Adding and Deleting Nodes and Relationships

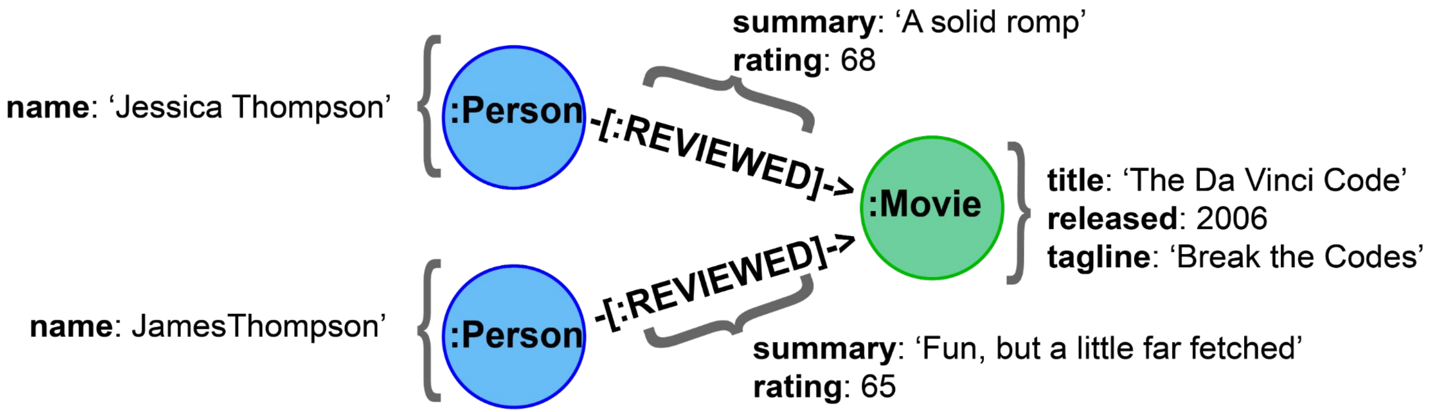
Dr. Sirasit Lochanachit

Today’s Outline

- Create a node
    - Add and remove node labels
    - Add and remove node properties
    - Update node properties
- Create a relationship
- Delete a node and a relationship

Syntax: Create a node

CREATE (optionalVariable optionalLabels {optionalProperties})



Ref: https://neo4j.com/graphacademy/training-querying-40/01-querying40-introduction-to-cypher/

Syntax: Create a node

```
CREATE (:Movie {title: 'Batman Begins'})
```

Syntax: Create multiple nodes

CREATE 2

```
(:Person {name: 'Michael Caine', born: 1933}),

(:Person {name: 'Liam Neeson', born: 1952}),

(:Person {name: 'Katie Holmes', born: 1978}),

(:Person {name: 'Benjamin Melniker', born: 1913})
```

Syntax: Add label(s) to a node

SET x:Label 6

SET x:Label1:Label2

```
match (p:Person)
where p.name starts with 'Robin'
set p:Female
return p.name
```

Example

```
MATCH (m:Movie)

WHERE m.title = 'Batman Begins'
```

Syntax: Remove label(s) from a node

REMOVE x:Label

REMOVE x:Label1, x:Label2

Example

MATCH (m:Action)

