Neo4j - Importação de Dados

Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação

Agenda

Importando Dados de um Arquivo CSV

- O arquivo CSV a ser usado deve ter as seguintes características:
  1. a codificação de caracteres deve ser UTF-8;
  2. a terminação da linha final é dependente do sistema, por exemplo, é \n no Unix ou \r\n no Windows;
  3. o terminador de campo padrão é **vírgula**;
  4. *Strings* entre aspas são permitidas no arquivo CSV e as aspas são eliminadas ao ler os dados;
  5. o caractere para citação de *String* é aspas duplas;
  6. uma aspa dupla deve estar em uma *String* entre aspas e ter escape, seja com o caractere de escape ou uma segunda aspa dupla.

Arquivo CSV local

artists.csv None

```
1,ABBA,1992
2,Roxette,1986
3,Europe,1979
4,The Cardigans,1992
```

Query Cypher

```
LOAD CSV FROM 'file:///artists.csv' AS line
CREATE (:Artist {name: line[1], year: toInteger(line[2])})
```

Arquivo CSV Remoto

data.neo4j.com/bands/artists.csv None — Copy to Clipboard

```
1,ABBA,1992
2,Roxette,1986
3,Europe,1979
4,The Cardigans,1992
```

Query Cypher — Copy to Clipboard — Run in Neo4j Browser

```
LOAD CSV FROM 'https://data.neo4j.com/bands/artists.csv' AS line CREATE (:Artist {name: line[1],
  year: toInteger(line[2])})
```

- **Obs:** Isso pode ser aplicado a todas os comandos de importação

Arquivo CSV com Cabeçalho

- Quando seu arquivo CSV tem cabeçalhos, você pode visualizar cada linha do arquivo como sendo um objeto.

artists-with-headers.csv None — Copy to Clipboard

```
Id,Name,Year
1,ABBA,1992
2,Roxette,1986
3,Europe,1979
4,The Cardigans,1992
```

Query Cypher — Copy to Clipboard — Run in Neo4j Browser

```
LOAD CSV WITH HEADERS FROM 'file:///artists-with-headers.csv' AS line
CREATE (:Artist {name: line.Name, year: toInteger(line.Year)})
```

Arquivo CSV com Delimitador Personalizado

- Às vezes, seu arquivo CSV tem outros delimitadores de campo além de vírgulas.
- Você pode especificar qual delimitador seu arquivo usa, usando **FIELDTERMINATOR**.
- A representação hexadecimal da codificação de caracteres unicode pode ser usada se precedida por \u.

artists-fieldterminator.csv None — Copy to Clipboard

```
1;ABBA;1992
2;Roxette;1986
3;Europe;1979
4;The Cardigans;1992
```

Query Cypher — Copy to Clipboard — Run in Neo4j Browser

```
LOAD CSV FROM 'file:///artists-fieldterminator.csv' AS line FIELDTERMINATOR ';'
CREATE (:Artist {name: line[1], year: toInteger(line[2])})
```

Importando dados com caracteres de escape

- Neste exemplo, ambos temos aspas adicionais em torno dos valores, bem como aspas com escape dentro de um valor

artists-with-escaped-char.csv None — Copy to Clipboard

```
"1","The ""Symbol""","1992"
```

Query Cypher — Copy to Clipboard — Run in Neo4j Browser

```
LOAD CSV FROM 'file:///artists-with-escaped-char.csv' AS line
CREATE (a:Artist {name: line[1], year: toInteger(line[2])})
RETURN
  a.name AS name,
  a.year AS year,
  size(a.name) AS size
```

Usando *linenumber()* com LOAD CSV

- Para determinados cenários, como depurar um problema com um arquivo csv, pode ser útil obter o número da linha atual em que LOAD CSV está operando.
- A função *linenumber()* fornece exatamente isso ou **null**

artists.csv None — Copy to Clipboard

```
1,ABBA,1992
2,Roxette,1986
3,Europe,1979
4,The Cardigans,1992
```

Query Cypher — Copy to Clipboard — Run in Neo4j Browser

```
LOAD CSV FROM 'file:///artists.csv' AS line
RETURN linenumber() AS number, line
```

```
+-------------------------------------+
| number | line                       |
+-------------------------------------+
| 1      | ["1","ABBA","1992"]        |
| 2      | ["2","Roxette","1986"]     |
| 3      | ["3","Europe","1979"]      |
| 4      | ["4","The Cardigans","1992"] |
```

