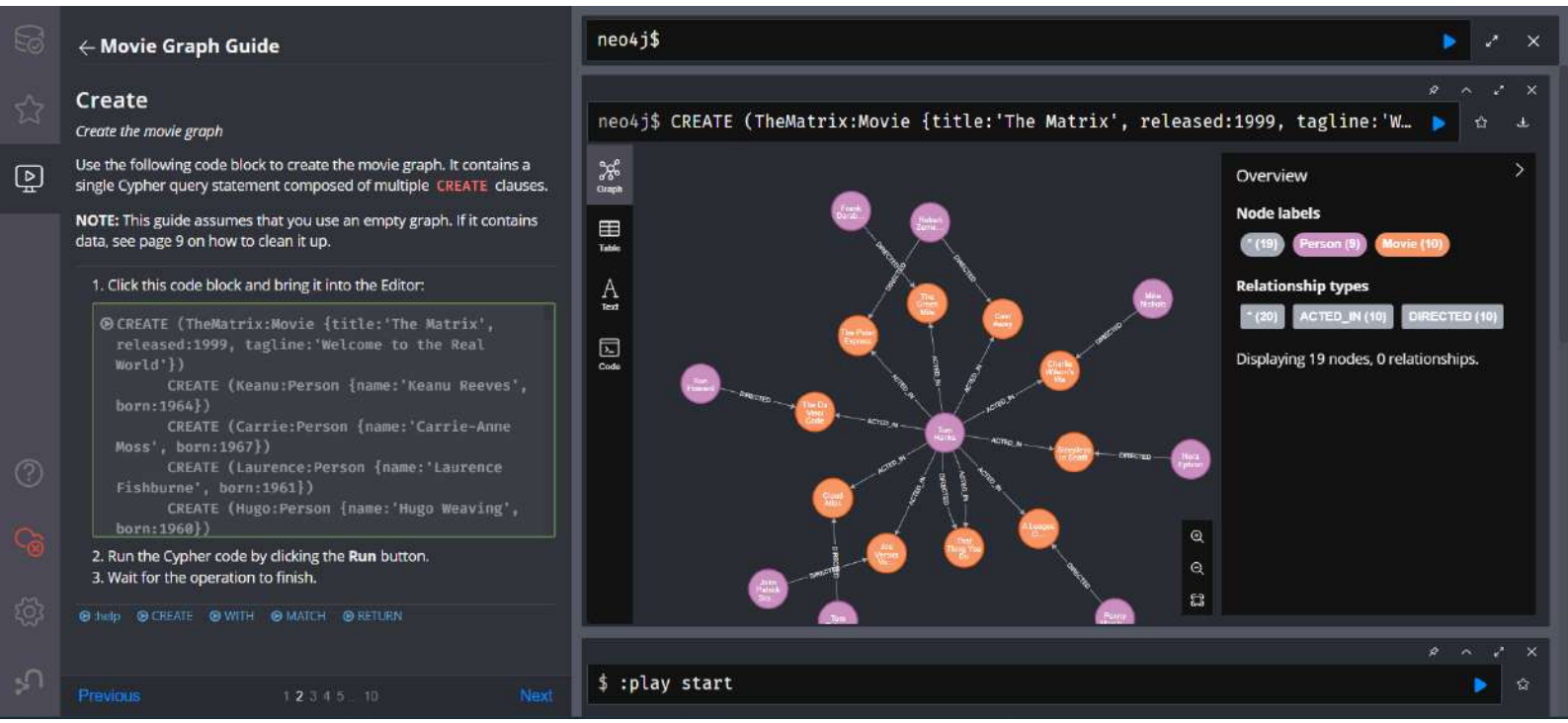
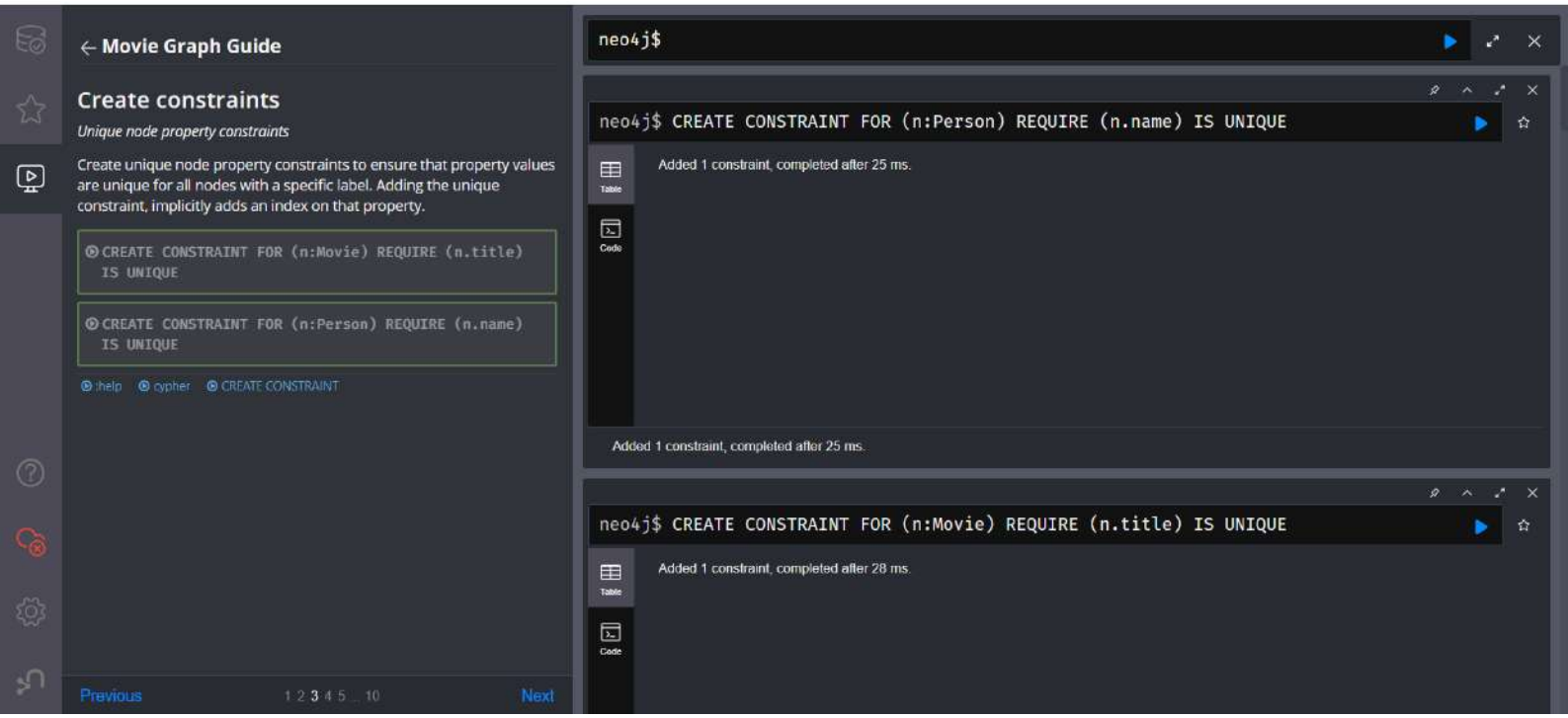
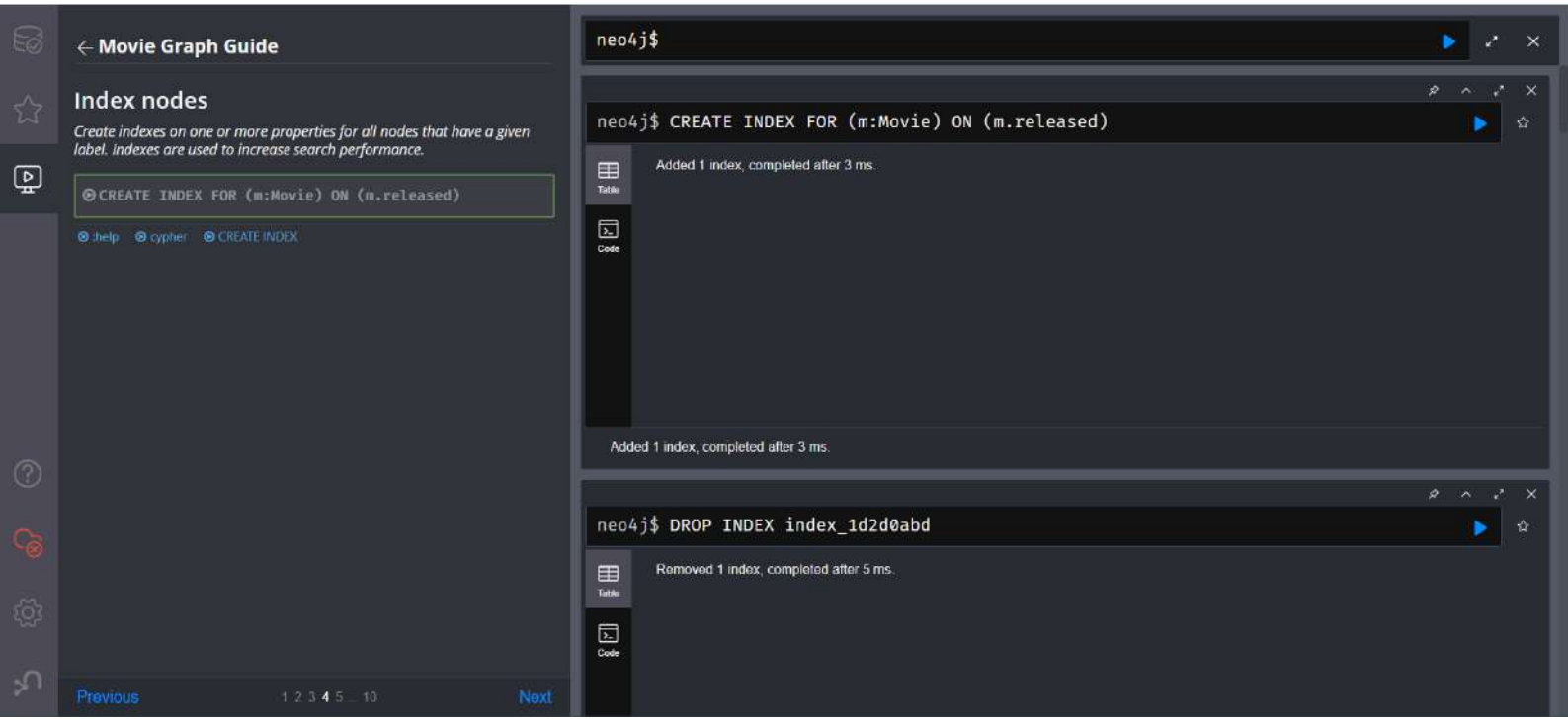
