

数据导入neo4j方法总结

#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"gzh"：

`match(n:gzh) delete n`

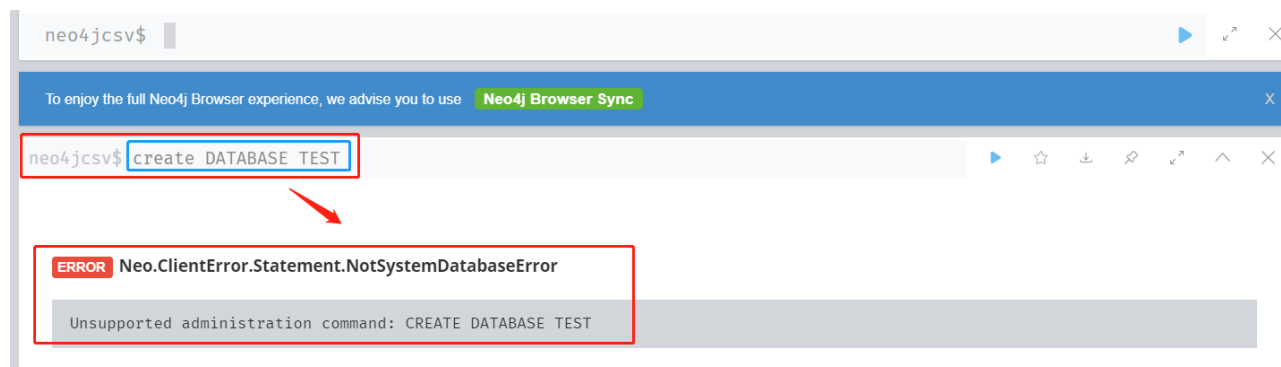
一、neo4j-admin import导入CSV文件说明

1. 确认neo4j已经加入环境变量
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：

```
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
```

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，我们执行是会报错的： 运行：create DATABASE TEST



二、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？

关键原因：neo4j社区版本不支持建多个数据库。默认的数据库如下。查看配置文件 \conf\neo4j.cof

```
1 #*****
2 # Neo4j configuration
3 #
4 # For more details and a complete list of settings, please see
5 # https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/
6 #*****
7
8 # The name of the default database
9 dbms.default_database=neo4j
10
11 # Paths of directories in the installation.
12 #dbms.directories.data=data
13 #dbms.directories.plugins=plugins
```

我们更改如下，并取消注释：

```
1 #*****
2 # Neo4j configuration
3 #
4 # For more details and a complete list of settings, please see
5 # https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/
6 #*****
7
8 # The name of the default database
9 dbms.default_database=neo4jcsv
10
11 # Paths of directories in the installation.
12 #dbms.directories.data=data
13 #dbms.directories.plugins=plugins
14 #dbms.directories.logs=logs
15 #dbms.directories.lib=lib
16 #dbms.directories.run=run
17 #dbms.directories.transaction.logs.root=data/transactions
```

