**从csv中导入数据到Neo4j**

**Neo4j版本号：neo4j-community-4.2.6**

#删除全部数据方法一 ：

match (n) detach delete n

#删除数据方法二:

MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r

#删除某一个Node Label，比如删除"teacher":

match(n:teacher) delete n

**一、导入文件说明**

在neo4j的配置文件conf/neo4j.conf中有一行代码：

dbms.directories.import=import

即导入csv文件时，只能从${NEO4J\_HOME}/import中导入已有的文件，如果注释后可则可以从任意路径读取文件，官方不推荐这样使用，毕竟要保证安全性。我们将在接下来的案例中举例说明。

另外有一样注释#dbms.security.allow\_csv\_import\_from\_file\_urls=true即不允许从远程URL来load csv。

**二、导入方式**

（1）Cypher CREATE 语句，为每一条数据写一个CREATE。

（2）Cypher LOAD CSV 语句，将数据转成CSV格式，通过LOAD CSV读取数据。

（3）官方提供的 neo4j-import 工具。

（4）官方提供的Java API —— Batch Inserter。

**三、Cypher CREATE**

Cypher CREATE这种方式就是通过创建节点和关系来生成数据的。

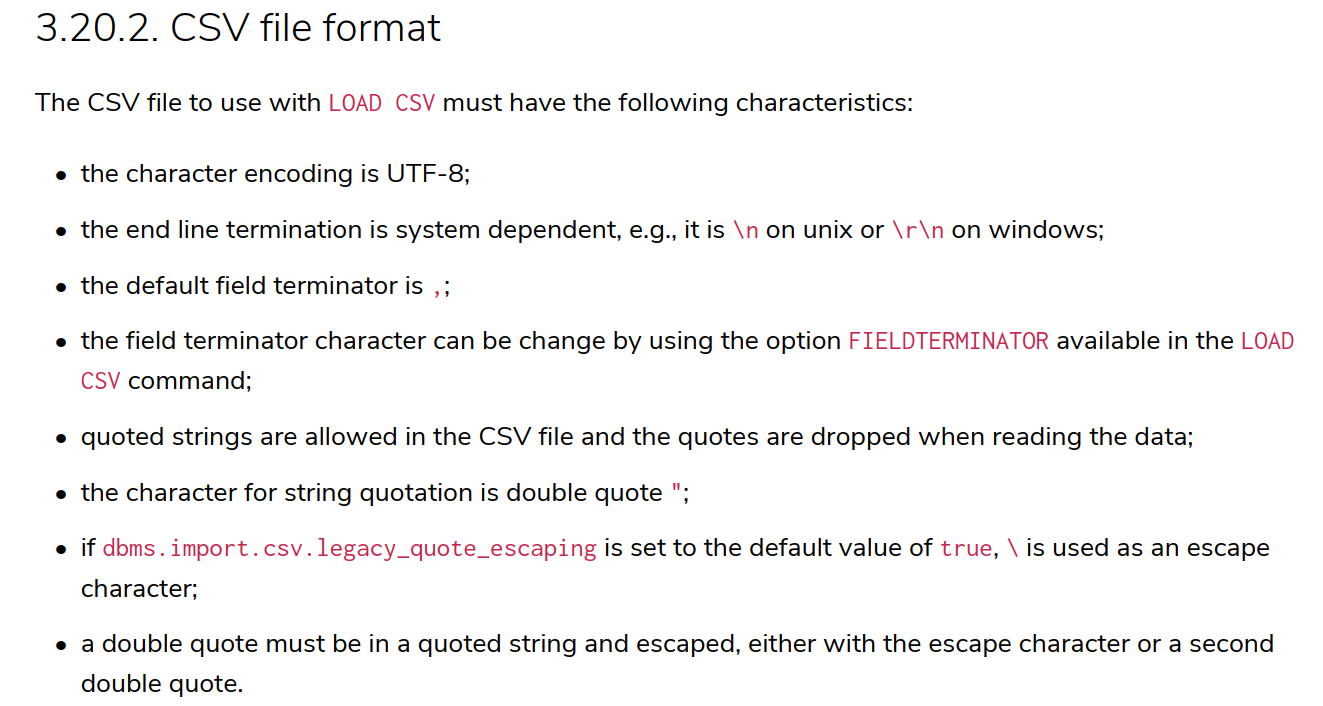
**四、LOAD CSV方式**

导入的注意事项：

（1）导入文件必须是csv格式，文件可以是本地的，或通过http、https、ftp等url指定的位置。

（2）如果文件中有中文，可使用VSCode或NotePad++等软件修改为utf-8格式文件，防止导入的数据中出现中文乱码。

官方文档的CSV文件说明：



假设我们有文件：D:\neo4j-community-4.2.6\import\files\_3\node.csv。

文件的内容大致如下：

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

可以看出上述文件文件是存在于neo4j的安装目录import下的file\_3中：此时我们的导入语句可以为：

:auto using periodic commit 1000 LOAD CSV

FROM 'file:///file\_3/artists.csv' AS line

CREATE (:Artist { name: line[1], year: toInteger(line[2])})

