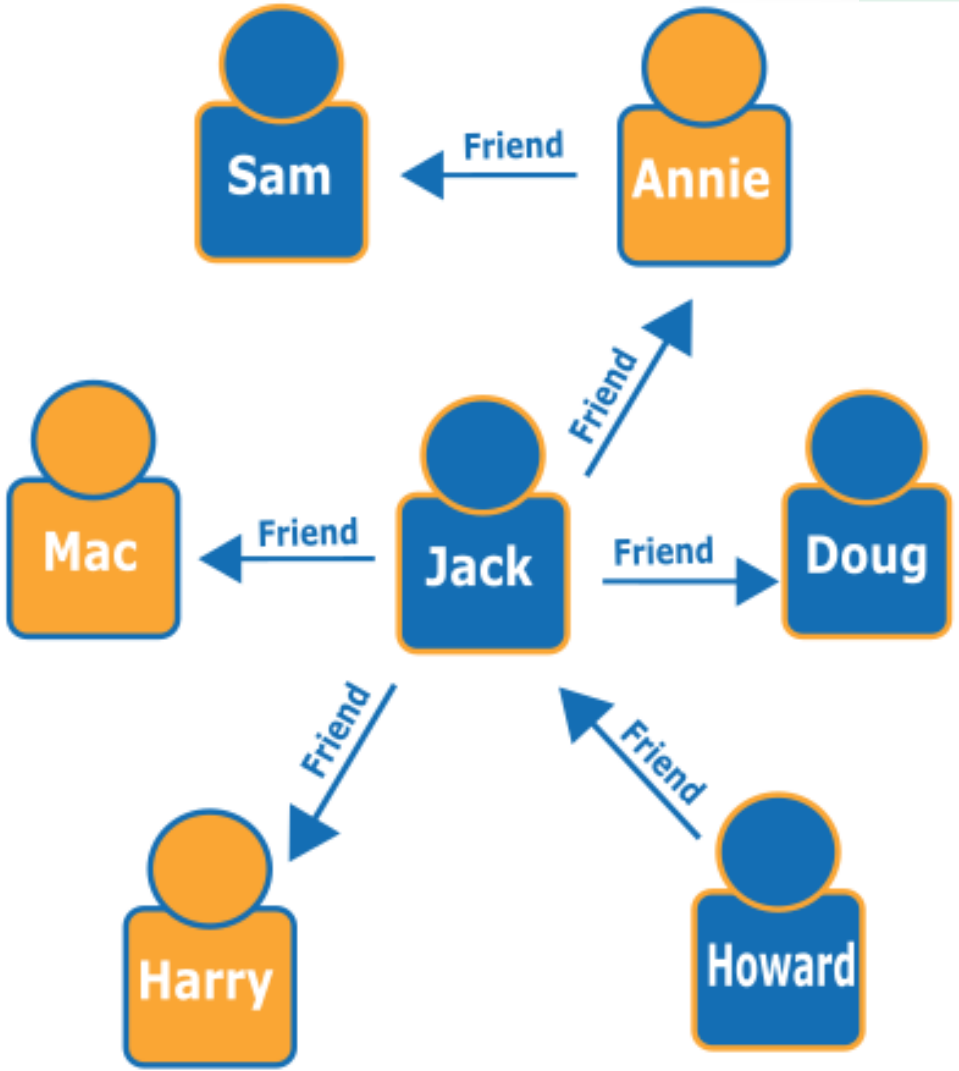
NoSQL
Grafos

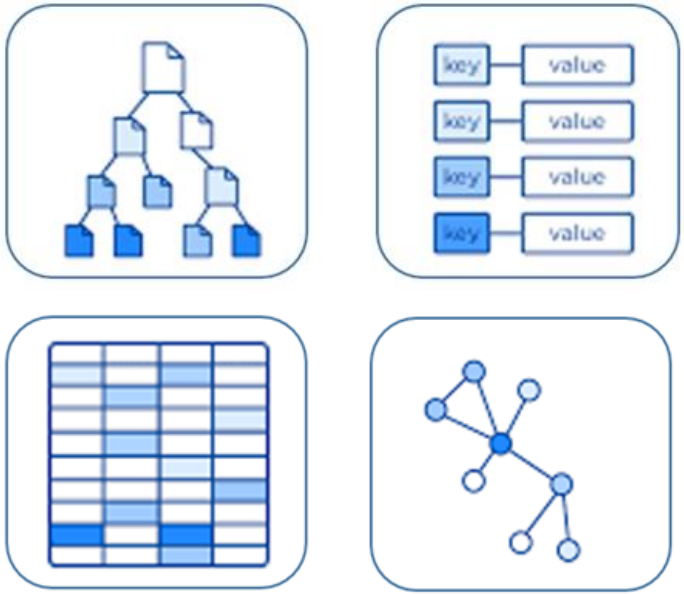


Profa. Carolina Torres

INSTITUTO FEDERAL
Pernambuco

Modelos de Dados

- Modelos de Dados mais comuns:
  - Chave-valor
  - Orientado a Documentos
  - Orientado a Colunas
  - **Grafos**



INSTITUTO FEDERAL
Pernambuco

Grafos

- Conceitos Relacionados:
  - Nó/vértices
  - Arestas/relacionamentos
  - Propriedades
  - Rótulos

INSTITUTO FEDERAL
Pernambuco

Grafos

- Conceitos sobre grafos:
  - https://www.youtube.com/watch?v=MC0u4f334ml
- Tipos de grafos:
  - https://www.inf.ufsc.br/grafos/definicoes/definicao.html

INSTITUTO FEDERAL
Pernambuco

BD orientado a Grafos

INSTITUTO FEDERAL
Pernambuco

BD orientados a grafos

Prós

- Garantia do ACID sem definição prévia de esquema.
- Dinâmico
- Rápida aprendizagem

Contras

- Nem todas as relações fazem sentido serem salvas como grafos
- Não há controle do estado atual da base

INSTITUTO FEDERAL
Pernambuco

Prática

NoSQL de Grafos

Utilizando o Neo4J

Neo4J

- Implementa o modelo grafo de forma eficiente até o nível de armazenamento.
- Fornece características de um BD completo, como ACID e suporte de cluster (tornando-o adequado para usar dados grafos em produção)

Neo4J

