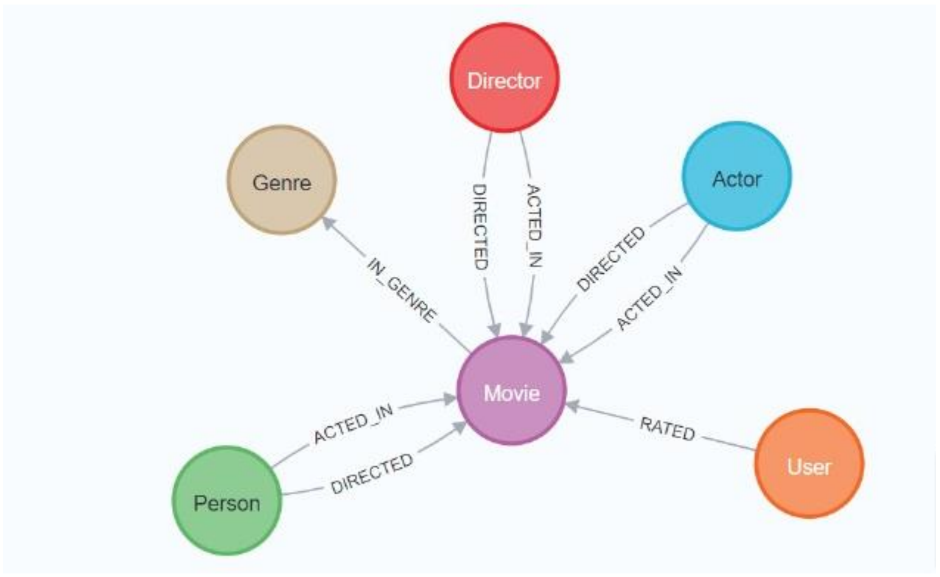
REPORT

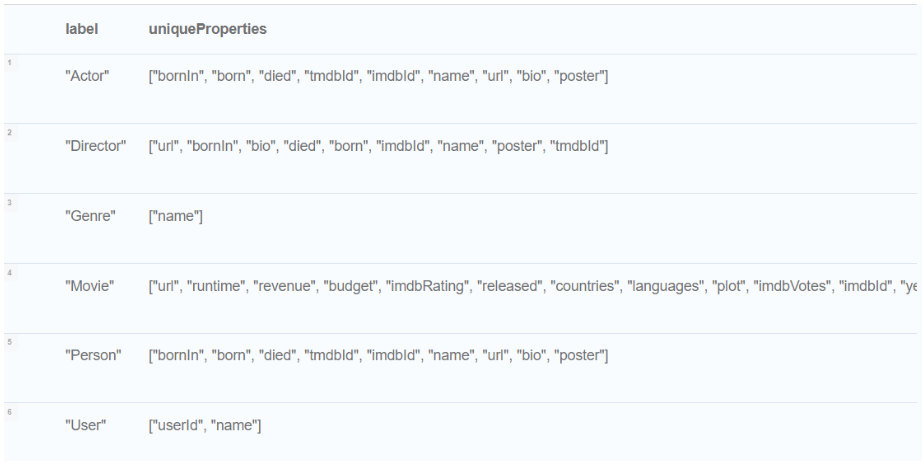
For our project, we chose to work with the movie recommendation dataset and we imported the corresponding dump file. The dataset schema is shown below:



Next, we will write queries to uncover insights about your data that might be very difficult to retrieve in other forms of databases.

To identify all node properties:

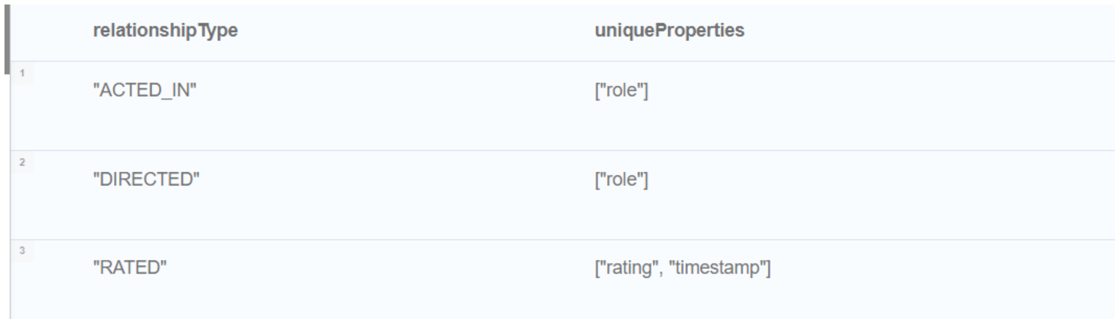
```
CALL {
  MATCH (n)
  RETURN labels(n) AS labels, keys(n) AS properties
}
UNWIND labels AS label
WITH label, properties
UNWIND properties AS property
WITH label, COLLECT(DISTINCT property) AS uniqueProperties
RETURN label, uniqueProperties ORDER BY label
```



	label	uniqueProperties
1	"Actor"	["bornIn", "born", "died", "tmdbId", "imdbId", "name", "url", "bio", "poster"]
2	"Director"	["url", "bornIn", "bio", "died", "born", "imdbId", "name", "poster", "tmdbId"]
3	"Genre"	["name"]
4	"Movie"	["url", "runtime", "revenue", "budget", "imdbRating", "released", "countries", "languages", "plot", "imdbVotes", "imdbId", "ye
5	"Person"	["bornIn", "born", "died", "tmdbId", "imdbId", "name", "url", "bio", "poster"]
6	"User"	["userId", "name"]

To identify all relationships properties:

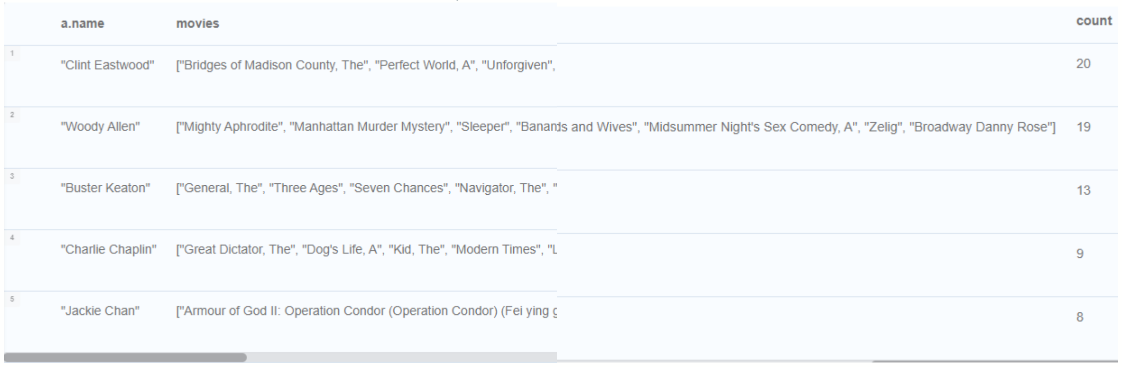
```
CALL {
  MATCH ()-[r]->()
  RETURN type(r) as relationshipType, keys(r) AS properties
}
UNWIND properties AS property
WITH relationshipType, COLLECT(DISTINCT property) AS uniqueProperties
RETURN relationshipType, uniqueProperties ORDER BY relationshipType
```



	relationshipType	uniqueProperties
1	"ACTED_IN"	["role"]
2	"DIRECTED"	["role"]
3	"RATED"	["rating", "timestamp"]

1. **Find the person who acted and directed the most movies**

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE (a)-[:DIRECTED]->(m)
RETURN a.name, collect(m.title) as movies, count(*) as count
ORDER BY count DESC LIMIT 5;
```

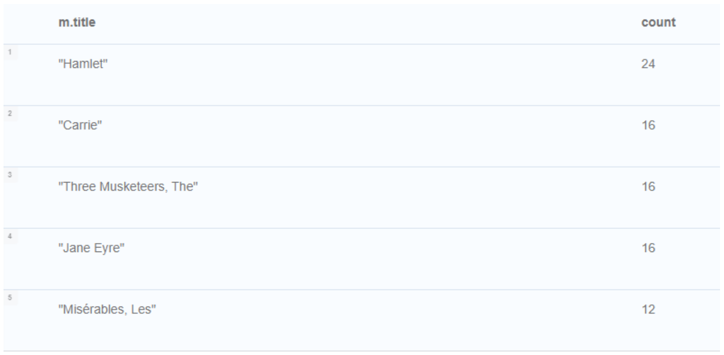


	a.name	movies	count
1	"Clint Eastwood"	["Bridges of Madison County, The", "Perfect World, A", "Unforgiven",	20
2	"Woody Allen"	["Mighty Aphrodite", "Manhattan Murder Mystery", "Sleeper", "Banands and Wives", "Midsummer Night's Sex Comedy, A", "Zelig", "Broadway Danny Rose"]	19
3	"Buster Keaton"	["General, The", "Three Ages", "Seven Chances", "Navigator, The", "	13
4	"Charlie Chaplin"	["Great Dictator, The", "Dog's Life, A", "Kid, The", "Modern Times", "L	9
5	"Jackie Chan"	["Armour of God II: Operation Condor (Operation Condor) (Fei ying g	8

We find that Clint Easten is the one that directed and acted in the most movies.

2. **Find the movies with the most actors:**

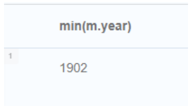
```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
RETURN m.title, count(a.name) as count
ORDER BY count DESC LIMIT 5;
```



	m.title	count
1	"Hamlet"	24
2	"Carrie"	16
3	"Three Musketeers, The"	16
4	"Jane Eyre"	16
5	"Misérables, Les"	12

3. **Find the year of the oldest movie:**

```
MATCH (m:Movie) RETURN min(m.year)
```



	min(m.year)
1	1902

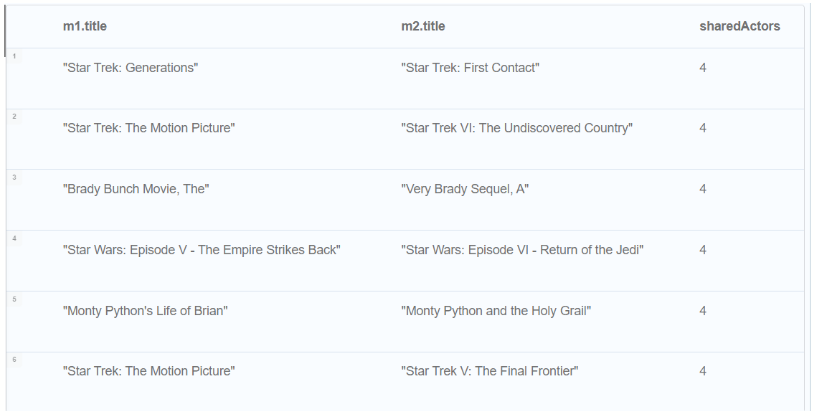
It is 1902

4. **Find movies that share the most actors:**

```
MATCH (m1:Movie)<-[:ACTED_IN]-(a:Actor)-[:ACTED_IN]->(m2:Movie)
WHERE id(m1) < id(m2)
WITH m1, m2, COUNT(a) AS sharedActors
ORDER BY sharedActors DESC
RETURN m1.title, m2.title, sharedActors
LIMIT 6
```

This query identifies pairs of movies that share the most actors, which might indicate similar casting preferences or genres.

Result:



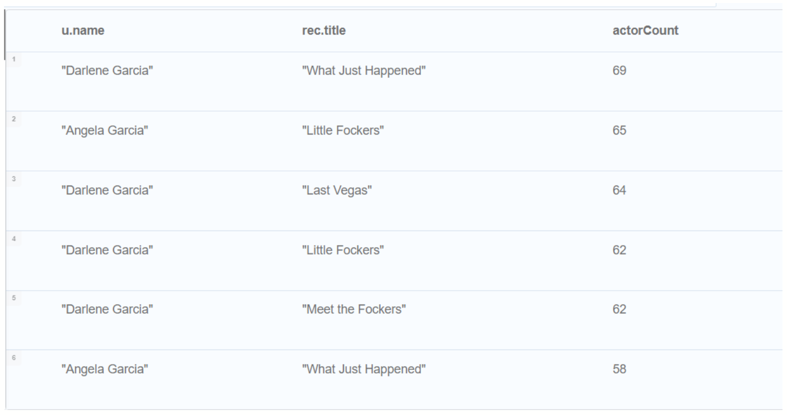
	m1.title	m2.title	sharedActors
1	"Star Trek: Generations"	"Star Trek: First Contact"	4
2	"Star Trek: The Motion Picture"	"Star Trek VI: The Undiscovered Country"	4
3	"Brady Bunch Movie, The"	"Very Brady Sequel, A"	4
4	"Star Wars: Episode V - The Empire Strikes Back"	"Star Wars: Episode VI - Return of the Jedi"	4
5	"Monty Python's Life of Brian"	"Monty Python and the Holy Grail"	4
6	"Star Trek: The Motion Picture"	"Star Trek V: The Final Frontier"	4

5. **Recommend Movies to Users Based on Their Favorite Actors:**

```
MATCH (u:User)-[:RATED]->(m:Movie)<-[:ACTED_IN]-(a:Actor),
(a)-[:ACTED_IN]->(rec:Movie)
WHERE NOT (u)-[:RATED]->(rec) AND m <> rec
WITH u, rec, COUNT(a) AS actorCount
ORDER BY actorCount DESC
RETURN u.name, rec.title, actorCount
LIMIT 6
```

This query recommends movies to users that feature actors from other movies the user has rated highly.

Results:



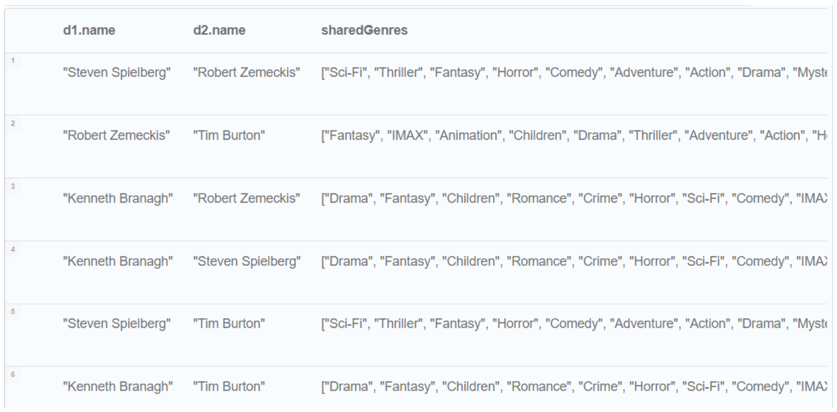
	u.name	rec.title	actorCount
1	"Darlene Garcia"	"What Just Happened"	69
2	"Angela Garcia"	"Little Fockers"	65
3	"Darlene Garcia"	"Last Vegas"	64
4	"Darlene Garcia"	"Little Fockers"	62
5	"Darlene Garcia"	"Meet the Fockers"	62
6	"Angela Garcia"	"What Just Happened"	58

6. **Detect Potential Collaboration Between Directors Based on Genre Preference**:

```
MATCH (d1:Director)-[:DIRECTED]->(:Movie)-[:IN_GENRE]->(g:Genre)<-[:IN_GENRE]-
(:Movie)<-[:DIRECTED]-(d2:Director)
WHERE id(d1) < id(d2)
WITH d1, d2, COLLECT(DISTINCT g.name) AS sharedGenres
RETURN d1.name, d2.name, sharedGenres, SIZE(sharedGenres) AS genreOverlap
ORDER BY genreOverlap DESC
LIMIT 6
```

This query identifies directors who have not worked together but have directed movies in similar genres, suggesting potential for future collaboration.

Result:



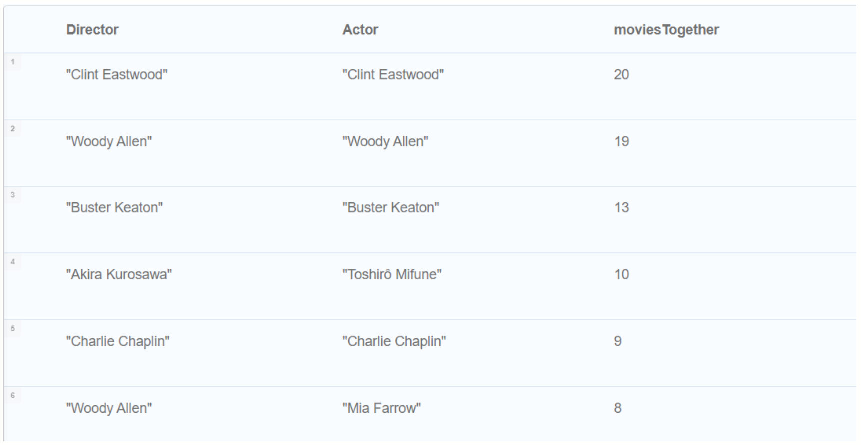
	d1.name	d2.name	sharedGenres
1	"Steven Spielberg"	"Robert Zemeckis"	["Sci-Fi", "Thriller", "Fantasy", "Horror", "Comedy", "Adventure", "Action", "Drama", "Myst
2	"Robert Zemeckis"	"Tim Burton"	["Fantasy", "IMAX", "Animation", "Children", "Drama", "Thriller", "Adventure", "Action", "H
3	"Kenneth Branagh"	"Robert Zemeckis"	["Drama", "Fantasy", "Children", "Romance", "Crime", "Horror", "Sci-Fi", "Comedy", "IMA
4	"Kenneth Branagh"	"Steven Spielberg"	["Drama", "Fantasy", "Children", "Romance", "Crime", "Horror", "Sci-Fi", "Comedy", "IMA
5	"Steven Spielberg"	"Tim Burton"	["Sci-Fi", "Thriller", "Fantasy", "Horror", "Comedy", "Adventure", "Action", "Drama", "Myst
6	"Kenneth Branagh"	"Tim Burton"	["Drama", "Fantasy", "Children", "Romance", "Crime", "Horror", "Sci-Fi", "Comedy", "IMA

7. **Director-Actor frequent partnership**:

```
MATCH (d:Director)-[:DIRECTED]->(m:Movie)<-[:ACTED_IN]-(a:Actor)
WITH d, a, COUNT(m) AS moviesTogether
ORDER BY moviesTogether DESC
RETURN d.name AS Director, a.name AS Actor, moviesTogether
LIMIT 6
```

This query ranks actor-director pairs by the number of movies they have worked on together, highlighting strong professional relationships or influences.

Result:



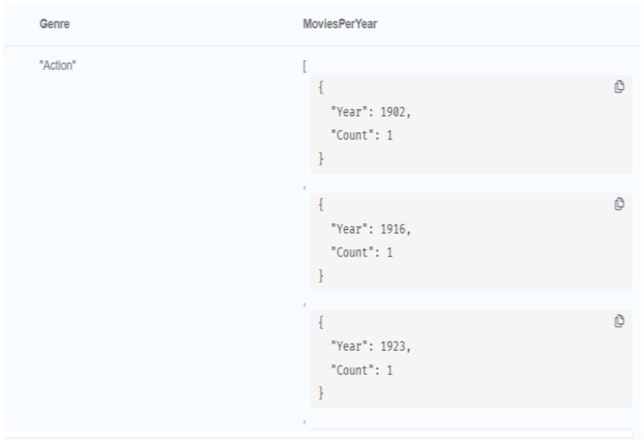
	Director	Actor	moviesTogether
1	"Clint Eastwood"	"Clint Eastwood"	20
2	"Woody Allen"	"Woody Allen"	19
3	"Buster Keaton"	"Buster Keaton"	13
4	"Akira Kurosawa"	"Toshirô Mifune"	10
5	"Charlie Chaplin"	"Charlie Chaplin"	9
6	"Woody Allen"	"Mia Farrow"	8

8. **Genre Trends Over Time**:

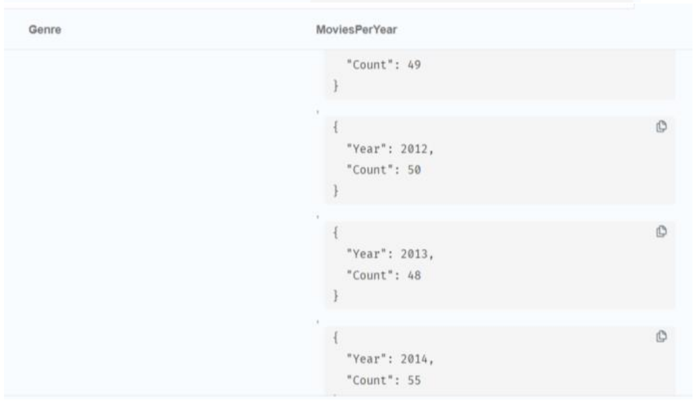
```
MATCH (m:Movie)-[:IN_GENRE]->(g:Genre)
WHERE m.year IS NOT NULL
WITH g.name AS Genre, m.year AS Year, COUNT(m) AS MoviesCount
ORDER BY Year
RETURN Genre, collect({Year: Year, Count: MoviesCount}) AS MoviesPerYear
```

This query maps out the popularity of genres over the years by counting the number of movies in each genre per year.

Result:



```
Genre        MoviesPerYear
"Action"     [
               {
                 "Year": 1902,
                 "Count": 1
               }
             ,
               {
                 "Year": 1916,
                 "Count": 1
               }
             ,
               {
                 "Year": 1923,
                 "Count": 1
               }
             ,
```

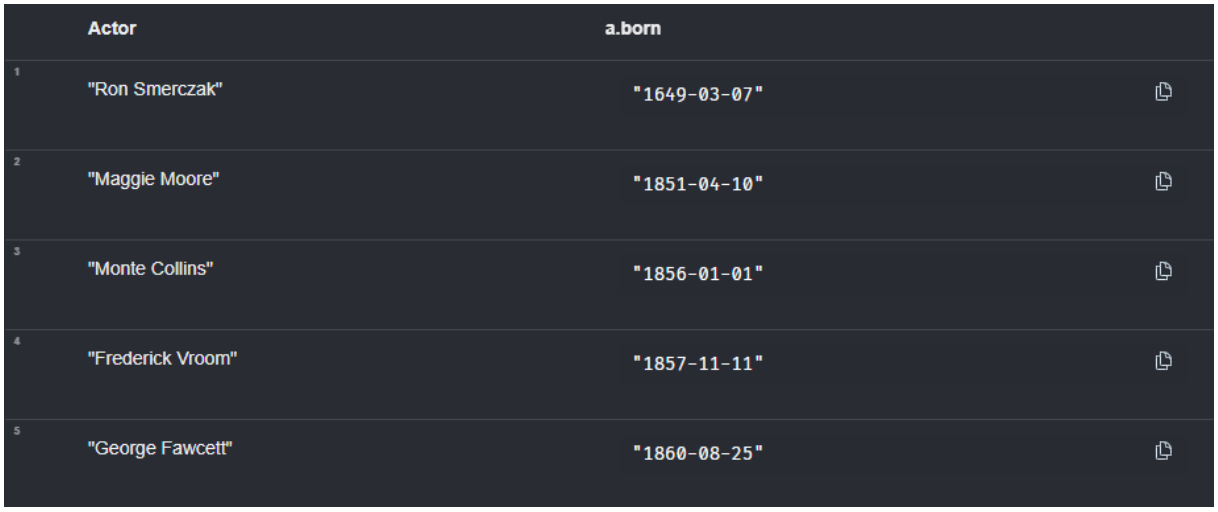


```
Genre        MoviesPerYear
                 "Count": 49
               }
             ,
               {
                 "Year": 2012,
                 "Count": 50
               }
             ,
               {
                 "Year": 2013,
                 "Count": 48
               }
             ,
               {
                 "Year": 2014,
                 "Count": 55
```

The results showed a huge increase in movie production throughout the years

9. Names of actors who played in comedies

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)-[:IN_GENRE]->(g:Genre)
WHERE g.name = 'Comedy'
RETURN DISTINCT a.name AS Actor, a.born
ORDER BY a.born ASC
LIMIT 5;
```