Trabalhando com Grandes Quantidades de Dados

- Se o arquivo CSV contiver um número significativo de linhas (aproximando-se de centenas de milhares ou milhões), CALL IN TRANSACTIONS pode ser usado para instruir o Neo4j a confirmar uma transação

Query Cypher | Copy to Clipboard | Run in Neo4j Browser

```
LOAD CSV FROM 'file:///artists.csv' AS line
CALL {
  WITH line
  CREATE (:Artist {name: line[1], year: toInteger(line[2])})
} IN TRANSACTIONS
```

- **Obs:** Para grandes quantidades de dados é recomendar criar índices. Ex: CREATE INDEX ON :Artist(code)

Definindo o número de linhas

- Você pode definir o número de linhas como no exemplo, onde é definido como 500 linhas

Query Cypher | Copy to Clipboard | Run in Neo4j Browser

```
LOAD CSV FROM 'file:///artists.csv' AS line
CALL {
  WITH line
  CREATE (:Artist {name: line[1], year: toInteger(line[2])})
} IN TRANSACTIONS OF 500 ROWS
```

Usando *file()* com LOAD CSV

- Para determinados cenários, como depurar um problema com um arquivo csv, pode ser útil obter o caminho absoluto do arquivo que LOAD CSVestá operando.
- A função *file()* fornece exatamente isso ou **null**

artists.csv None — Copy to Clipboard

```
1,ABBA,1992
2,Roxette,1986
3,Europe,1979
4,The Cardigans,1992
```

Query Cypher — Copy to Clipboard — Run in Neo4j Browser

```
LOAD CSV FROM 'file:///artists.csv' AS line
RETURN DISTINCT file() AS path
```

Criando Arestas oriundas de Arquivo CSV

- Quando um arquivo CSV conecta dois outros arquivos, pode ser usado o MATCH para encontrar nós que já estão no grafo e ligá-los.
- Por exemplo, no código a seguir é lida uma tabela que indica drogas que foram usadas para tratar doenças.
  - Para cada ocorrência, é criada uma aresta entre a droga e a respectiva doença:

```
LOAD CSV WITH HEADERS FROM 'https://raw.githubusercontent.com/santanche/lab2learn/master/data/faers-2017/drug-use.csv' AS
MATCH (d:Drug {code: line.codedrug})
MATCH (p:Pathology {code: line.codepathology})
CREATE (d)-[:Treats {person: line.idperson}]->(p)
```

Veja o resultado:

```
MATCH (d)-[:Treats]->(p)
RETURN d, p
LIMIT 50
```

MERGE

- No caso acima, não computamos o número de vezes que uma relação acontece.
- Portanto, iremos refazer as arestas de outra maneira, usando uma sentença MERGE.

1. Primeiro apagaremos as arestas feitas anteriormente:

```
MATCH (d:Drug)-[t:Treats]->(p:Pathology)
DELETE t
```

MERGE

2. Depois criaremos uma *query* para refazê-las usando o MERGE.
   - Ela define um atributo peso (*weight*) na relação e incrementa um para cada relação encontrada:

```
LOAD CSV WITH HEADERS FROM 'https://raw.githubusercontent.com/santanche/lab2learn/master/data/faers-2017/drug-use.csv' AS
MATCH (d:Drug {code: line.codedrug})
MATCH (p:Pathology {code: line.codepathology})
MERGE (d)-[t:Treats]->(p)
ON CREATE SET t.weight=1
ON MATCH SET t.weight=t.weight+1
```

MERGE

Agora que temos peso (*weight*) na aresta podemos fazer as seguintes consultas:

3. Vamos inspecionar as relações que têm peso maior que cinquenta:

```
MATCH (d)-[t:Treats]->(p)
WHERE t.weight > 50
RETURN d,p
```

4. . . . e aquelas com peso maior que vinte:

```
MATCH (d)-[t:Treats]->(p)
WHERE t.weight > 20
RETURN d,p
```

Neo4j - Importação de Dados

Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação