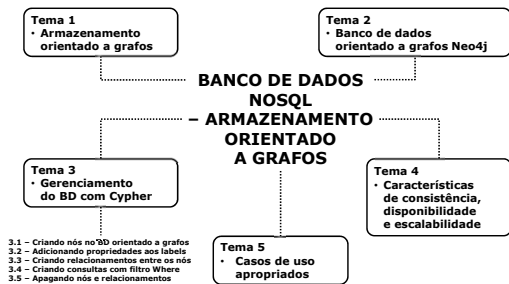


Aula 5

Bancos de Dados NoSQL

Prof. Alex Mateus Porn

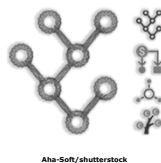
Conversa Inicial



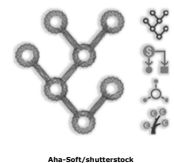
Armazenamento Orientado a Grafos

Armazenamento orientado a grafos

- Apresenta uma estrutura baseada na teoria dos grafos:
 - Vértices para representar os dados armazenados
 - Arestas para representar os relacionamentos entre os dados

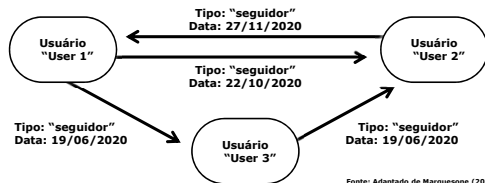


- Destacam-se em aplicações que precisam traçar os caminhos existentes nos relacionamentos entre os dados:
 - Identificar como que um conjunto de amigos está conectado em uma rede
 - Descobrir a melhor rota para se chegar a um local em menor tempo



Exemplificando...

▪ Estrutura de um banco de dados orientado a grafos



Classificação dos bancos de dados orientados a grafos

▪ Nativos:

- Usam listas de adjacências
- Cada vértice mantém referências diretas para seus vértices adjacentes formando uma espécie de microíndice para os vértices próximos
- A estrutura de grafo é considerada tanto no armazenamento físico quanto no processamento de consultas

▪ Não nativos:

- Modelam logicamente os dados como grafos
- Armazenam os dados por meio de outros modelos
 - ✓ Tabelas relacionais
 - ✓ Estrutura chave-valor

▪ Exemplo:

- No modelo relacional, as relações de triplas vértice-aresta-vértice em um grafo são armazenadas como tuplas em tabelas

Modelagem orientada a grafos

▪ Grafo simples-relacional:

- Modelo bem simples e limitado
- Todos os vértices denotam o mesmo tipo de objeto
- Todas as arestas denotam o mesmo tipo de relacionamento

▪ Grafo multirrelacional:

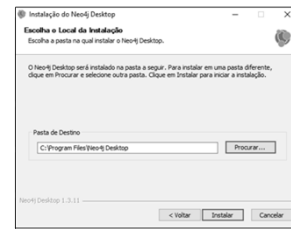
- Permite um conjunto variado de tipos de objetos e de relacionamentos
- Possibilita múltiplas relações e um maior poder de modelagem

▪ Grafo de propriedades:

- Grafo multirrelacional com atributos e arestas direcionadas
- Uma aresta pode ser direcionada e/ou rotulada e/ou valorada com um peso em um modelo
- Adicionalmente, arestas e vértices podem ter propriedades com valores associados

- Recomenda-se não instalar em diretórios que contêm "espaço" no nome:
 - "Arquivos de Programas"
- Criação dos seguintes diretórios:
 - C:\Neo4jDesktop
 - C:\Neo4jDesktop\Data

Alterando o diretório de instalação do Neo4j



<https://neo4j.com/download/>

Seleção do diretório para armazenamento dos dados da aplicação

Please choose path where you want to store application data

C:\Users\intellegi\Neo4jDesktop

Choose

Confirm

<https://neo4j.com/download/>

Registrando o Neo4j

Software registration

Neo4j Desktop is always free. Registration lets us know who has accepted this gift of graphs.

Register yourself with the following contact information.

Already registered? Add your software key here to activate this installation.

Name *

Email *

Organization *

Software key *

Software keys look like a long block of hexadecimal characters.

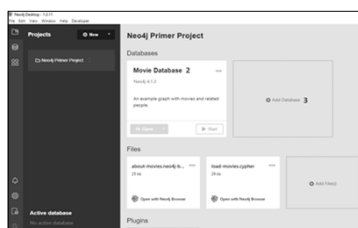
Read about our privacy policy.