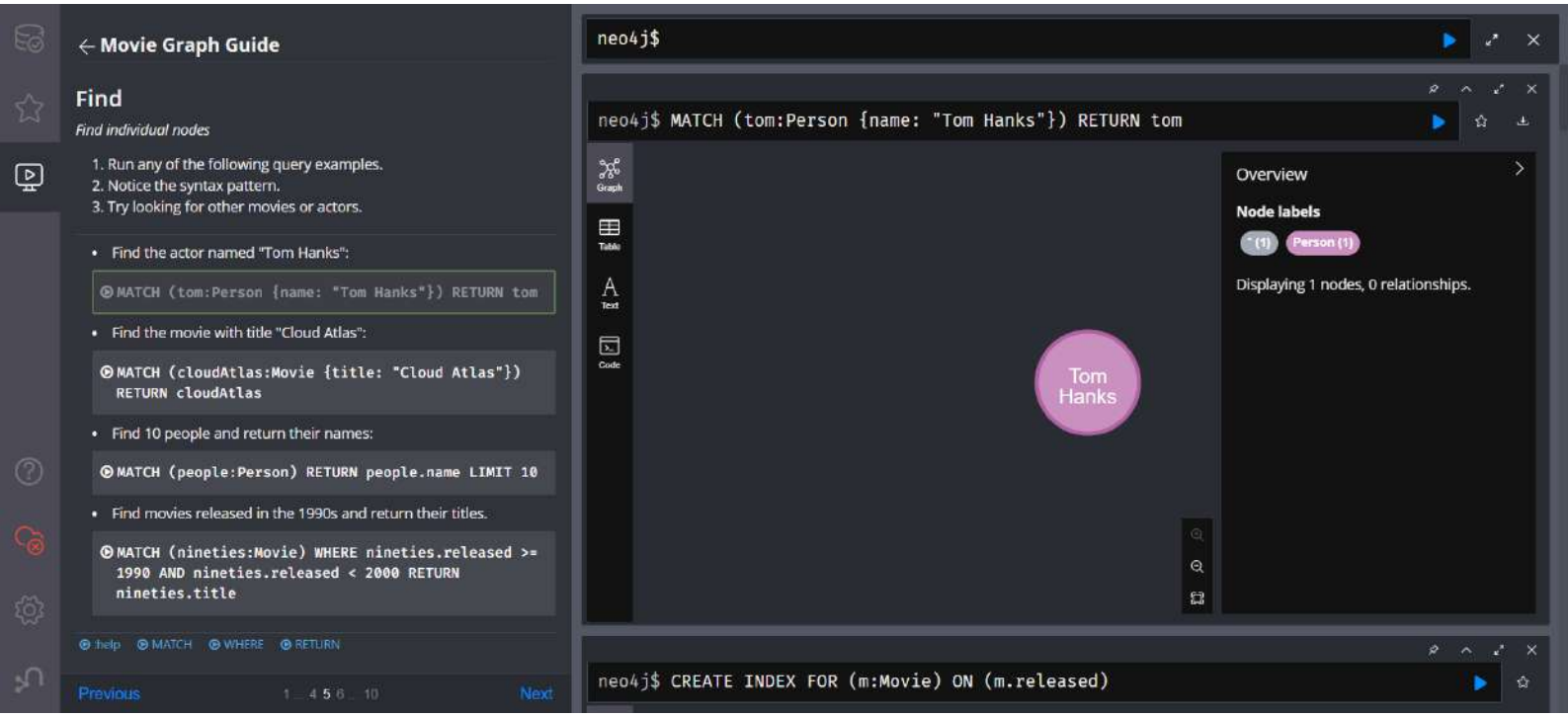
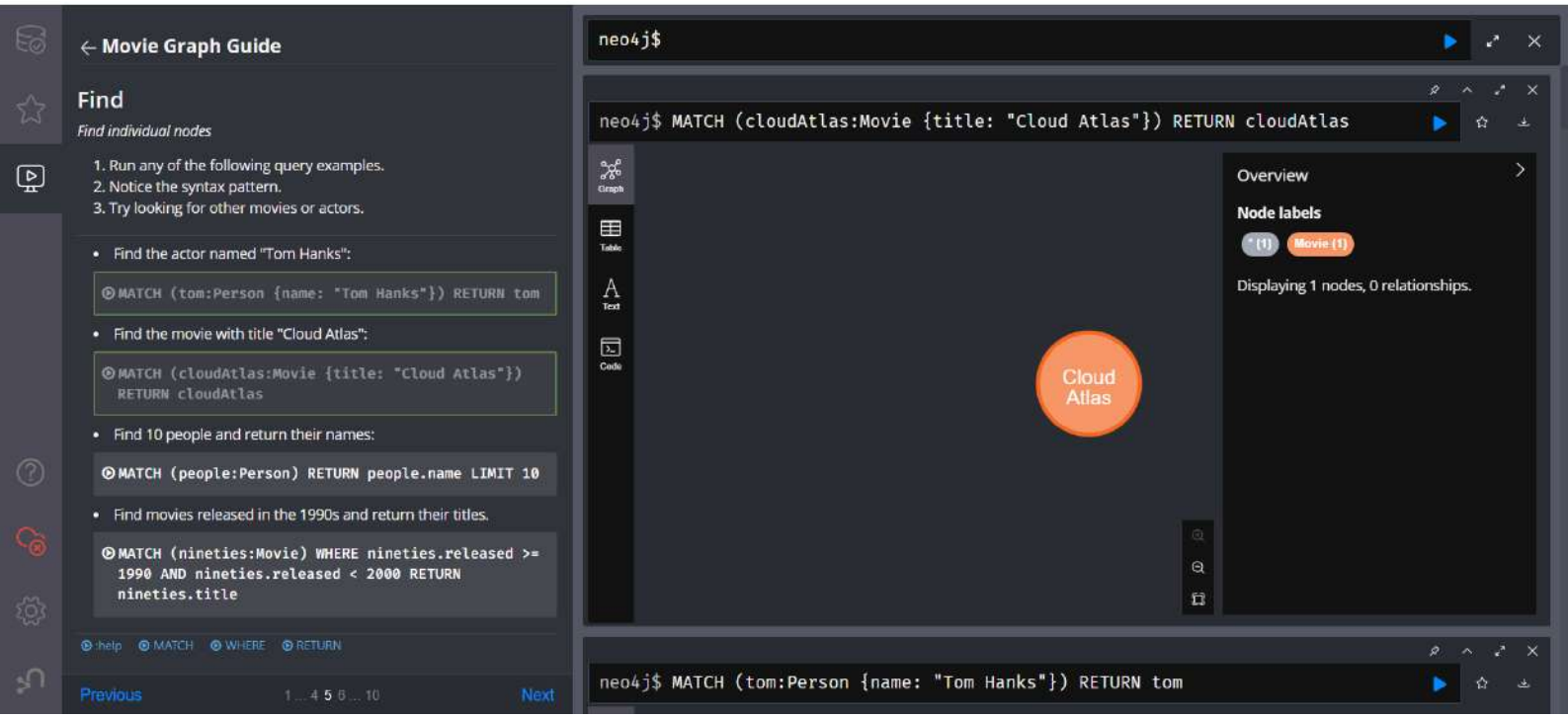
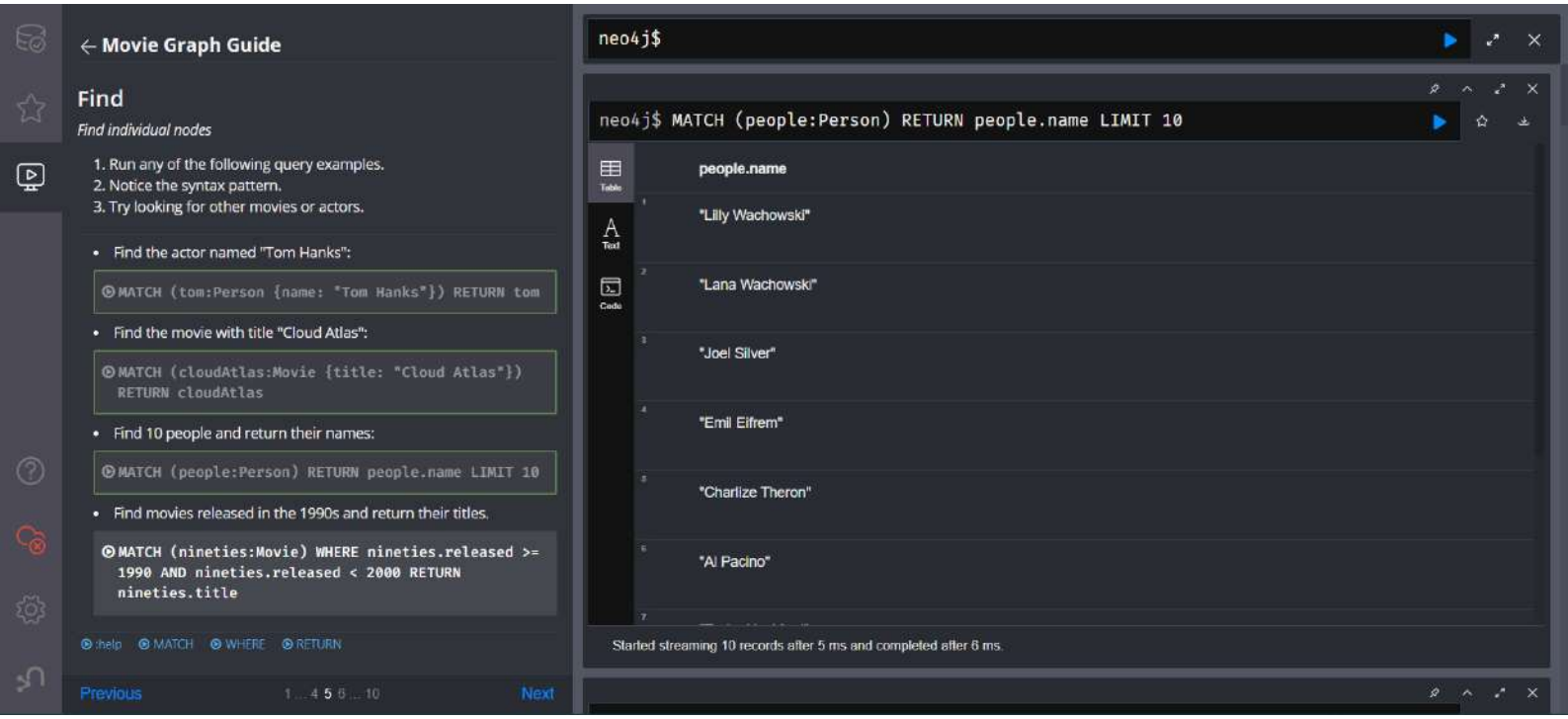

