



CERTIFICATE

This is to certify that

Mr./Miss/Ms. _____ student of
B.Sc.(CS) /B.Sc(Animation)/B.Com. /BCA / MCA (Comm.) /
M.Sc(CS) /MCA(Sci.) Semester _____ having Seat No.
_____ at Suryadatta College of Management Information
Research & Technology (SCMIRT), Pune, has successfully
completed the assigned practical in
_____ prescribed by the University of
Pune During the academic year _____.

Internal Examiner

External Examiner

Principa

1 Place: Pune

Date:

COMPUTER APPLICATION

Submitted on_____InCharge_____	
Name _____ _____	
Course_____	Semester_____
Batch_____	
Expt.No._____Title_____	



CERTIFICATE

This is to certify that

Mr./Miss/Ms._____

Student of B.Sc. (CS) / B.Sc(Animation)/ BCA / MCA (Comm.) /

MSC (CS) / MCA (Sci.) Semester_____ having Seat No.

_____at

Suryadatta College of Management Information Research &
Technology

(SCMIRT), Pune has successfully completed the assigned Project titled _____ as per the norms of University of Pune During the academic year_____.

Internal Examiner

External Examiner

Project Guide

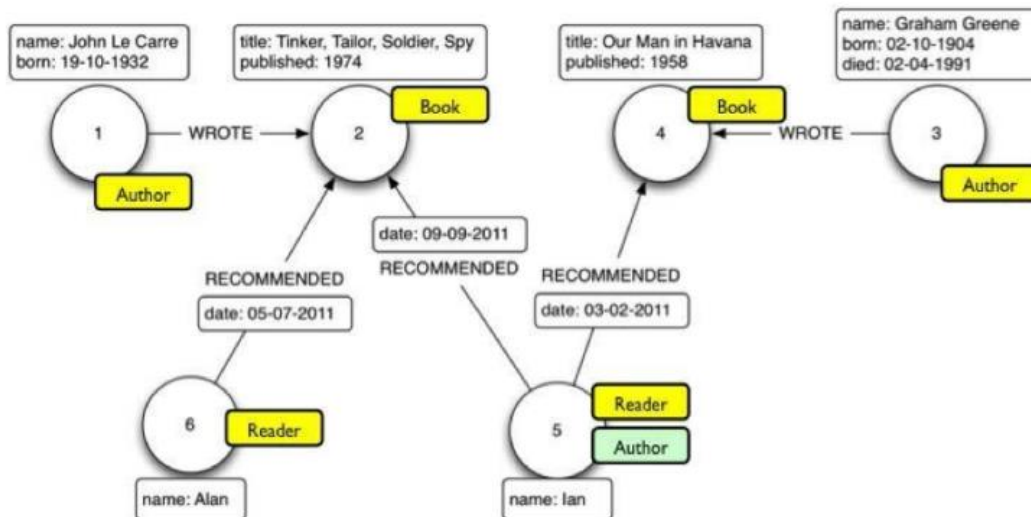
Principal

Place: Pune

Date:

SUBJECT: Neo4j

Practical Assignment



```
CREATE (library:author{name:"jon le  
carre",born:"19/10/1932"})
```

```
CREATE  
(l:book{title:["tinker","tailor","soldier","spy"],publish:"1974"})
```

