实验 4：Neo4j

1. 阅读“知识存储与查询-书稿.pdf”的“5.2.2 Cypher 语言”，执行其中的查询 20-31：（新建知识图谱的语句见文件 code-5-pg.txt）

(1) 查询 20

查询结果截图：

battleName
1 "四渡赤水"

(2) 查询 21

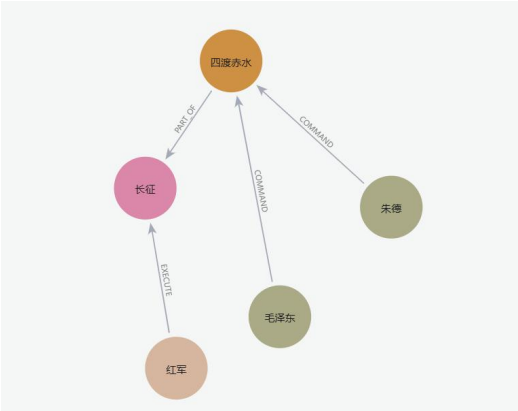
查询结果截图：

b	r
1 (:Battle { endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19" })	(:Basin { name: "赤水河", length: 436.5 })

(3) 查询 22

查询结果截图：

p	b	c	o
1 (:Person {name: "毛泽东"})	(:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Campaign {name: "长征"})	(:Organization {name: "红军"})
2 (:Person {name: "朱德"})	(:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Campaign {name: "长征"})	(:Organization {name: "红军"})



(4) 查询 23

查询结果截图：

	battleName	start
1	"四渡赤水"	"1935-01-19"

(5) 查询 24

查询结果截图：

	commander	role	e
1	"毛泽东"	"战略指挥"	null
2	"朱德"	"军事领导"	null

(6) 查询 25

查询结果截图：

	entityName
1	"四渡赤水"
2	"长征"

(7) 查询 26

查询结果截图：

originalName	shortName	daysElapsed
"四渡赤水"	"四渡"	62

(8) 查询 27

查询结果截图：

	province	battleCount
1	"贵州省"	1
2	"四川省"	1

(9) 查询 28

查询结果截图：

basinName	newLength
"赤水河"	440.0

Set 1 property Started streaming 1 record after 98 ms and comple

(10) 查询 29

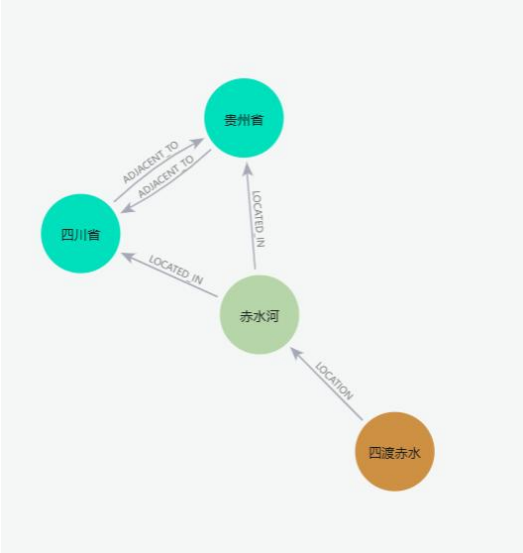
查询结果截图：

```
1 MATCH (b:Battle)
2 RETURN b.name AS battleName, b.startDate AS start
3 ORDER BY b.startDate DESC
4 SKIP 5
5 LIMIT 5
```