← Movie Graph Guide
Create
Create the movie graph
Use the following code block to create the movie graph. It contains a single Cypher query statement composed of multiple CREATE clauses.
NOTE: This guide assumes that you use an empty graph. If it contains data, see page 9 on how to clean it up.
1. Click this code block and bring it into the Editor:
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})
CREATE (Laurence:Person {name:'Laurence Fishburne', born:1961})
CREATE (Hugo:Person {name:'Hugo Weaving', born:1960})
2. Run the Cypher code by clicking the Run button.
3. Wait for the operation to finish.
:help
CREATE
WITH
MATCH
RETURN
Previous
1 2 3 4 5 … 10
Next
neo4j$
neo4j$ CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'W…
Graph
Table
Text
Code
Overview
Node labels
* (19)
Person (9)
Movie (10)
Relationship types
* (20)
ACTED_IN (10)
DIRECTED (10)
Displaying 19 nodes, 0 relationships.
$:play start

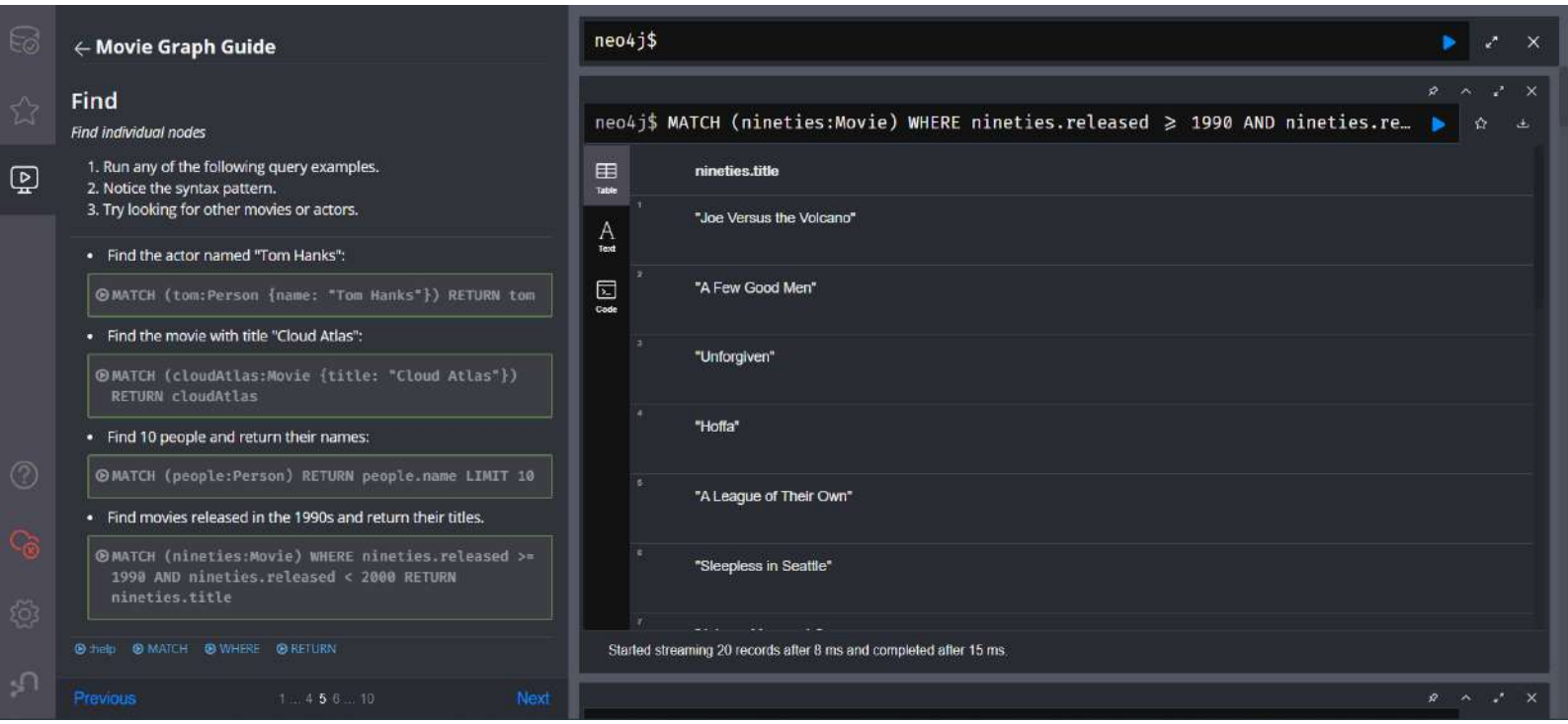

Movie Graph Guide
Create constraints
Unique node property constraints
Create unique node property constraints to ensure that property values are unique for all nodes with a specific label. Adding the unique constraint, implicitly adds an index on that property.
CREATE CONSTRAINT FOR (n:Movie) REQUIRE (n.title) IS UNIQUE
CREATE CONSTRAINT FOR (n:Person) REQUIRE (n.name) IS UNIQUE
:help
cypher
CREATE CONSTRAINT
Previous
1 2 3 4 5 ... 10
Next
neo4j$
neo4j$ CREATE CONSTRAINT FOR (n:Person) REQUIRE (n.name) IS UNIQUE
Table
Code
Added 1 constraint, completed after 25 ms.
Added 1 constraint, completed after 25 ms.
neo4j$ CREATE CONSTRAINT FOR (n:Movie) REQUIRE (n.title) IS UNIQUE
Table
Code
Added 1 constraint, completed after 28 ms.

