

# Knowledge Querying(II)

# 一、Neo4j安装

若电脑没有java11环境，请查阅Knowledge Querying(1) Java安装教程

# Neo4j 安装

官网下载链接：<https://neo4j.com/download-center>  
(已提供安装文件)。

Partners ▾ Company ▾ Contact Us Support 🔍

neo4j Products Solutions Learn Developers Data Scientists [Get Started](#)

## Neo4j Download Center

[Download Neo4j Desktop](#) ⬇

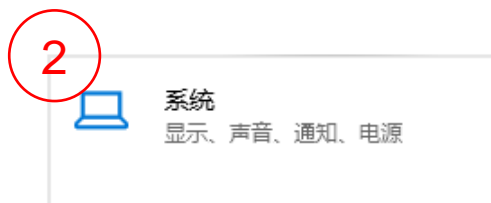
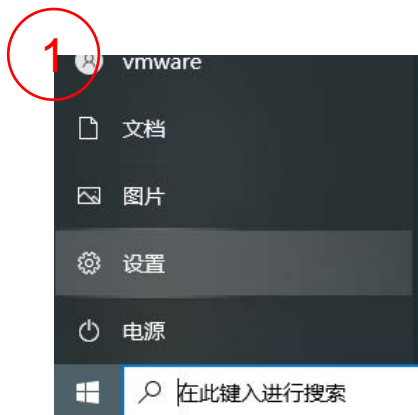
### Current Releases

选择社区版

Enterprise Server	Community Server	Neo4j Desktop
<b>Neo4j Community Edition 4.4.1</b> 14 December 2021 <a href="#">Release Notes</a>   <a href="#">Read More</a>		
OS	Download	
Linux/Mac	<a href="#">Neo4j 4.4.1 (tar)</a> SHA-256	根据系统选择对应的安装包
Windows	<a href="#">Neo4j 4.4.1 (zip)</a> SHA-256	