Syntax: Add property(s) to a node

SET x.propertyName = value

SET x.propertyName1 = value1, x.propertyName2 = value2

SET x = {propertyName1: value1, propertyName2: value2}

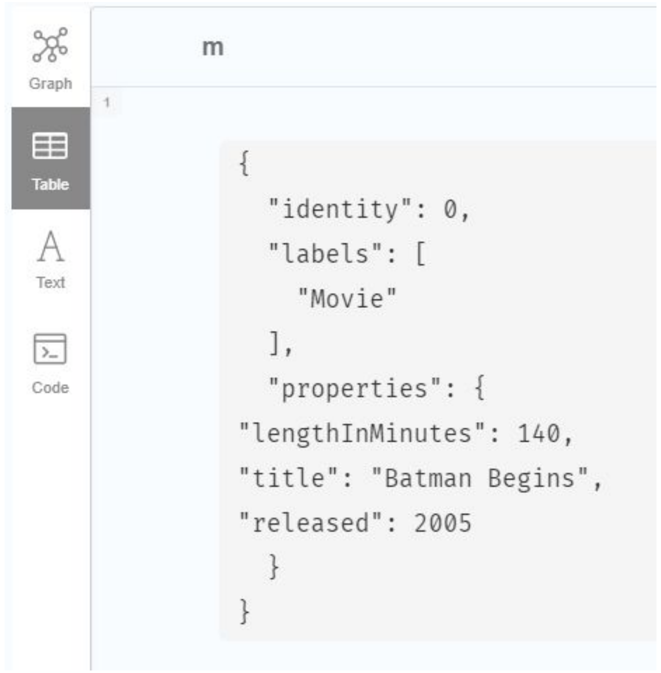
SET x += {propertyName1: value1, propertyName2: value2}

Syntax: Add property(s) to a node

Example

MATCH (m:Movie)

WHERE m.title = 'Batman Begins'



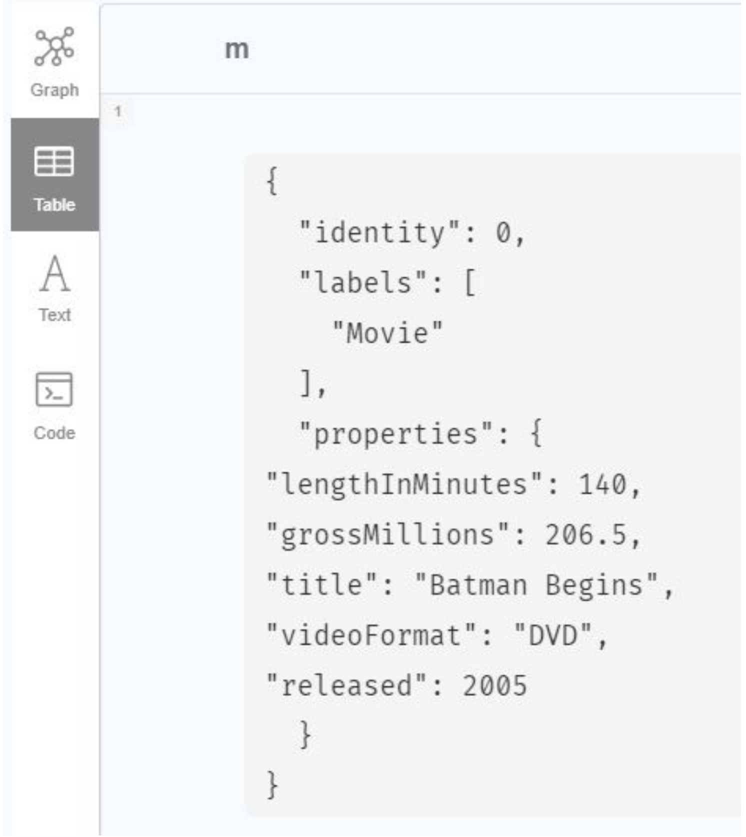
```
{
  "identity": 0,
  "labels": [
    "Movie"
  ],
  "properties": {
"lengthInMinutes": 140,
"title": "Batman Begins",
"released": 2005
  }
}
```

Syntax: Add property(s) to a node

Example

MATCH (m:Movie)

WHERE m.title = 'Batman Begins'



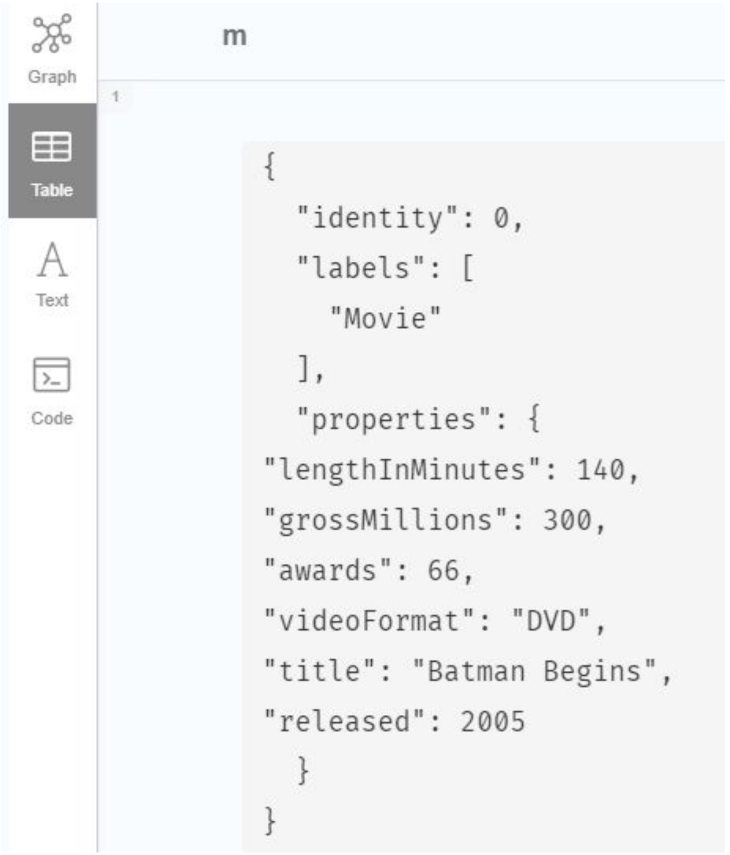
```
{
  "identity": 0,
  "labels": [
    "Movie"
  ],
  "properties": {
"lengthInMinutes": 140,
"grossMillions": 206.5,
"title": "Batman Begins",
"videoFormat": "DVD",
"released": 2005
  }
}
```

Syntax: Add property(s) to a node

Example

MATCH (m:Movie)

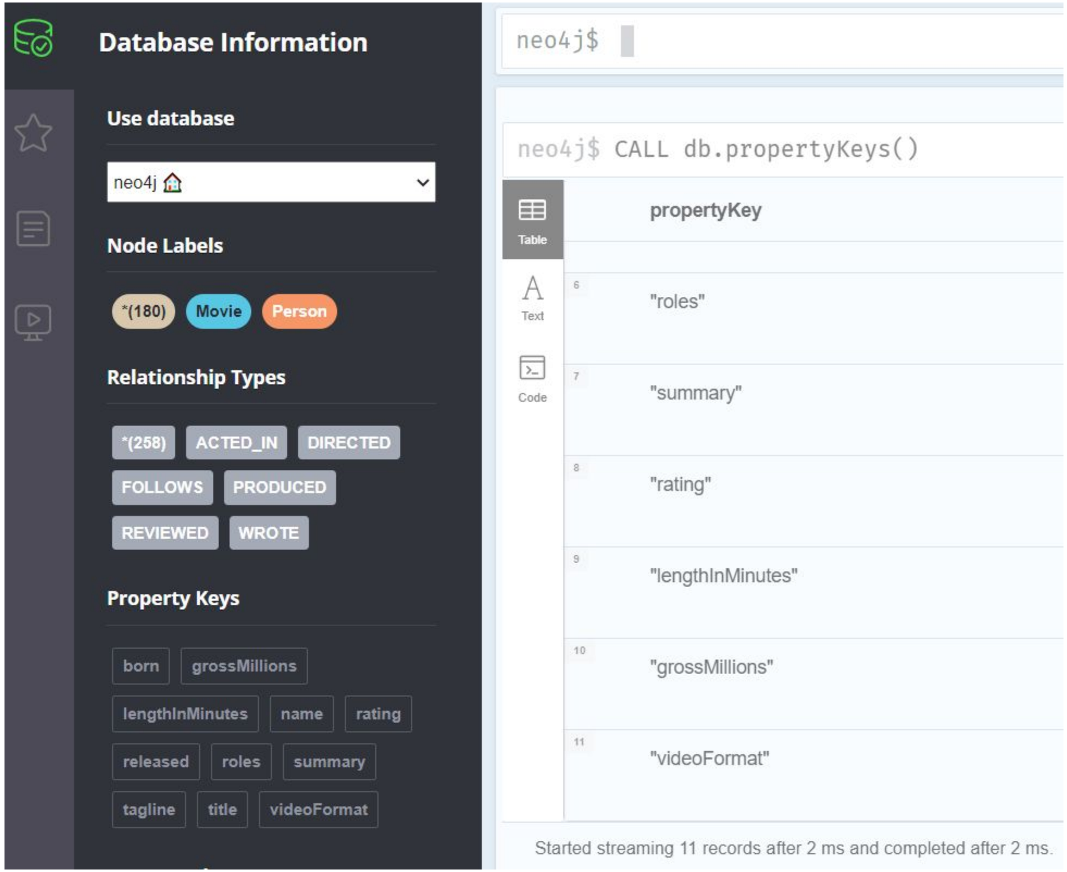
WHERE m.title = 'Batman Begins'



```
{
  "identity": 0,
  "labels": [
    "Movie"
  ],
  "properties": {
"lengthInMinutes": 140,
"grossMillions": 300,
"awards": 66,
"videoFormat": "DVD",
"title": "Batman Begins",
"released": 2005
  }
}
```

Dr. Sirasit Lochanachit

Viewing property Keys



Database Information

Use database

neo4j

Node Labels

*(180) Movie Person

Relationship Types

*(258) ACTED_IN DIRECTED FOLLOWS PRODUCED REVIEWED WROTE

Property Keys

born grossMillions lengthInMinutes name rating released roles summary tagline title videoFormat

```
neo4j$
```

```
neo4j$ CALL db.propertyKeys()
```

	propertyKey
6	"roles"
7	"summary"
8	"rating"
9	"lengthInMinutes"
10	"grossMillions"
11	"videoFormat"

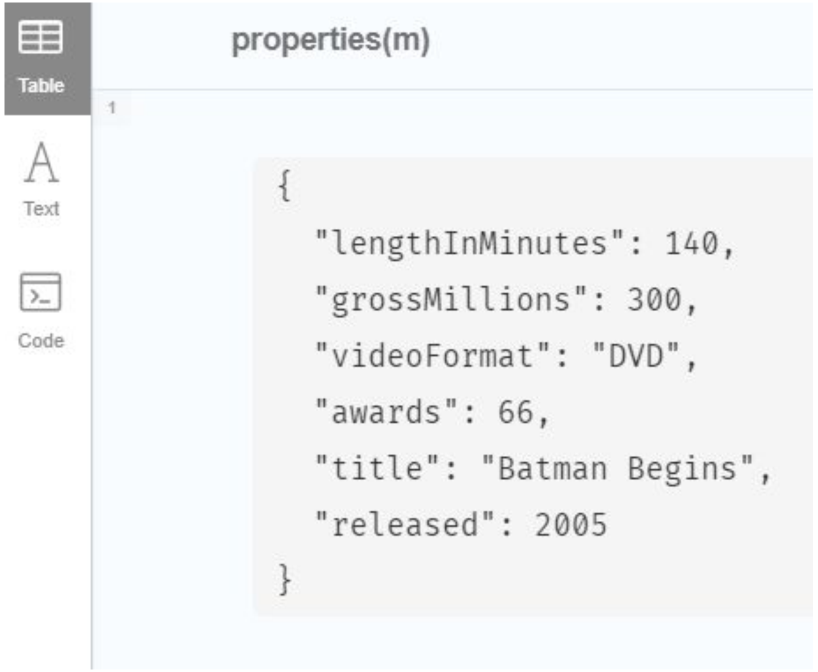
Started streaming 11 records after 2 ms and completed after 2 ms.

Retrieve Properties of a node

Example

MATCH (m:Movie)

WHERE m.title = 'Batman Begins'



```
{
  "lengthInMinutes": 140,
  "grossMillions": 300,
  "videoFormat": "DVD",
  "awards": 66,
  "title": "Batman Begins",
  "released": 2005
}
```

Syntax: Remove property(s) from a node

```
REMOVE x.propertyName
```

9.10

```
SET x.propertyName = null
```

Example

```
MATCH (m:Movie)
WHERE m.title = 'Batman Begins'
```

Exercise 9: Creating Nodes

1) Create a Movie node for the movie with the title, *Forrest Gump*, and return its title.
2) Create Person node for the person with the name, *Robin Wright*, and return its name.
3) Add the label *OlderMovie* to any *Movie* node that was released before 2010, then return distinct labels.
4) Retrieve all older movie nodes to test that the label was indeed added to these nodes. Return movie title and year released.