**说明：**

（1）我们的导入路径为: file:/files\_3/artists.csv这里没有填写的完整的路径是因为我们在neo4j的配置文件中设置了dbms.directories.import=import。

（2）环境变量设置了neo4j的安装目录(如：D:\neo4j-community-4.2.6\bin)

**如果我们注释那一行代码，且不设置环境变量我们可以这样导入数据：**

:auto using periodic commit 10000 LOAD CSV

FROM 'file:///C:/Users/Desktop/neo4j/artrists.csv' AS line

CREATE (:Artist { name: line[1], year: toInteger(line[2])})

即我们可以导入任意位置的满足条件的CSV文件。

如果想导入远程的文件可以这样：

LOAD CSV FROM 'http://data.neo4j.com/bands/artists.csv'

AS line CREATE (:Artist {name: line[1],year: toInteger(line[2])})

**包含标题的CSV**

上述导入的CSV文件是不含头标题的，使用LOAD CSV导入含有表头的CSV文件案例如下：

LOAD CSV WITH HEADERS FROM 'file:///file\_3/artists-with-headers.csv'

AS line CREATE (:Artist {name: line.Name, year: toInteger(line.Year)})

如果我们的CSV文件不是逗号分割的，比如使用;分割的，此时可以使用这样的语句：

artists.csv文件以“;”分割时：

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

LOAD CSV FROM 'file:///artists.csv' AS line

FIELDTERMINATOR ';' CREATE (:Artist {name: line[1], year: toInteger(line[2])})

另外，当我们的数据很大的时候，官方提供了一种导入数据的方式即：

USING PERIODIC COMMIT LOAD CSV FROM xxxfile

**五、LOAD　CSV导入文件总结：**

1.using periodic commit的作用是分段式创建顶点，数值为1000即读取1000行数据就写入数据库中，分行读取的目的是就是防止溢出内存导致的错误。

2.create语句可以替换成merge，防止导入的数据重复。

3.文件中存在其他的引用符号的时候，此处就不在阐述，请看官方文档。

4.适用场景数据量在1000w內

5.速度：一般 5000/s

6.优点：可以加载本地/远程CSV;可实时插入

7.缺点: 导入速度较慢;需要将数据转换成csv

**六、neo4j-admin import导入CSV**

1.确认neo4j已经加入环境变量(或者保证一下步骤能进行即可)

2.cd 到neo4j路径下的import所在目录

3.确认import 路径下的文件是否存在

4.导入之前我们需要将neo4j服务停止，即neo4j stop .

5.这里使用的是4.2社区版本neo4j导入的结果的（cmd窗口输入）语句如下：

neo4j-admin import --database=neo4jcsv --delimiter=";" --array-delimiter="|" --quote="'" --nodes=import/files\_2/movies2.csv --nodes=import/files\_2/actors2.csv --relationships=import/files\_2/roles2.csv --multiline-fields=true