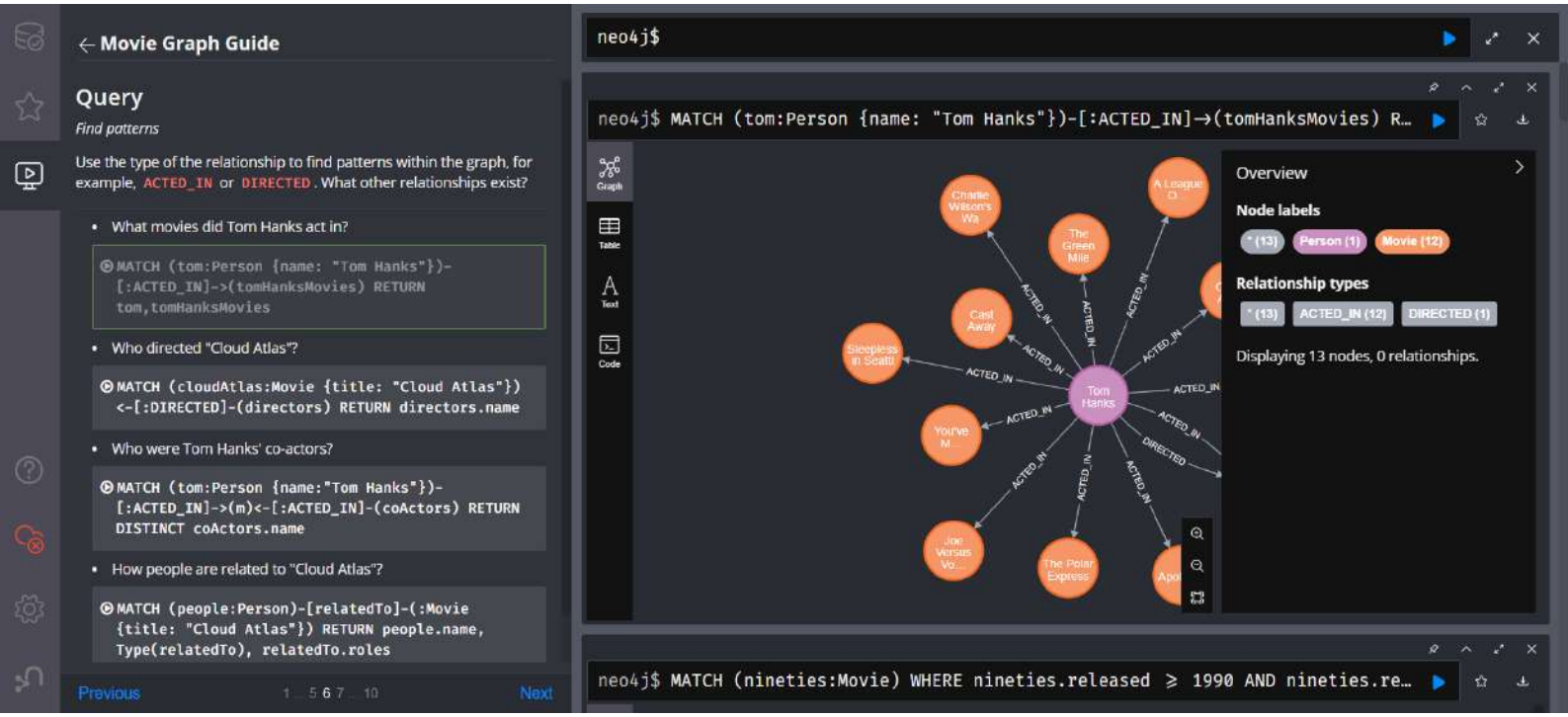

Movie Graph Guide
Index nodes
Create indexes on one or more properties for all nodes that have a given label. Indexes are used to increase search performance.
CREATE INDEX FOR (m:Movie) ON (m.released)
:help
cypher
CREATE INDEX
Previous
1 2 3 4 5 … 10
Next
neo4j$
neo4j$ CREATE INDEX FOR (m:Movie) ON (m.released)
Table
Code
Added 1 index, completed after 3 ms.
Added 1 index, completed after 3 ms.
neo4j$ DROP INDEX index_1d2d0abd
Table
Code
Removed 1 index, completed after 5 ms.

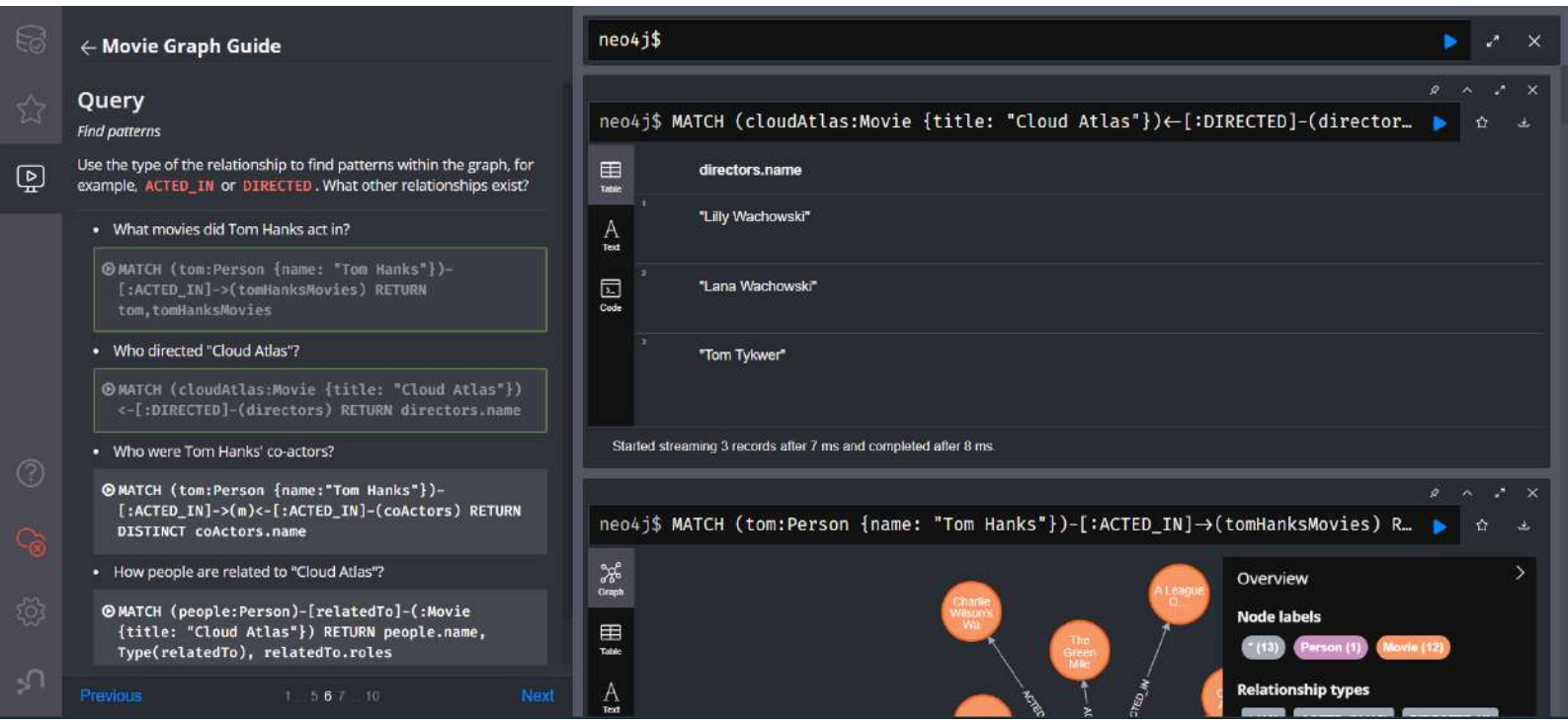

Movie Graph Guide
Find
Find individual nodes
1. Run any of the following query examples.
2. Notice the syntax pattern.
3. Try looking for other movies or actors.
Find the actor named "Tom Hanks":
MATCH (tom:Person {name: "Tom Hanks"}) RETURN tom
Find the movie with title "Cloud Atlas":
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})
RETURN cloudAtlas
Find 10 people and return their names:
MATCH (people:Person) RETURN people.name LIMIT 10
Find movies released in the 1990s and return their titles.
MATCH (nineties:Movie) WHERE nineties.released >=
1990 AND nineties.released < 2000 RETURN
nineties.title
:help MATCH WHERE RETURN
Previous
1 4 5 6 10
Next
neo4j$
neo4j$ MATCH (tom:Person {name: "Tom Hanks"}) RETURN tom
Graph
Table
Text
Code
Tom Hanks
Overview
Node labels
(1)
Person (1)
Displaying 1 nodes, 0 relationships.
neo4j$ CREATE INDEX FOR (m:Movie) ON (m.released)

