

1. Visão geral do banco de dados escolhido:

• Descrição geral do banco de dados:

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:

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

Enterprise Edition: opção paga com recursos avançados, como alta disponibilidade, melhorias de segurança e muito mais.

AuraDB: serviço totalmente gerenciado baseado em nuvem para implantação e escalabilidade perfeitas.

• Descrição do modelo de dados (principal) utilizado pelo SGDB:

O núcleo do Neo4j é seu modelo gráfico. Isso consiste em:

Nós: representam entidades ou objetos em seus dados (por exemplo, pessoas, lugares, coisas).

Relacionamentos: conecte nós e defina as conexões entre eles (por exemplo, "WORKS_FOR," "FRIENDS_WITH").

Propriedades: pares de valores-chave associados a nós e relacionamentos, armazenando atributos e características (por exemplo, "nome", "idade", "cidade").

Este modelo gráfico permite navegação e análise eficientes de relacionamentos complexos dentro de seus dados.

• Características principais:

Armazenamento gráfico nativo: os dados são armazenados em uma estrutura semelhante a um gráfico, otimizada para atravessar relacionamentos.

Cypher Query Language: Uma linguagem de consulta declarativa projetada especificamente para bancos de dados gráficos.

Altamente escalável: Pode lidar com conjuntos de dados massivos e consultas complexas.

Análise em tempo real: permite análise rápida e eficiente de dados conectados.

Forte apoio comunitário: Comunidade grande e ativa com amplos recursos e documentação.

- **Público-alvo ou áreas de aplicação:**

Neo4j é usado em vários campos, incluindo:

Redes Sociais: Analisando conexões e relacionamentos sociais.

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

Detecção de fraude: identificação de padrões e anomalias em dados de transações.

Gráficos de conhecimento: Representando conhecimento e relações entre conceitos.

Biotecnologia: Modelagem de redes biológicas complexas.

- **URL do desenvolvedor e da documentação:**

URL do desenvolvedor: <https://neo4j.com/>

URL da documentação: <https://neo4j.com/docs/>

- **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.

2. Instalação e/ou configuração:

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.

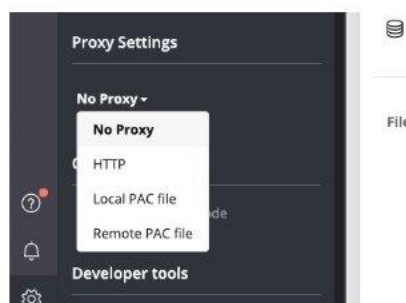
Observe que esta etapa é opcional, você pode pular ou registrar-se posteriormente sem quaisquer limitações.

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.



3. Caso de uso:

• 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:

1. Redes Sociais:

- Modelagem de Relacionamentos: Representação de usuários, suas conexões, interações e interesses.
- Recomendações: Sugestões de amigos, conteúdos, ou grupos com base em conexões e interesses comuns.

2. Detecção de Fraudes:

- Análise de Transações: Identificação de padrões suspeitos em transações financeiras.
- Rastreamento de Cadeias de Fraude: Descoberta de redes de contas ou entidades fraudulentas através da análise dos relacionamentos entre elas.

3. Gerenciamento de Redes de Telecomunicações:

- Monitoramento de Infraestrutura: Representação de equipamentos, conexões e fluxos de dados para otimização e manutenção.
- Análise de Falhas: Identificação de pontos críticos e predição de falhas na rede.

4. Gerenciamento de Dados Mestres (MDM):

- Consolidação de Dados: Integração de dados de diferentes fontes para fornecer uma visão unificada e limpa dos dados mestres.
- Rastreamento de Linhagem de Dados: Monitoramento da origem e transformação de dados através de diferentes sistemas.

5. Gerenciamento de Cidades Inteligentes:

- Planejamento Urbano: Modelagem de infraestrutura urbana, tráfego e uso do solo.
- Análise de Dados de Sensores: Integração de dados de diversos sensores para otimização de recursos e serviços urbanos.

O Neo4j é particularmente eficaz nesses casos de uso devido à sua capacidade de realizar consultas complexas de grafos de forma rápida e eficiente, permitindo análises profundas e em tempo real de dados interconectados.