Register later

Activate

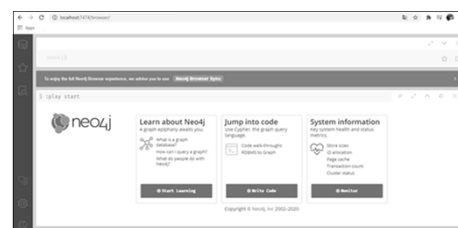
<https://neo4j.com/download/>

Interface inicial do Neo4j



<https://neo4j.com/download/>

Interface web de gerenciamento do Neo4j



<https://localhost:7474>

Gerenciamento do Banco de Dados Neo4j com Cypher

Linguagem Cypher

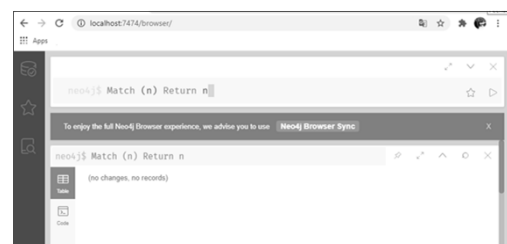
- Desenvolvida para uso exclusivo do Neo4j
- Posteriormente adotada por outros bancos de dados de grafo por meio do projeto *openCypher*
- Linguagem oficial de consultas do Neo4j e permite que se crie, modifique e procure dados em uma estrutura baseada em um grafo Meyrelles (2015)
- Como o SQL está para o modelo relacional, o Cypher está para os bancos em grafos

Comparação entre SQL e Cypher

SQL	Cypher
Create	Create
Create if not exists	Merge
Select	Match/Return
Where	Where
Update	Set
Delete	Delete/Remove

- Select * From Nome_da_Tabela
- Match (n) Return n

Execução das consultas em Cypher

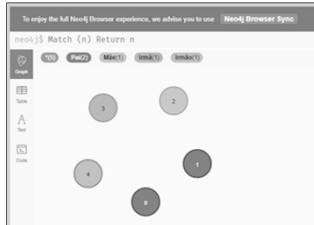


Criando nós no banco de dados Neo4j

- Criando nós isoladamente:
 - Sintaxe
 - ✓ Create (nome do nó)
 - Exemplo
 - ✓ Create (Carlos)

- Criando nós em um label:
 - Sintaxe
 - ✓ Create (nome do nó :label)
 - Exemplo
 - ✓ Create (Carlos :Pai)
 - ✓ Create (Carlos :Pessoa :Pai)

Representação dos nós no banco de dados Neo4j



Adicionando propriedades

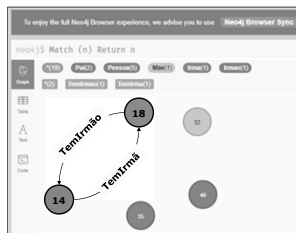
- Uma propriedade é similar a um atributo em uma tabela de um banco de dados relacional
- Sintaxe:
 - Create (n :Label :Sublabel {propriedade:'Valor'}) Return n
- Exemplo:
 - Create (n :Pessoa :Pai {nome:'Carlos', idade:'35'}) Return n
 - Create (t :Pessoa :Mãe {nome:'Maria', idade:'32'}) Return t

Criando relacionamentos entre os nós no banco de dados Neo4j

- Um relacionamento entre nós é similar a um relacionamento entre duas tabelas em um banco de dados relacional
- Sintaxe:
 - Match (p :Label), (c :Label)
 - Where p.propriedade = 'Valor' and c.propriedade = 'Valor'
 - Create (p) - [r :Relacionamento] -> (c)
 - Return p, c, r

- Exemplo:
 - Match (p :Irmã), (c :Irmão)
 - Where p.nome = 'Julia' and c.nome = 'Mario'
 - Create (p) - [r :TemIrmão] -> (c)
 - Return p, c, r

Representação dos nós e relacionamentos no banco de dados Neo4j



Criando consultas sem Cypher

- Sintaxe:
 - Match (n) Where n.propriedade = 'Valor' Return n
- Exemplo:
 - Match (n) Where n.idade = '35' Return n
 - Match (n :Pai), (y :Mãe) Where n.idade = '35' and y.idade = '32' Return n, y

Apagando nós e relacionamentos

- Apagar todos os nós com relacionamentos:
 - Match (n) Detach Delete n
- Apagar um relacionamento:
 - Match (p :Irmã) - [r :TemIrmão] -> (c :Irmão)
 - Where p.nome = 'Julia' and c.nome = 'Mario'
 - Delete r
 - Return p, c

Características de Consistência, Disponibilidade e Escalabilidade

Características


- Destacam-se a robustez, escalabilidade e o alto desempenho
- Capaz de garantir as propriedades ACID, que são uma das mais importantes características dos bancos de dados relacionais
- Todas as operações que modificam dados ocorrem dentro de uma transação, garantindo que os dados permaneçam em estado consistente
- Permite alta disponibilidade por meio de vários servidores distribuídos

Casos de Uso Apropriados

Casos de uso apropriados

- Redes sociais
- Aplicados a situações em que os dados são altamente conectados e devem ser representados a partir da forma como se conectam ou se correlacionam com outros dados
- Recomendações de compras em lojas virtuais
- Sistemas que exploram dados químicos e biológicos para detecção de padrões (Penteado et al., 2014)

Referências

- 
- **MARQUESONE, R.** Big Data: Técnicas e tecnologias para extração de valor dos dados. São Paulo: Casa do Código, 2017.
 - **MEYRELLES, M.** Banco de dados orientados a grafos com Neo4j. Accendis Tech, 2015.
 - **NEO4J.** The Neo4j Manual. Disponível em: <http://neo4j.com/docs/stable/>. Acesso em: 30 maio 2021.
 - **PENTEADO, R. R. M. et al.** Um estudo sobre bancos de dados em grafos nativos. Escola Regional de Banco de Dados ERBD, São Francisco do Sul, 2014.