- Download da versão desktop disponível em: https://neo4j.com/download/
  - No momento da instalação, deu erro quando alterei a pasta destino da instalação. Só funcionou quando deixei instalar na pasta padrão selecionada pelo app.
- Utilização online: https://neo4j.com/sandbox/
- Utiliza uma linguagem própria para manipulação dos dados: Cypher

INSTITUTO FEDERAL
Pernambuco

Neo4J

Quem usa?

- NASA
- Airbnb
- ebay
- ...

Quando usar?

- Redes Sociais
- Recomendações
- ...

Concorrentes

- ArangoDB
- AllegroGraph
- Filament
- InfiniteGraph
- HyperGraphDb
- Oracle Spatial
- ...

INSTITUTO FEDERAL
Pernambuco

Terminologia

- **Labels:** uma espécie de *template* para os nós
- **Propriedades:** demonstram características ou descrições em um rótulo ou relacionamento
  - Descritos baseando-se em parâmetros chave-valor
- **Nó:** entidades baseadas em um rótulo
- **Arestas/relacionamentos:** representa uma conexão entre nós

Neo4J	OO
Label/Rótulo	Classe
Propriedade	Atributo
Nó	entidade

INSTITUTO FEDERAL
Pernambuco

Operações Básicas

CRUD

Criar dados (CREATE)

- Criação de nós

```
CREATE (x)
```

```
CREATE (a), (b)
```

- Criação de nó com label

```
CREATE (p:Aluno)
```

- Criação de nó com propriedades

```
CREATE({logradouro:'Rua X',numero:345,cidade:'Recife'})
```