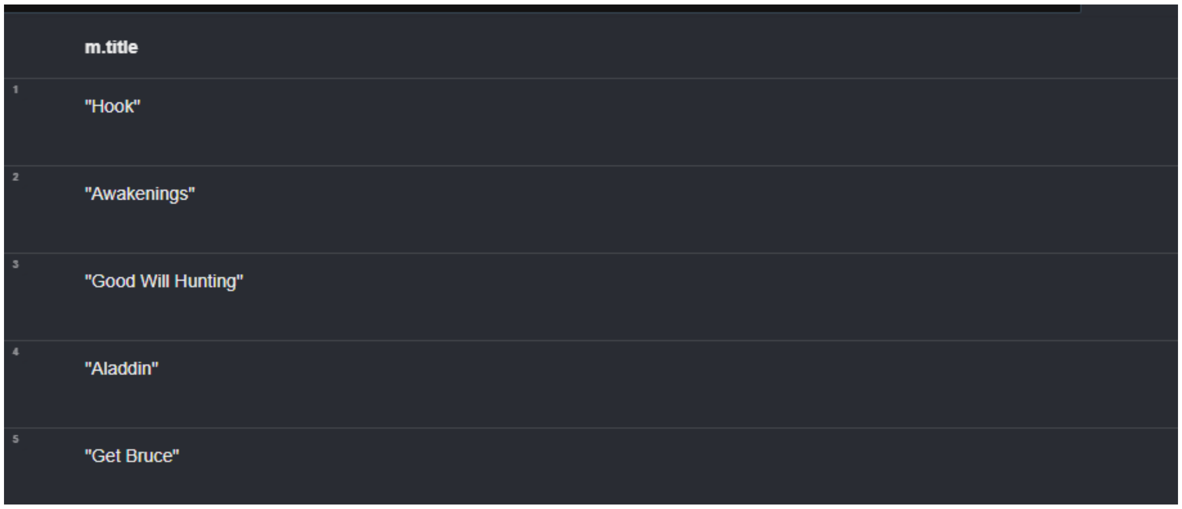


	Actor	a.born
1	"Ron Smerczak"	"1649-03-07"
2	"Maggie Moore"	"1851-04-10"
3	"Monte Collins"	"1856-01-01"
4	"Frederick Vroom"	"1857-11-11"
5	"George Fawcett"	"1860-08-25"

We order the results by the year of birth of the actors

10. Get all the actors from "Jumanji" and for each actor get a movie that the actor has acted in:

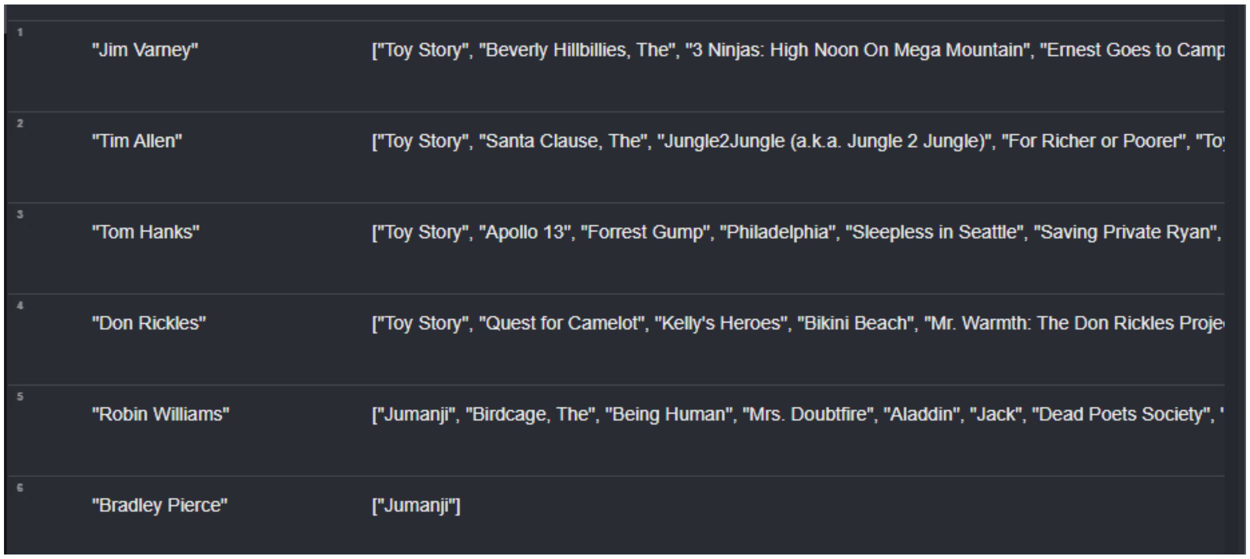
```
MATCH (a:Person)-[:ACTED_IN]->(:Movie {title: "Jumanji"})
MATCH (a)-[ACTED_IN]->(m:Movie)
RETURN m.title
LIMIT 5
```



	m.title
1	"Hook"
2	"Awakenings"
3	"Good Will Hunting"
4	"Aladdin"
5	"Get Bruce"

11. Get the names of all the movies that each actor had a role in:

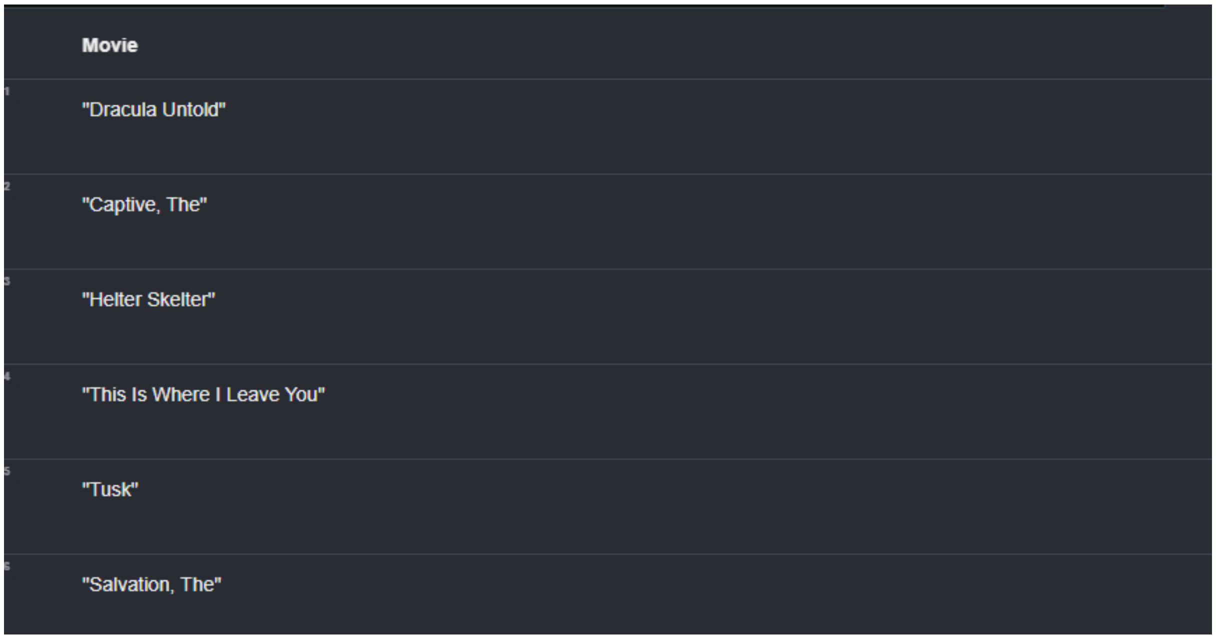
```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
RETURN a.name AS Actors, COLLECT(m.title) AS movies
```



```
1  "Jim Varney"       ["Toy Story", "Beverly Hillbillies, The", "3 Ninjas: High Noon On Mega Mountain", "Ernest Goes to Camp
2  "Tim Allen"        ["Toy Story", "Santa Clause, The", "Jungle2Jungle (a.k.a. Jungle 2 Jungle)", "For Richer or Poorer", "To
3  "Tom Hanks"        ["Toy Story", "Apollo 13", "Forrest Gump", "Philadelphia", "Sleepless in Seattle", "Saving Private Ryan",
4  "Don Rickles"      ["Toy Story", "Quest for Camelot", "Kelly's Heroes", "Bikini Beach", "Mr. Warmth: The Don Rickles Proje
5  "Robin Williams"   ["Jumanji", "Birdcage, The", "Being Human", "Mrs. Doubtfire", "Aladdin", "Jack", "Dead Poets Society", '
6  "Bradley Pierce"   ["Jumanji"]
```

