04

Análise em tempo real

permite análise rápida e eficiente de dados conectados.



Casos de uso de utilização do SGDB

O Neo4j é um banco de dados orientado a grafos que se destaca por sua capacidade de modelar e consultar relacionamentos complexos de maneira eficiente. Aqui estão alguns casos de uso comuns e onde o Neo4j pode ser especialmente benéfico:



Áreas de aplicação

01

Redes Sociais

Analizando conexões e relacionamentos sociais.

02

Sistemas de recomendação

sugestão de conteúdo relevante com base nas preferências e conexões do usuário.

03

Deteccão de fraude

identificação de padrões e anomalias em dados de transações.

04

Análise em tempo real

permite análise rápida e eficiente de dados conectados.



CRUD no NEO4j



Create

Para criar um nó no Neo4j, é utilizado a cláusula CREATE. Por exemplo, para criar um nó representando uma pessoa com nome "João" e idade 30:

```
CREATE (p:Pessoa {nome: "João", idade: 30})
```

Para criar um relacionamento entre dois nós existentes, é utilizado a cláusula MATCH para encontrar os nós e a cláusula CREATE para criar o relacionamento. Por exemplo, para criar um relacionamento de "AMIGO" entre os nós "João" e "Maria":

```
MATCH (joao:Pessoa {nome: "João"}), (maria:Pessoa {nome: "Maria"})  
CREATE (joao)-[:AMIGO]->(maria)
```


Read

Para ler dados no Neo4j, é utilizado a cláusula MATCH. Por exemplo, para encontrar todos os nós com o rótulo "Pessoa":

```
MATCH (p:Pessoa)  
RETURN p
```

Para encontrar um relacionamento específico, é utilizado a cláusula MATCH para encontrar os nós e especificar o tipo de relacionamento. Por exemplo, para encontrar todos os amigos de "João":

```
MATCH (joao:Pessoa {nome: "João"})-[:AMIGO]->(amigo)  
RETURN amigo
```

Update

Para atualizar propriedades de um nó existente, é utilizado a cláusula SET. Por exemplo, para atualizar a idade de "João" para 35:

```
MATCH (p:Pessoa {nome: "João"})  
SET p.idade = 35
```

Para atualizar um relacionamento, é utilizado a cláusula MATCH para encontrar o relacionamento e a cláusula SET para atualizar suas propriedades. Por exemplo, para mudar o tipo de relacionamento entre "João" e "Maria" para "CASADO":

```
MATCH (joao:Pessoa {nome: "João"})-[:AMIGO]->(maria:Pessoa {nome: "Maria"})  
SET [:CASADO] = [:AMIGO]
```

Delete

Para excluir um nó, é utilizado a cláusula MATCH para encontrar o nó e a cláusula DELETE. Por exemplo, para excluir o nó "João":

```
MATCH (p:Pessoa {nome: "João"})  
DELETE p
```

Para excluir um relacionamento específico, é utilizado a cláusula MATCH para encontrar o relacionamento e a cláusula DELETE. Por exemplo, para excluir o relacionamento "AMIGO" entre "João" e "Maria":

```
MATCH (joao:Pessoa {nome: "João"})-[:AMIGO]->(maria:Pessoa {nome: "Maria"})  
DELETE (:AMIGO)
```



Posição e evolução do SGDB no ranking do DB-Engines:


Neo4j está consistentemente classificado entre os principais sistemas de gerenciamento de banco de dados gráfico no DB-Engines.

A sua popularidade e ampla adoção em vários domínios destacam a sua relevância e capacidades.

Atualmente ele reside em 1º lugar no ranking de banco de dados gráfico.

Posicionamento do SGDB no teorema CAP:

O Neo4j prioriza consistência e tolerância de partição, posicionando-o como um banco de dados CP. Isso significa que mesmo na presença de partições de rede, o Neo4j garante a consistência dos dados. Embora não seja excelente em disponibilidade (A), seu foco em CP o torna ideal para cenários onde a integridade dos dados e relacionamentos precisos são fundamentais.



Instalação



Baixar e instalar

O Neo4j Desktop pode ser encontrado no Centro de implantação do Neo4j e está disponível para Mac, Linux e Windows.

Vá para [o Deployment Center](#), faça o download e veja as instruções abaixo:

Mac OSLinuxWindows

Primeiro, localize o arquivo recém-baixado e clique duas vezes nele para iniciar a instalação. Em seguida, siga as etapas na tela para concluir a instalação.

Assim que o Neo4j Desktop for lançado, você poderá registrar-se ou fornecer uma chave de ativação para ativar sua cópia do Desktop para acordos comerciais e suporte. Copie e cole a chave, que pode ser encontrada no topo da página de download, na caixa do aplicativo Desktop.

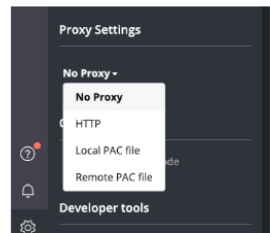
Observe que esta etapa é opcional, você pode pular ou registrar-se posteriormente sem quaisquer limitações.