No changes, no records Completed after 62 ms

(11) 查询 30

查询结果截图：

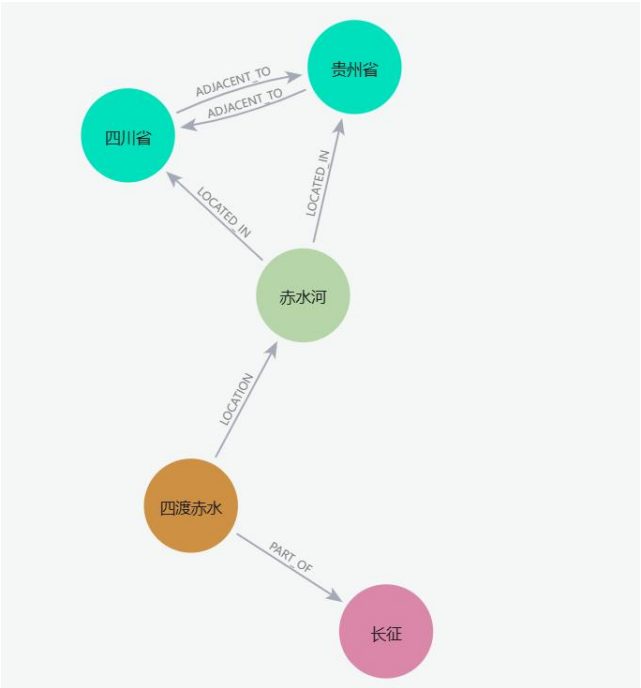


path

```
1 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})-[:LOCATION]->(:Basin {name: "赤水河", length: 440.0})-[:LOCATED_IN]->(:Provi
2 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})-[:LOCATION]->(:Basin {name: "赤水河", length: 440.0})-[:LOCATED_IN]->(:Provi
3 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})-[:LOCATION]->(:Basin {name: "赤水河", length: 440.0})-[:LOCATED_IN]->(:Provi
4 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})-[:LOCATION]->(:Basin {name: "赤水河", length: 440.0})-[:LOCATED_IN]->(:Provi
```

(12) 查询 31

查询结果截图：



start	end
1 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Basin {name: "赤水河", length: 440.0})
2 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Province {name: "贵州省"})
3 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Province {name: "四川省"})
4 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Province {name: "贵州省"})
5 (:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})	(:Province {name: "四川省"})

```
(:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})  (:Province {name: "贵州省"})
(:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})  (:Province {name: "四川省"})
(:Battle {endDate: "1935-03-22", name: "四渡赤水", startDate: "1935-01-19"})  (:Campaign {name: "长征"})
```

2. 使用 Neo4j 导入 IMDB 电影知识图谱数据集（imdb-data）的 actors.csv、movies.csv、roles.csv（imdb-data.zip 压缩包解压，具体导入命令见文件“neo4j import cmd.txt”），执行课程视频“实战-Neo4j.mp4”中所演示的所有 Cypher 查询，并给出查询结果截图。

给出执行 neo4j-admin import 批量导入数据集成功截图：

```
D:\neo4j\neo4j-community-5.26.4-windows\neo4j-community-5.26.4\import>D:\neo4j\neo4j-community-5.26.4-windows\neo4j-community-5.26.4\bin\neo4j-admin database import full --nodes=Movie=import/movies.csv --nodes=Actor=import/actors.csv --relationships=PLAYED_IN=import/roles.csv
WARNING! You are using an unsupported Java runtime.
* Please use Java(TM) 17 or Java(TM) 21 to run Neo4j.
* Please see https://neo4j.com/docs/ for Neo4j installation instructions.
Neo4j version: 5.26.4
Importing the contents of these files into D:\neo4j\neo4j-community-5.26.4-windows\neo4j-community-5.26.4\data\databases\neo4j:
Nodes:
  [Movie]:
  D:\neo4j\neo4j-community-5.26.4-windows\neo4j-community-5.26.4\import\movies.csv

  [Actor]:
  D:\neo4j\neo4j-community-5.26.4-windows\neo4j-community-5.26.4\import\actors.csv

Relationships:
  PLAYED_IN:
  D:\neo4j\neo4j-community-5.26.4-windows\neo4j-community-5.26.4\import\roles.csv


Available resources:
  Total machine memory: 15.73GiB
  Free machine memory: 2.989GiB
  Max heap memory : 3.497GiB
  Max worker threads: 16
  Configured max memory: 1.392GiB
  High parallel IO: true
```

```
IMPORT DONE in 10s 318ms.
Imported:
  2614151 nodes
  13103222 relationships
  2614151 properties
Peak memory usage: 1.057GiB
```

(1) 查询 1

Cypher 语句：

```
MATCH (n)
RETURN labels(n) AS labels, keys(n) AS properties, COUNT(*) AS total
ORDER BY total DESC;
```