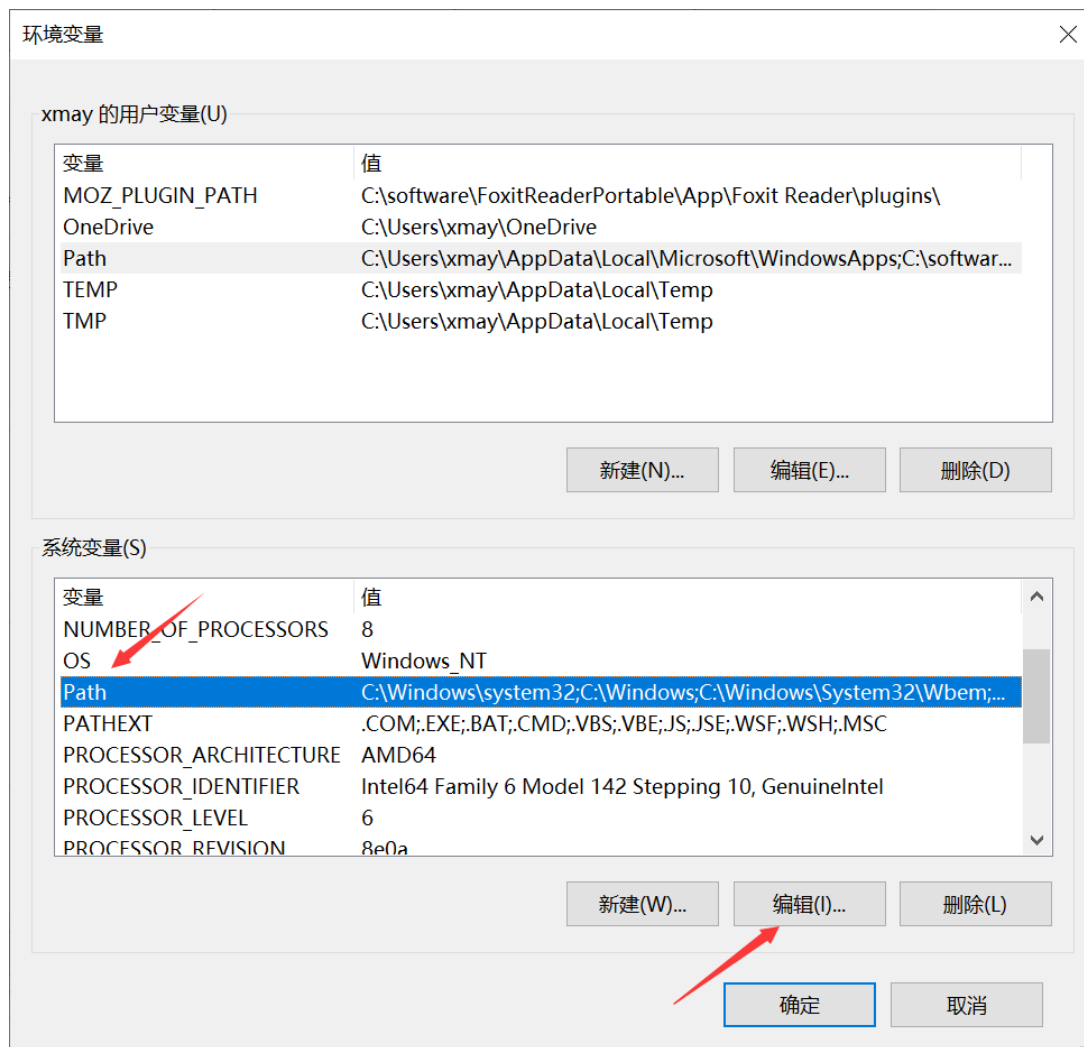
# 环境配置

打开 设置 --> 系统 --> 关于 --> 高级系统设置 --> 环境变量



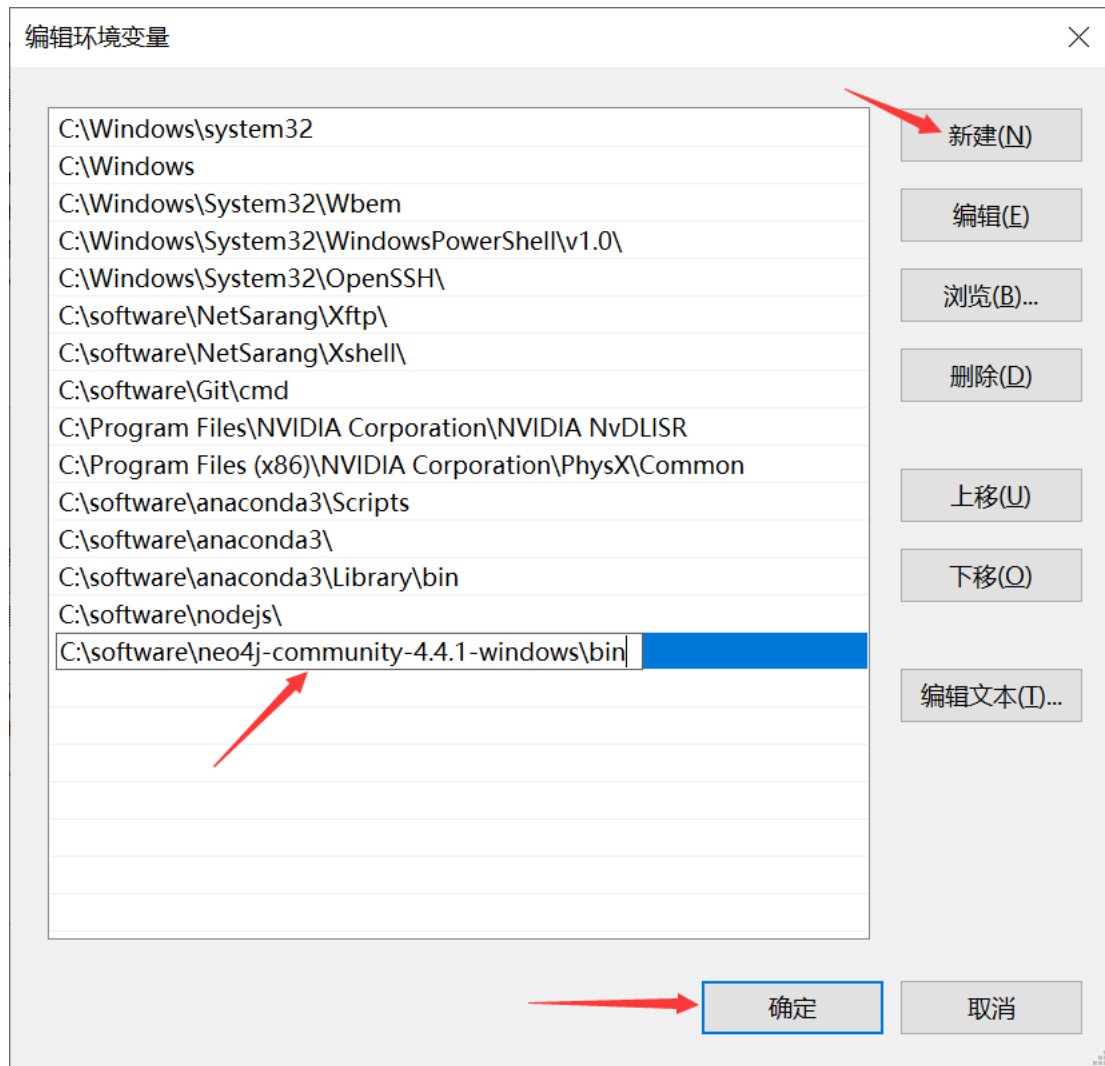
# 环境配置

- 选择Path系统环境变量，点击编辑。



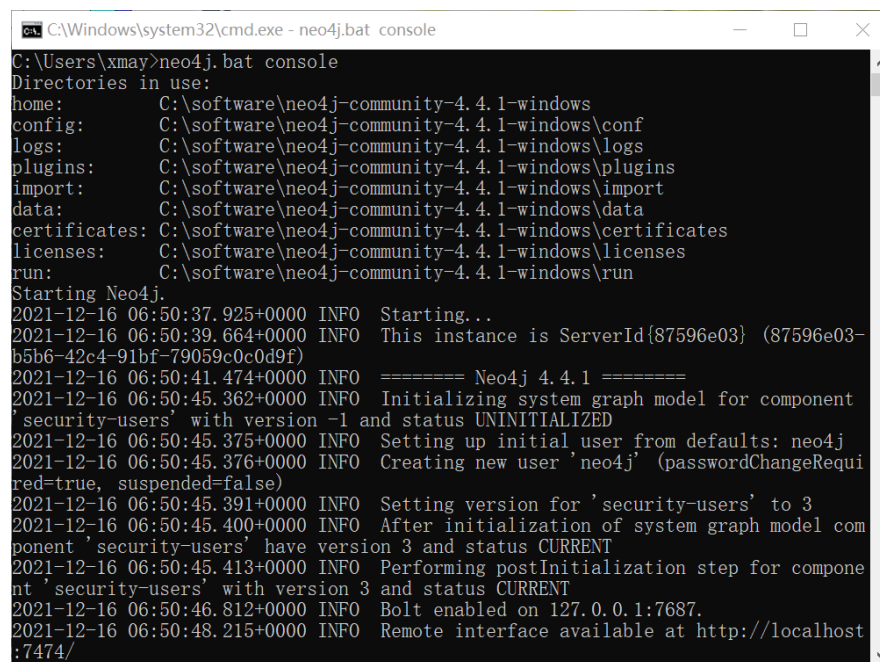
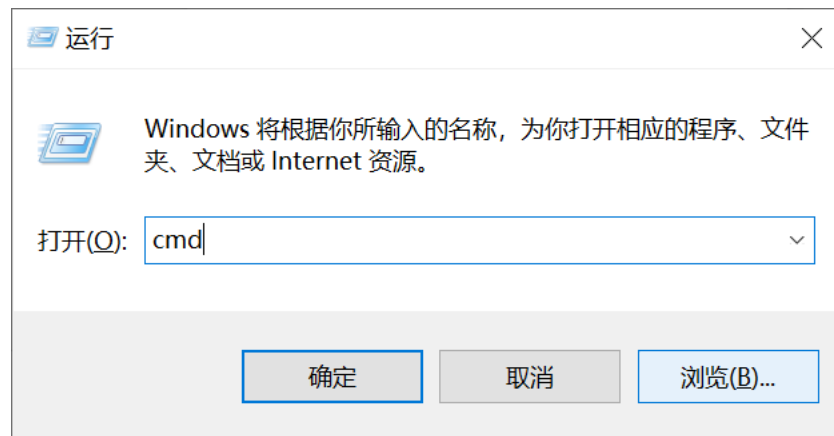
# 环境配置

- 解压下载好的Neo4j压缩包；
- 点击新建；
- 进入到解压后的Neo4j\bin所在文件夹，复制该路径粘贴到新的一行；
- 点击确定。



# 环境配置

- Win+R 打开运行窗口，打开cmd；
- 输入neo4j.bat console，执行命令；
- 出现右边第二张图即成功；
- 关闭cmd窗口。



# 环境配置-可能遇到的问题

- 如果在安装后报错java jdk版本问题
  - 建议重新在官网下载对应版本Neo4j版本
  - 下载链接: <https://neo4j.com/download-center>
- 如果在安装后报错: powershell命令提示符出现“不是内部或外部命令, 也不是可运行的程序或批处理”
  - 将如下三个路径添加到环境变量中  
C:\Windows\System32\WindowsPowerShell\v1.0  
C:\Windows\System32\wbem  
C:\Windows
  - 重启cmd, 再次输入neo4j.bat console



# 安装服务

- 新打开一个cmd窗口
- cd 进入到neo4j根目录，示例：cd C:\software\neo4j-community-4.4.1-windows\bin
- 执行neo4j install-service，将neo4j安装为服务