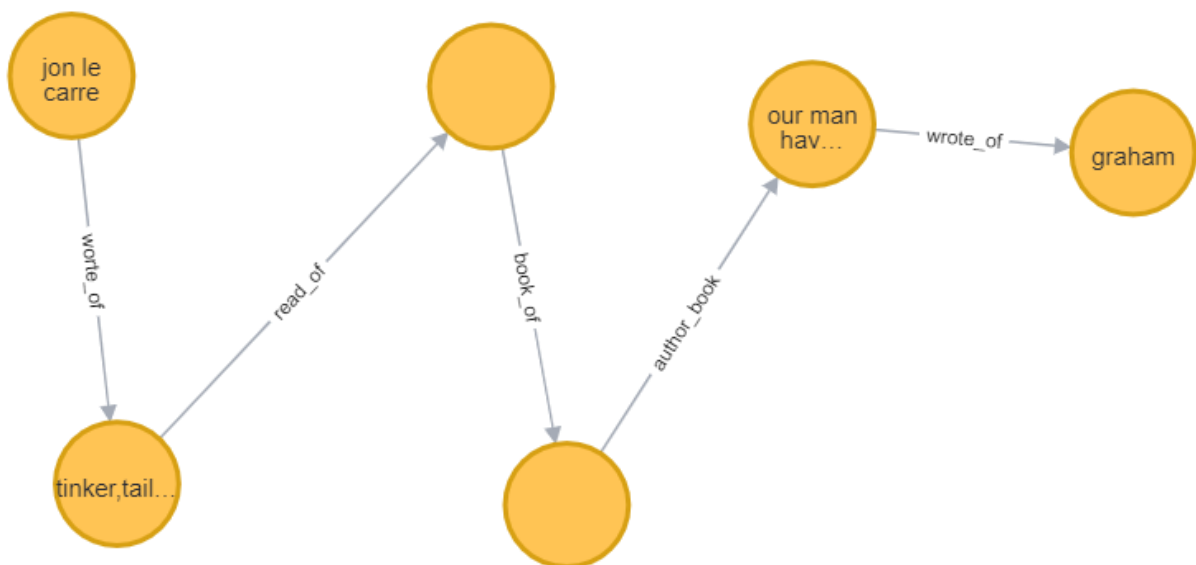
```
CREATE (d:book{title:"our man in havana",publish:"1958"})
```

CREATE

(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}
)

CREATE (library)-[r:wrote_of]->(l)-[t:read_of]->(b)-
[f:book_of]->(c)-[j:author_book]->(d)-[k:wrote_of]->(g)

RETURN library,l,b,c,d,g



The image displays two screenshots of the Neo4j Browser interface, showing the process of creating a database and viewing node/relationship details.

Top Screenshot:

- Database Information:** The left sidebar shows the database name "neo4j" and various node labels (e.g., "Book", "Country", "Person", "Reader", "author") and relationship types (e.g., "ACTED_IN", "FOLLOWS", "PRODUCED", "REVIEWED", "SAYS").
- Query Editor:** The main area shows a Cypher query:


```
CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE
(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-
[r:wornte_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(
g) RETURN library,l,b,c,d,g
```
- Results:** The bottom section shows a table with 6 columns (library, l, b, c, d, g) and 6 rows of node details, including identity, labels, and properties.

Bottom Screenshot:

- Database Information:** Similar to the top screenshot, showing the database name "neo4j" and various node labels and relationship types.
- Query Editor:** The main area shows the same Cypher query as the top screenshot.
- Results:** The bottom section shows a table with 6 columns (library, l, b, c, d, g) and 6 rows of node details, including identity, labels, and properties.

Consider a Song database, with labels as Artists, Song, Recording_company, Recording_studio, song author etc. Relationships can be as follows Artist → [Performs] → Song → [Written by] → Song_author. Song → [Recorded in] → Recording Studio → [managed by] → recording Company Recording Company → [Finances] → Song You may add more

labels and relationship and their properties, as per assumptions.

ANS:

```
CREATE(song_actorsong:singer{name:"song_actorsong",yod:"4/6/7"})
```

```
CREATE(recording_studio:name{name:"recording_studio"})
```

```
CREATE(recordingcompanyCREATE(artis:actor{name:"artis",yod:"2/4/1985"})
```

```
CREATE (song:mus{musicname:"song",date:"6/7/11"})
```

```
ny:SAMPLE{name:"recordingcompany"})
```

