



Table of Content

1. Introduction:
2. Business Objective(s):
3. Business Question(s):
4. Data Exploration Database Queries, Results & Analysis:
5. Decisions & Conclusions:

1. Introduction:

In today's rapidly evolving movie industry, understanding the relationships and collaborations between actors and directors is essential for making data-driven decisions. This business report aims to explore the potential of the Neo4j graph NoSQL database for analyzing and visualizing these intricate connections within the film industry. By utilizing Neo4j's Sandbox, an accessible and user-friendly platform, we will dive into the available data and showcase how it can provide valuable insights to drive strategic business decisions.

2. Business Objective:

Analyze and understand the film industry landscape by identifying key players and their level of influence. Identify successful actors and directors based on their experience and output in the industry. Leverage social media presence to gauge the popularity and influence of actors, directors, and other film industry professionals. Discover potential partnerships, collaborations, or marketing opportunities with high-profile individuals in the film industry.

3. Business Question

1. What is the number of movies each actor was in?
2. What is the most number of movies a person has directed?
3. Which person has most followers?

4. Data Exploration Database Queries, Results & Analysis:

Looking for total relationships and nodes we have in the data

The screenshot shows the Neo4j Browser interface with two separate query sections.

Top Query:

```
neo4j$ // Count all relationships MATCH ()—>() RETURN count(*);
```

Result:

count(*)
1 253

Started streaming 1 records after 24 ms and completed after 25 ms.

Bottom Query:

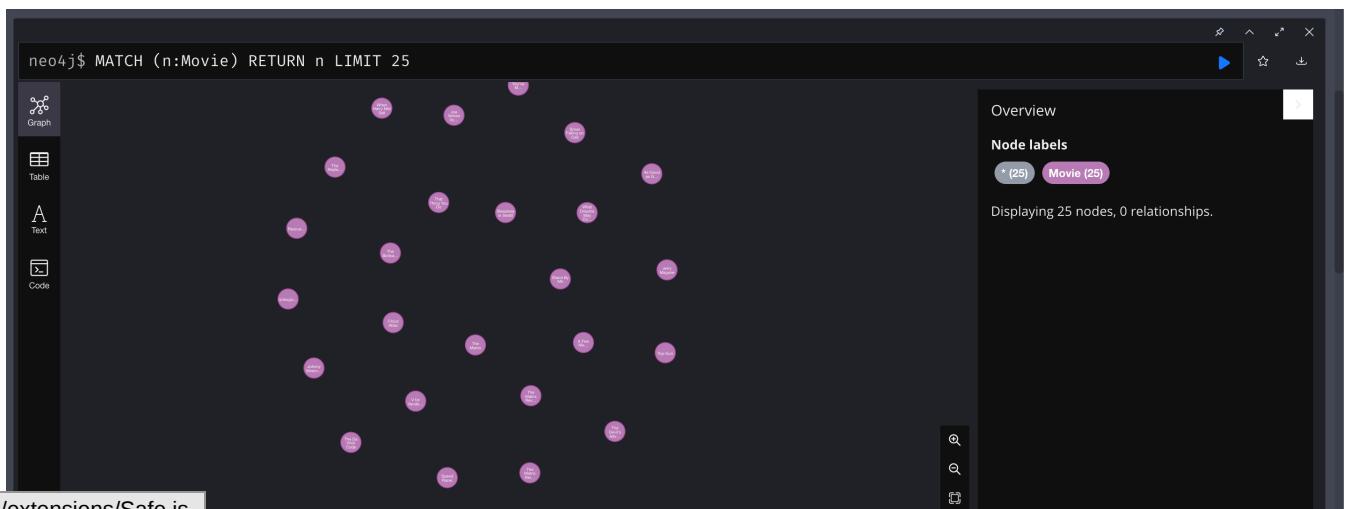
```
neo4j$ // Count all nodes MATCH (n) RETURN count(n)
```

Result:

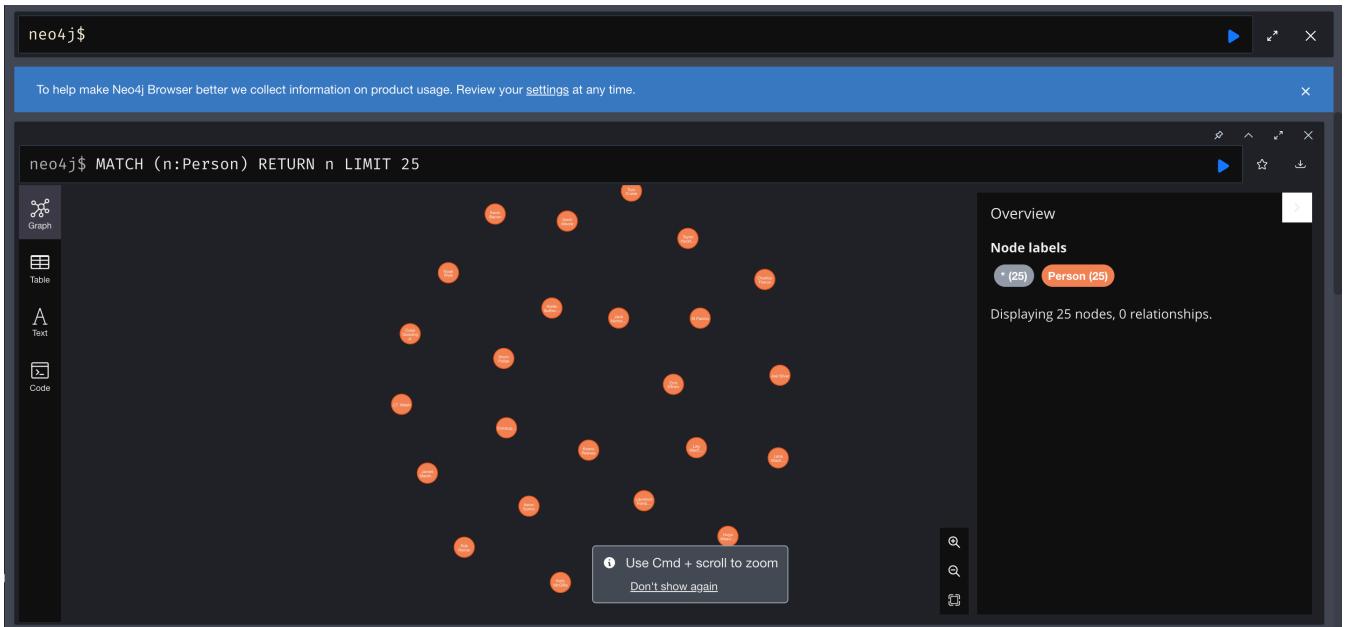
count(n)
1 171

Started streaming 1 records after 33 ms and completed after 34 ms.