Exercise 9: Creating Nodes

5) Add the label *Female* to all Person nodes that has a person whose name starts with *Robin* and return their names (non-list).

6) Retrieve all Female nodes and return their names (non-list).

7) Remove the Female label from the nodes that have this label and return their names (non-list).

7.5) Display the current schema of the graph

Exercise 9: Creating Nodes

8) Add *OlderMovie* label and add the following properties to the movie, *Forrest Gump*:

- released: 1994
- tagline: Life is like a box of chocolates...you never know what you're gonna get.
- lengthInMinutes: 142

Then return this Movie node and check the Table format.

10) Remove the *lengthInMinutes* property from the movie, *Forrest Gump*.

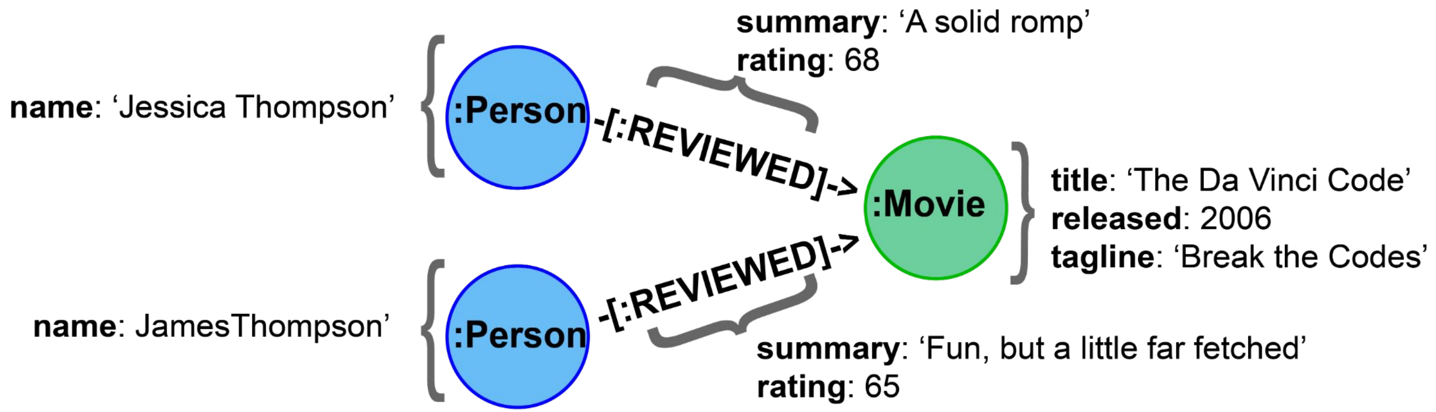
Then return this Movie node and check the Table format.

Syntax: Create a relationship

CREATE (x) -[:REL_TYPE]-> (y)

10.1

CREATE (x) <-[:REL_TYPE]- (y)



Ref: https://neo4j.com/graphacademy/training-querying-40/01-querying40-introduction-to-cypher/

Syntax: Create a relationship

- Connect the actor, *Michael Caine* with the movie, *Batman Begins*

```
MATCH (a:Person), (m:Movie)

WHERE a.name = 'Michael Caine' AND m.title = 'Batman Begins'
```

Syntax: Create multiple relationships

```
MATCH (a:Person), (m:Movie), (p:Person)
WHERE a.name = 'Liam Neeson' AND
    m.title = 'Batman Begins' AND
    p.name = 'Benjamin Melniker'
```

Syntax: Creating a relationship with properties

```
MATCH (a:Person), (m:Movie)
WHERE a.name = 'Katie Holmes' AND m.title = 'Batman Begins'
```