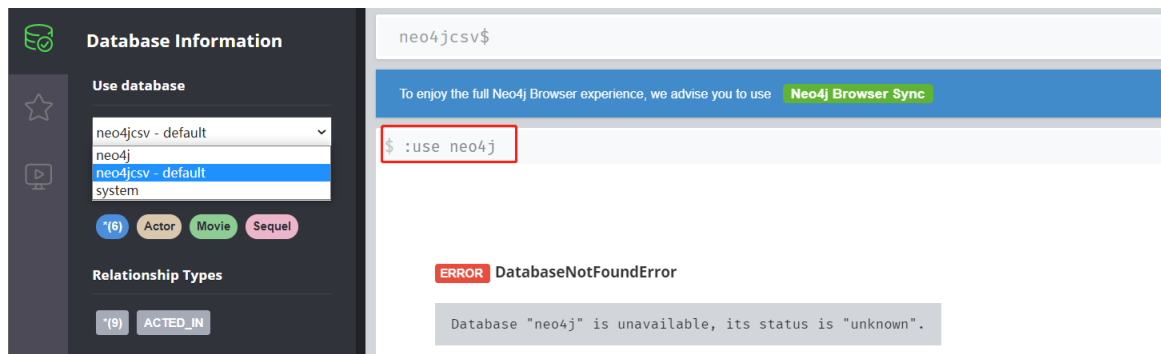
然后重启：neo4j restart

之后更新一下服务：neo4j update-service

然后neo4j restart 即可

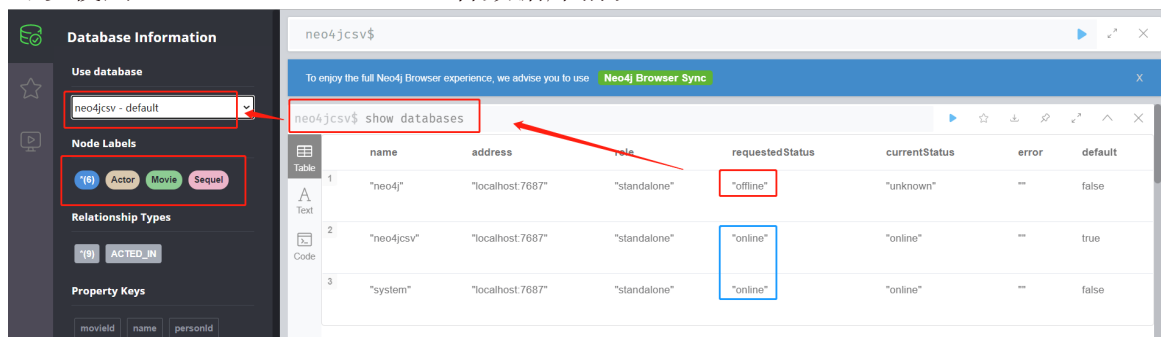
之后我们在web端打开，就可以看到数据文件了：
可以看出：

1. 默认的neo4j 数据库已经不可用了



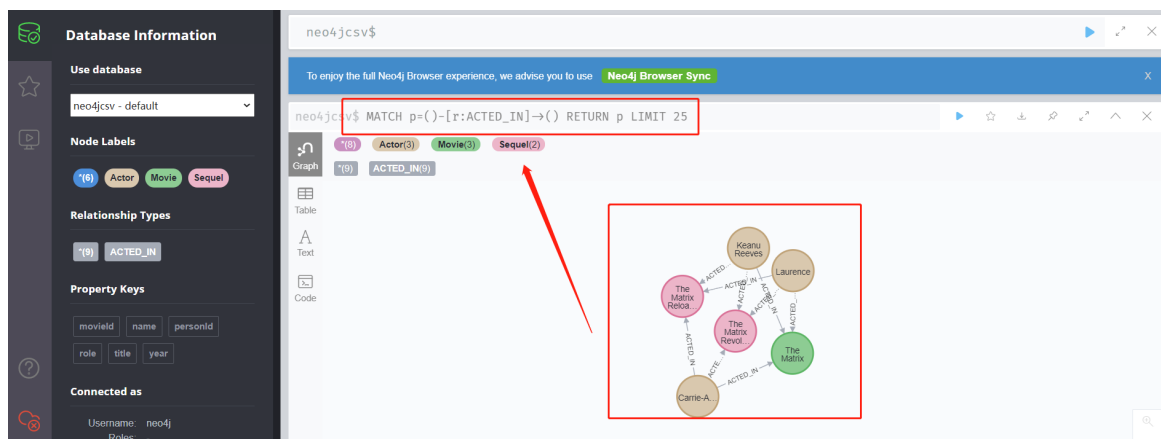
2. 左侧出现三个默认的数据库，此时默认的为neo4jcsv

3. 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，笔者使用create语句创建来的快。



三、 neo4j-admin import导入CSV总结：

- 适用场景：千万以上 nodes导入
- 速度：非常快 (数万nodes/s)
- 优点：占用更少的资源

- 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。

四、load csv 方式导入数据

neo4j-admin import 导入必须停止数据库的服务, 只能生成新的数据库, 不能在已有数据库中插入数据, load csv方式可插入数据, 即使用cypher语法中的LOAD CSV。