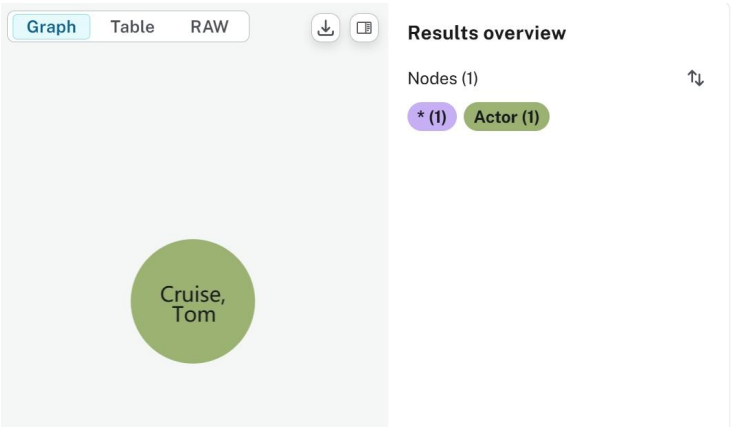
查询结果截图：

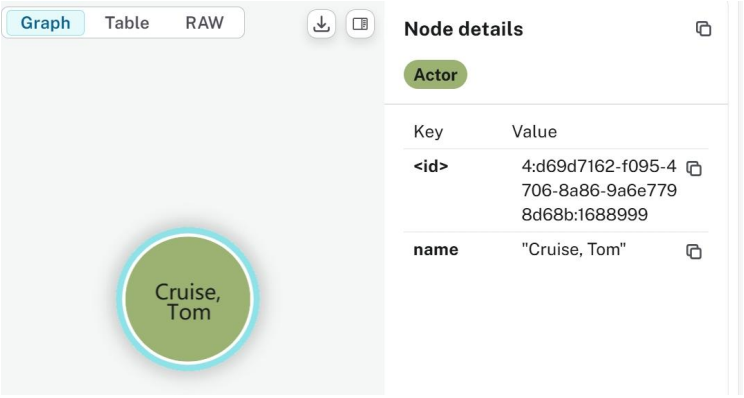
(2) 查询 2

Cypher 语句：

```
neo4j$ MATCH (x:Actor{name:"Cruise, Tom"}) RETURN x;
```

查询结果截图：





(3) 查询 3

Cypher 语句：

```
neo4j$ MATCH (x:Actor) RETURN COUNT(x);
```

查询结果截图：

COUNT(x)
1854870

(4) 查询 4

Cypher 语句：

```
match (m)-[r]->(n)
return labels(m), type(r), labels(n), count(*) as total
order by total desc;
```

查询结果截图：

labels(m)	type(r)	labels(n)	total
["Actor"]	"PLAYED_IN"	["Movie"]	13103222

(5) 查询 5

Cypher 语句：

```
MATCH (x:Actor{name:"Cruise, Tom"})-[:PLAYED_IN]->(m:Movie)
RETURN x,m;
```

