NEO4J E CASSANDRA

Como funcionam as ferramentas e exemplos praticos

Pedro Henrique

INTRODRUÇÃO

BANCOS DE DADOS NOSQL

- Utilizado para manipular um grande volume de dados (big data)
- Facilita a distribuiçao e manipulação de dados em cluster
- Pode ser dividida em 4 categorias; Chave-valor, grafo, colunas e documentos

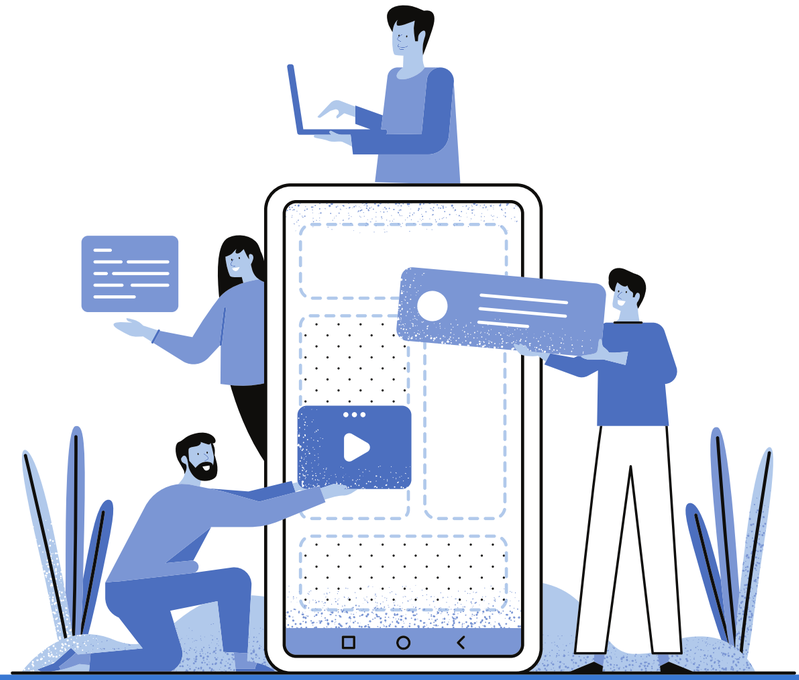
CASSANDRA

ASPECTOS GERAIS E CARACTERÍSTICAS

- É do tipo colunar, considerado os bancos que mais se assemelham aos bancos relacionais
- Basicamente considerado uma grande tabela
- Diferencia-se dos relacionais pois seus dados são armazenados em colunas e não em linhas
- Cassandra tema capacidade de lidar com altas cargas de trabalho e de suportar consultas de baixa latência

CASSANDRA

- Utiliza linguagem propria CQL (cassandra query language)
- Se assemelha com SQL e foi projetada em java



EXEMPLO PRATICO

PROBLEMA DE NEGOCIO

- Uma instituiçao academica precisa cadastrar seus alunos e professores

```
$ cqlsh
cqlsh> CREATE KEYSPACE unipe WITH REPLICATION = {‘class’:
‘SimpleStrategy’, ‘replication_factor’: ‘3’}
cqlsh> use unipe
```

1) Criando o que seria equivalante ao database

2) Criando a column family, ou também chamada de tabela

```
cqlsh:unipe> CREATE TABLE cadastro (
... id uuid PRIMARY KEY,
... nome text,
... cargo text);
```

EXEMPLO PRATICO

PROBLEMA DE NEGOCIO


```
cqlsh:unipe> INSERT INTO cadastro (id, nome, cargo) VALUES
(1a8d6a80-33df-11e5-a151-feff819cdc9f, ‘Thyago’, ‘professor’);
cqlsh:unipe> INSERT INTO cadastro (id, nome, cargo) VALUES
(1a8d649a-33df-11e5-a151-feff819cdc9f, ‘Afonso’, ‘professor’);
cqlsh:unipe> INSERT INTO cadastro (id, nome, cargo) VALUES
(a70ca7ff-6d57-4f89-be89-08421c432bb7, ‘Fernanda’, ‘aluna’);
cqlsh:unipe> INSERT INTO cadastro (id, nome, cargo) VALUES
(04b57f0e-33df-11e5-a151-feff819cdc9f, ‘Theo’, ‘aluna’);
cqlsh:unipe> INSERT INTO cadastro (id, nome, cargo) VALUES
(c4f408dd-00f3-488e-8800-050d2775bbc7, ‘Sophia’, ‘funcionario’);
cqlsh:unipe> INSERT INTO cadastro (id, nome, cargo) VALUES
(04b57c98-33df-11e5-a151-feff819cdc9f, ‘Leonardo’, ‘funcionario’);
```