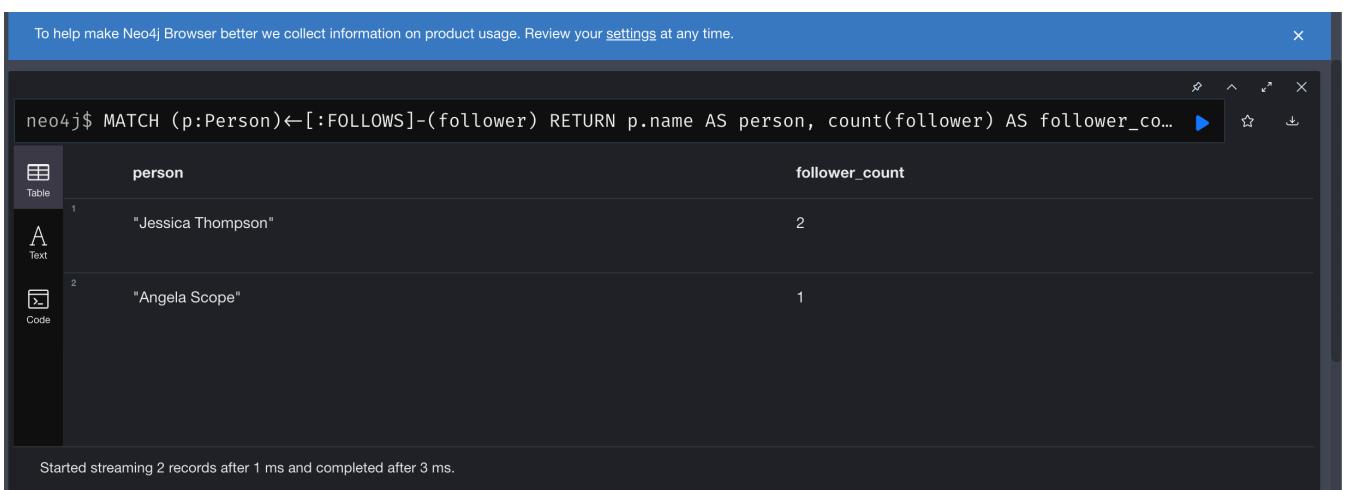
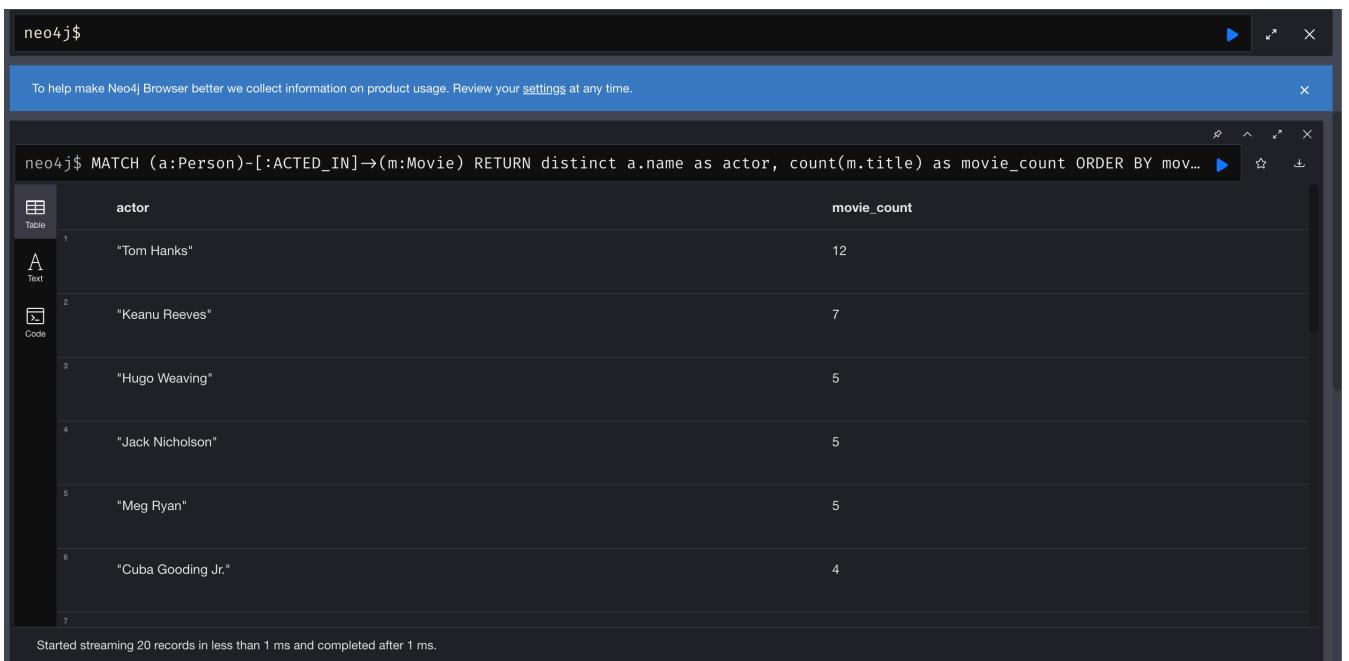
Visualizing movies



Visualizing persons



Running the query on Neo4j browser console



neo4j\$

To help make Neo4j Browser better we collect information on product usage. Review your [settings](#) at any time.

	actor	movie_count
1	"Lana Wachowski"	5
2	"Lilly Wachowski"	5
3	"Rob Reiner"	3
4	"Ron Howard"	3
5	"Mike Nichols"	2
6	"Nora Ephron"	2
7		

Started streaming 20 records after 2 ms and completed after 4 ms.