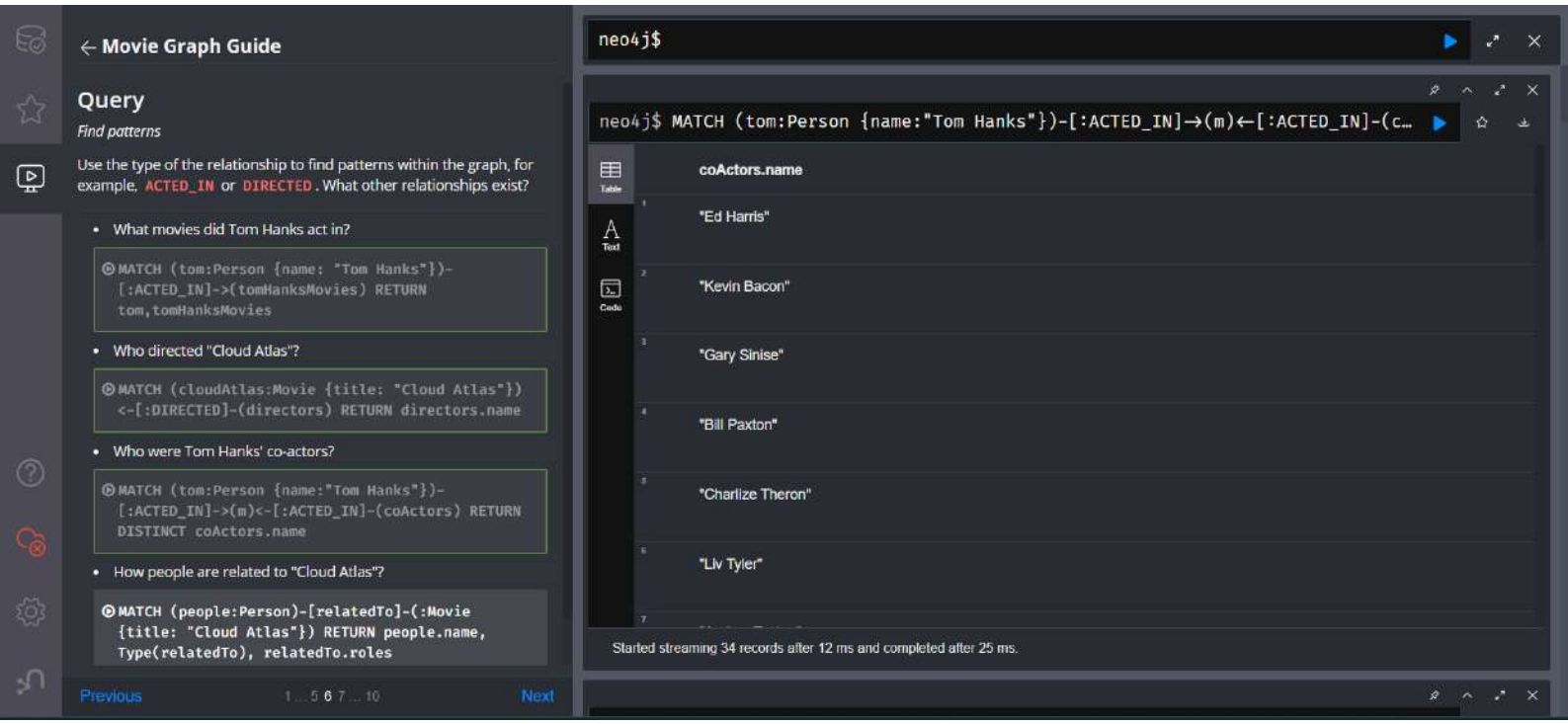

Movie Graph Guide
Find
Find individual nodes
1. Run any of the following query examples.
2. Notice the syntax pattern.
3. Try looking for other movies or actors.
Find the actor named "Tom Hanks":
MATCH (tom:Person {name: "Tom Hanks"}) RETURN tom
Find the movie with title "Cloud Atlas":
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})
RETURN cloudAtlas
Find 10 people and return their names:
MATCH (people:Person) RETURN people.name LIMIT 10
Find movies released in the 1990s and return their titles.
MATCH (nineties:Movie) WHERE nineties.released >=
1990 AND nineties.released < 2000 RETURN
nineties.title
:help
MATCH
WHERE
RETURN
Previous
1 ... 4 5 6 ... 10
Next
neo4j$
neo4j$ MATCH (cloudAtlas:Movie {title: "Cloud Atlas"}) RETURN cloudAtlas
Graph
Table
Text
Code
Cloud Atlas
Overview
Node labels
* (1)
Movie (1)
Displaying 1 nodes, 0 relationships.
neo4j$ MATCH (tom:Person {name: "Tom Hanks"}) RETURN tom

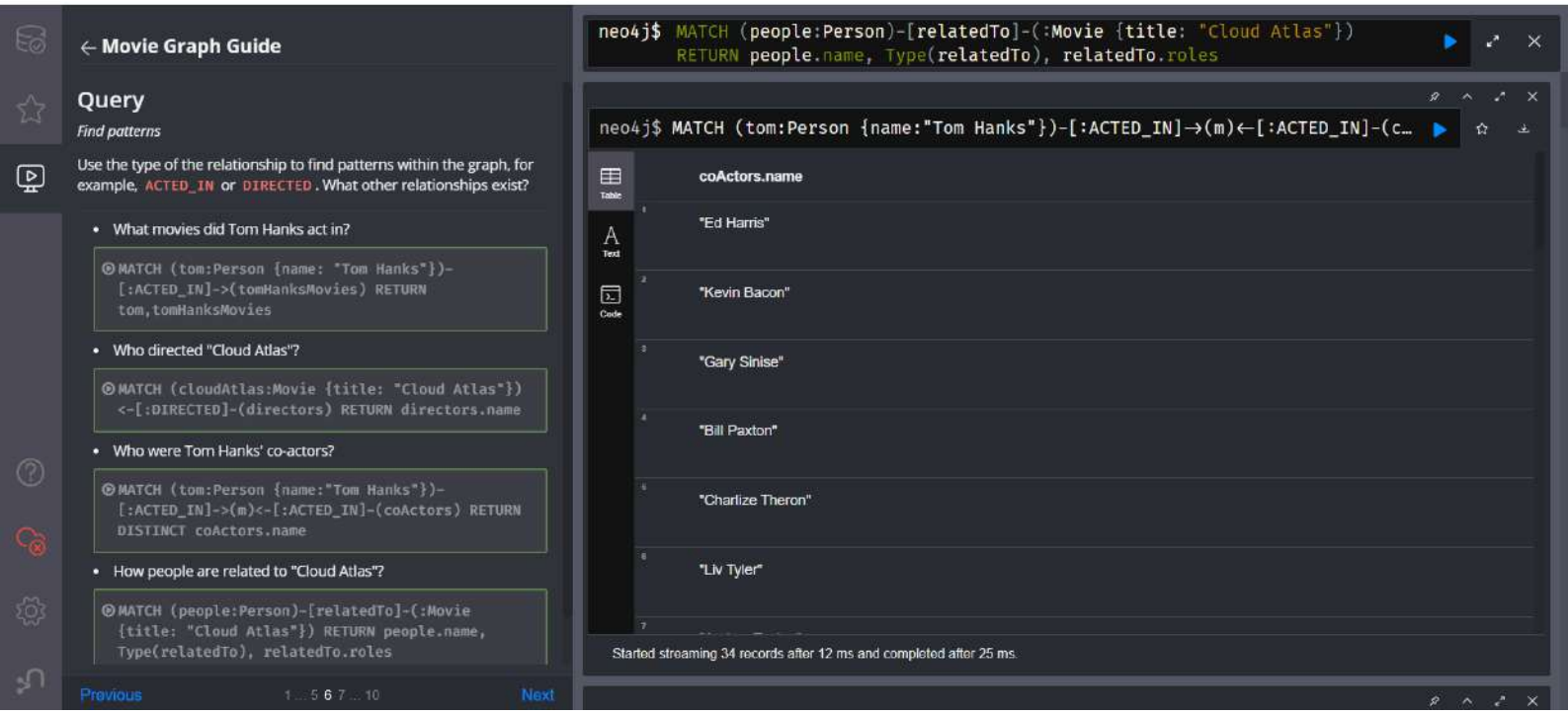

Movie Graph Guide
Find
Find individual nodes
1. Run any of the following query examples.
2. Notice the syntax pattern.
3. Try looking for other movies or actors.
Find the actor named "Tom Hanks":
MATCH (tom:Person {name: "Tom Hanks"}) RETURN tom
Find the movie with title "Cloud Atlas":
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})
RETURN cloudAtlas
Find 10 people and return their names:
MATCH (people:Person) RETURN people.name LIMIT 10
Find movies released in the 1990s and return their titles.
MATCH (nineties:Movie) WHERE nineties.released >=
1990 AND nineties.released < 2000 RETURN
nineties.title
:help
MATCH
WHERE
RETURN
Previous
1 … 4 5 6 … 10
Next
neo4j$
neo4j$ MATCH (people:Person) RETURN people.name LIMIT 10
Table
Text
Code
people.name
"Lilly Wachowski"
"Lana Wachowski"
"Joel Silver"
"Emil Eifrem"
"Charlize Theron"
"Al Pacino"
Started streaming 10 records after 5 ms and completed after 6 ms.