• Estudo de caso em indústrias específicas (e-commerce, inteligência artificial, análise de dados, mídia social etc.):

Instituto: German Centre for Diabetes Research (DZD)

Uso de Caso: Pesquisa sobre diabetes.

Solução: Utilização de Neo4j para criar grafos de conhecimento que conectam dados de diferentes estudos e fontes.

Resultados: Aceleração da pesquisa e descoberta de novas conexões entre dados que podem levar a tratamentos inovadores (Graph Database & Analytics).

Empresa: Lockheed Martin Space

Uso de Caso: Gerenciamento de ciclo de vida de dados dispersos.

Solução: Neo4j ajuda a integrar dados de diferentes fontes para criar uma visão unificada dos dados do produto.

Resultados: Melhor tomada de decisões e maior eficiência na gestão de projetos (Graph Database & Analytics).

Empresa: Caterpillar

Uso de Caso: Processamento de linguagem natural (NLP) em escala.

Solução: Neo4j é usado para melhorar a eficiência do reparo de equipamentos, analisando grandes volumes de dados de manutenção.

Resultados: Aumento da eficiência operacional e redução do tempo de inatividade dos equipamentos (Graph Database & Analytics).

Empresa: eBay

Uso de Caso: eBay usa Neo4j para alimentar seu aplicativo no Google Assistant, permitindo uma experiência de comércio inteligente.

Solução: Neo4j é usado para analisar o comportamento do cliente em tempo real e fornece recomendações personalizadas.

Resultados: A integração de Neo4j resultou em recomendações mais precisas e um aumento na satisfação do cliente (Graph Database & Analytics).

4. CRUD no SGDB selecionado:

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
```

Atualizando (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]
```

Excluindo (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)
```

5. Definição de esquemas:

- **Como o NEO4J trabalha com a especificação/imposição de esquemas de dados (caso trabalhe).**

O Neo4j é um banco de dados de grafo que difere dos bancos de dados relacionais tradicionais em relação à especificação de esquemas. No Neo4j, os esquemas são mais flexíveis e dinâmicos.

Em vez de tabelas rigidamente definidas com colunas fixas, o Neo4j usa nós e relacionamentos para modelar os dados. Os nós representam entidades e os relacionamentos conectam essas entidades.

No Neo4j, não precisa definir explicitamente esquemas antes de inserir dados. Os nós e relacionamentos podem ser criados conforme necessário, e suas propriedades podem ser adicionadas dinamicamente.

No entanto, você pode impor restrições de esquema usando validações de negócios em seu código de aplicativo ou por meio de regras de negócios específicas.

- **Como o Neo4j implementa o relacionamento entre os dados. (similar a ideia de chaves e estrangeira em Neo4j relacionais).**

O Neo4j implementa relacionamentos entre os dados por meio arestas (ou relacionamentos) que conectam nós.

Cada aresta tem um tipo e direção. Por exemplo, se temos um nó “Pessoa” e um nó “Filme”, podemos criar uma aresta “ATUOU_EM” que conecta a pessoa ao filme.

Os relacionamentos no Neo4j são flexíveis e podem ter propriedades. Por exemplo, a aresta “ATUOU_EM” pode ter uma propriedade “papel” para indicar o papel desempenhado pela pessoa no filme.

A consulta Cypher é usada para criar, consultar e manipular esses relacionamentos.

6. Questões adicionais:

- **Método de segurança e controle de acesso: apresentar o método de controle de acesso e medidas de segurança implementada pelo Neo4j.**

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.

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.

Criptografia: O tráfego entre o cliente e o servidor pode ser criptografado usando SSL/TLS.

Auditoria: O Neo4j registra atividades de acesso e alterações no banco de dados afins auditoria.

Restrições de Acesso a Dados: Você pode definir restrições de acesso específicas para nós e relacionamentos com base em regras de negócios.

- **Suporte a Transações: Descrição do suporte a transações ACID no Neo4j.**

O 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.

7. Referências:

“Explorando o Neo4j: O Banco de Dados de Grafos”. Blog - Gaspar Barancelli,
<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>