THIAGO ADRIANO

PÓS TECH

MACHINE LEARNING ENGINEERING

BANCOS DE DADOS

AULA 03

SUMÁRIO

O QUE VEM POR AÍ?

Nesta aula, você aprenderá o que são os bancos de grafos, uma estrutura de armazenamento de dados que representa as relações entre diferentes entidades por meio de nós e arestas, e entenderá como os grafos podem ser utilizados para modelar e analisar dados complexos e interconectados.

Para acessar o GitHub desta aula, clique aqui.

HANDS ON

Nessa aula prática, os professores e professoras demonstram como trabalhar com os bancos de dados Neo4j utilizando o Docker. Através de exemplos práticos você aprenderá como um banco de grafos pode ser utilizado para modelar e analisar dados complexos e interconectados.

SAIBA MAIS

O sistema de banco de dados baseado em grafos apresenta uma abordagem de design relativamente simples, diferindo significativamente dos bancos de dados relacionais, que adotam tabelas como seu modelo básico de estrutura e relações.

No contexto dos bancos de dados em grafos, o modelo fundamental é composto por grafos. Nele, a inserção de dados ocorre de maneira mais descomplicada, sem a necessidade imediata de especificar os relacionamentos.

A simplicidade desse modelo reflete na representação explícita dos relacionamentos entre os dados, proporcionando uma modelagem mais direta e eficaz.

Isso não apenas resulta em melhor desempenho, mas também torna a linguagem de consulta mais natural e intuitiva.

Nos grafos, os dados são representados por:

- **Nós**: os nós, também conhecidos como vértices, são unidades fundamentais que armazenam os objetos de dados. Cada nó pode estabelecer uma variedade de relacionamentos, sendo versátil na representação de diferentes tipos de entidades. Importante destacar que não há limitação quanto ao número de conexões que um nó pode ter.
- **Arestas**: as arestas, por sua vez, são responsáveis por representar os relacionamentos entre os nós. Essas conexões podem denotar diversas relações, como laços familiares, interações sociais, ações específicas ou qualquer outra relação concebível. Uma aresta é caracterizada por um nó inicial, um nó final, um tipo e uma direção, o que proporciona uma riqueza adicional na modelagem de interações.
- **Propriedades**: cada nó pode conter propriedades ou atributos que o descrevem. Essas características são cruciais para enriquecer a compreensão do nó em questão. Além disso, em alguns casos, as arestas também podem ter propriedades, tornando o grafo mais flexível e capaz de representar informações mais detalhadas. Quando os grafos incorporam

propriedades, eles são frequentemente denominados "grafos de propriedade".

Um banco de dados baseado em grafos é, fundamentalmente, uma estrutura composta por nós e arestas que representam uma maneira intuitiva de armazenar e relacionar dados. Cada nó nesse contexto representa uma entidade distinta, como uma pessoa, empresa ou qualquer elemento relevante para o domínio do banco de dados. Por sua vez, as arestas são responsáveis por representar as conexões ou relacionamentos entre dois nós específicos.

De forma detalhada, cada nó em um banco de dados baseado em grafos possui atributos específicos, incluindo um identificador exclusivo que o diferencia de outros nós, um conjunto de arestas de saída que indicam suas conexões com outros nós, e/ou arestas de entrada que revelam as conexões que apontam para ele.

Além disso, cada nó possui um conjunto de propriedades expressas como pares de chave/valor, proporcionando informações adicionais sobre a entidade representada.

Da mesma forma, cada aresta em um banco de dados baseado em grafos é caracterizada por um identificador exclusivo, um ponto inicial e/ou final que determina os nós conectados por ela e um conjunto de propriedades que oferece detalhes sobre a natureza do relacionamento entre os nós.

Esse modelo permite uma representação mais rica e contextualizada dos dados, tornando a estrutura do banco de dados mais adaptável a cenários complexos.

Para ficar mais claro, vejamos uma imagem demonstrando um conjunto de colegas e seus relacionamentos, representados como um grafo de propriedades.