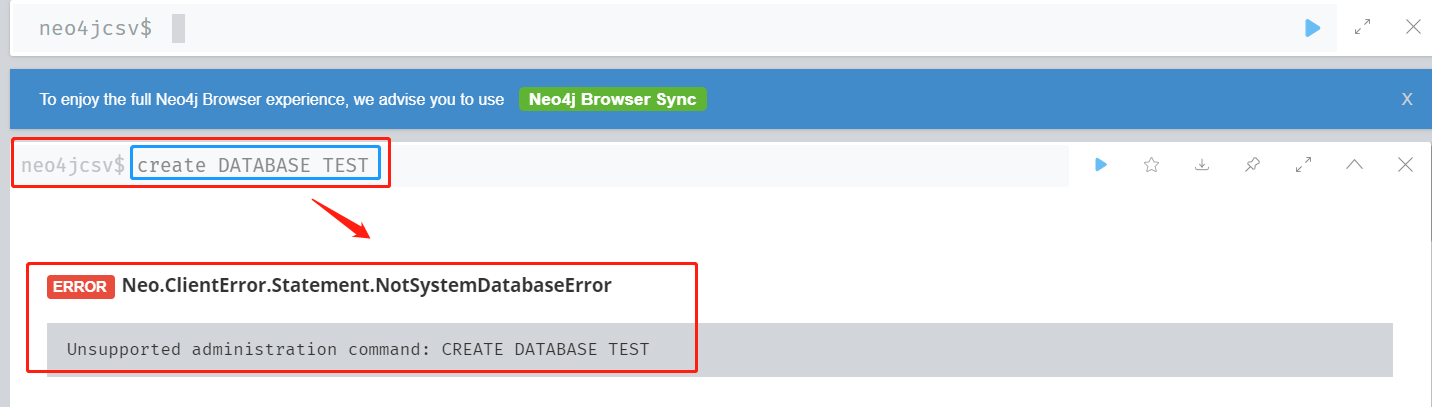
如果没有加入环境变量则可以这样：

D:\neo4j-community-4.2.6\bin\neo4j-admin import --database=neo4jcsv --delimiter=";" --array-delimiter="|" --quote="'" --nodes=import/files\_2/movies2.csv --nodes=import/files\_2/actors2.csv --relationships=import/files\_2/roles2.csv --multiline-fields=true

另外大家注意到上述语句指明了数据库：--database=neo4jcsv此种方式导入数据将使用新的数据库，此处为neo4jcsv。 但是这种方式导入数据之后，我们是看不见我们的数据的，因为我们默认使用的数据库是neo4j，而我们导入的数据是到neo4jcsv中的。