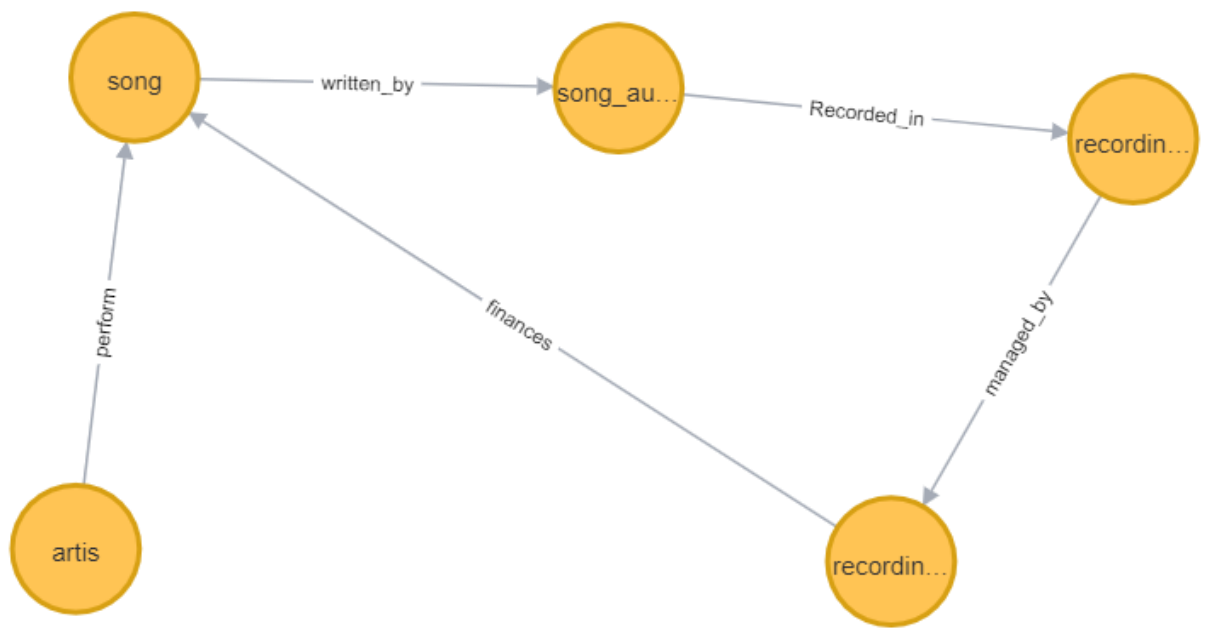
```
CREATE(song:song{name:"song"})
```

```
CREATE (artis)-[:perform]->(song)-[:written_by]-  
>(song_actorsong)-[:Recorded_in]->(recording_studio)-  
[:managed_by]->(recordingcompany)-[:finances]->(song)
```

```
RETURN
```

```
artis,song,song_actorsong,recording_studio,recordingcompa  
ny
```

OUTPUT :



neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(590) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahu son subject

yadav yadav11

Relationship types

(788) ACTED_IN BATSMAN_OF

DIRECTED FOLLOWS

PRODUCED REVIEWED SAYS

WROTE author_book

batmans_of book_of college_of

read_of student_college

wrote_of wrote_of

neo4j\$

```
neo4j$ CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE
(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-
[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(
g) RETURN library,l,b,c,d,g
```

MAX COLUMN WIDTH: 100

neo4j\$ CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham..."

Node properties

book

<id> 578

72°F Haze

19:26 07-01-2023

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(590) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahu son subject

yadav yadav11

Relationship types

(788) ACTED_IN BATSMAN_OF

DIRECTED FOLLOWS

PRODUCED REVIEWED SAYS

WROTE author_book

batmans_of book_of college_of

read_of student_college

wrote_of wrote_of

neo4j\$

```
neo4j$ CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE
(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-
[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(
g) RETURN library,l,b,c,d,g
```

MAX COLUMN WIDTH: 100

neo4j\$ CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham..."

Node properties

book

<id> 578

72°F Haze

19:26 07-01-2023

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1590) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahul son subject

yadav yadav11

Relationship types

(788) ACTED_IN BATSMAN_OF

DIRECTED FOLLOWED

PRODUCED REVIEWED SAYS

WROTE author_book

batmans_of book_of college_of

read_of student_college

write_of wrote_of

neo4j\$

```
CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE
(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-
[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]->
(g) RETURN library,l,b,c,d,g
```

Server address localhost:7687

Query

```
CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE (g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE
(library)-[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(g) RETURN library,l,b,c,d,g
```

Summary

Response

Added 4 labels, created 6 nodes, set 9 properties, created 5 relationships, started streaming 1 records after 1 ms and completed after 3 ms.

neo4j\$ CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham"...