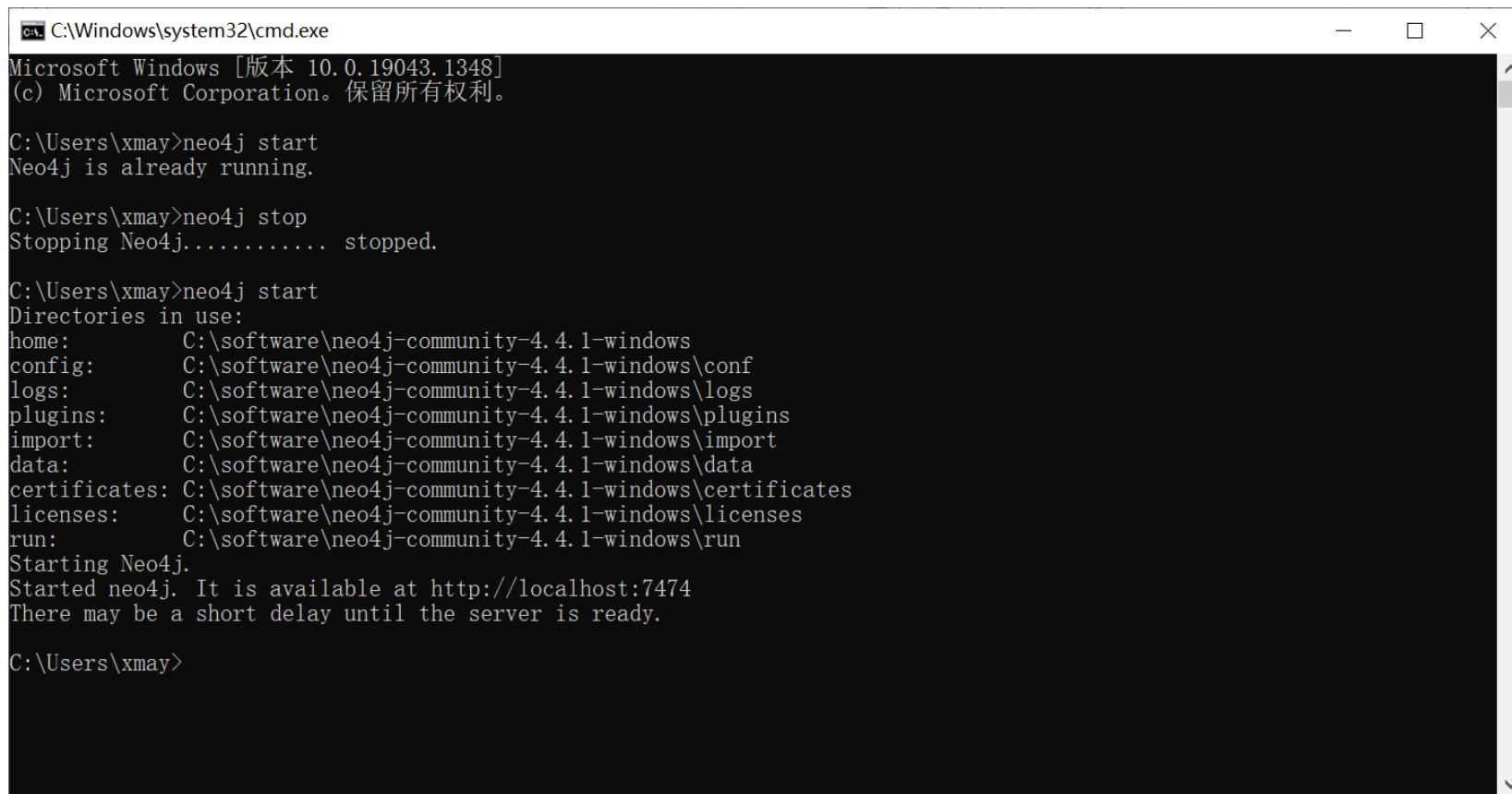
```
C:\Users\xmay>cd C:\software\neo4j-community-4.4.1-windows\bin  
C:\software\neo4j-community-4.4.1-windows\bin>neo4j install-service  
Neo4j service is already installed  
C:\software\neo4j-community-4.4.1-windows\bin>start neo4j  
C:\software\neo4j-community-4.4.1-windows\bin>
```

# 使用neo4j服务

新打开一个cmd窗口，开始使用neo4j服务

启动服务：neo4j start

关闭服务：neo4j stop

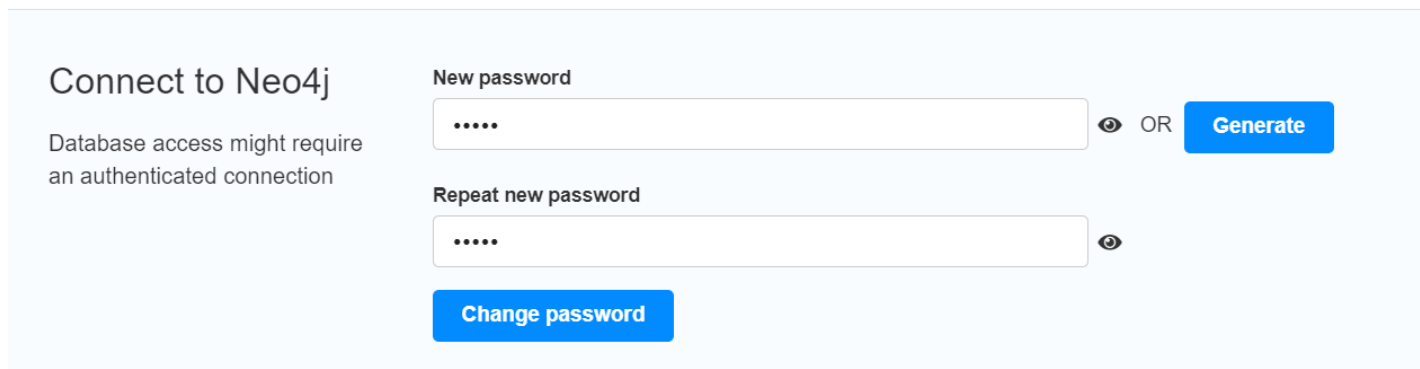
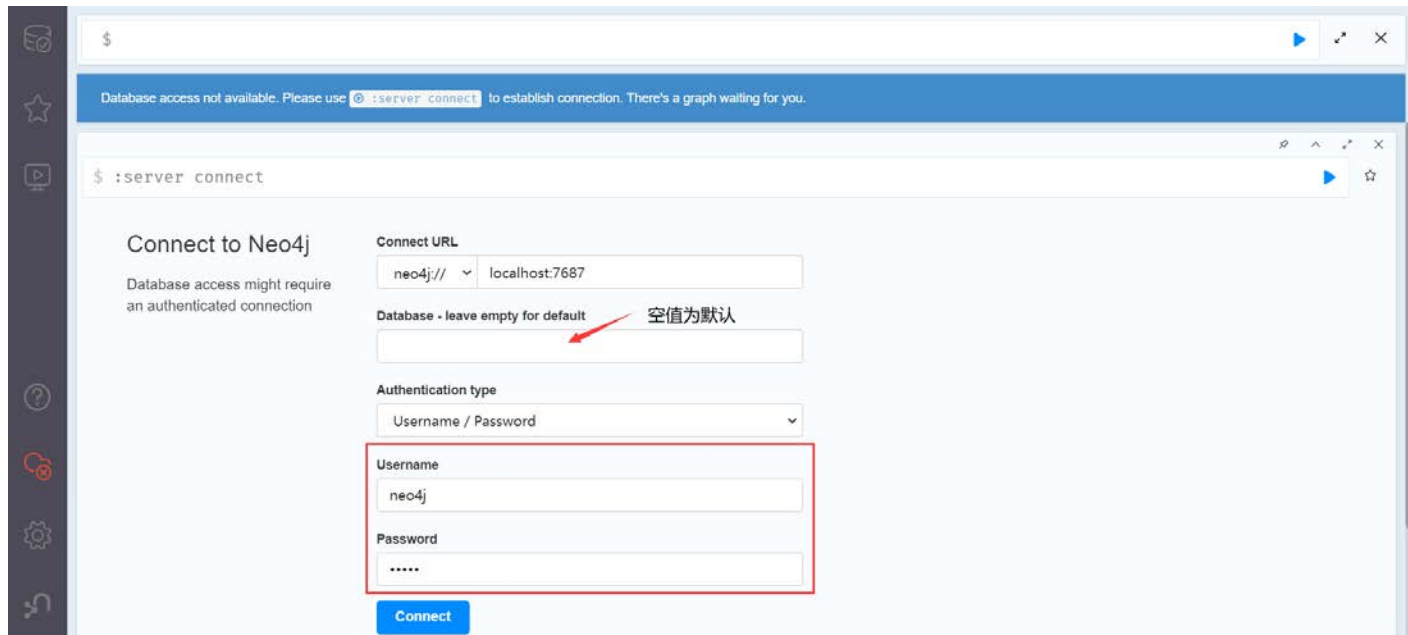
A screenshot of a Windows Command Prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [版本 10.0.19043.1348]  
(c) Microsoft Corporation. 保留所有权利。  
  
C:\Users\xmay>neo4j start  
Neo4j is already running.  
  