由于关键原因：neo4j社区版本不支持建多个数据库（不能建库）。

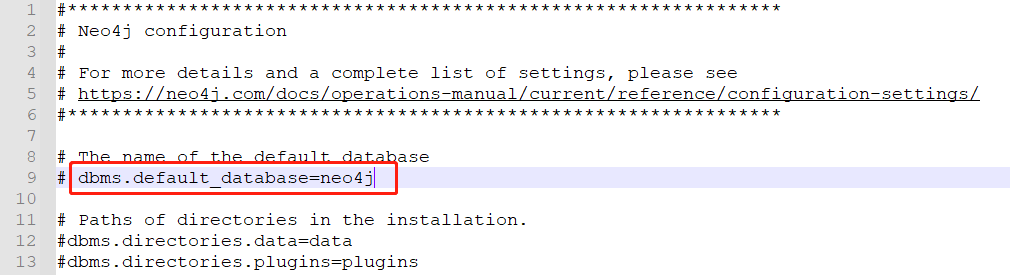
运行：create DATABASE TEST 我们执行是会报错的：



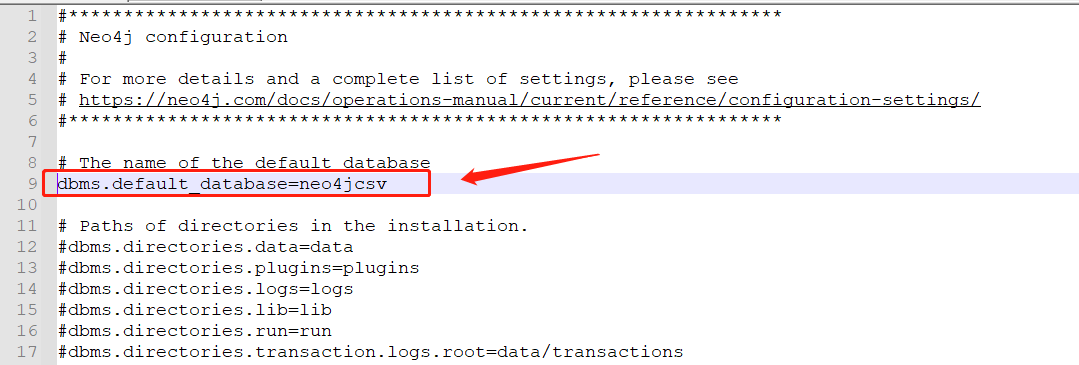
**七、neo4j-admin import导入CSV后文件数据的展示**

创建的数据该怎么在网页中展示呢？

默认的数据库如下。查看配置文件\conf\neo4j.cof



我们更改这一行并取消注释：



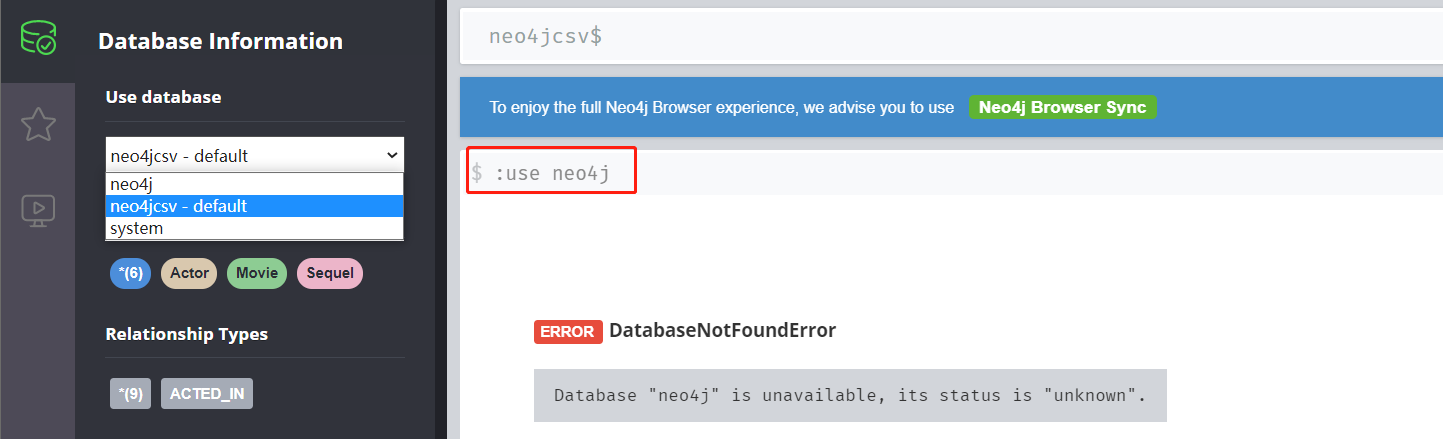
然后重启：neo4j restart

之后更新一下服务：neo4j update-service

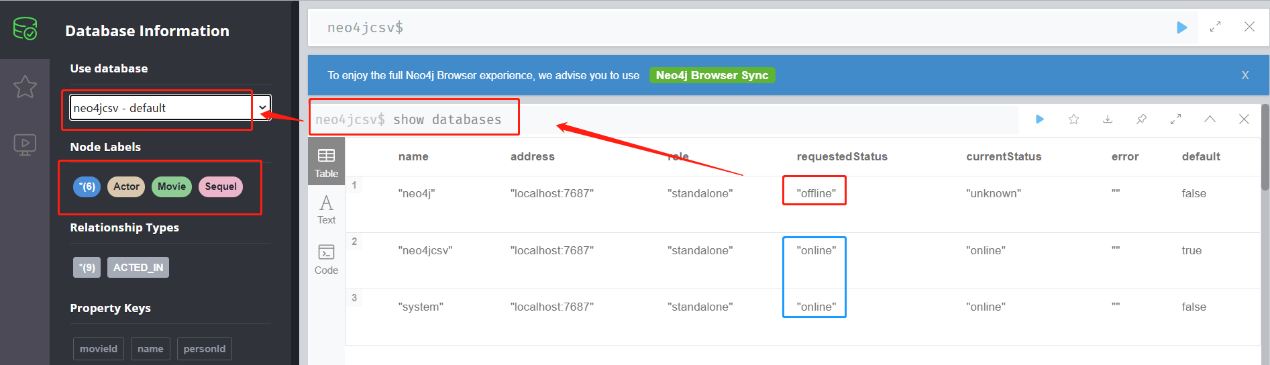