Node properties

book

<id> 578

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1548) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahul son subject

yadav yadav11

Relationship types

(1,304) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWED PRODUCED

REVIEWED SAYS WROTE

author_book batmans_of

book_of college_of read_of

student_college wrote_of

neo4j\$

```
MATCH (n:author) RETURN n LIMIT 25
```

n

```
{
  "identity": 565,
  "labels": [
    "author"
  ],
  "properties": {
    "born": "19/10/1932",
    "name": "jon le carre"
  },
  "elementId": "565"
}
```

Started streaming 7 records in less than 1 ms and completed after 1 ms.

neo4j\$ // load movies // This is an initialization script for the movie graph. // Run it only o...

Added 179 labels, created 179 nodes, set 500 properties, created 250 relationships, completed after 34 ms.

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1940) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahul son subject

yadav yadav11

Relationship types

(1,304) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED SAYS WROTE

author_book batmans_of

book_of college_of read_of

student_college words_of

neo4j\$

neo4j\$ MATCH (n:author) RETURN n LIMIT 25

Server address localhost:7687

Query MATCH (n:author) RETURN n LIMIT 25

Summary [{"query": {"text": "MATCH (n:author) RETURN n LIMIT 25", ...

Response [{"keys": [...

Started streaming 7 records in less than 1 ms and completed after 1 ms.

neo4j\$ // load movies // This is an initialization script for the movie graph. // Run it only o...

Added 179 labels, created 179 nodes, set 580 properties, created 258 relationships, completed after 34 ms.

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(980) Book Country

Database MSCSFY Message

Movie Person Reader

SAMPLE actor author book

child college cricket

grandgrandparent grandparent

mus name parent person

place player player rahul

singer son subject yadav

yadav11

Relationship types

(1,331) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED Recorded_in SAYS

WROTE author_book

neo4j\$

neo4j\$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE (song:mus{musicname:"song",date:"6/7/11"}) CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"}) CREATE(recording_studio:name{name:"recording_studio"}) CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]→ (song)-[:written_by]→(song_autorsong)-[:Recorded_in]→(recording_studio)-[:managed_by]→(recordingcompany)-[:finances]→(song) RETURN artis,song,song_autorsong,recording_studio,recordingcompany

Graph

Table

Text

Code

Node properties

actor

<id> 975

name artis

yod 2/4/1985

song

song au

recordin

Recorded_in

managed_by

finances

perform

artis

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1990) Book Country

Database MSCSFY Message

Movie Person Reader

SAMPLE actor author book

child college cricket

grandgrandparent grandparent

mus name parent person

place player player rahul

singer son subject yadav

yadav11

Relationship types

(1,331) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED Recorded_in SAYS

WROTE author_book

```
neo4j$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE (song:mus{musicname:"song",date:"6/7/11"}) CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"}) CREATE(recording_studio:name{name:"recording_studio"}) CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]→(song)-[:written_by]→(song_autorsong)-[:Recorded_in]→(recording_studio)-[:managed_by]→(recordingcompany)-[:finances]→(song) RETURN artis,song,song_autorsong,recording_studio,recordingcompany
```

artis	song	song_autorsong	recording_studio	recordingcompany
{ "identity": 975, "labels": ["actor"], "properties": { "name": "artis", "yod": "2/4/1985" }, "elementId": 975 }	{ "identity": 976, "labels": ["mus"], "properties": { "date": "6/7/11", "musicname": "song" }, "elementId": 976 }	{ "identity": 977, "labels": ["singer"], "properties": { "name": "song_autorsong", "yod": "4/6/7" }, "elementId": 977 }	{ "identity": 978, "labels": ["name"], "properties": { "name": "recording_studio", "elementId": "978" }, "elementId": 978 }	{ "identity": 979, "labels": ["SAMPLE"], "properties": { "name": "recordingcompany", "elementId": "979" }, "elementId": 979 }