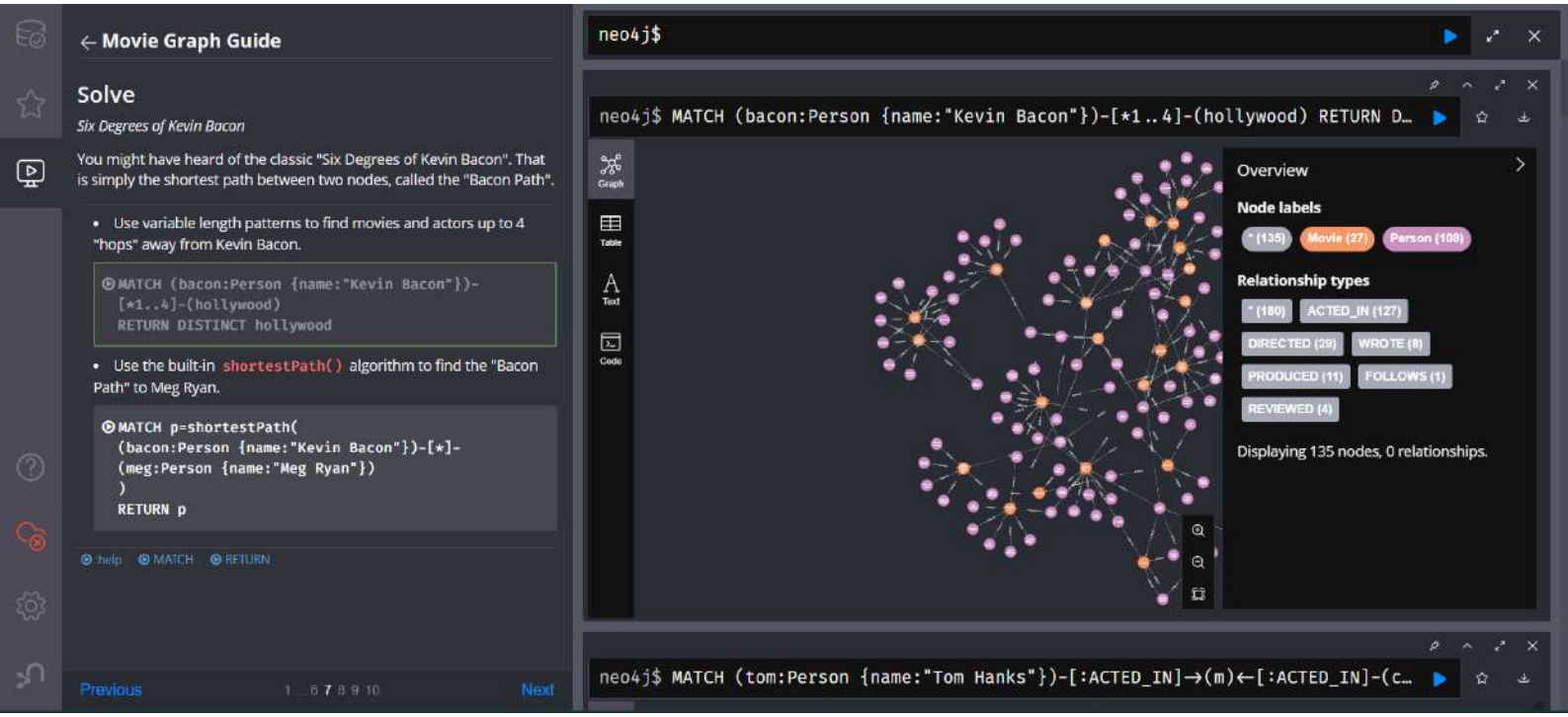

Movie Graph Guide
Find
Find individual nodes
1. Run any of the following query examples.
2. Notice the syntax pattern.
3. Try looking for other movies or actors.
Find the actor named "Tom Hanks":
MATCH (tom:Person {name: "Tom Hanks"}) RETURN tom
Find the movie with title "Cloud Atlas":
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"}) RETURN cloudAtlas
Find 10 people and return their names:
MATCH (people:Person) RETURN people.name LIMIT 10
Find movies released in the 1990s and return their titles.
MATCH (nineties:Movie) WHERE nineties.released >= 1990 AND nineties.released < 2000 RETURN nineties.title
:help MATCH WHERE RETURN
Previous 1 ... 4 5 6 ... 10 Next
neo4j$
neo4j$ MATCH (nineties:Movie) WHERE nineties.released ⩾ 1990 AND nineties.re…
Table
Text
Code
nineties.title
"Joe Versus the Volcano"
"A Few Good Men"
"Unforgiven"
"Hoffa"
"A League of Their Own"
"Sleepless in Seattle"
Started streaming 20 records after 8 ms and completed after 15 ms.