In [1]:

```
#installing py2neo
!pip install py2neo
```

Requirement already satisfied: py2neo in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (2021.2.3)

Requirement already satisfied: certifi in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (2021.10.8)

Requirement already satisfied: monotonic in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (1.6)

Requirement already satisfied: urllib3 in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (1.26.9)

Requirement already satisfied: pansi>=2020.7.3 in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (2020.7.3)

Requirement already satisfied: pygments>=2.0.0 in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (2.11.2)

Requirement already satisfied: packaging in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (21.3)

Requirement already satisfied: interchange~=2021.0.4 in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (2021.0.4)

Requirement already satisfied: six>=1.15.0 in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from py2neo) (1.16.0)

Requirement already satisfied: pytz in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from interchange~=2021.0.4->py2neo) (2021.3)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /Users/farazahmed/opt/anaconda3/lib/python3.9/site-packages (from packaging->py2neo) (3.0.4)

In [2]:

```
import py2neo
from py2neo import Graph, NodeMatcher
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [15]:

```
#Connecting to the sandbox
uri='bolt://3.219.31.164:7687'
graph=py2neo.Graph(uri,user="neo4j", password="verse-restraints-twirls" )
```

What is the number of movies each actor was in

```
In [5]: cypher="""
```

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
RETURN distinct a.name as actor, count(m.title) as movie_count
ORDER BY movie_count descending LIMIT 20
```

```
"""
```

```
In [6]: df = graph.run(cypher).to_data_frame()
```

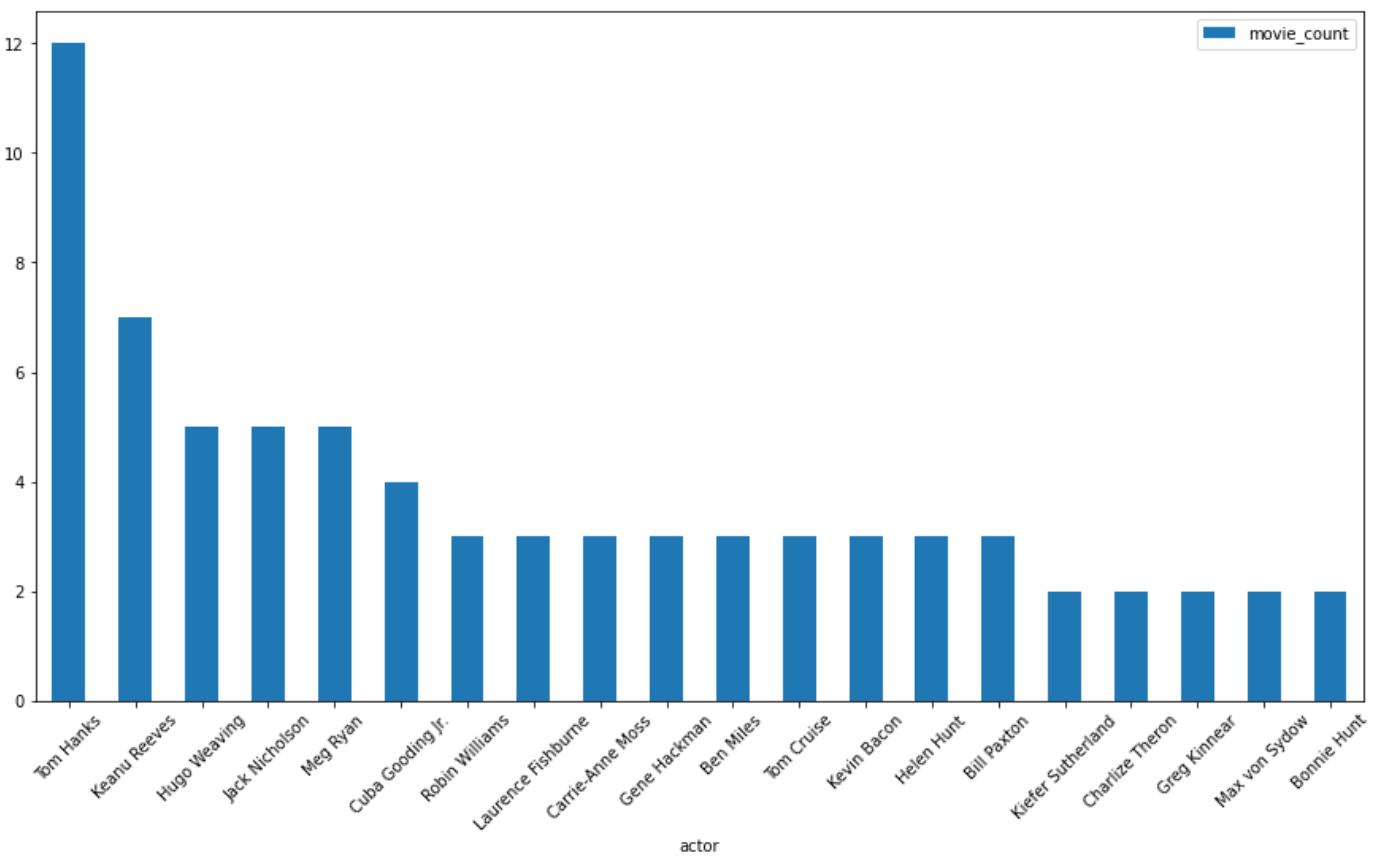
```
In [7]: df.head(20)
```

```
Out[7]:
```