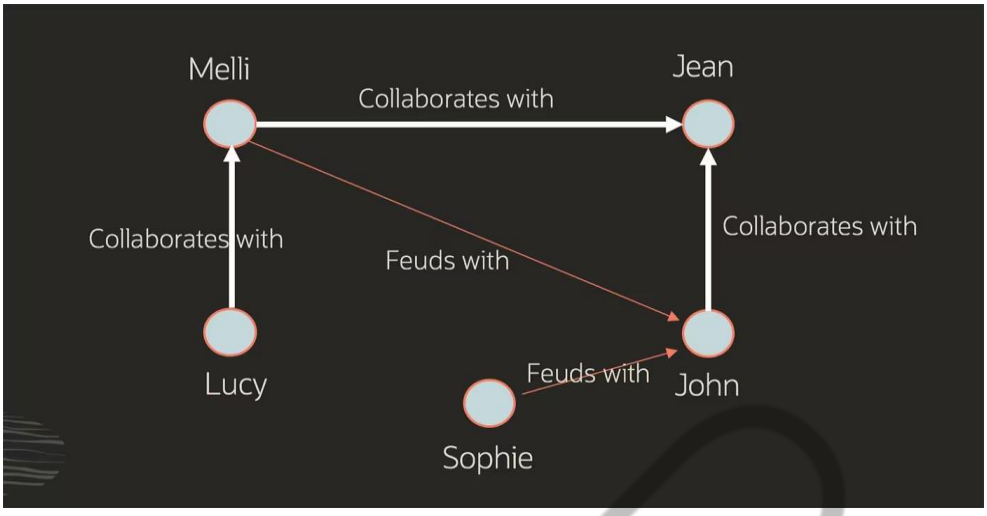

Figura 1 - Relacionamento
Fonte: Oracle (2020)

Devido à sua adaptabilidade, os grafos de propriedades encontram aplicação em diversos setores e indústrias, abrangendo áreas como finanças, manufatura, segurança pública, varejo e uma variedade de outros segmentos.

Bom, tivemos um breve resumo sobre grafos e bancos de dados de grafos, mas como eles funcionam?

Grafos e bancos de dados baseados em grafos oferecem modelos que representam relacionamentos nos dados por meio de uma estrutura de nós e arestas. Esses modelos possibilitam a execução de consultas de travessia com base em conexões, além da aplicação de algoritmos de grafos para descobrir padrões, caminhos, comunidades, influenciadores, pontos únicos de falha e outros relacionamentos. Essa abordagem resulta em uma análise eficiente, especialmente ao lidar com grandes volumes de dados, proporcionando insights valiosos e a capacidade de conectar diversas fontes de dados.

Ao analisar grafos, os algoritmos exploram caminhos e distâncias entre vértices, a importância dos próprios vértices e o agrupamento de vértices. Por exemplo, para determinar a importância, os algoritmos geralmente examinam bordas de entrada, a relevância de vértices vizinhos e outros indicadores.

Os algoritmos de grafos, especialmente projetados para analisar relacionamentos e comportamentos em dados gráficos, permitem compreender aspectos que podem ser difíceis de perceber por outros métodos. Eles ajudam na

identificação de indivíduos ou itens mais conectados em redes sociais ou processos comerciais, revelando comunidades, anomalias, padrões comuns e caminhos entre indivíduos ou transações relacionadas.

Graças ao armazenamento explícito de relacionamentos, os bancos de dados de grafos possibilitam a execução de consultas e algoritmos com grande rapidez, muitas vezes em subsegundos, evitando a necessidade de realizar inúmeras junções e permitindo o uso eficiente dos dados para análise e aprendizado de máquina. Essa abordagem facilita a descoberta de informações significativas sobre o mundo ao nosso redor.

Mas qual seria a vantagem de trabalhar com banco de dados de grafos? O formato de grafo oferece uma estrutura mais flexível para identificar conexões distantes e analisar dados com base em critérios como a força ou qualidade dos relacionamentos.

Essa abordagem permite a exploração e descoberta de padrões em diversas áreas, como redes sociais, Internet das Coisas (IoT), big data, data warehouses e dados de transações complexas.

Os grafos têm aplicações variadas em casos de uso empresarial, incluindo detecção de fraudes em instituições financeiras, descoberta de conexões em redes sociais e criação de perfis abrangentes de clientes (clientes 360). Atualmente, os grafos estão se tornando cada vez mais essenciais na ciência de dados, facilitando a compreensão de conexões em relacionamentos complexos.

A vantagem dos bancos de dados de grafos reside na forma explícita como armazenam relacionamentos, permitindo que consultas e algoritmos que utilizam a conectividade entre vértices sejam executados em subsegundos, em contraste com as horas ou dias necessários em abordagens mais tradicionais. Essa eficiência elimina a necessidade de realizar múltiplas junções, proporcionando facilidade no uso dos dados para análises e aprendizado de máquina, contribuindo para uma compreensão mais profunda do mundo.

Os bancos de dados de grafos representam uma ferramenta altamente flexível e poderosa. Devido à estrutura do grafo, é possível determinar relacionamentos complexos para obter insights mais profundos com um esforço consideravelmente reduzido. Geralmente, esses bancos executam consultas em linguagens específicas,

como Property Graph Query Language (PGQL), simplificando a interação e a análise. A comparação entre consultas em PGQL e SQL ilustra como os bancos de dados de grafos oferecem uma abordagem mais intuitiva e eficiente para analisar e explorar dados interconectados.