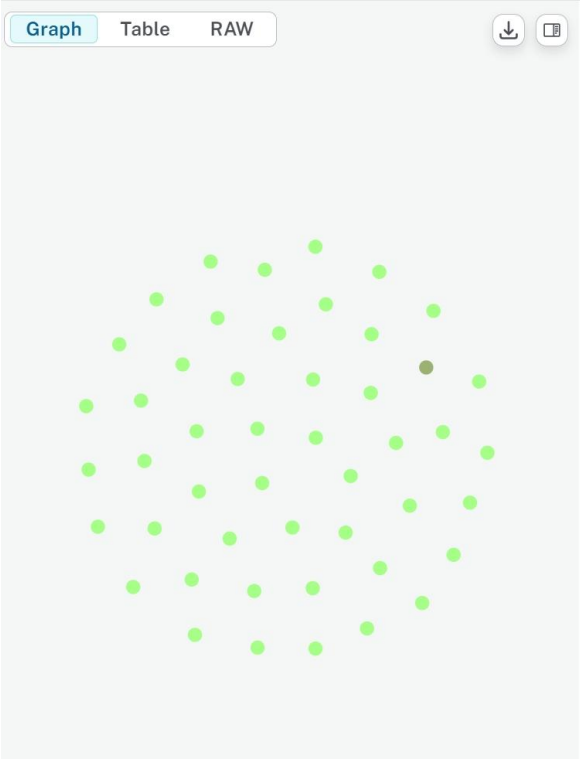
查询结果截图：

Graph Table RAW

	x	m
1	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "War of the Worlds (2005)"})
2	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Vanilla Sky (2001)"})
3	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Valkyrie (2008)"})
4	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Tropic Thunder (2008)"})
5	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Top Gun 2 (????)"})
6	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Top Gun (1986)"})
7	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "The Outsiders (1983)"})
8	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "The Mummy (2017)"})
9	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "The Last Samurai (2003)"})
10	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "The Firm (1993)"})
11	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "The Color of Money (1986)"})
12	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Taps (1981/I)"})
13	(:Actor {name: "Cruise, Tom"})	(:Movie {title: "Space Station 3D (2002)"})

Started streaming 47 records after 31 ms and completed after 359 ms.

Graph Table RAW



Results overview

Nodes (48)

* (48) Actor (1) Movie (47)

(6) 查询 6

Cypher 语句：

```
MATCH (x:Actor{name:"Cruise, Tom"})-[:PLAYED_IN]->(:Movie)<-[:PLAYED_IN]-(y:Actor)
WHERE x.name<>y.name
RETURN x.name as name1, y.name as name2, COUNT(*) AS total
ORDER BY total DESC;
```

查询结果截图：

Table RAW

	name1	name2	total
1	"Cruise, Tom"	"Mapother, William"	5
2	"Cruise, Tom"	"Rhames, Ving"	5
3	"Cruise, Tom"	"Pegg, Simon"	4
4	"Cruise, Tom"	"Kehoe, Michael G."	3
5	"Cruise, Tom"	"Voight, Jon"	3
6	"Cruise, Tom"	"Dean, Ron (I)"	3
7	"Cruise, Tom"	"Oliver, Jay (I)"	3
8	"Cruise, Tom"	"Armijo, John L."	3
9	"Cruise, Tom"	"Gleeson, Brendan"	3
10	"Cruise, Tom"	"Renner, Jeremy"	3
11	"Cruise, Tom"	"Dye, Dale"	3
12	"Cruise, Tom"	"Bronson, Greg"	3
13	"Cruise, Tom"	"Simpkins, Ty"	2
14	"Cruise, Tom"	"Robbins, Tim (I)"	2

Started streaming 2501 records after 58 ms and completed after 425 ms.

(7) 查询 7

Cypher 语句：

```
MATCH (x:Actor{name:"Cruise, Tom"})-[:PLAYED_IN]->(:Movie)<-[:PLAYED_IN]-(y:Actor)
WHERE x.name<>y.name
RETURN x.name as name1, y.name as name2, COUNT(*) AS total
ORDER BY total DESC
LIMIT 10;
```

查询结果截图：

Table RAW

	name1	name2	total
1	"Cruise, Tom"	"Mapother, William"	5
2	"Cruise, Tom"	"Rhames, Ving"	5
3	"Cruise, Tom"	"Pegg, Simon"	4
4	"Cruise, Tom"	"Gleeson, Brendan"	3
5	"Cruise, Tom"	"Oliver, Jay (I)"	3
6	"Cruise, Tom"	"Voight, Jon"	3
7	"Cruise, Tom"	"Armijo, John L."	3
8	"Cruise, Tom"	"Kehoe, Michael G."	3
9	"Cruise, Tom"	"Dean, Ron (I)"	3
10	"Cruise, Tom"	"Renner, Jeremy"	3

(8) 查询 8

Cypher 语句：

```
MATCH (x:Actor{name:"Cruise, Tom"})-[:PLAYED_IN*2]-(y:Actor)
WHERE x.name<>y.name
RETURN x.name as name1, y.name as name2, COUNT(*) AS total
ORDER BY total DESC;