3) Insirindo os dados de nome e cargo na tabela criada, e o id obrigatorio

4) fazendo as consultas, updates e deletes necessarios

```
cqlsh:unipe> UPDATE cadastro SET cargo = ‘funcionario’ WHERE
id = 04b57f0e-33df-11e5-a151-feff819cdc9f;


cqlsh:unipe> DELETE from cadastro WHERE id = 1a8d649a-33df-11e5-a151-feff819cdc9


cqlsh:unipe> SELECT * FROM cadastro WHERE cargo = ‘professor’;
cqlsh:unipe> SELECT * FROM cadastro WHERE cargo = ‘aluno’;
```


neo4j

INTRODUÇÃO AO NEO4J

- O Neo4J é um banco de dados não relacional, orientado à grafos. Ele e os demais bancos noSql se distinguem dos bancos tradicionais por não possuírem um esquema fixo, como linhas e colunas. Sendo sua principal característica, simplificar a análise dos dados através de grafos e relacionamentos entre eles.

LINGUAGEM

- A sua linguagem chama-se cypherQuery,
e a curva de aprendizagem não é tão grande
devido a sua semelhança com o SQL.
- Ela permite que os usuários realizem consultas complexas de maneira simples e intuitiva, além de ter uma sintaxe clara e legível que torna fácil o seu entendimento.

COMANDOS

- CREATE - Cria nós e relacionamentos;
- MATCH - utilizado para buscar padrões em um banco de dados de grafos.
- RETURN - Mostra (retorna) o valor criado;
- WHERE - Seleciona dados através de uma determinada condição,filtra os resultados;
- SET - Atualização;
- REMOVE - Deletar;

EX DE COMANDOS (CREATE)

```
CREATE (alice:Aluno {nome: 'Alice'}),
(bob:Aluno {nome: 'Bob'}),
(cursoMatematica:Curso {nome:
'Matemática'}),
(cursoHistoria:Curso {nome: 'História'}),
(turmaMatematica:Turma {codigo: 'MAT101'}),
(turmaHistoria:Turma {codigo: 'HIS201'}),
(professorJoao:Professor {nome: 'João'}),
(professorMaria:Professor {nome: 'Maria'})
```

EX DE COMANDOS (MATCH + CREATE)

```
MATCH (alice:Aluno {nome: 'Alice'}), (bob:Aluno {nome: 'Bob'}),
(cursoMatematica:Curso {nome: 'Matemática'}),
(cursoHistoria:Curso {nome: 'História'})

CREATE (alice)-[:MATRICULADO_EM]->(cursoMatematica),
(bob)-[:MATRICULADO_EM]->(cursoHistoria)
```

EX DE COMANDOS (WHERE)

```
MATCH (professor:Professor)-[:MINISTRA]->(:Turma)-
[:OFERECIDO_EM]->(curso:Curso)

WHERE curso.nome CONTAINS 'Matemática'
RETURN DISTINCT professor.nome
```

VANTAGENS DO USO DO NEO4J

Modelagem Natural: Representação direta e intuitiva de dados conectados.
Consultas Eficientes: Desempenho superior em consultas envolvendo múltiplos relacionamentos e travessias complexas.
Flexibilidade: Facilidade em adicionar novos tipos de relacionamentos e entidades sem reestruturar o esquema.
Ferramentas Avançadas: Conjunto robusto de ferramentas para visualização, análise e integração de dados de grafos.

neo4j