72°F Haze

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(590) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahul son subject

yadav yadav11

Relationship types

(788) ACTED_IN BATSMAN_OF

DIRECTED FOLLOWS

PRODUCED REVIEWED SAYS

WROTE author_book

batmans_of book_of college_of

read_of student_college

wrote_of wrote_of

```
neo4j$ CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-[r:wrote_of]→(l)-[t:read_of]→(b)-[f:book_of]→(c)-[j:author_book]→(d)-[k:wrote_of]→(g) RETURN library,l,b,c,d,g
```

"library"	"l"	"b"	"c"	"d"
{ "born": "19/10/1932", "name": "jon le carre" }	{ "publish": "1974", "title": ["tinker", "tailor", "soldier", "spy"] }	{ }	{ }	{ "publish": "1958", "title": "our man in havana" }

MAX COLUMN WIDTH: 100%

```
neo4j$ CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"})
```

Node properties

book

<id> 578

19:26 07-01-2023

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(590) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahu son subject

yadav yadav11

Relationship types

(788) ACTED_IN BATSMAN_OF

DIRECTED FOLLOWS

PRODUCED REVIEWED SAYS

WROTE author_book

batmans_of book_of college_of

read_of student_college

wrote_of wrote_of

neo4j\$

```
neo4j$ CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE
(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-
[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(
g) RETURN library,l,b,c,d,g
```

	"b"	"c"	"d"	"g"
sier","spy"]}] {} {} {} {"publish":"1958","title":"our man in havana"} {"born":"2/10/1904","name":"graham","died":"2/4/1991"}				

MAX COLUMN WIDTH: 100%

neo4j\$ CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham"...

Node properties

book

<id> 578

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(590) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahu son subject

yadav yadav11

Relationship types

(788) ACTED_IN BATSMAN_OF

DIRECTED FOLLOWS

PRODUCED REVIEWED SAYS

WROTE author_book

batmans_of book_of college_of

read_of student_college

wrote_of wrote_of

neo4j\$

```
neo4j$ CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:
["tinker","tailor","soldier","spy"],publish:"1974"}) CREATE (d:book{title:"our man in
havana",publish:"1958"}) CREATE
(g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE (library)-
[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(
g) RETURN library,l,b,c,d,g
```

Server address localhost:7687

Query

```
CREATE (library:author{name:"jon le carre",born:"19/10/1932"}) CREATE (l:book{title:["tinker","tailor","soldier","spy"],publish:"1974"})
CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham",born:"2/10/1904",died:"2/4/1991"}) CREATE
(library)-[r:wrote_of]-(l)-[t:read_of]-(b)-[f:book_of]-(c)-[j:author_book]-(d)-[k:wrote_of]-(g) RETURN library,l,b,c,d,g
```

Summary

Response

```
[{"keys": [...
```

Added 4 labels, created 6 nodes, set 9 properties, created 5 relationships, started streaming 1 records after 1 ms and completed after 3 ms.

neo4j\$ CREATE (d:book{title:"our man in havana",publish:"1958"}) CREATE (g:author{name:"graham"...

Node properties

book

<id> 578

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1948) Book Country Database MSCSFY Message Movie Person Reader author book child college cricket grandgrandparent grandparent parent person place player player rahu son subject yadav yadav11

Relationship types

(1,304) ACTED_IN BATSMAN_OF DIRECTED FOLLOWS PRODUCED REVIEWED SAYS WROTE author_book batmans_of book_of college_of read_of student_college wrote_of

```
neo4j$ MATCH (n:author) RETURN n LIMIT 25
```

Started streaming 7 records in less than 1 ms and completed after 1 ms.

```
neo4j$ // load movies // This is an initialization script for the movie graph. // Run it only o...
```

Added 179 labels, created 179 nodes, set 580 properties, created 258 relationships, completed after 34 ms.