	actor	movie_count
0	Tom Hanks	12
1	Keanu Reeves	7
2	Hugo Weaving	5
3	Jack Nicholson	5
4	Meg Ryan	5
5	Cuba Gooding Jr.	4
6	Robin Williams	3
7	Laurence Fishburne	3
8	Carrie-Anne Moss	3
9	Gene Hackman	3
10	Ben Miles	3
11	Tom Cruise	3
12	Kevin Bacon	3
13	Helen Hunt	3
14	Bill Paxton	3
15	Kiefer Sutherland	2
16	Charlize Theron	2
17	Greg Kinnear	2
18	Max von Sydow	2
19	Bonnie Hunt	2

```
In [8]: df.plot(kind='bar', x='actor', y='movie_count', figsize=(15,8), rot=45)
```

```
Out[8]: <AxesSubplot:xlabel='actor'>
```



```
In [9]: cypher=""
```

```
MATCH (a:Person)-[:DIRECTED]->(m:Movie)
RETURN distinct a.name as actor, count(m.title) as movie_count
ORDER BY movie_count DESCENDING LIMIT 20
```

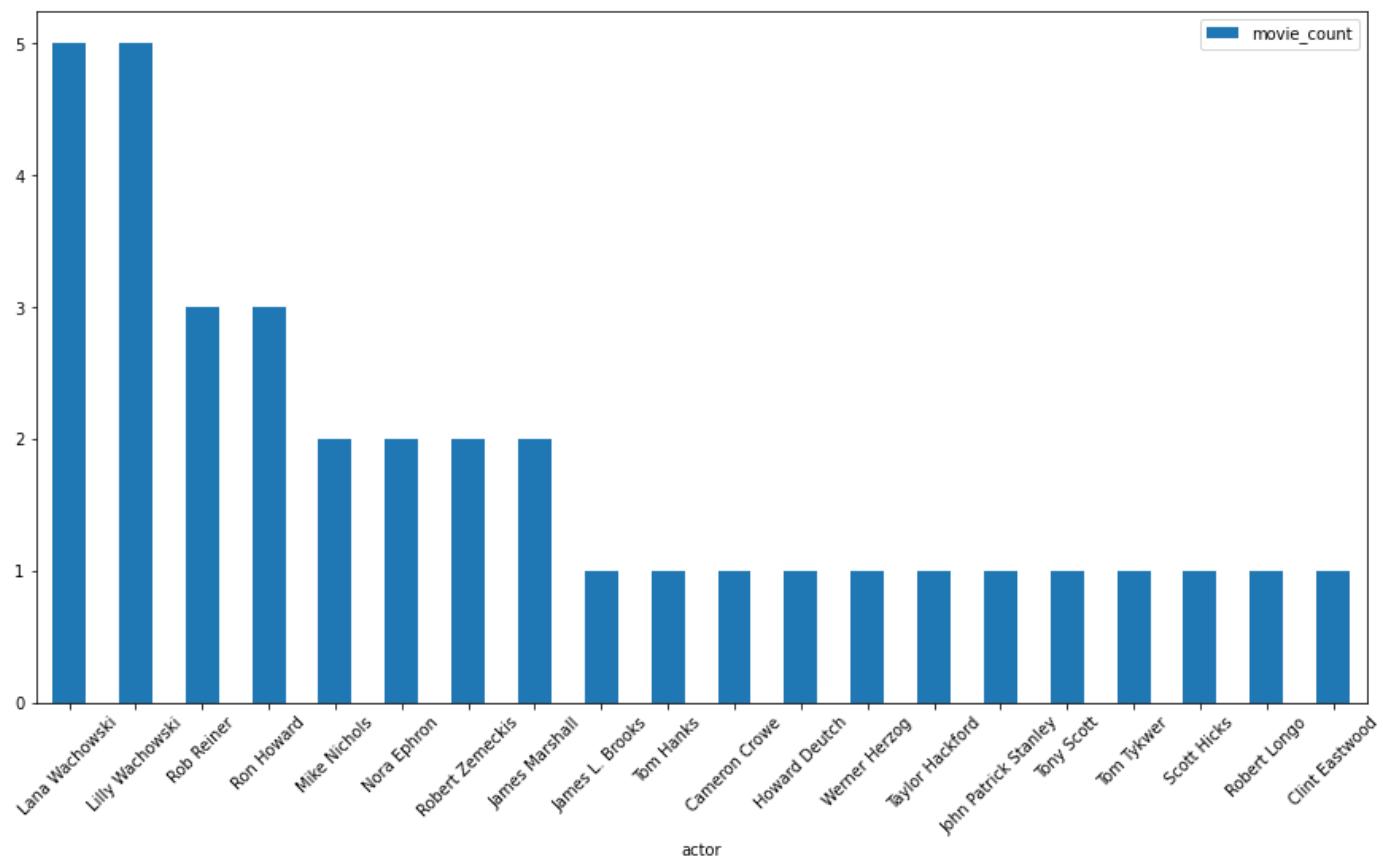
```
"""
```

```
In [10]: df = graph.run(cypher).to_data_frame()
df.head(20)
```