12. **Get the movies where the genre is drama**

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)-[:IN_GENRE]->(g:Genre)
WHERE g.name ="Drama"
RETURN DISTINCT  m.title AS Movie
```



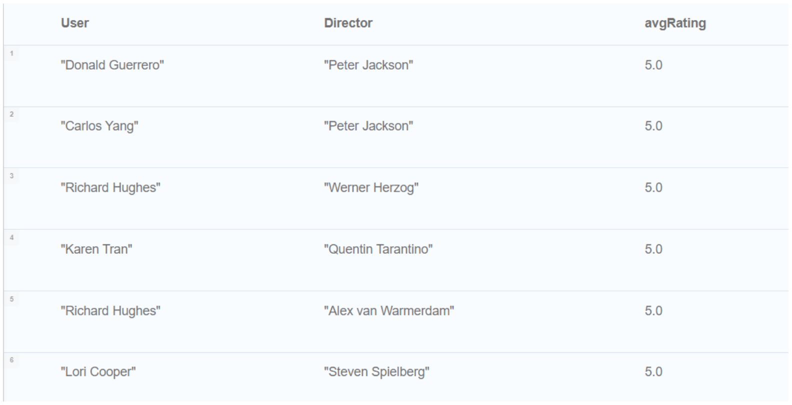
```
   Movie
1  "Dracula Untold"
2  "Captive, The"
3  "Helter Skelter"
4  "This Is Where I Leave You"
5  "Tusk"
6  "Salvation, The"
```

13. **User Rating Patterns for Directors**:

```
MATCH (u:User)-[r:RATED]->(m:Movie)<-[:DIRECTED]-(d:Director)
WITH u, d, AVG(r.rating) AS avgRating, COUNT(r) AS ratingsCount
WHERE ratingsCount > 5
RETURN u.name AS User, d.name AS Director, avgRating
ORDER BY avgRating DESC
LIMIT 6
```

This query finds average ratings users have given to movies, grouped by director, indicating user preferences for certain directors' styles.

Result:



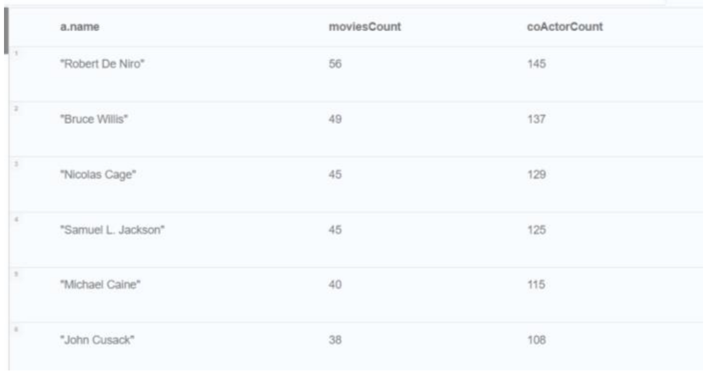
	User	Director	avgRating
1	"Donald Guerrero"	"Peter Jackson"	5.0
2	"Carlos Yang"	"Peter Jackson"	5.0
3	"Richard Hughes"	"Werner Herzog"	5.0
4	"Karen Tran"	"Quentin Tarantino"	5.0
5	"Richard Hughes"	"Alex van Warmerdam"	5.0
6	"Lori Cooper"	"Steven Spielberg"	5.0

14. **Central Actors in the Movie Network**:

```
MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)
WITH a, COUNT(m) AS moviesCount
MATCH (a)-[:ACTED_IN]->()<-[:ACTED_IN]-(coActor)
WITH a, moviesCount, COUNT(DISTINCT coActor) AS coActorCount
RETURN a.name, moviesCount, coActorCount
ORDER BY coActorCount DESC, moviesCount DESC
LIMIT 10
```

This query identifies actors who have worked with many different co-actors, which might suggest their central role in the movie industry.

Result:



	a.name	moviesCount	coActorCount
1	"Robert De Niro"	56	145
2	"Bruce Willis"	49	137
3	"Nicolas Cage"	45	129
4	"Samuel L. Jackson"	45	125
5	"Michael Caine"	40	115
6	"John Cusack"	38	108

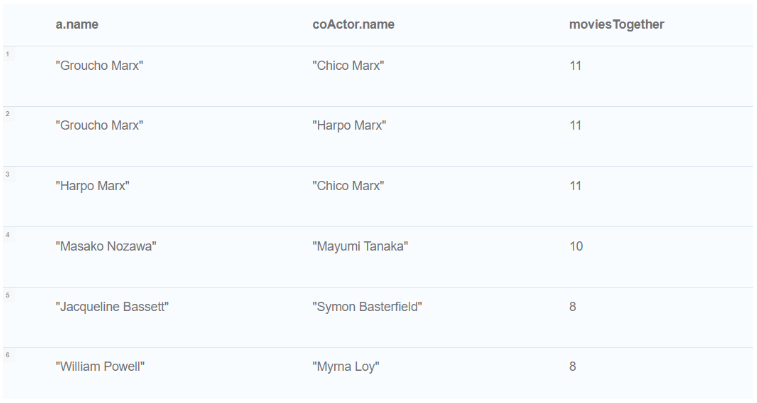
The results verifies the purpose behind this query, as we can see the output lists top stars in Hollywood.

15. **Cluster of Actors Who Frequently Work Together**:

```
MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(coActor:Actor)
WITH a, coActor, COUNT(m) AS moviesTogether
WHERE moviesTogether > 3 AND id(a) < id(coActor)
RETURN a.name, coActor.name, moviesTogether
ORDER BY moviesTogether DESC
LIMIT 10
```

This query looks for pairs of actors who frequently appear in the same movies, hinting at strong collaboration networks.

Result:



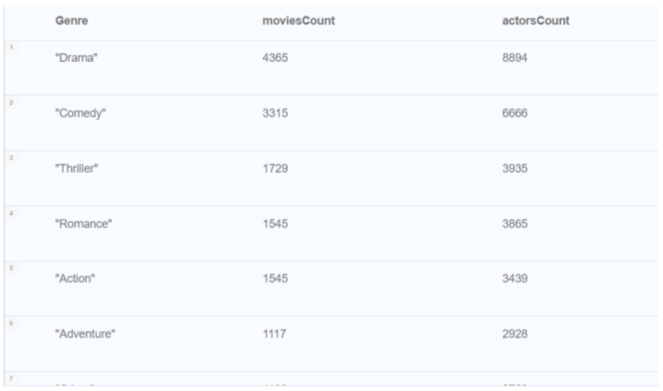
	a.name	coActor.name	moviesTogether
1	"Groucho Marx"	"Chico Marx"	11
2	"Groucho Marx"	"Harpo Marx"	11
3	"Harpo Marx"	"Chico Marx"	11
4	"Masako Nozawa"	"Mayumi Tanaka"	10
5	"Jacqueline Bassett"	"Symon Basterfield"	8
6	"William Powell"	"Myrna Loy"	8

16. **Influential Genres in the Film Industry**:

```
MATCH (g:Genre)<-[:IN_GENRE]-(m:Movie)
WITH g, COUNT(m) AS moviesCount
MATCH (g)<-[:IN_GENRE]-(m)<-[:ACTED_IN]-(a:Actor)
WITH g, moviesCount, COUNT(DISTINCT a) AS actorsCount
ORDER BY moviesCount DESC, actorsCount DESC
RETURN g.name AS Genre, moviesCount, actorsCount
LIMIT 10
```

Reveals which genres have the most movies and actors involved, indicating their influence in the industry.

Result:



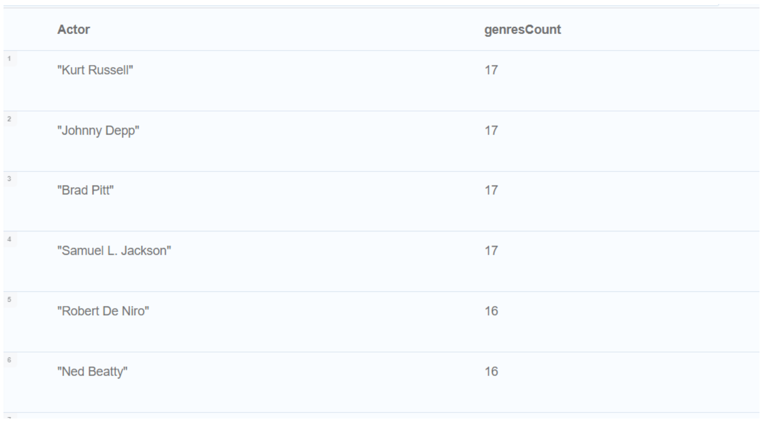
Genre	moviesCount	actorsCount
"Drama"	4365	8894
"Comedy"	3315	6666
"Thriller"	1729	3935
"Romance"	1545	3865
"Action"	1545	3439
"Adventure"	1117	2928

17. Actor Versatility:

```
MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)-[:IN_GENRE]->(g:Genre)
WITH a, COUNT(DISTINCT g) AS genresCount
ORDER BY genresCount DESC
RETURN a.name AS Actor, genresCount
LIMIT 10
```

Finds actors who have acted in the most diverse range of genres, showing their versatility.

Result:



Actor	genresCount
"Kurt Russell"	17
"Johnny Depp"	17
"Brad Pitt"	17
"Samuel L. Jackson"	17
"Robert De Niro"	16
"Ned Beatty"	16

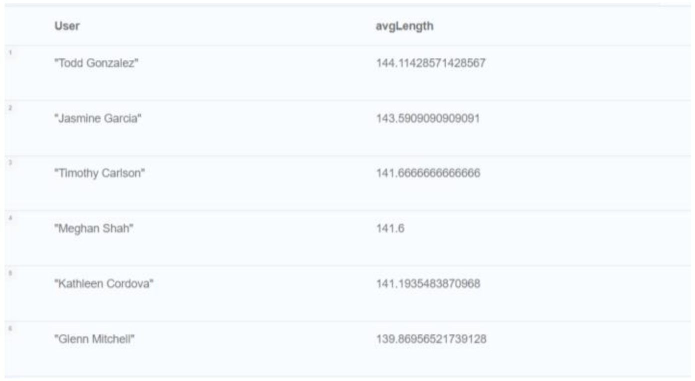
It shows that Kurt Russell, Johnny Depp, Brad Pitt and Samuel L. Jackson are the most versatile actors.

18. **User Preferences for Movie Length**:

```
MATCH (u:User)-[:RATED]->(m:Movie)
WITH u, AVG(m.runtime) AS avgLength
RETURN u.name AS User, avgLength
ORDER BY avgLength DESC
LIMIT 10
```

Discovers if certain users have a preference for longer or shorter movies based on the average length of movies they've rated.

Result:



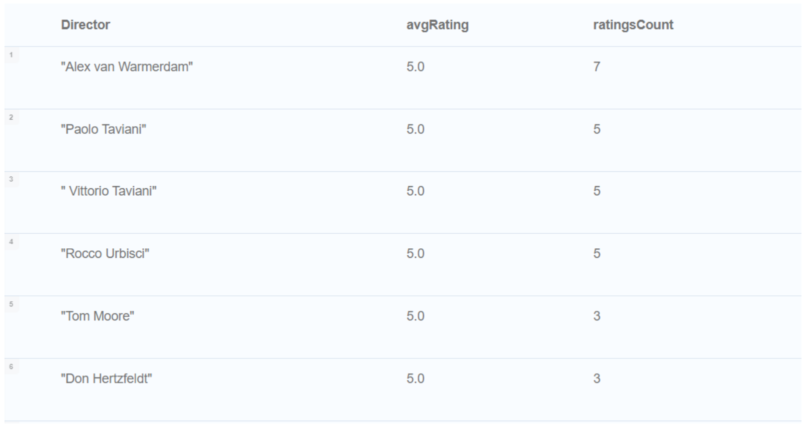
	User	avgLength
1	"Todd Gonzalez"	144.11428571428567
2	"Jasmine Garcia"	143.5909090909091
3	"Timothy Carlson"	141.66666666666666
4	"Meghan Shah"	141.6
5	"Kathleen Cordova"	141.1935483870968
6	"Glenn Mitchell"	139.86956521739128

19. Directors with the highest user ratings:

```
MATCH (d:Director)-[:DIRECTED]->(m:Movie)<-[r:RATED]-(u:User)
WITH d, AVG(r.rating) AS avgRating, COUNT(r) AS ratingsCount
ORDER BY avgRating DESC, ratingsCount DESC
RETURN d.name AS Director, avgRating, ratingsCount
LIMIT 10
```

Identifies directors whose movies receive the highest average user ratings, adjusted for the number of ratings

Result:

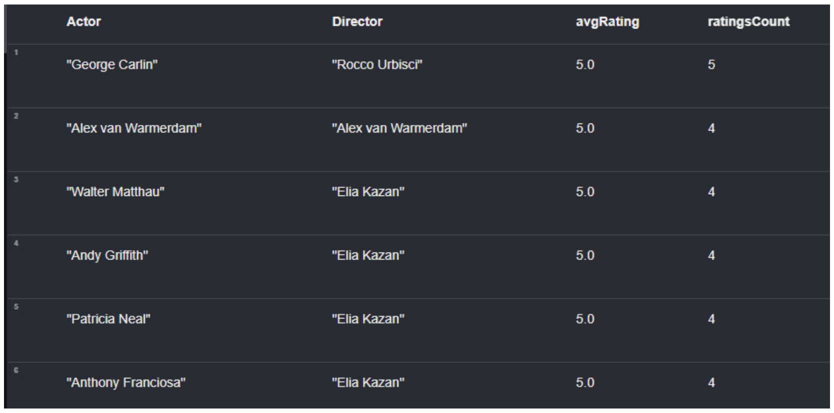


	Director	avgRating	ratingsCount
1	"Alex van Warmerdam"	5.0	7
2	"Paolo Taviani"	5.0	5
3	" Vittorio Taviani"	5.0	5
4	"Rocco Urbisci"	5.0	5
5	"Tom Moore"	5.0	3
6	"Don Hertzfeldt"	5.0	3

20. Actor-Director pair with best rating:

```
MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)<-[r:RATED]-(u:User),
    (d:Director)-[:DIRECTED]->(m)
WITH a, d, AVG(r.rating) AS avgRating, COUNT(r) AS ratingsCount
ORDER BY avgRating DESC, ratingsCount DESC
RETURN a.name AS Actor, d.name AS Director, avgRating, ratingsCount
LIMIT 10
```

Finds actor-director pairs that seem to be a hit with the audience, based on average ratings

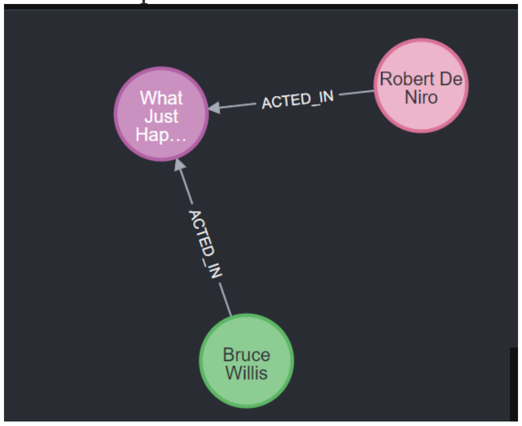


	Actor	Director	avgRating	ratingsCount
1	"George Carlin"	"Rocco Urbisci"	5.0	5
2	"Alex van Warmerdam"	"Alex van Warmerdam"	5.0	4
3	"Walter Matthau"	"Elia Kazan"	5.0	4
4	"Andy Griffith"	"Elia Kazan"	5.0	4
5	"Patricia Neal"	"Elia Kazan"	5.0	4
6	"Anthony Franciosa"	"Elia Kazan"	5.0	4

21. Shortest path between actors:

If two actors have acted together, the shortest path should be the movie they've both acted in. But if two actors never acted together, the shortest path between them would be a chain of movies and actors that connect them together.

```
MATCH path=shortestPath((a1:Actor)-[:ACTED_IN*]-(a2:Actor))
WHERE a1.name = 'Bruce Willis' AND a2.name = 'Robert De Niro'
RETURN path
```



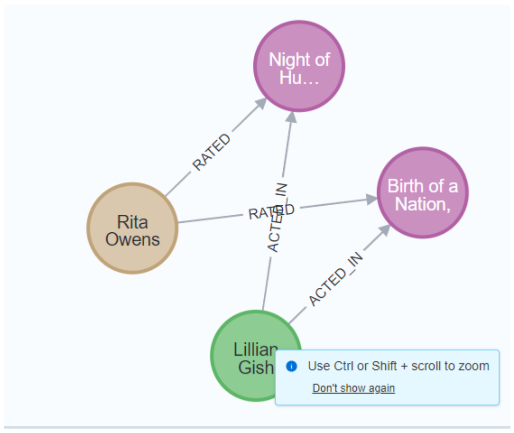
The shortest path between Robert De Niro and Bruce Willis was found to be a movie they both were part of which is "What Just Happened."

22. Loop search

We are looking for a path of three to five hops where we end up with the person that we started with.

```
MATCH path = (p:Person)-[*3..5]-(p)
RETURN path
LIMIT 1
```

Result:


Night of Hu…
Birth of a Nation,
Rita Owens
Lillian Gish
RATED
RATED
ACTED_IN
ACTED_IN
Use Ctrl or Shift + scroll to zoom
Don't show again