72°F Haze

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1948) Book Country Database MSCSFY Message Movie Person Reader author book child college cricket grandgrandparent grandparent parent person place player player rahu son subject yadav yadav11

Relationship types

(1,304) ACTED_IN BATSMAN_OF DIRECTED FOLLOWS PRODUCED REVIEWED SAYS WROTE author_book batmans_of book_of college_of read_of student_college wrote_of

```
neo4j$ MATCH (n:author) RETURN n LIMIT 25
```

"n"
{"born": "19/10/1932", "name": "jon le carre"}
{"born": "19/10/1932", "name": "jon le carre"}
{"date": "3/2/11", "name": "lan"}
{"born": "2/10/1904", "name": "graham", "died": "2/4/1991"}
{"born": "2/10/1904", "name": "graham", "died": "2/4/1991"}
{"born": "19/10/1932", "name": "jon le carre"}
{"born": "2/10/1904", "name": "graham", "died": "2/4/1991"}

MAX COLUMN WIDTH: 100%

```
neo4j$ // load movies // This is an initialization script for the movie graph. // Run it only o...
```

Added 179 labels, created 179 nodes, set 580 properties, created 258 relationships, completed after 34 ms.

72°F Haze

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1940) Book Country

Database MSCSFY Message

Movie Person Reader author

book child college cricket

grandgrandparent grandparent

parent person place player

player rahul son subject

yadav yadav11

Relationship types

(1,304) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED SAYS WROTE

author_book batmans_of

book_of college_of read_of

student_college words_of

neo4j\$

neo4j\$ MATCH (n:author) RETURN n LIMIT 25

Server address localhost:7687

Query MATCH (n:author) RETURN n LIMIT 25

Summary [{"query": {"text": "MATCH (n:author) RETURN n LIMIT 25", ...

Response [{"keys": [...

Started streaming 7 records in less than 1 ms and completed after 1 ms.

neo4j\$ // load movies // This is an initialization script for the movie graph. // Run it only o...

Added 179 labels, created 179 nodes, set 580 properties, created 258 relationships, completed after 34 ms.

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(980) Book Country

Database MSCSFY Message

Movie Person Reader

SAMPLE actor author book

child college cricket

grandgrandparent grandparent

mus name parent person

place player player rahul

singer son subject yadav

yadav11

Relationship types

(1,331) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED Recorded_in SAYS

WROTE author_book

neo4j\$

neo4j\$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE (song:mus{musicname:"song",date:"6/7/11"}) CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"}) CREATE(recording_studio:name{name:"recording_studio"}) CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]→ (song)-[:written_by]→(song_autorsong)-[:Recorded_in]→(recording_studio)-[:managed_by]→(recordingcompany)-[:finances]→(song) RETURN artis,song,song_autorsong,recording_studio,recordingcompany

Graph

Table

Text

Code

Node properties

actor

<id> 975

name artis

yod 2/4/1985

song

song au

recordin

Recorded_in

managed_by

finances

perform

artis

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1980) Book Country Database MSCSFY Message

Movie Person Reader

SAMPLE actor author book

child college cricket

grandgrandparent grandparent

mus name parent person

place player player rahul

singer son subject yadav

yadav11

Relationship types

(1,331) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED Recorded_in SAYS

WROTE author_book

```
neo4j$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE
(song:mus{musicname:"song",date:"6/7/11"})
CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"})
CREATE(recording_studio:name{name:"recording_studio"})
CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]→
(song)-[:written_by]→(song_autorsong)-[:Recorded_in]→(recording_studio)-
[:managed_by]→(recordingcompany)-[:finances]→(song) RETURN
artis,song,song_autorsong,recording_studio,recordingcompany
```

artis	song	song_autorsong	recording_studio	recordingcompany
{ "identity": 975, "labels": ["actor"], "properties": { "name": "artis", "yod": "2/4/1985" }, "elementId": 975 }	{ "identity": 976, "labels": ["mus"], "properties": { "date": "6/7/11", "musicname": "song" }, "elementId": 976 }	{ "identity": 977, "labels": ["singer"], "properties": { "name": "song_autorsong", "yod": "4/6/7" }, "elementId": 977 }	{ "identity": 978, "labels": ["name"], "properties": { "name": "recording_studio" }, "elementId": 978 }	{ "identity": 979, "labels": ["SAMPLE"], "properties": { "name": "recordingcompany" }, "elementId": 979 }

72°F Haze

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1980) Book Country Database MSCSFY Message