C:\Users\xmay>neo4j stop  
Stopping Neo4j..... stopped.  
  
C:\Users\xmay>neo4j start  
Directories in use:  
home:      C:\software\neo4j-community-4.4.1-windows  
config:    C:\software\neo4j-community-4.4.1-windows\conf  
logs:      C:\software\neo4j-community-4.4.1-windows\logs  
plugins:   C:\software\neo4j-community-4.4.1-windows\plugins  
import:    C:\software\neo4j-community-4.4.1-windows\import  
data:      C:\software\neo4j-community-4.4.1-windows\data  
certificates: C:\software\neo4j-community-4.4.1-windows\certificates  
licenses:  C:\software\neo4j-community-4.4.1-windows\licenses  
run:       C:\software\neo4j-community-4.4.1-windows\run  
Starting Neo4j.  
Started neo4j. It is available at http://localhost:7474  
There may be a short delay until the server is ready.  
  
C:\Users\xmay>
```

## 二、Neo4j使用

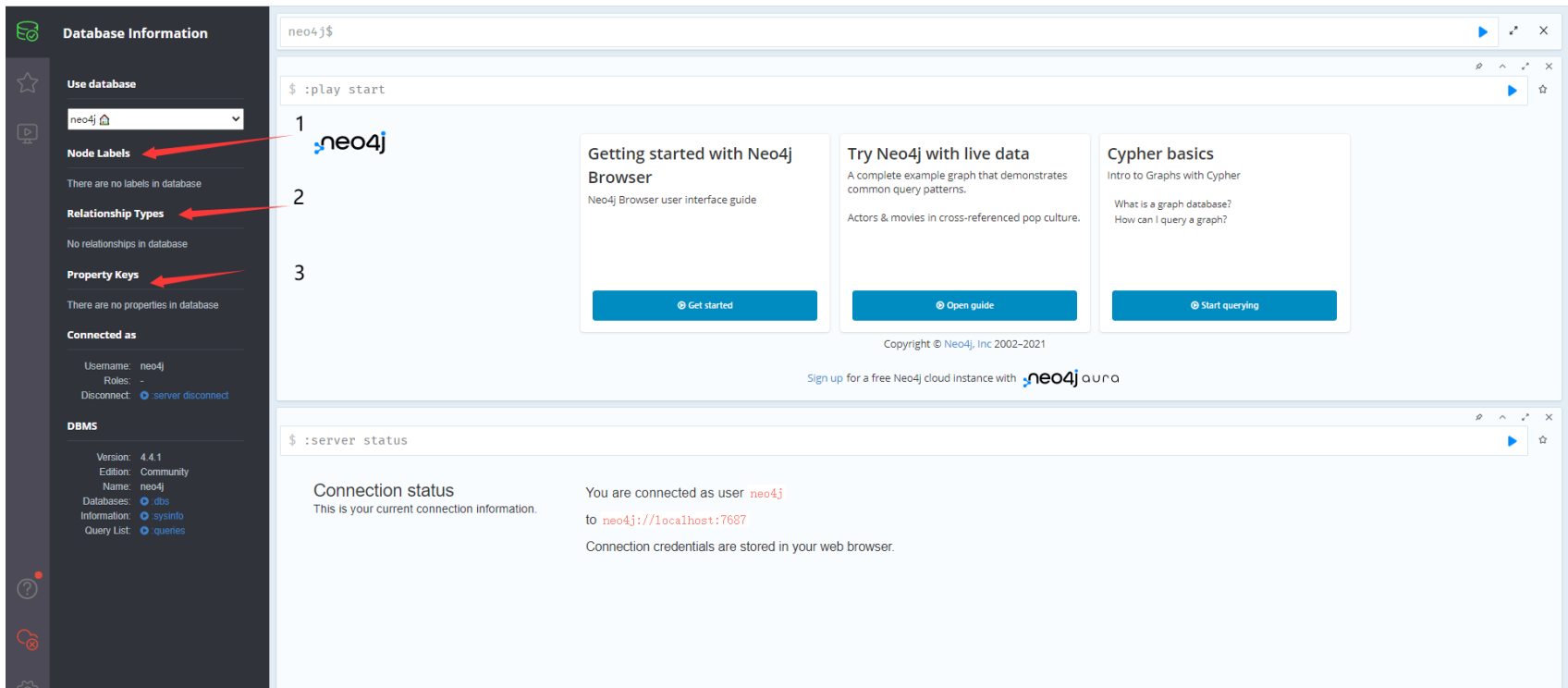
# Neo4j 使用

- 打开浏览器，输入<http://localhost:7474/>
- neo4j的原始账号密码都是：neo4j，登录后会提示修改密码（记住修改后的密码）。



# Neo4j 使用

- 1. Node Labels: 展示数据库中的Labels;
- 2. Relationship Types: 展示数据库中的所有关系类型;
- 3. Property Keys: 展示所有属性名称。



# Neo4j 导入图谱数据

- 导入数据前需要先关闭数据库，在cmd中执行neo4j stop;
- 将数据库文件movies-43.dump 放在neo4j根目录import文件夹下;
- cd 进入到neo4j根目录，示例为：cd C:\software\neo4j-community-4.4.1-windows;
- 执行如下导入命令，将数据导入到图谱中

**neo4j-admin load --from=import\movies-43.dump --database=graph.db --force**

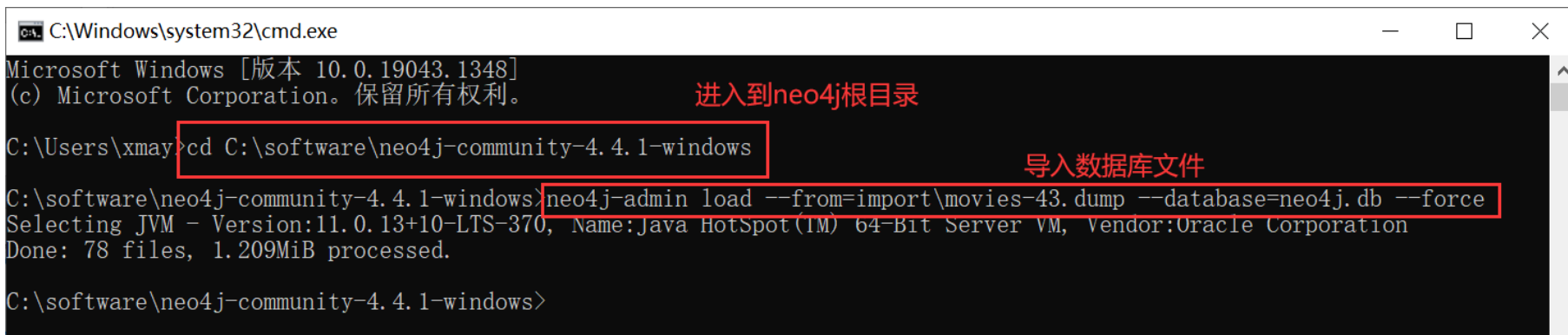
命令说明：

neo4j-admin: noe4j管理命令，我们当前是导入，所以参数为load

--from: 导入数据源路径， import\movies-43.dump

--database: 导入到哪个数据库， graph.db

--force: 若数据库不为空，则强制覆盖原有数据库中的数据



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.1348]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\xmay>cd C:\software\neo4j-community-4.4.1-windows
C:\software\neo4j-community-4.4.1-windows>neo4j-admin load --from=import\movies-43.dump --database=neo4j.db --force
Selecting JVM - Version:11.0.13+10-LTS-370, Name:Java HotSpot(TM) 64-Bit Server VM, Vendor:Oracle Corporation
Done: 78 files, 1.209MiB processed.

C:\software\neo4j-community-4.4.1-windows>
```