导入的注意事项: (1) 导入的文件必须是csv文件, 位置可以是本地的, 或通过http、https、ftp等url指定位置。(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, 防止中文乱码。

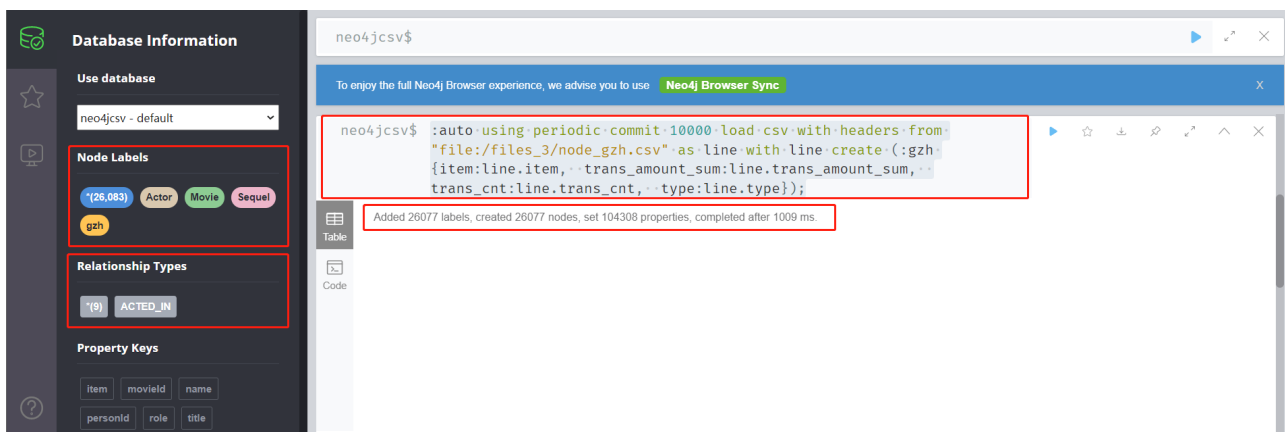
假设我们在neo4j的安装目录下有一个要导入的文件: D:\neo4j-community-4.2.6\import\files_3\node_gzh.csv。文件的内容大致如下:

```
D: > neo4j-community-4.2.6 > import > files_3 > node_gzh.csv
1  item,trans_amount_sum,trans_cnt,type
2  1,2184.6,10,有风险
3  2,4000.12,3,有风险
4  3,65050,15,有风险
5  4,32491.99,5,有风险
6  5,0,0,有风险
7  6,747232,595,有风险
8  7,53258,38,有风险
9  8,22763.23,11,有风险
10 9,127229,25,有风险
```

我们将这个数据导入到数据库neo4jcsv中(此时默认的就是neo4jcsv数据库): 我们执行语句如下:

```
:auto using periodic commit 10000 load csv with headers from
"file:/files_3/node_gzh.csv" as line with line create (:gzh
{item:line.item, trans_amount_sum:line.trans_amount_sum,
trans_cnt:line.trans_cnt, type:line.type});
```

之后, 我们可以看到数据库中的数据了:



导入的数据的说明：

1. using periodic commit的作用是分段式创建顶点，数值为10000即读取10000行数据就写入数据库中，分行读取的目的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. gzh代表节点的标签名，可以在Node Labels中查看。item、trans_amount_sum等代表节点的属性。
4. 上述方法创建了很多节点（节点有item、trans_amount_sum等属性）。

五、load csv 方式导入数据总结

- 适用场景：0 ~ 1000w
- 速度：一般 5000/s
- 优点：可以加载本地/远程CSV;可实时插入
- 缺点：导入速度较慢;需要将数据转换成csv