Movie Person Reader

SAMPLE actor author book

child college cricket

grandgrandparent grandparent

mus name parent person

place player player rahul

singer son subject yadav

yadav11

Relationship types

(1,331) ACTED_IN

BATSMAN_OF DIRECTED

FOLLOWS PRODUCED

REVIEWED Recorded_in SAYS

WROTE author_book

```
neo4j$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE
(song:mus{musicname:"song",date:"6/7/11"})
CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"})
CREATE(recording_studio:name{name:"recording_studio"})
CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]→
(song)-[:written_by]→(song_autorsong)-[:Recorded_in]→(recording_studio)-
[:managed_by]→(recordingcompany)-[:finances]→(song) RETURN
artis,song,song_autorsong,recording_studio,recordingcompany
```

artis	song	song_autorsong	recording_studio
{ "name": "artis", "yod": "2/4/1985" }	{ "date": "6/7/11", "musicname": "song" }	{ "name": "song_autorsong", "yod": "4/6/7" }	{ "name": "recording_studio" }

MAX COLUMN WIDTH: 100

```
neo4j$ CREATE (artis)-[:perform]→(song)-[:written_by]→(song_autorsong)-[:Recorded_in]→(recor...
```

Node properties

966

72°F Haze

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1980) Book Country
Database MSCSFY Message
Movie Person Reader
SAMPLE actor author book
child college cricket
grandgrandparent grandparent
mus name parent person
place player player rahul
singer son subject yadav
yadav11

Relationship types

(1,331) ACTED_IN
BATSMAN_OF DIRECTED
FOLLOWS PRODUCED
REVIEWED Recorded_in SAYS
WROTE author_book

```
neo4j$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE
(song:mus{musicname:"song",date:"6/7/11"})
CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"})
CREATE(recording_studio:name:"recording_studio")
CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]->
(song)-[:written_by]->(song_autorsong)-[:Recorded_in]->(recording_studio)-
[:managed_by]->(recordingcompany)-[:finances]->(song) RETURN
artis,song,song_autorsong,recording_studio,recordingcompany
```

Server address localhost:7687

Query

```
CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE (song:mus{musicname:"song",date:"6/7/11"})
CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"}) CREATE(recording_studio:name:"recording_studio")
CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]->(song)-[:written_by]->(song_autorsong)-
[:Recorded_in]->(recording_studio)-[:managed_by]->(recordingcompany)-[:finances]->(song) RETURN
artis,song,song_autorsong,recording_studio,recordingcompany
```

Summary

Response

Added 5 labels, created 5 nodes, set 8 properties, created 5 relationships, started streaming 1 records after 1 ms and completed after 3 ms.

Overview

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j

Node labels

(1980) Book Country
Database MSCSFY Message
Movie Person Reader
SAMPLE actor author book
child college cricket
grandgrandparent grandparent
mus name parent person
place player player rahul
singer son subject yadav
yadav11

Relationship types

(1,331) ACTED_IN
BATSMAN_OF DIRECTED
FOLLOWS PRODUCED
REVIEWED Recorded_in SAYS
WROTE author_book

```
neo4j$ CREATE(artis:actor{name:"artis",yod:"2/4/1985"}) CREATE
(song:mus{musicname:"song",date:"6/7/11"})
CREATE(song_autorsong:singer{name:"song_autorsong",yod:"4/6/7"})
CREATE(recording_studio:name:"recording_studio")
CREATE(recordingcompany:SAMPLE{name:"recordingcompany"}) CREATE (artis)-[:perform]->
(song)-[:written_by]->(song_autorsong)-[:Recorded_in]->(recording_studio)-
[:managed_by]->(recordingcompany)-[:finances]->(song) RETURN
artis,song,song_autorsong,recording_studio,recordingcompany
```

artis	song	song_autorsong	recording_studio
{name:"artis",yod:"2/4/1985"}	{date:"6/7/11",musicname:"song"}	{name:"song_autorsong",yod:"4/6/7"}	{name:"recording_studio"}

MAX COLUMN WIDTH: 100

```
neo4j$ CREATE (artis)-[:perform]->(song)-[:written_by]->(song_autorsong)-[:Recorded_in]->(recording_studio)-
[:managed_by]->(recordingcompany)-[:finances]->(song) RETURN
artis,song,song_autorsong,recording_studio,recordingcompany
```

Node properties