进入到neo4j根目录

导入数据库文件

# Neo4j 检查数据库文件

- 出现Done , Processed之后表示数据导入成功;
- 进入neo4j根目录data\databases文件夹下, 出现了新创建的数据库: graph.db;
- 如果出现导入错误, 可将数据库文件夹删除再重新执行导入命令。

**注:**

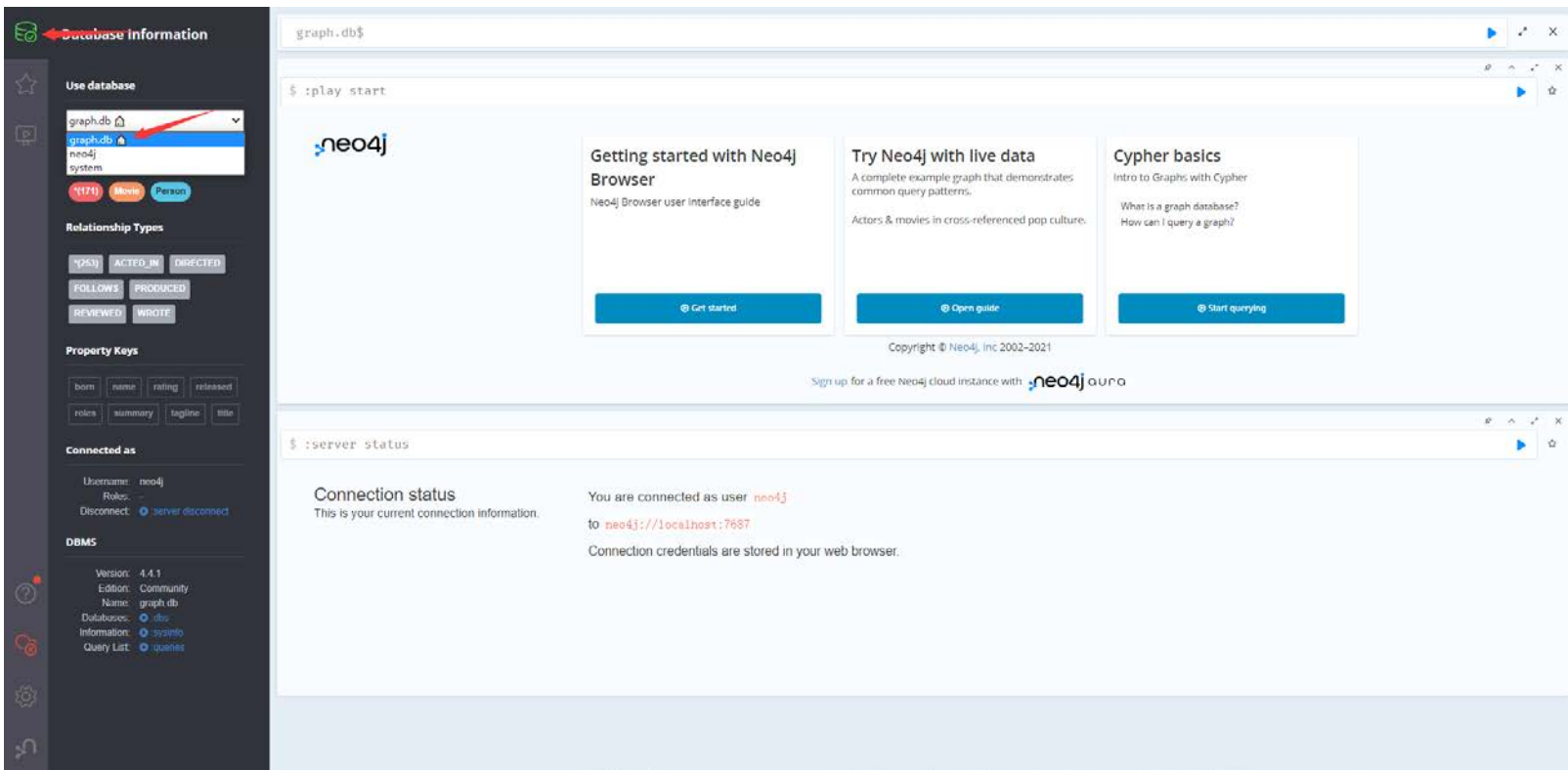
neo4j-admin导入命令要求数据库为空或数据库未创建, 因此使用前应先清空数据库;

如果数据库不为空, 可使用--force强制覆盖原有数据库, 因此使用时需要确保原数据库不再使用, 在工程中谨慎使用。

名称	修改日期	类型
profiles	2021/12/16 19:51	文件夹
schema	2021/12/16 19:51	文件夹
database_lock	2021/12/16 19:51	文件
neostore	2021/12/16 19:51	文件
neostore.counts.db	2021/12/16 19:56	Data Ba
neostore.id	2021/12/16 19:51	ID 文件
neostore.indexstats.db	2021/12/16 19:56	Data Ba
neostore.labelscanstore.db	2021/12/16 19:56	Data Ba
neostore.labeltokenstore.db	2021/12/16 19:51	Data Ba
neostore.labeltokenstore.db.id	2021/12/16 19:56	ID 文件
neostore.labeltokenstore.db.names	2021/12/16 19:51	NAMES
neostore.labeltokenstore.db.names.id	2021/12/16 19:56	ID 文件
neostore.nodestore.db	2021/12/16 19:51	Data Ba
neostore.nodestore.db.id	2021/12/16 19:56	ID 文件
neostore.nodestore.db.labels	2021/12/16 19:51	LABELS
neostore.nodestore.db.labels.id	2021/12/16 19:56	ID 文件
neostore.propertystore.db	2021/12/16 19:51	Data Ba
neostore.propertystore.db.arrays	2021/12/16 19:51	ARRAYS
neostore.propertystore.db.arrays.id	2021/12/16 19:56	ID 文件

# Neo4j启动服务

- 通过cmd执行neo4j start启动数据库服务；
- 通过浏览器访问数据库， <http://localhost:7474/browser/>；
- 点击数据库图标，选择下拉框graph.db作为本次实验的数据库。

