

# neo4j图数据库搭建使用说明文档

---

## 1.neo4j搭建(windows 10)

1. 从neo4j官网下载neo4j-community-4.1.4-windows.zip压缩包并解压，在该目录下运行bin/neo4j console，并在window上安装服务bin\neo4j install-service,开启服务查看服务器运行状况，并设置初始密码

## 2.数据导入

### 1. 数据处理

将数据一共分成3个节点和2个关系，分别为User, Product, Movie, 关系为(User)-[REVIEW]-(Product)-[BELONG]-(Movie)

前半部分由800w的评论集完成，提取每一个User到Product的评论关系，后半部分由爬虫得到的productId到movieName的映射关系完成，并且根据第一次作业中的去重工作，不是所有的Product都能连接到Movie，同时会有多个Product连接到同一个Movie上

由于数据量较大，使用neo4j admin import tool进行导入,对于csv头处理如下：

- User:userId:ID,profileName,:LABEL
- Product:productId:ID,:LABEL
- Movie: Movie:ID,:LABEL
- REVIEW: :START\_ID,helpfulness,score:double,time,:END\_ID,:TYPE
- BELONG: :START\_ID,:END\_ID,:TYPE

按照上述规则在python中处理得到对应的.csv文件

### 2. 数据导入

运行bin\neo4j stop,停止数据库

得到数据后将.csv放入import文件夹下，运行命令：

```
bin/neo4j-admin import
--nodes=import/user-header.csv,import/user.csv
--nodes=import/product-header.csv,import/product.csv
--nodes=import/movie-header.csv,import/movie.csv
--relationships=import/belong-header.csv,import/belong.csv
--relationships=import/review-header.csv,import/review.csv
```

重启数据库bin\neo4j start

在图形化页面中,运行

```
match p=(:User)-[:REVIEW]-(:Product)-[:Belong]-(:Movie) return p limit 25
```

检查结果正确性

### 3. 查询

1. 在python中使用py2neo，连接数据库，输入需要查询的关键词，根据关键词正则匹配对应的Movie节点，根据User连接的关系数量降序排序得到用户和用户评论的数量。
2. 为了得到评论最多的用户合集，读取降序排序的结果，只将关系数最多的User加入集合，源码如下:

```
from py2neo import Graph

graph = Graph(
    "http://localhost:7474",
    username="neo4j",
    password="NEO4J"
)

mname = input("请输入要查询的关键词：")
print("结果是：")

data1 = graph.run(
    "match (u:User)-[r:REVIEW]-(p:Product)-[b:Belong]-(m:Movie) where m.Movie=~\".*\" + mname + \".*\" return u,count(r) order by count(r) desc"
)
maxCount = 0
data = data1.data()
result = set()
for item in data:
    if item['count(r)'] >= maxCount:
        maxCount = item['count(r)']
        result.add(item['u'])
        print(item['u'])
    else:
        break
```