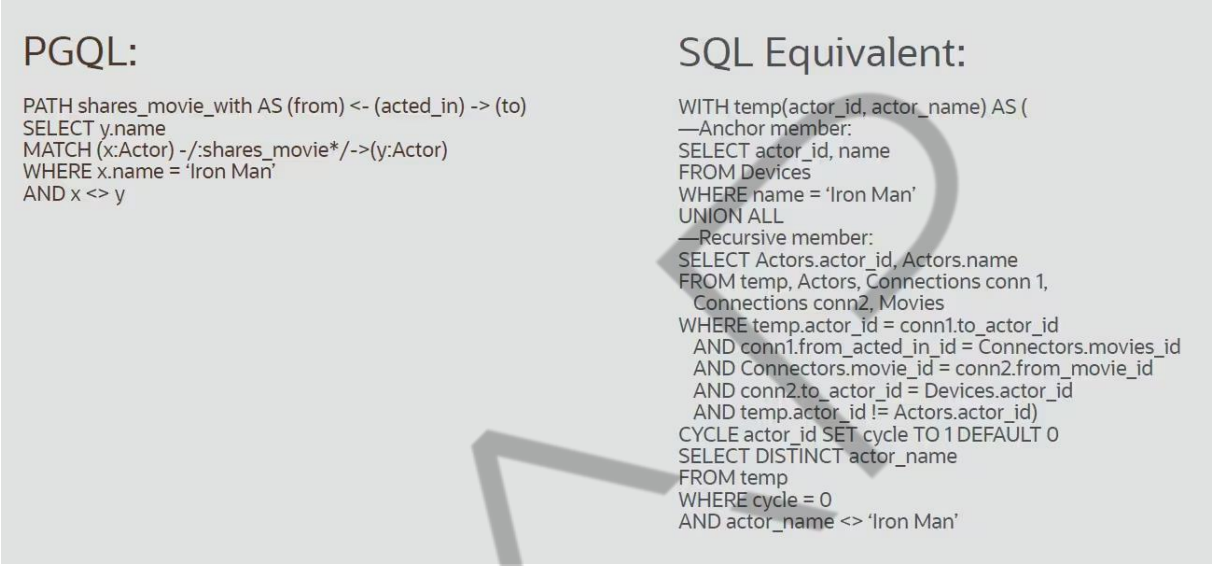

Figura 2 — Exemplo PGQL x SQL
Fonte: Oracle (2020)

Agora que já temos um conhecimento básico sobre este assunto, vamos conhecer o banco sobre o qual aprenderemos nesta aula: o Neo4j.

O Neo4j é um banco de dados de grafos que se destaca por sua capacidade de representar e processar dados complexos e inter-relacionados. Ele adota uma abordagem única, utilizando estruturas de grafos para modelar dados e suas conexões.

Caso tenha interesse em saber mais sobre esse banco e sua versão open source, eu recomendo a leitura da página de licensing do Neo4j.

Para criar um banco de dados Neo4j, basta executar o seguinte comando no seu terminal:

```
docker run --name myneo4j -p 7474:7474 -p 7687:7687 -e
NEO4J_AUTH=neo4j/my-password -d neo4j
```

Em seguida, você pode abrir o seguinte endereço no seu navegador para acessar o seu banco de dados: http://localhost:7474/browser/.

O QUE VOCÊ VIU NESTA AULA?

Ao longo desta aula, você adquiriu conhecimentos sobre bancos de grafos, uma estrutura de armazenamento de dados que representa as inter-relações entre diversas entidades por meio de nós e arestas. Além disso, compreendeu como os grafos se revelam instrumentais na modelagem e análise de dados complexos e interconectados.

REFERÊNCIAS

AWS. **O que é um banco de dados de grafos?** 2024. Disponível em: https://aws.amazon.com/pt/nosql/graph/. Acesso em: 26 mar. 2024.

BORGES, R. **Banco de Dados Baseado em Grafos e suas Principais Características**. 2021. Disponível em: https://micreiros.com/banco-de-dados-baseado-em-grafos-e-suas-principais-caracteristicas/. Acesso em: 26 mar. 2024.

NEO4j. **Neo4j Licensing**. [s.d.]. Disponível em: <https://neo4j.com/licensing/>. Acesso em: 26 mar 2024.

ORACLE. **Banco de dados de grafos definido**. 2023. Disponível em: https://www.oracle.com/br/autonomous-database/what-is-graph-database/. Acesso em: 26 mar. 2024.