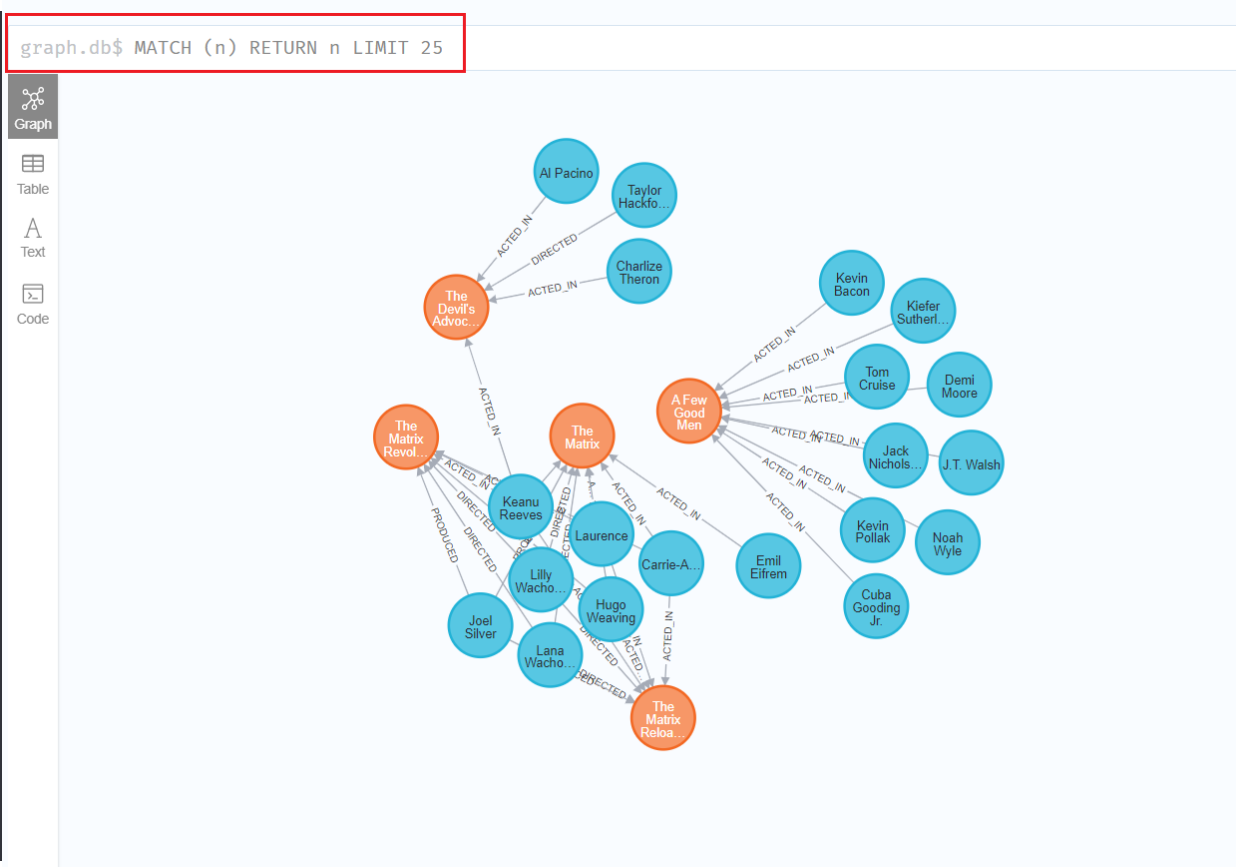




# Neo4j启动服务

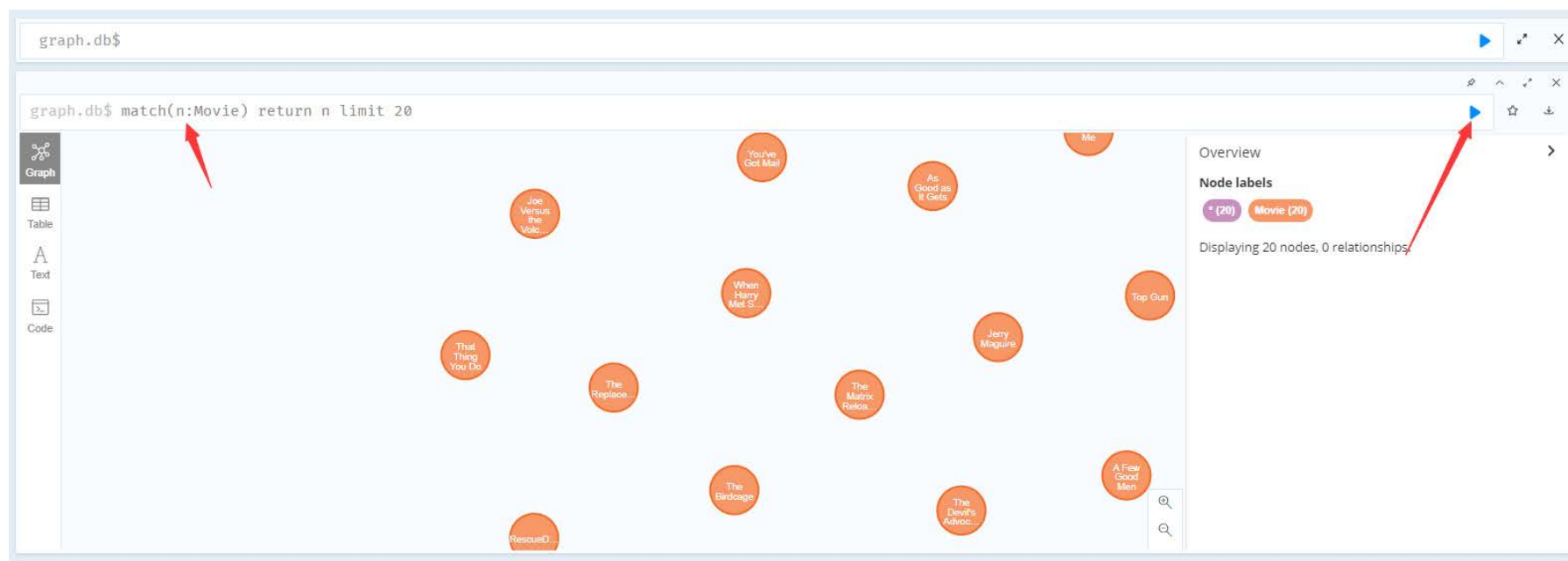
- 可看到左侧节点、关系、属性已有数据；
- 通过鼠标点击图标，可以快速获取到相关节点、关系或属性信息；
- 点击左侧\*(171)（表示当前数据库所有节点数量），可快速浏览当前数据库的节点信息，红色框选中是自动生成的cypher语句。

The screenshot shows the Neo4j Desktop interface. On the left, the 'Database Information' panel is visible. It includes a 'Use database' dropdown set to 'graph.db'. Below this, the 'Node Labels' section shows three labels: '\* (171)' (highlighted with a red arrow), 'Movie', and 'Person'. The 'Relationship Types' section lists several types: '\* (253)', 'ACTED\_IN', 'DIRECTED', 'FOLLOWS', 'PRODUCED', 'REVIEWED', and 'WROTE'. The 'Property Keys' section lists 'born', 'name', 'rating', 'released', 'roles', 'summary', 'tagline', and 'title'.



# Cypher的使用

- 键入cypher语句：  
match(n:Movie) return n limit 20  
查询电影节点，并返回前20个；
- 点击右侧蓝色按钮运行，可以看到结果；
- 单击节点可看到节点相关属性，双击节点可看到节点相关联的关系（请自行探索）。

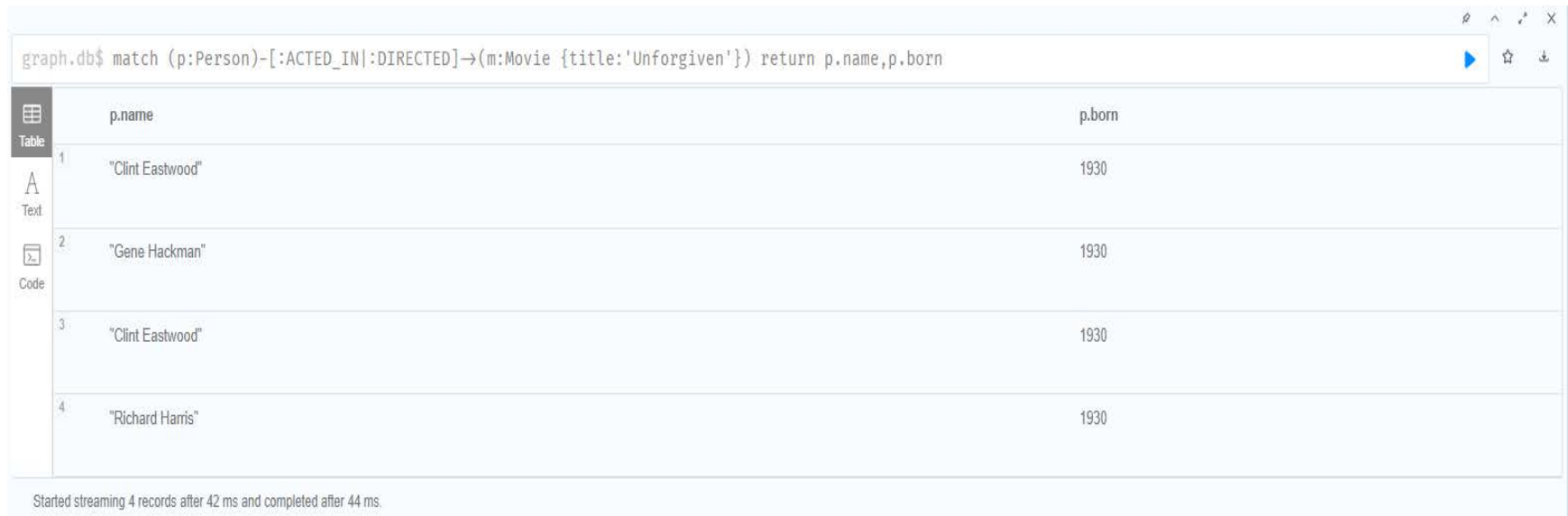


# Cypher的使用

- 示例

出演或导演电影Unforgiven的人物姓名和出生时间：

```
match (p:Person)-[:ACTED_IN|DIRECTED]->(m:Movie {title:'Unforgiven'}) return p.name,p.born
```