然后neo4j restart 即可

之后我们在web端打开，就可以看到数据文件了，可以看出：

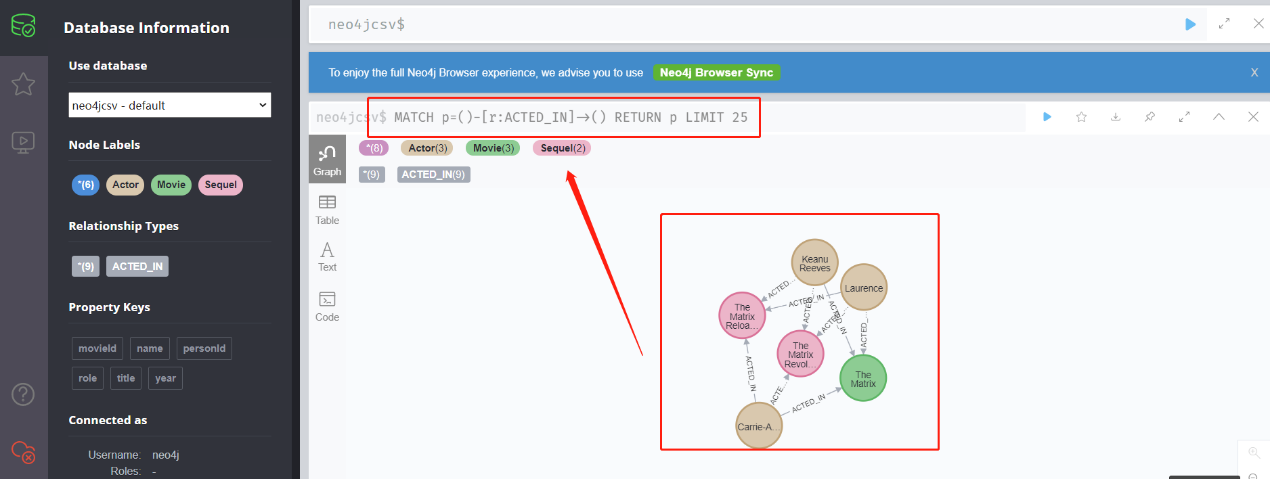
1. 默认的neo4j 数据库已经不可用了



1. 左侧出现三个默认的数据库，此时默认的为neo4jcsv
2. 可以使用SHOW DATABASES 查看数据库情况：此时共有三个数据库，之前默认的数据量neo4j已经离线。



4. 此时我们查看一下导入的数据,可以看出我们使用csv文件就可以导入数据了，笔使用create语句创建来的快。



**八、neo4j-admin import导入CSV总结：**

1.适用场景：千万以上 nodes导入

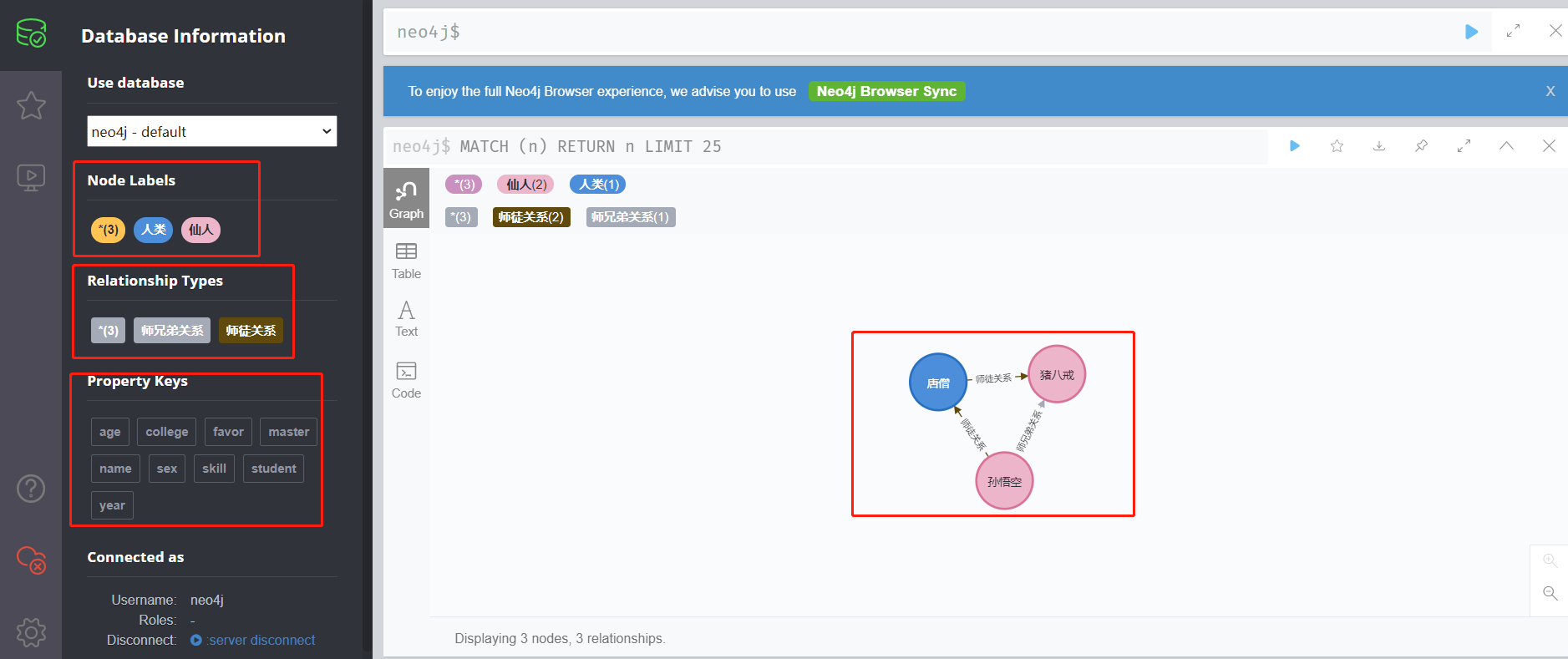
2.速度：非常快 (数万nodes/s)