ADRIANO, T. S. **[Dica rápida]Conhecendo o Relational Migrator**. 2024. Disponível em: https://programadriano.medium.com/dica-r%C3%A1pida-conhecendo-o-relational-migrator-956dddcb7e5e. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Comandos básicos Docker**. 2017. Disponível em: https://programadriano.medium.com/principais-comandos-docker-f9b02e6944cd. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Comparando os termos utilizados no NoSQL com SQL**. 2024. Disponível em: https://medium.com/xp-inc/comparando-os-termos-utilizados-no-nosql-com-sql-e862788e2374. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Trabalhando com AWS em um ambiente localhost**. 2024. Disponível em: https://dev.to/programadriano/trabalhando-com-aws-em-um-ambiente-localhost-20i8. Acesso em: 14 jun. 2024.

AMAZON. **Download NoSQL Workbench for DynamoDB**. 2024. Disponível em: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html. Acesso em: 14 jun. 2024.

AMAZON.COM. **Nível gratuito do Amazon** RDS. 2024. Disponível em: https://aws.amazon.com/pt/rds/free/. Acesso em: 14 jun. 2024.

DIAS, L. A. Grafos, teoria e aplicações. 2019. Disponível em: https://medium.com/xp-inc/grafos-teoria-e-aplica%C3%A7%C3%B5es-2a87444df855. Acesso em: 14 jun. 2024.

GITHUB. **Criação do container postgres**. 2024. Disponível em: https://gist.github.com/programadriano/2171fe63e6a2c1b08985bccf263fe980. Acesso em: 14 jun. 2024.

GITHUB. **Docker + Neo4J**. 2024. Disponível em: https://gist.github.com/programadriano/0224c1a1d389af25177eb397d912dbe7. Acesso em: 14 jun. 2024.

GITHUB. **Estrutura**. 2024. Disponível em: https://app.diagrams.net/#G1_-jdpuomwUwU3KD01rSRGepNwA3kYA9X#%7B%22pageId%22%3A%22apt8lNFUySjbP2aItylb%22%7D. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (1)**. 2024. Disponível em: https://gist.github.com/programadriano/01508490e0f53745f2778a045e1cc070. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (2)**. 2024. Disponível em: https://gist.github.com/programadriano/e22931ff2bc032dcc68498bc3597d1a7. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (3)**. 2024. Disponível em: https://gist.github.com/programadriano/8c8e7ea613acc4e854a7f2ffe9f44111. Acesso em: 14 jun. 2024.

GITHUB. **Query X Scan**. 2024. Disponível em: https://gist.github.com/programadriano/8169d065da7e0cf8494435ca8cacae70. Acesso em: 14 jun. 2024.

GITHUB.COM. **Carga e Fluxo Neo4J (1)**. 2024. Disponível em: https://gist.github.com/programadriano/e7c314539571c811a48940deca8c8491. Acesso em: 14 jun. 2024.

GITHUB.COM. **Carga e Fluxo Neo4J (2)**. 2024. Disponível em: https://gist.github.com/programadriano/c70281293a5720f1ab8719d65fe5e410. Acesso em: 14 jun. 2024.

IBM. **O que é o Docker?** 2024. Disponível em: https://www.ibm.com/br-pt/topics/docker. Acesso em: 14 jun. 2024.

LOCALSTACK.CLOUD. **Abrir e mostrar o ambiente rodando**. 2024. Disponível em: https://app.localstack.cloud/dashboard. Acesso em: 14 jun. 2024.

LOCALSTACK.CLOUD. **LocalStack Desktop**. 2024. Disponível em: https://app.localstack.cloud/download. Acesso em: 14 jun. 2024.

MIRO. **Fluxo Miro**. 2024. Disponível em: https://miro.com/welcomeonboard/Mzh1bEhNczFWeWtFeFpUS0tXelJlQ1RjZnJjZHhmbmhFU0kyd3RjenpjZVhTdVZqajJPN3NxdFpuR3lkMWc5NHwzNDU4NzY0NTY0Nzg5NTc4MjI1fDI=?share_link_id=113865211181. Acesso em: 14 jun. 2024.

MONGODB. **MongoDB Relational Migrator Download**. 2024. Disponível em: https://www.mongodb.com/try/download/relational-migrator. Acesso em: 14 jun. 2024.

NEO4J. **Exploring LinkedIn in Neo4j**. 2013. Disponível em: https://neo4j.com/blog/exploring-linkedin-in-neo4j/. Acesso em: 14 jun. 2024.

NEO4J. **Use Cases**: Social Media and Social Network Graphs. 2024. Disponível em: https://neo4j.com/use-cases/social-network/. Acesso em: 14 jun. 2024.

PGADMIN.ORG. **PgAdmin4**. 2024. Disponível em: https://www.pgadmin.org/download/pgadmin-4-windows/. Acesso em: 14 jun. 2024.

PALAVRAS-CHAVE

Palavras-chave: Grafos, Neo4j, banco de dados.