Out[10]:

	actor	movie_count
0	Lana Wachowski	5
1	Lilly Wachowski	5
2	Rob Reiner	3
3	Ron Howard	3
4	Mike Nichols	2
5	Nora Ephron	2
6	Robert Zemeckis	2
7	James Marshall	2
8	James L. Brooks	1
9	Tom Hanks	1
10	Cameron Crowe	1
11	Howard Deutch	1
12	Werner Herzog	1
13	Taylor Hackford	1
14	John Patrick Stanley	1
15	Tony Scott	1
16	Tom Tykwer	1
17	Scott Hicks	1
18	Robert Longo	1
19	Clint Eastwood	1

In [11]: `df.plot(kind='bar', x='actor', y='movie_count', figsize=(15,8), rot=45)`Out[11]: `<AxesSubplot:xlabel='actor'>`



```
In [20]: cypher="""
MATCH (p:Person)<-[:FOLLOWS]-(follower)
RETURN p.name AS person, count(follower) AS follower_count
ORDER BY follower_count DESC
LIMIT 23

"""
```

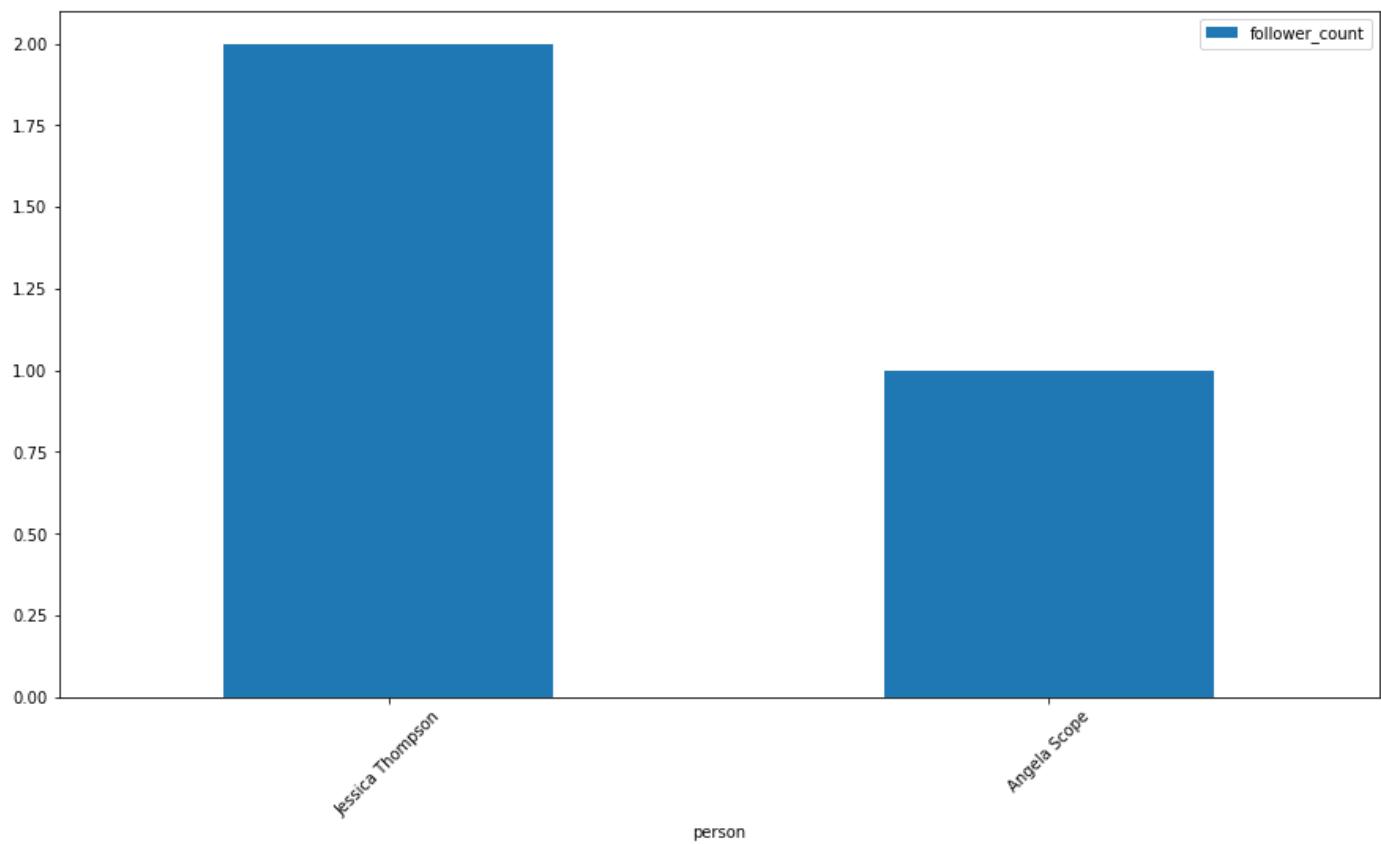
```
In [21]: df = graph.run(cypher).to_data_frame()
df.head(20)
```

```
Out[21]:
```

	person	follower_count
0	Jessica Thompson	2
1	Angela Scope	1

```
In [22]: df.plot(kind='bar', x='person', y='follower_count', figsize=(15,8), rot=45)
```

```
Out[22]: <AxesSubplot:xlabel='person'>
```



5. Decisions & Conclusions:

Tom Hanks is the most prolific actor in the list, with 12 movies to his name. He could be a valuable asset for future projects due to his extensive experience in the film industry.

Keanu Reeves, with 7 movies, and other actors with 5 or more movies, such as Hugo Weaving, Jack Nicholson, and Meg Ryan, can also be considered for potential projects or partnerships due to their notable experience.

Lana and Lilly Wachowski have directed the highest number of movies (5 each) among the listed directors. They could be targeted for potential collaborations or partnerships, given their directing expertise.

The social media presence of the individuals in our dataset seems to be very low, with Jessica Thompson having only 2 followers and Angela Scope having just 1 follower. It might be necessary to verify the follower count data for accuracy or gather more data on other industry professionals' social media presence to make informed decisions about potential partnerships based on online influence.

To identify potential collaborations, it might be useful to explore the relationships between successful actors and directors, such as which directors actors frequently work with and vice versa. This could help identify synergistic pairs or teams that have a track record of successful projects.

Since the data on social media presence is limited, it would be beneficial to gather more data on industry professionals' online influence, as this could provide valuable insights for marketing campaigns or promotional strategies.

In summary, the data suggests that Tom Hanks, Keanu Reeves, and the Wachowski siblings are among the most experienced and successful professionals in the dataset. However, more information on social media

presence is needed to make informed decisions about potential partnerships based on popularity and online influence.