3.优点：占用更少的资源

4.缺点: 需要转成CSV；必须停止neo4j；只能生成新的数据库，而不能在已存在的数据库中插入数据。

1. **from** py2neo **import** Graph, Node, Relationship
3. # 连接neo4j数据库
4. graph = Graph("http://localhost:7474", username="neo4j", password='KG-neo4j')
6. # 清除neo4j中原有的结点等所有信息
7. graph.delete\_all()
9. # 创建结点
10. # 该结点语义类型是仙人  结点名字是孙悟空  也是它的属性
11. node1 = Node('仙人', name='孙悟空')
12. node2 = Node('仙人', name='猪八戒')
13. node3 = Node('人类', name='唐僧')

16. # 添加相应的属性
17. node1['skill'] = "72般变化"
18. node2['favor'] = "eating"
19. node3['year'] = 1986
21. # 把结点实例化 在Neo4j中显示出来
22. graph.create(node1)
23. graph.create(node2)
24. graph.create(node3)
26. # 创建关系
27. maojor = Relationship(node1, '师兄弟关系', node2)
28. friends = Relationship(node1, '师徒关系', node3)
29. maojor1 = Relationship(node3, '师徒关系', node2)
31. # 把关系实例化 在Neo4j中显示出来
32. graph.create(maojor)
33. graph.create(maojor1)
34. graph.create(friends)



2. 导入文件

Import\_neo4j\_test.csv内容如下：

Id,name,性别,技能

1,程咬金,男,回血

2,李元芳,男,飞镖

3,安琪拉,女,火球

4,兰陵王,男,隐身

1. **from** py2neo **import** Graph, Node, Relationship
2. **import** csv
3. # 连接neo4j数据库，输入地址、用户名、密码
4. graph = Graph("http://localhost:7474", username="neo4j", password='KG-neo4j')
5. graph.delete\_all()

8. **def** import\_scvfile\_to\_neo4j():
10. with open(r"C:\Users\LEGION\Desktop\neo4j使用\import\_neo4j\_test.csv", 'r', encoding="UTF-8") as f:
11. reader = csv.reader(f)
12. data = list(reader)
13. **print**(f'数据行数为：{len(data)}')
15. **for** i **in** range(1, len(data)):
16. node = Node('person', id=data[i][0], name=data[i][1])
17. relation = Node('sex', sex=data[i][2])
18. relation1 = Node('skill', name=data[i][3])
20. graph.create(node)
21. graph.create(relation)
22. graph.create(relation1)
24. sex = Relationship(node, '性别', relation)
25. skill = Relationship(node, '技能', relation1)
27. graph.create(sex)
28. graph.create(skill)
30. import\_scvfile\_to\_neo4j()

导入数据的方式：

neo4j的导入方式有很多，大概总结了一下：

（1）Cypher CREATE 语句，为每一条数据写一个CREATE

（2）Cypher LOAD CSV 语句，将数据转成CSV格式，通过LOAD CSV读取数据。

（3）官方提供的Java API —— Batch Inserter

（4）大牛编写的 Batch Import 工具

（5）官方提供的 neo4j-import 工具

**python 实现查询Neo4j多节点的多层关系**

https://neo4j.com/books/

<https://neo4j.com/docs/operations-manual/current/tutorial/neo4j-admin-import/>

https://www.jianshu.com/p/3acbf66bd0d0

https://zhuanlan.zhihu.com/p/93746655

https://cloud.tencent.com/developer/article/1434905

https://blog.csdn.net/qq\_32532663/article/details/107027458?utm\_medium=distribute.pc\_relevant.none-task-blog-2~default~baidujs\_title~default-1.no\_search\_link&spm=1001.2101.3001.4242