The screenshot shows a web-based interface for running Cypher queries. At the top, a text input field contains the query: `graph.db$ match (p:Person)-[:ACTED_IN|DIRECTED]->(m:Movie {title:'Unforgiven'}) return p.name,p.born`. To the right of the input are icons for running the query (a blue play button), saving (a star), and downloading (a download icon). Below the input is a table view. On the left side of the table, there are three icons: a table icon (selected), a text icon, and a code icon. The table has two columns: `p.name` and `p.born`. It displays four rows of results, numbered 1 to 4. The first row shows "Clint Eastwood" with birth year 1930. The second row shows "Gene Hackman" with birth year 1930. The third row shows "Clint Eastwood" with birth year 1930. The fourth row shows "Richard Harris" with birth year 1930. At the bottom of the interface, a status bar indicates: "Started streaming 4 records after 42 ms and completed after 44 ms."

	p.name	p.born
1	"Clint Eastwood"	1930
2	"Gene Hackman"	1930
3	"Clint Eastwood"	1930
4	"Richard Harris"	1930

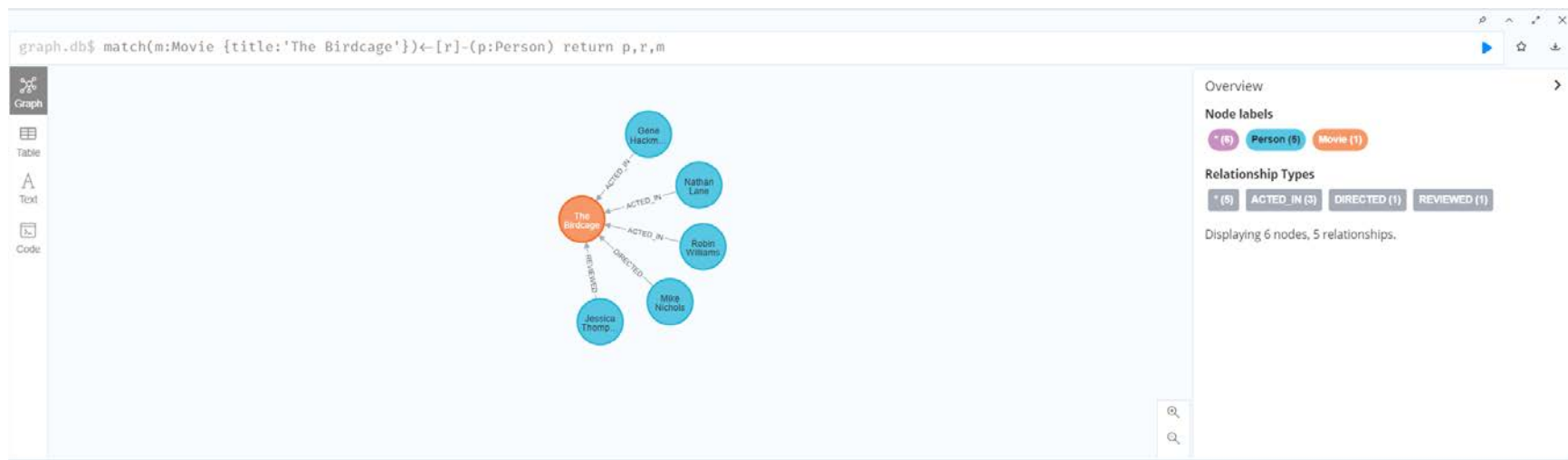
Started streaming 4 records after 42 ms and completed after 44 ms.

# Cypher的使用

## - 示例:

查询与电影**The Birdcage**相关联的所有人物和电影信息，返回所有相关联的信息

```
match(m:Movie {title:'The Birdcage'})<-[r]-(p:Person) return p,r,m
```

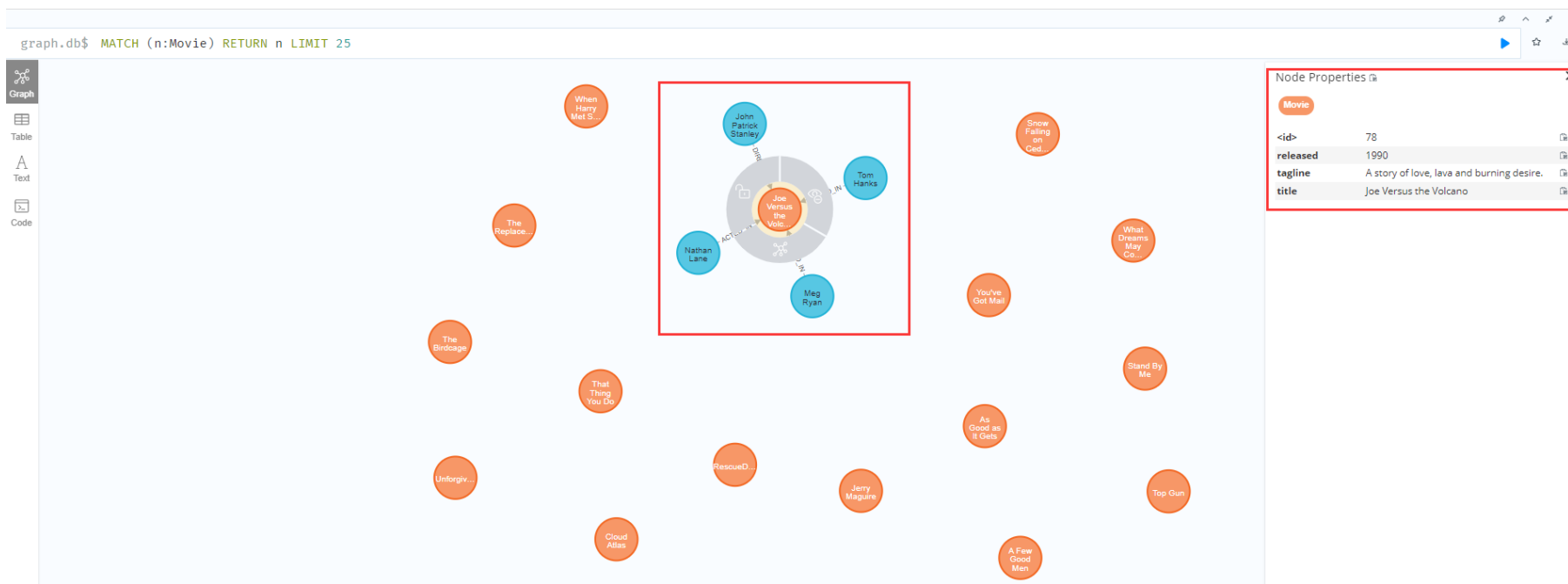


# Cypher的使用

- 任意选择一个电影节点；
- 单击节点查看其属性信息；
- 双击节点查看其关系信息；

本示例所选节点可通过以下语句查询：

```
match (m:Movie{title:'Joe Versus the Volcano'}) return m
```