```

查询结果截图：

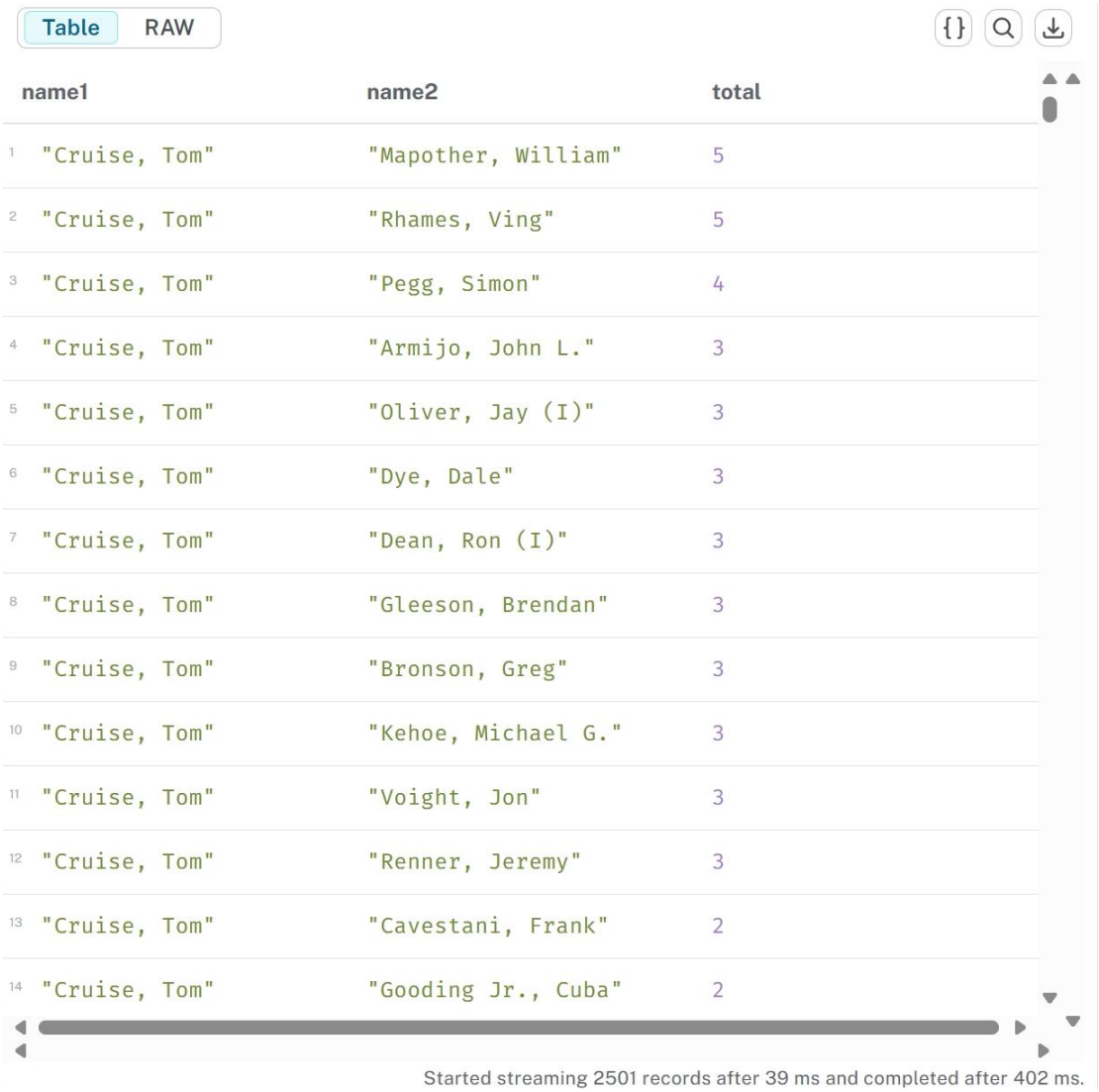


Table RAW

	name1	name2	total
1	"Cruise, Tom"	"Mapother, William"	5
2	"Cruise, Tom"	"Rhames, Ving"	5
3	"Cruise, Tom"	"Pegg, Simon"	4
4	"Cruise, Tom"	"Armijo, John L."	3
5	"Cruise, Tom"	"Oliver, Jay (I)"	3
6	"Cruise, Tom"	"Dye, Dale"	3
7	"Cruise, Tom"	"Dean, Ron (I)"	3
8	"Cruise, Tom"	"Gleeson, Brendan"	3
9	"Cruise, Tom"	"Bronson, Greg"	3
10	"Cruise, Tom"	"Kehoe, Michael G."	3
11	"Cruise, Tom"	"Voight, Jon"	3
12	"Cruise, Tom"	"Renner, Jeremy"	3
13	"Cruise, Tom"	"Cavestani, Frank"	2
14	"Cruise, Tom"	"Gooding Jr., Cuba"	2

Started streaming 2501 records after 39 ms and completed after 402 ms.