- Criação de nó com label e propriedades

```
CREATE(a:Aluno{nome:'Adriana Menezes',idade:21})
```

INSTITUTO FEDERAL
Pernambuco

Criar dados (CREATE)

- Criação de relacionamentos

```
CREATE(p:Professor{nome:'Carolina Torres',idade:33,formação:'CC'})
CREATE(d1:Disciplina{nome:'BD1',CH:60})
CREATE(d2:Disciplina{nome:'BD2',CH:80})
CREATE(p)-[:ministra]->(d1)
CREATE(p)-[:ministra]->(d2)
```

```
CREATE(p:Professor{nome:'Josino Rodrigues',idade:35,formação:'EC'})
CREATE(d1:Disciplina{nome:'Web 2',CH:60})
CREATE(p)-[:ministra]->(d1)
```

INSTITUTO FEDERAL
Pernambuco

Consultar dados

- **MATCH**: equivalente ao SELECT do banco de dados relacional, é o responsável por montar um padrão de busca para que o *engine* traga os nós e relações de interesse.
  - O padrão principal de um MATCH é (um nó)-[relacionado]->(com outro nó).
  - É possível modificar este padrão base à vontade para satisfazer uma consulta, e inclusive incluindo mais nós e ou relações;

INSTITUTO FEDERAL
Pernambuco

Consultar dados

- **RETURN**: responsável por trazer as consultas no formato esperado para o cliente.
  - Saber trabalhar com o RETURN é interessante porque é possível retornar linhas ou então, um grafo completo.
  - Além disso, é possível fazer agregações como somas, contagens, médias, concatenação em listas e muito mais...

Consultando dados

- Selecionar todos os dados já criados

```
MATCH (n) RETURN n
```

- Selecionar todos os professores

```
MATCH (n:Professor) RETURN n
```

- Selecionar todos os professores e disciplinas

```
MATCH (n:Professor),(d:Disciplina)
RETURN * LIMIT 25
```

INSTITUTO FEDERAL
Pernambuco

Consultando dados

- O professor Josino mora no endereço cadastrado e ensina BD1. Criar esses relacionamentos.

```
MATCH(L{cidade:'Recife'}),(p:Professor{nome:'Josino Rodrigues'}),(d:
Disciplina{nome:'BD1'})
create(p)-[:ministra]->(d)
create(p)-[:mora_em]->(L)
```

- Quais os nomes dos professores que ensinam BD1?

```
MATCH(p:Professor)-[:ministra]->(d:Disciplina{nome:'BD1'})
return p.nome
```

INSTITUTO FEDERAL
Pernambuco

Consultando dados

- Quantos professores tem cadastrados?

```
MATCH(p:Professor)
return count(*)
```

- Qual a média de idade dos professores?

```
MATCH(p:Professor)
return avg(p.idade)
```

INSTITUTO FEDERAL
Pernambuco

Atualizar propriedades

- **SET**: muda as propriedades de uma relação ou nó.
  - **Exemplo:** Carol casou e alterou seu sobrenome, além disso deve se criar a propriedade casada

```
match(p:Professor{nome:'Carolina Torres'})set p.nome='Carolina Ribeiro',
p.casada=True
return p
```

INSTITUTO FEDERAL
Pernambuco

Apagar dados

- **DELETE**: responsável por apagar nó ou relação inteira.
  - Pode ser que você não consiga apagar um nó que esteja envolvido em outras relações. Se quiser apagar também as relações deste nó a ser excluído, use o **DETACH DELETE**.
- **REMOVE**: remove uma propriedade de um nó ou relação. Também serve para remover um label do nó.