Configuração de proxy

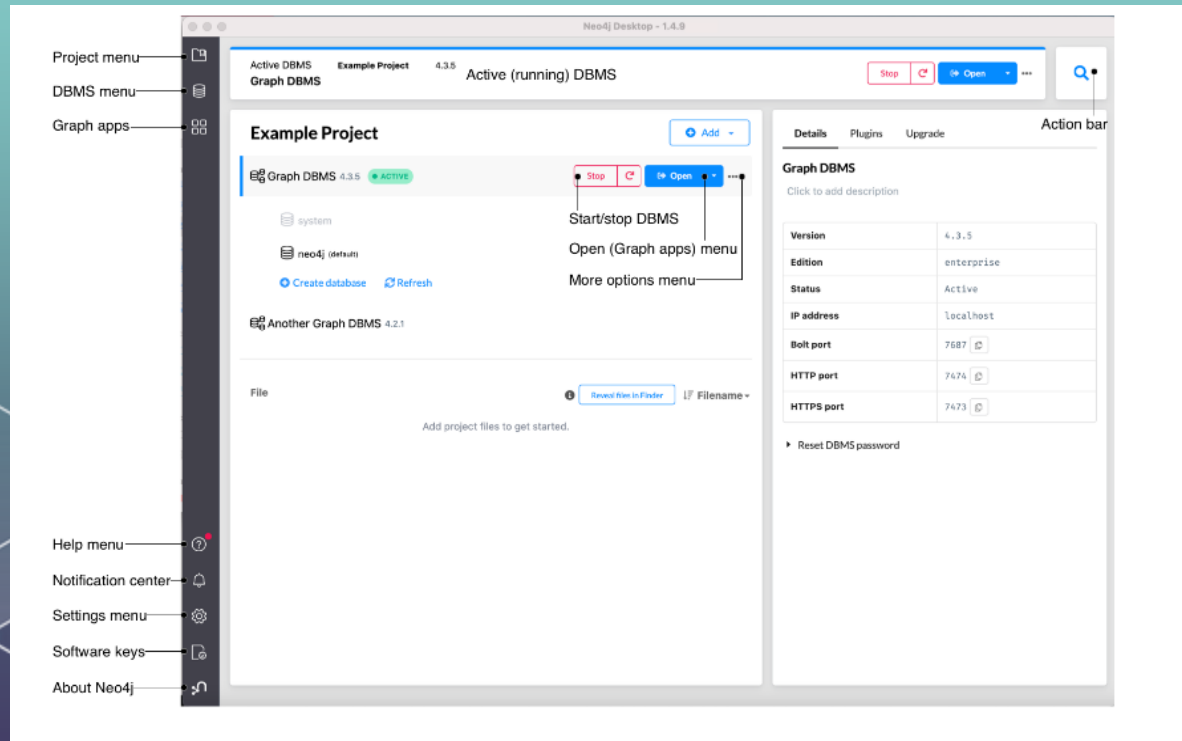
Às vezes você precisa usar um proxy para acesso à Internet ou acesso a sites externos. Desktop permite que você defina as configurações de proxy no **Settings** menu da barra lateral. Existem três maneiras diferentes de configurar um proxy:

- HTTP
- Arquivo PAC local
- Arquivo PAC remoto

Por padrão, isso é definido como **No Proxy** e, para configurar um proxy, clique no menu suspenso e selecione a configuração de sua preferência.



Visual



SEGURANÇA E CONTROLE DE ACESSO NO Neo4j

O Neo4j implementa várias medidas de segurança para proteger seus dados:

1. **Autenticação:**
Os usuários devem autenticar-se antes de acessar o banco de dados.
Isso pode ser feito por meio de senhas ou outros métodos de autenticação.
2. **Autorização:**
O Neo4j permite definir papéis e permissões para usuários e grupos.
Você pode controlar quem pode ler, gravar ou executar consultas no banco de dados.
3. **Criptografia:**
O tráfego entre o cliente e o servidor pode ser criptografado usando SSL/TLS.
4. **Auditoria:**
O Neo4j registra atividades de acesso e alterações no banco de dados para fins de auditoria.
5. **Restrições de Acesso a Dados:**
É possível definir restrições específicas para nós e relacionamentos com base em regras de negócios.

SUORTE A TRANSAÇÕES

ACID

Neo4j é ACID-compliant, o que significa que as transações são tratadas como uma unidade indivisível.
As propriedades ACID são:

- **Atomicidade:** Todas as operações em uma transação têm sucesso ou são revertidas completamente.
- **Consistência:** O banco de dados permanece em um estado consistente após a conclusão ou reversão de uma transação.
- **Isolamento:** As transações não interferem umas nas outras.
- **Durabilidade:** As alterações feitas por uma transação são permanentes, mesmo após falhas do sistema.



Referências

<https://gasparbarancelli.com//post/explorando-o-neo4j-o-banco-de-dados-de-grafos>

<https://en.wikipedia.org/wiki/Neo4j>

<https://www.geeksforgeeks.org/the-cap-theorem-in-dbms/>

<https://db-engines.com/en/system/Neo4j>