3. 在上题基础上，继续执行下列查询，编写 Cypher 语句并给出查询结果截图。

(1) 两个演员共同出演一部电影可以用如下关系表示:
(a)--[:PLAYED_IN*2]--(b)
称为演员 a 和 b 之间的合作距离为 1,
“贝肯数”是指与演员 Kevin Bacon（在 IMDB 数据集中 name 为“Bacon, Kevin (I)”）之间的合作距离。
查找贝肯数是 5 的 10 名演员，并输出由演员 Kevin Bacon 到这些演员的路径。

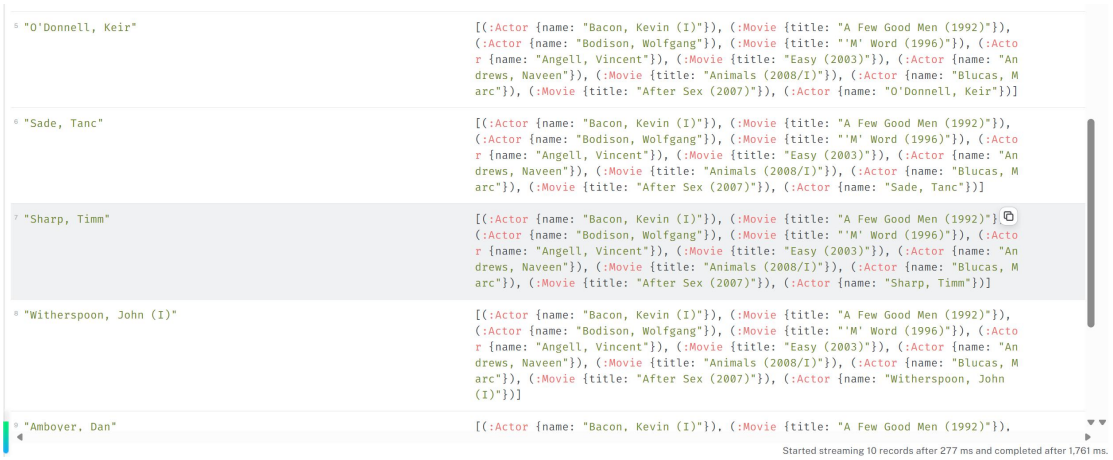
Cypher 语句：

```
MATCH path = (kevin:Actor {name: "Bacon, Kevin (I)"})-[:PLAYED_IN*10]-(actor:Actor)
WHERE length(path) = 10
WITH actor, path
LIMIT 10
RETURN actor.name, nodes(path) AS path;
```

查询结果截图：

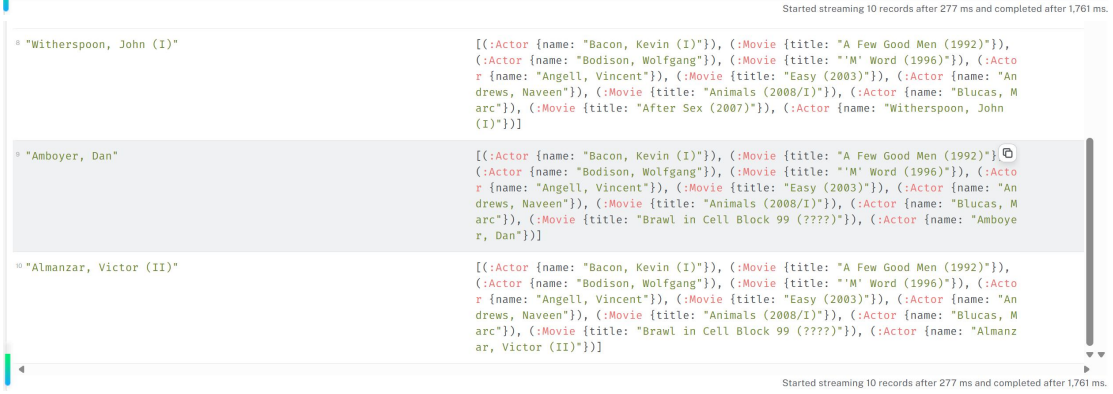
actor.name	path
"Cantillo, Jose Pablo"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Cantillo, Jose Pablo"})]
"DeBello, James"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "DeBello, James"})]
"Fisher, Noel (I)"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Fisher, Noel (I)"})]
"Franco, Dave (I)"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Franco, Dave (I)"})]
"O'Donnell, Keir"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}),

Started streaming 10 records after 277 ms and completed after 1,761 ms.