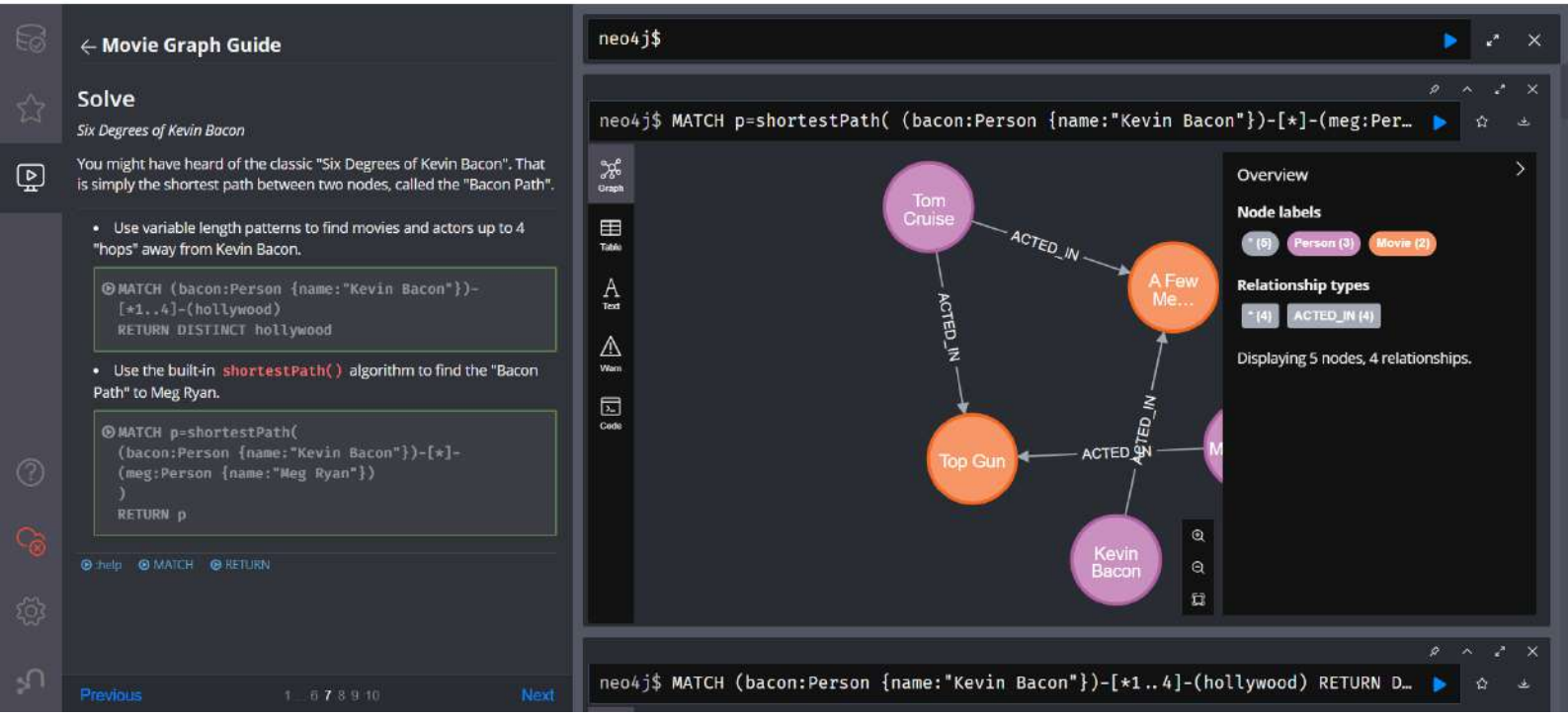

Movie Graph Guide
Query
Find patterns
Use the type of the relationship to find patterns within the graph, for example, ACTED_IN or DIRECTED. What other relationships exist?
What movies did Tom Hanks act in?
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
Who directed "Cloud Atlas"?
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})<-[:DIRECTED]-(directors) RETURN directors.name
Who were Tom Hanks' co-actors?
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors) RETURN DISTINCT coActors.name
How people are related to "Cloud Atlas"?
MATCH (people:Person)-[relatedTo]-(:Movie {title: "Cloud Atlas"}) RETURN people.name, Type(relatedTo), relatedTo.roles
Previous
1 5 6 7 10
Next
neo4j$
neo4j$ MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]→(tomHanksMovies) R...
Graph
Table
Text
Code
Charlie Wilson's Wa...
The Green Mile
A League Of...
Cast Away
Sleepless in Seattl...
You've M...
Joe Versus Vo...
The Polar Express
Tom Hanks
ACTED_IN
DIRECTED
Overview
Node labels
* (13)
Person (1)
Movie (12)
Relationship types
* (13)
ACTED_IN (12)
DIRECTED (1)
Displaying 13 nodes, 0 relationships.
neo4j$ MATCH (nineties:Movie) WHERE nineties.released ⩾ 1990 AND nineties.re...

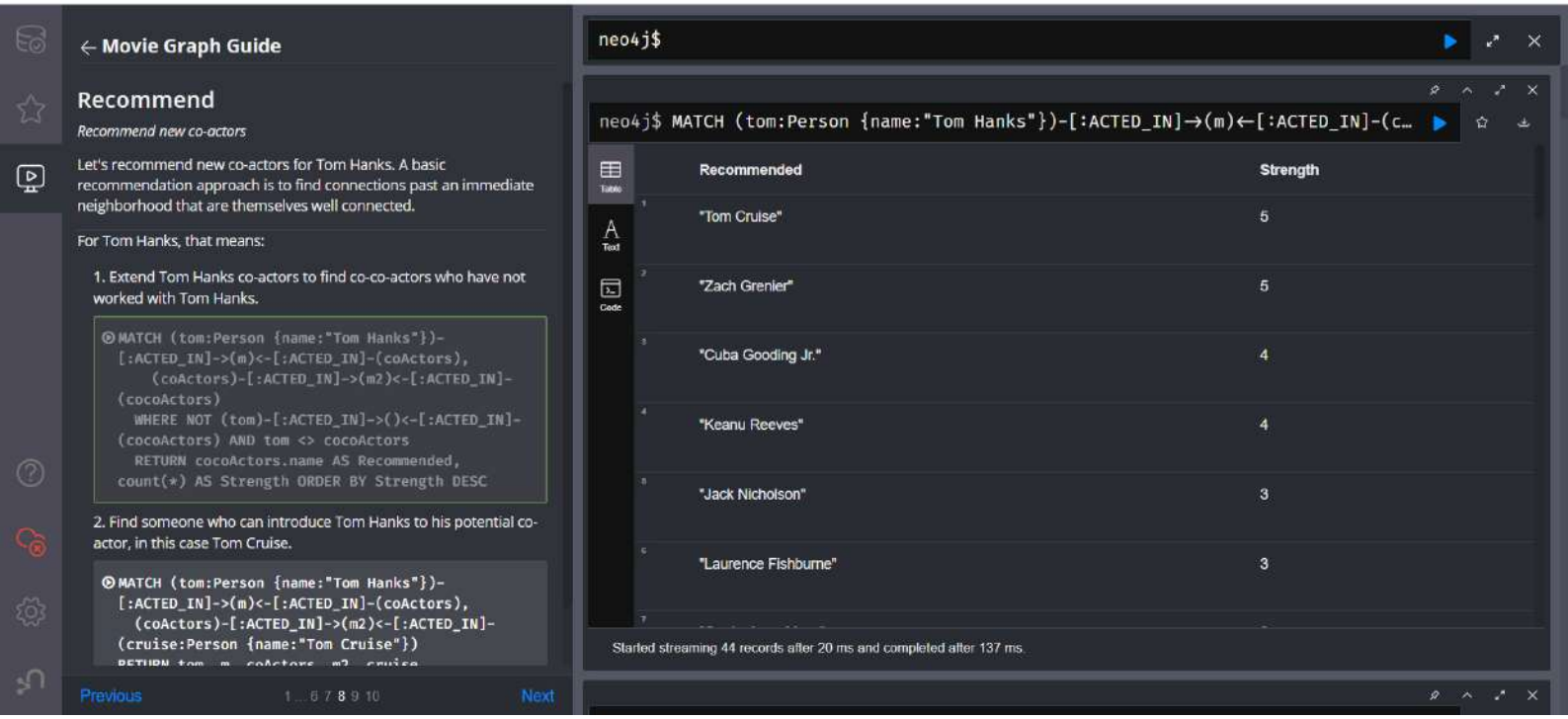

← Movie Graph Guide
Query
Find patterns
Use the type of the relationship to find patterns within the graph, for example, ACTED_IN or DIRECTED. What other relationships exist?
• What movies did Tom Hanks act in?
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
• Who directed "Cloud Atlas"?
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})<-[:DIRECTED]-(directors) RETURN directors.name
• Who were Tom Hanks' co-actors?
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors) RETURN DISTINCT coActors.name
• How people are related to "Cloud Atlas"?
MATCH (people:Person)-[relatedTo]-(:Movie {title: "Cloud Atlas"}) RETURN people.name, Type(relatedTo), relatedTo.roles
Previous
1 5 6 7 10
Next
neo4j$
neo4j$ MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})←[:DIRECTED]-(director…
Table
Text
Code
directors.name
"Lilly Wachowski"
"Lana Wachowski"
"Tom Tykwer"
Started streaming 3 records after 7 ms and completed after 8 ms
neo4j$ MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]→(tomHanksMovies) R…
Graph
Table
Text
Charlie Wilson's Wa
The Green Mile
A League O
Overview
Node labels
(13)
Person (1)
Movie (12)
Relationship types