# Cypher的使用

记录下选择节点的信息

- 属性信息

Key	value
released	1990
tagline	A story of love, lava and burning desire.
title	Joe Versus the Volcano

# Cypher的使用

记录下选择节点的信息

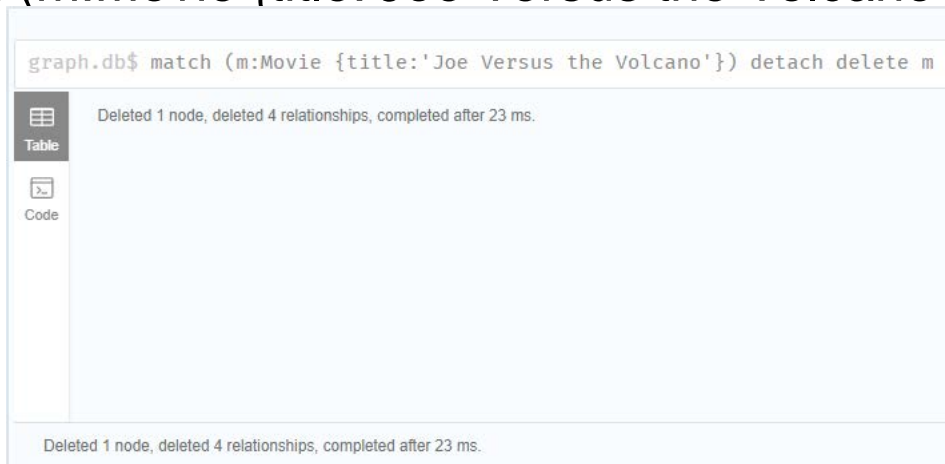
- 关系信息

Start_node_Label	Start_node_name	Relation_type	End_node_Label	End_node_title	Relation Properties
Person	John Patrick Stanley	DIRECTED	Movie	Joe Versus the Volcano	NULL
Person	Tom Hanks	ACTED_IN	Movie	Joe Versus the Volcano	NULL
Person	Nathan Lane	ACTED_IN	Movie	Joe Versus the Volcano	NULL
Person	Meg Ryan	ACTED_IN	Movie	Joe Versus the Volcano	NULL

# Cypher的使用

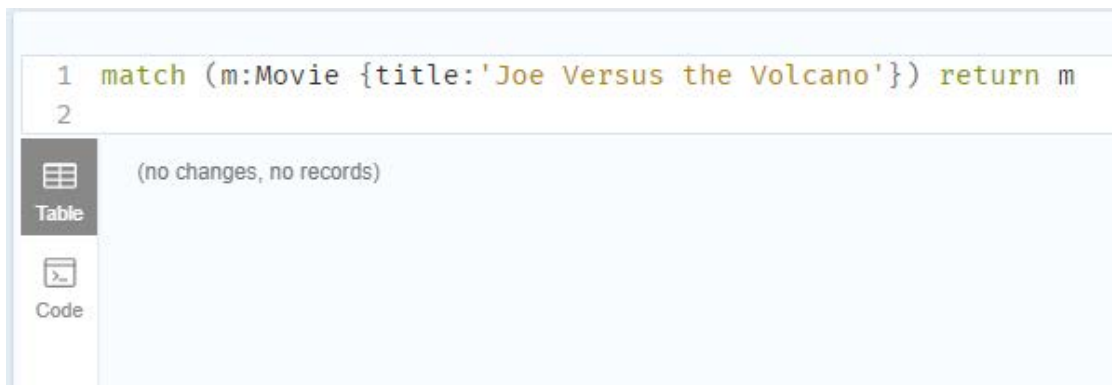
- 删除该节点及其相连接的关系

`match (m:Movie {title:'Joe Versus the Volcano'}) detach delete m`



- 查询该节点是否删除成功

`match (m:Movie {title:'Joe Versus the Volcano'}) return m`





# Cypher的使用

- 将上一步删除的节点及其节点属性重新导入到图谱中

Create

```
( m:Movie
```

```
  { title:'Joe Versus the Volcano',released:1990,tagline:'A story  
    of love, lava and burning desire.'}
```

```
)
```

```
return m
```



The screenshot shows the Neo4j Cypher query editor interface. The query bar at the top contains the following Cypher query:

```
graph.db$ create (m:Movie{title:'Joe Versus the Volcano',released:1990,tagline:'A story of love, lava and burning desire.'}) return m
```

The query is executed, and the result is displayed in the graph view. A single node is created, labeled "Joe Versus the Volcano", with properties "released: 1990" and "tagline: A story of love, lava and burning desire.". The node is shown in a circular graph view with a red border and a yellow center.

On the right side, the "Node Properties" panel shows the details of the created node:

Property	Value
<id>	78
released	1990
tagline	A story of love, lava and burning desire.
title	Joe Versus the Volcano

# Cypher的使用

- 增加前面删除的节点相连的边添加到图谱中

Start_node_Label	Start_node_name	Relation_type	End_node_Label	End_node_title	Relation Properties
Person	John Patrick Stanley	DIRECTED	Movie	Joe Versus the Volcano	NULL

match

(p:Person{name:'John Patrick Stanley'}),(m:Movie{title:'Joe Versus the Volcano'})

create

(p)-[r:DIRECTED]->(m)

return p,r,m

The screenshot shows a Cypher query interface. The query entered is: `graph.db$ match (p:Person{name:'John Patrick Stanley'}),(m:Movie{title:'Joe Versus the...})`. The graph visualization shows two nodes: 'Joe Versus the Volcano' (orange circle) and 'John Patrick Stanley' (blue circle), connected by a directed edge labeled 'DIRECTED'. The sidebar on the right shows the 'Overview' section with 'Node labels' (Person (1), Movie (1)) and 'Relationship Types' (DIRECTED (1)).

# Cypher的使用

- 更新电影的名称为” Joe Versus the Volcano2”

```
match
(m:Movie{title:'Joe Versus the Volcano'})
set
m.title= 'Joe Versus the Volcano2'
return m
```

The screenshot displays a Cypher query interface. At the top, a query editor contains the following code:

```
1 match (m:Movie{title:'Joe Versus the Volcano'}) set m.title= 'Joe Versus the Volcano2' return m
2
```

Below the query editor, a sidebar on the left shows four views: Graph (selected), Table, Text, and Code. The main area displays a graph visualization with a central orange node labeled "Joe Versus the Volcano2" surrounded by a light blue circle. The node is connected to four other nodes, each represented by a different icon (a lock, a speech bubble, a network, and a document).

On the right side, a "Node Properties" panel is open, showing the properties of the selected node:

Property	Value
<id>	78
released	1990
tagline	A story of love, lava and burning desire.
title	Joe Versus the Volcano2

The "title" property is highlighted with a red border.

### 三、作业

# 作业

1. 导入contact-tracing-43.dump文件到数据库graph.db中；
2. 查询名叫Madison Odonnell的人物节点，并记录下该节点的healthstatus、name、confirmedtimes属性和属性值；
3. 将该人物节点及与其相连的关系删除，并检查是否删除成功；
4. 重新创建该节点以及第2步记录下来的节点属性；
5. 重新创建关系： Madison Odonnell的人物节点与名为‘Place nr 40’的Place节点间的关系，不考虑关系属性；
6. Madison Odonnell不幸被确诊为新冠（healthstatus=‘sick’），对图谱进行更新。

完成以上问题并检验形成实验报告

# 参考资料

<https://neo4j.com/docs/cypher-manual/current/syntax/>

<https://neo4j.com/docs/cypher-manual/current/clauses/>