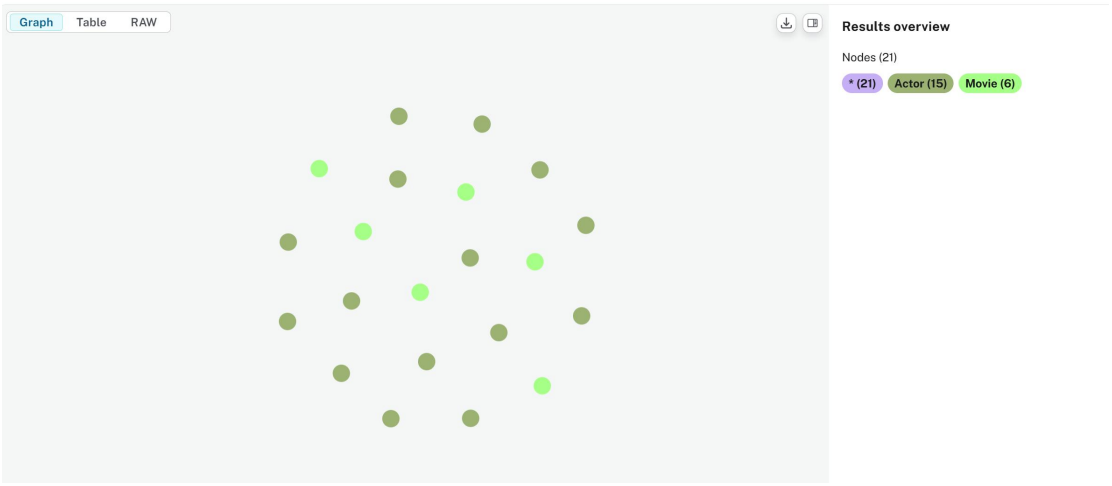
"O'Donnell, Keir"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "O'Donnell, Keir"})]
"Sade, Tanc"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Sade, Tanc"})]
"Sharp, Timm"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Sharp, Timm"})]
"Witherspoon, John (I)"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Witherspoon, John (I)"})]
"Amboyer, Dan"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}),

Started streaming 10 records after 277 ms and completed after 1,761 ms.



"Witherspoon, John (I)"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "After Sex (2007)"}), (:Actor {name: "Witherspoon, John (I)"})]
"Amboyer, Dan"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "Brawl in Cell Block 99 (????)"}), (:Actor {name: "Amboyer, Dan"})]
"Almanzar, Victor (II)"	[(:Actor {name: "Bacon, Kevin (I)"}), (:Movie {title: "A Few Good Men (1992)"}), (:Actor {name: "Bodison, Wolfgang"}), (:Movie {title: "'M' Word (1996)"}), (:Actor {name: "Angell, Vincent"}), (:Movie {title: "Easy (2003)"}), (:Actor {name: "Andrews, Naveen"}), (:Movie {title: "Animals (2008/I)"}), (:Actor {name: "Blucas, Marc"}), (:Movie {title: "Brawl in Cell Block 99 (????)"}), (:Actor {name: "Almanzar, Victor (II)"})]

Started streaming 10 records after 277 ms and completed after 1,761 ms.



(2) 查询出演电影数量最多的（降序排列）10 名演员及其出演的电影名称。

Cypher 语句：

```
MATCH (actor:Actor)-[:PLAYED_IN]->(movie:Movie)
WITH actor,count(movie) as movieCount,collect(movie.title) AS movieTitles
ORDER BY movieCount DESC
LIMIT 10
RETURN actor.name, movieTitles;
```

查询结果截图：

（只截图了一部分）