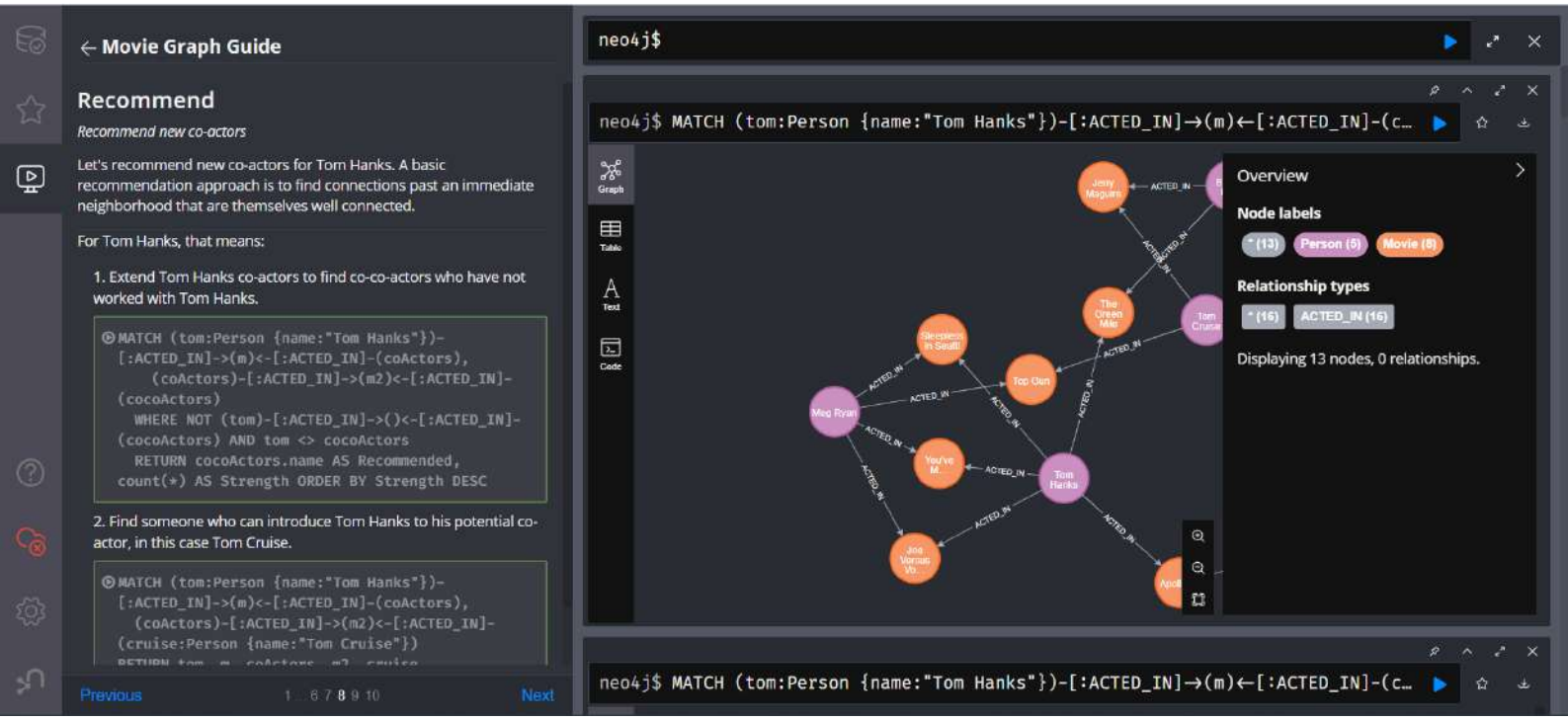

← Movie Graph Guide
Query
Find patterns
Use the type of the relationship to find patterns within the graph, for example, ACTED_IN or DIRECTED. What other relationships exist?
• What movies did Tom Hanks act in?
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
• Who directed "Cloud Atlas"?
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})<-[:DIRECTED]-(directors) RETURN directors.name
• Who were Tom Hanks' co-actors?
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors) RETURN DISTINCT coActors.name
• How people are related to "Cloud Atlas"?
MATCH (people:Person)-[relatedTo]-(:Movie {title: "Cloud Atlas"}) RETURN people.name, Type(relatedTo), relatedTo.roles
Previous
1 ... 5 6 7 ... 10
Next
neo4j$
neo4j$ MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]→(m)←[:ACTED_IN]-(c…
Table
Text
Code
coActors.name
"Ed Harris"
"Kevin Bacon"
"Gary Sinise"
"Bill Paxton"
"Charlize Theron"
"Liv Tyler"
Started streaming 34 records after 12 ms and completed after 25 ms.


← Movie Graph Guide
Query
Find patterns
Use the type of the relationship to find patterns within the graph, for example, ACTED_IN or DIRECTED. What other relationships exist?
• What movies did Tom Hanks act in?
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
• Who directed "Cloud Atlas"?
MATCH (cloudAtlas:Movie {title: "Cloud Atlas"})<-[:DIRECTED]-(directors) RETURN directors.name
• Who were Tom Hanks' co-actors?
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors) RETURN DISTINCT coActors.name
• How people are related to "Cloud Atlas"?
MATCH (people:Person)-[relatedTo]-(:Movie {title: "Cloud Atlas"}) RETURN people.name, Type(relatedTo), relatedTo.roles
Previous
1 … 5 6 7 … 10
Next
neo4j$ MATCH (people:Person)-[relatedTo]-(:Movie {title: "Cloud Atlas"})
RETURN people.name, Type(relatedTo), relatedTo.roles
neo4j$ MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]→(m)←[:ACTED_IN]-(c…
Table
Text
Code
coActors.name
"Ed Harris"
"Kevin Bacon"
"Gary Sinise"
"Bill Paxton"
"Charlize Theron"
"Liv Tyler"
Started streaming 34 records after 12 ms and completed after 25 ms.


Movie Graph Guide
Solve
Six Degrees of Kevin Bacon
You might have heard of the classic "Six Degrees of Kevin Bacon". That is simply the shortest path between two nodes, called the "Bacon Path".
Use variable length patterns to find movies and actors up to 4 "hops" away from Kevin Bacon.
MATCH (bacon:Person {name:"Kevin Bacon"})-[*1..4]-(hollywood)
RETURN DISTINCT hollywood
Use the built-in shortestPath() algorithm to find the "Bacon Path" to Meg Ryan.
MATCH p=shortestPath(
(bacon:Person {name:"Kevin Bacon"})-[*]-(meg:Person {name:"Meg Ryan"})
)
RETURN p
:help
MATCH
RETURN
Previous
1 6 7 8 9 10
Next
neo4j$
neo4j$ MATCH (bacon:Person {name:"Kevin Bacon"})-[*1..4]-(hollywood) RETURN D…
Graph
Table
Text
Code
Overview
Node labels
* (135)
Movie (27)
Person (108)
Relationship types
* (180)
ACTED_IN (127)
DIRECTED (29)
WROTE (8)
PRODUCED (11)
FOLLOWS (1)
REVIEWED (4)
Displaying 135 nodes, 0 relationships.
neo4j$ MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]→(m)←[:ACTED_IN]-(c…


← Movie Graph Guide
Solve
Six Degrees of Kevin Bacon
You might have heard of the classic "Six Degrees of Kevin Bacon". That is simply the shortest path between two nodes, called the "Bacon Path".
• Use variable length patterns to find movies and actors up to 4 "hops" away from Kevin Bacon.
MATCH (bacon:Person {name:"Kevin Bacon"})-[*1..4]-(hollywood)
RETURN DISTINCT hollywood
• Use the built-in shortestPath() algorithm to find the "Bacon Path" to Meg Ryan.
MATCH p=shortestPath((bacon:Person {name:"Kevin Bacon"})-[*]-(meg:Person {name:"Meg Ryan"}))
RETURN p
:help MATCH RETURN
Previous
1 … 6 7 8 9 10
Next
neo4j$
neo4j$ MATCH p=shortestPath((bacon:Person {name:"Kevin Bacon"})-[*]-(meg:Per…
Graph
Table
Text
Warn
Code
Tom Cruise
ACTED_IN
A Few Me…
ACTED_IN
Top Gun
ACTED_IN
ACTED_IN
Kevin Bacon
Overview
Node labels
* (5)
Person (3)
Movie (2)
Relationship types
* (4)
ACTED_IN (4)
Displaying 5 nodes, 4 relationships.
neo4j$ MATCH (bacon:Person {name:"Kevin Bacon"})-[*1..4]-(hollywood) RETURN D…


← Movie Graph Guide
Recommend
Recommend new co-actors
Let's recommend new co-actors for Tom Hanks. A basic recommendation approach is to find connections past an immediate neighborhood that are themselves well connected.
For Tom Hanks, that means:
1. Extend Tom Hanks co-actors to find co-co-actors who have not worked with Tom Hanks.
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors), (coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cocoActors) WHERE NOT (tom)-[:ACTED_IN]->()<-[:ACTED_IN]-(cocoActors) AND tom <> cocoActors RETURN cocoActors.name AS Recommended, count(*) AS Strength ORDER BY Strength DESC
2. Find someone who can introduce Tom Hanks to his potential co-actor, in this case Tom Cruise.
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors), (coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cruise:Person {name:"Tom Cruise"})
Previous
1 … 6 7 8 9 10
Next
neo4j$
neo4j$ MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]→(m)←[:ACTED_IN]-(c…
Table
Text
Code
Recommended
Strength
"Tom Cruise"
5
"Zach Grenier"
5
"Cuba Gooding Jr."
4
"Keanu Reeves"
4
"Jack Nicholson"
3
"Laurence Fishburne"
3
Started streaming 44 records after 20 ms and completed after 137 ms.


← Movie Graph Guide
Recommend
Recommend new co-actors
Let's recommend new co-actors for Tom Hanks. A basic recommendation approach is to find connections past an immediate neighborhood that are themselves well connected.
For Tom Hanks, that means:
1. Extend Tom Hanks co-actors to find co-co-actors who have not worked with Tom Hanks.
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors), (coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cocoActors) WHERE NOT (tom)-[:ACTED_IN]->()<-[:ACTED_IN]-(cocoActors) AND tom <> cocoActors RETURN cocoActors.name AS Recommended, count(*) AS Strength ORDER BY Strength DESC
2. Find someone who can introduce Tom Hanks to his potential co-actor, in this case Tom Cruise.
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors), (coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cruise:Person {name:"Tom Cruise"})
Previous
1 6 7 8 9 10
Next
neo4j$
neo4j$ MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]→(m)←[:ACTED_IN]-(c…
Graph
Table
Text
Code
Overview
Node labels
* (13)
Person (5)
Movie (5)
Relationship types
* (16)
ACTED_IN (16)
Displaying 13 nodes, 0 relationships.
Jerry Maguire
The Green Mile
Sleepless in Seattl
Top Gun
Meg Ryan
You've M…
Tom Hanks
Joe Versus Vo…
ACTED_IN
neo4j$ MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]→(m)←[:ACTED_IN]-(c…