BANCO DE DADOS ORIENTADO A GRAFOS

Equipe: Diego Henrique Nogueira

Diego Santos Castro

Jonathan de Paula Damas

**Introdução**

A vasta maioria dos sistemas que foram ou estão sendo criados hoje guardam seus dados em bancos de dados relacionais como o Oracle, SQL Server, MySQL, Postgres e muitos outros. Eles usam o SQL como linguagem para fazer suas consultas e manipulação de dados.

Diferentemente de um banco de dados relacional comum, o banco orientado a grafos é muito mais simples de desenhar. Não precisa de tabelas. Não precisa de chaves primárias, Não precisa de um design complexo de tabelas para começar a incluir os dados.

Você consegue começar simplesmente… incluindo novos dados da forma que você quiser.

**Apresentando o Neo4j**

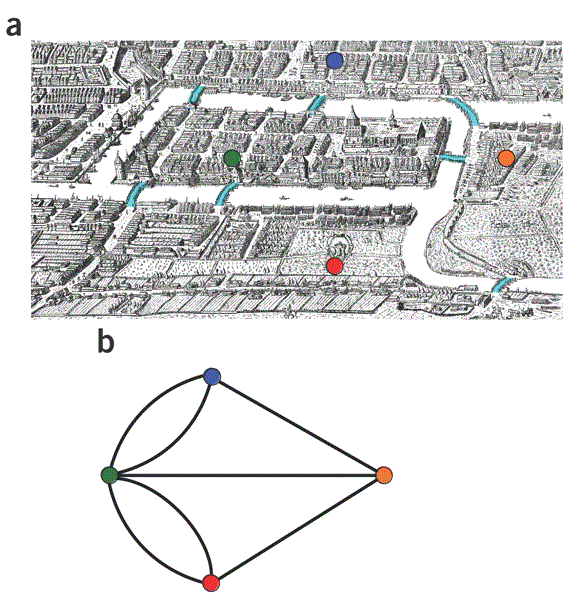
O **Neo4j** é um banco de dados Open Source baseado no conceito **NoSQL** (Banco de Dados que não utiliza os conceitos estruturados).

As informações não são armazenadas em tabelas, mas sim na forma de Grafos e suas estruturas são representadas de forma que o conhecimento é representado pelos conceitos matemáticos da Teoria de Grafos.

**Teoria dos Grafos**

A teoria de grafos é uma ferramenta simples, acessível e poderosa para construção de modelos para inúmeros problemas que requerem a construção de sistemas complexos que vão desde o mapeamento de processos industriais, logística, sistemas de comunicação, fluxo de redes, escolha de rotas… e possui ampla aplicação em diversas áreas do conhecimento, tais como, engenharias, computação, genética, física, química, antropologia, linguística, etc.

A ideia de Grafo surgiu independente das diversas áreas de conhecimento, no entanto é considerada como uma área da matemática aplicada. A mais antiga menção sobre o assunto ocorreu no trabalho de Euler (pronuncia-se Óiler), no ano de 1736 para modelar e explicar um problema chamado “Pontes de Königsberg.

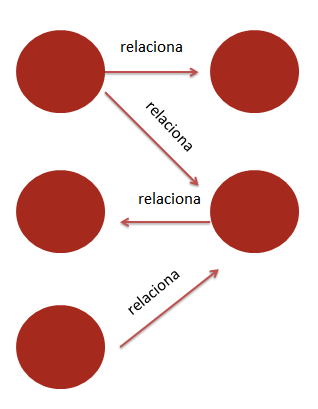


O problema consistia em verificar se seria possível percorrer todas as sete pontes da cidade passando uma única vez em cada ponte.

Euler verificou (e provou) através de um diagrama associando nós (vértices) e arcos (arestas) que não havia solução para o problema. É justamente este o conceito simples a essência usado pelo **Neo4j**.

Neste tipo de Banco de Dados, os registros são gravados em vértices (nós) que possuem propriedades definidas conforme a necessidade. Estes vértices por sua vez se relacionam com outros vértices através de arestas (arcos) que se interligam criando caminhos entre os vértices de maneira organizada com relações explicitas.

Em um banco de dados de grafos, relacionamentos são mais naturais. Temos as entidades chamadas de vértices (ou node) que são ligadas entre elas pelas arestas (ou relationships) cada um podendo guardar dados entre os relacionamentos e cada relacionamento pode ter uma direção.



A imagem mostra um exemplo da ideia de grafos. As esferas vermelhas são os vértices e as arestas seus relacionamentos. Apesar de todos aqui possuírem a indicação “relaciona”, cada relacionamento pode guardar dados diferentes sobre o relacionamento. Por exemplo: no twitter podemos ligar uma pessoa a outra pelo relacionamento de seguir. Podemos guardar junto a data de início da relação e a pessoa pode seguir de volta. Podemos também ter vértices de outros tipos, como tweets. Uma pessoa se relaciona a um tweet escrevendo ele ou dando RT.

Baseado na teoria de grafos, essas bases são um tipo dentro do mundo das bases noSQL. Ao pensar em noSQL já lembramos que esses são bancos mais rápidos (potencialmente menos seguros) e fáceis de escalar. A velocidade de se comparar um banco de dados relacional com um banco de dados de grafos vem na hora de se comparar uma busca cheia de joins index em um banco SQL e a simplicidade de se buscar relacionamentos em grafos.

**Vértice:**

Também pode ser chamado de nó é a nossa unidade de dados, um conjunto de propriedades do tipo chave valor que representam uma entidade. Um exemplo seria um usuário o Twitter:

* Nome: “Priscila Sato”
* Arroba: “MayogaX”

**Aresta:**

São os nossos relacionamentos. Eles ligam os vértices por meio de uma rede semântica. Uma aresta pode possuir um sentido, uma orientação e, se necessário, dados sobre esse relacionamento. Exemplo baseado no Twitter: vértice “Priscila” segue o vértice “Lucas” desde 2012. “Priscila” e “Lucas” são vértices ligados pelo relacionamento “segue” que possui um sentido: do primeiro ao segundo, e possui também dados: a data de início do relacionamento.

## Como manipulo acesso a dados em bases de dados de grafos?

A maioria das bases de dados de grafos disponibilizam uma API REST para você poder manipular os dados. Algumas feitas em java disponibilizam um meio amigável para acessar suas classes e fazer buscas em suas bibliotecas. No caso de Neo4J, você pode usar uma linguagem de query específica chamada cypher.

Exemplos de banco de dados de grafos existentes:

* AllegroGraph
* ArangoDb
* Bitsy (java)
* BrightstarDB (feito para .Net)
* DEX (possui api .Net e roda em windows)
* Filament
* InfiniteGraph
* InfoGrid
* HyperGraphDb
* Oracle Spatial
* Titan
* Neo4J (provavelmente o mais famoso de todos).

**Bibliografia**

<https://medium.com/accendis-tech/uma-gentil-introdu%C3%A7%C3%A3o-ao-uso-de-banco-de-dados-orientados-a-grafos-com-neo4j-ca148df2d352>

<http://www.tiselvagem.com.br/artigos-cientificos/neo4j-banco-de-dados-orientado-a-grafos/>

https://imasters.com.br/banco-de-dados/graphdb-series-o-que-e-um-banco-de-dados-de-grafos/?trace=1519021197&source=single