INSTITUTO FEDERAL
Pernambuco

Apagar dados

- Apagar nós isolados

```
MATCH (p) where ID(p) = 11 OR ID(p) =9 OR ID(p) =7 OR ID(p) =10
DELETE p
```

- Apagar nós com relacionamentos

```
MATCH (L {cidade:'Recife'})
DETACH DELETE L
```

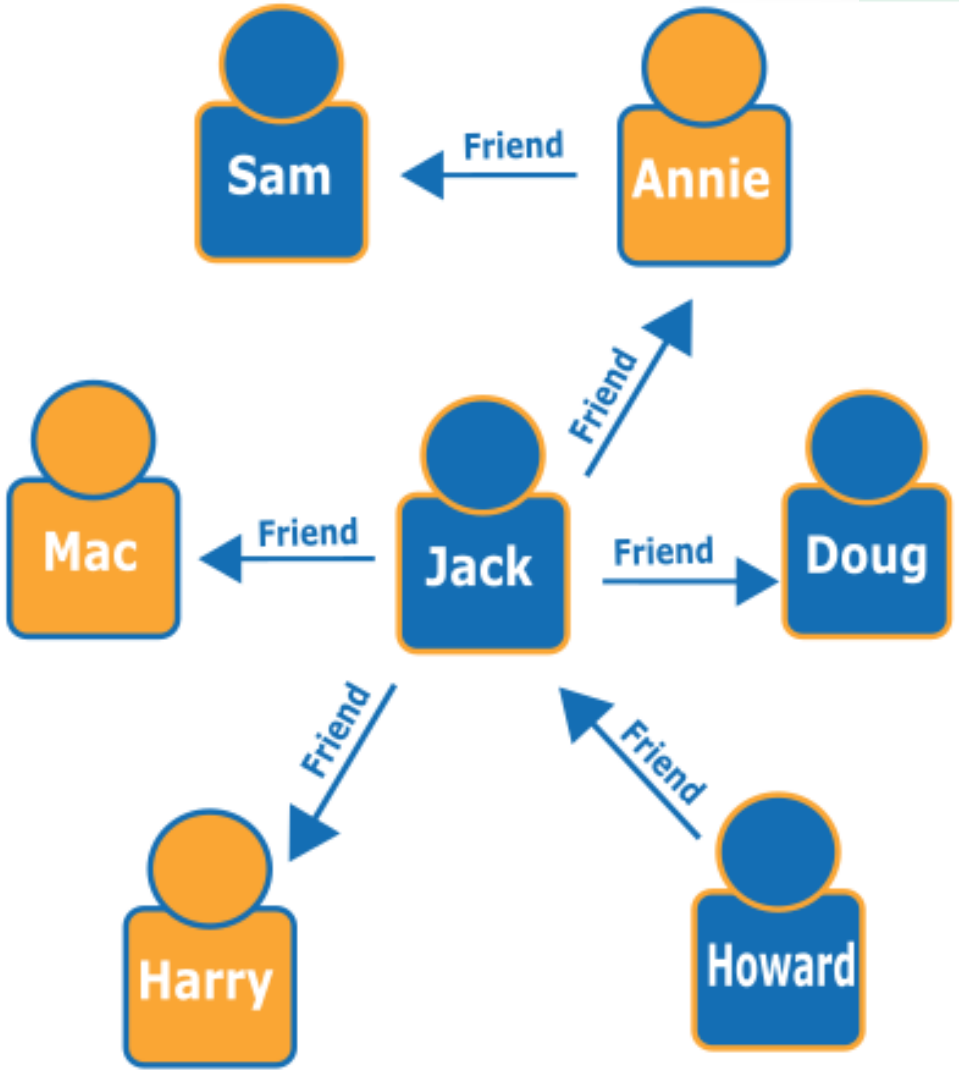
INSTITUTO FEDERAL
Pernambuco

Links interessantes

- Tutorial prático completo de um CRUD, inclusive com carga de dados: AQUI
- Manual Cypher completo: https://neo4j.com/docs/cypher-manual/current/
  - RETURN: https://neo4j.com/docs/cypher-manual/current/clauses/return/
  - Funções de agregação: https://neo4j.com/docs/cypher-manual/current/functions/aggregating/

INSTITUTO FEDERAL
Pernambuco

NoSQL
Grafos



Profa. Carolina Torres

INSTITUTO FEDERAL
Pernambuco