Syntax: Creating a node and a relationship

- Create *Gary Oldman* with the *ACTED_IN* relationship to *Batman Begins*

```
MATCH (m:Movie)

WHERE m.title = 'Batman Begins'
```

Syntax: Create a unique node and relationship

Create a new node or update existing node:

Create a new relationship or update existing relationship:

Syntax: Add property(s) to a relationship(s)

```
SET r.propertyName = value

SET r.propertyName1 = value1, r.propertyName2 = value2

SET r = {propertyName1: value1, propertyName2: value2}

SET r += {propertyName1: value1, propertyName2: value2}
```

Syntax: Add property(s) to a relationship(s)

- Add the *roles* property to the *ACTED_IN* relationship from *Christian Bale* to *Batman Begins*

```
MATCH (a:Person), (m:Movie)
WHERE a.name = 'Christian Bale' AND m.title = 'Batman Begins'
```

Syntax: Add property(s) to a relationship(s)

- Test if the relationship exists before creating it

```
MATCH (a:Person), (m:Movie)
WHERE a.name = 'Christian Bale' AND
      m.title = 'Batman Begins' AND


CREATE (a) -[rel:ACTED_IN]-> (m)
SET rel.roles = ['Bruce Wayne','Batman']
RETURN a, rel, m
```

Syntax: Remove property(s) from a relationship

```
REMOVE r.propertyName

SET r.propertyName = null
```

Example

```
MATCH (a:Person) -[rel:ACTED_IN]-> (m:Movie)

WHERE a.name = 'Christian Bale' AND m.title = 'Batman Begins'



RETURN a, rel, m
```

Exercise 10: Creating Relationships

1) Create the *ACTED_IN* relationship between the actors, *Robin Wright*, *Tom Hanks*, and *Gary Sinise* and the movie, *Forrest Gump*. Return person and movie nodes.
2) Create the *DIRECTED* relationship between *Robert Zemeckis* and the movie, *Forrest Gump*. Return person and movie nodes.
3) Create a new relationship, *HELPED* from *Tom Hanks* to *Gary Sinise*. Return both nodes.

Exercise 10: Creating Relationships

4) Write a Cypher query to return all nodes connected to the movie, *Forrest Gump*, along with their relationships.

5) Add a new property, *research* to the *HELPED* relationship between *Tom Hanks* and *Gary Sinise* and set this property's value to *war history*. Return both person nodes and relationships.

5.5) Display the current schema of the graph.

Exercise 10: Creating Relationships

6) Add the *roles* property to the three *ACTED_IN* relationships that you just created to the movie, *Forrest Gump* using this information:

Tom Hanks played the role, *Forrest Gump*.

Robin Wright played the role, *Jenny Curran*.

Gary Sinise played the role, *Lieutenant Dan Taylor*.

Return person nodes and movie nodes with relationships.

Hint: https://neo4j.com/docs/cypher-manual/current/syntax/expressions/#syntax-simple-case

Exercise 10: Creating Relationships

7) Query the graph to return the names and roles of actors in the movie, *Forrest Gump*.

8) Modify the roles that *Gary Sinise* played in the movie, *Forrest Gump* from *Lieutenant Dan Taylor* to *Lt. Dan Taylor*. Return the name and roles of an actor.

9) Remove the *research* property from the *HELPED* relationship from *Tom Hanks* to *Gary Sinise*. Return both person nodes and relationships.

9.5) Query the graph to confirm changes made to the graph.

Syntax: Delete a node

Suppose

```
CREATE (p:Person {name: 'Jane Doe'})
```

To delete this node:

```
MATCH (p:Person)
WHERE p.name = 'Jane Doe'
DELETE p
```

Syntax: Delete a relationship

Suppose

```
MATCH (a:Person), (m:Movie)
WHERE a.name = 'Katie Holmes' AND m.title = 'Batman Begins'
CREATE (a) -[:WROTE]-> (m)
CREATE (a) -[:DIRECTED]-> (m)
WITH a
MATCH (a) -[rel]- ()
RETURN type(rel)
```

Syntax: Delete a relationship

To delete relationships:

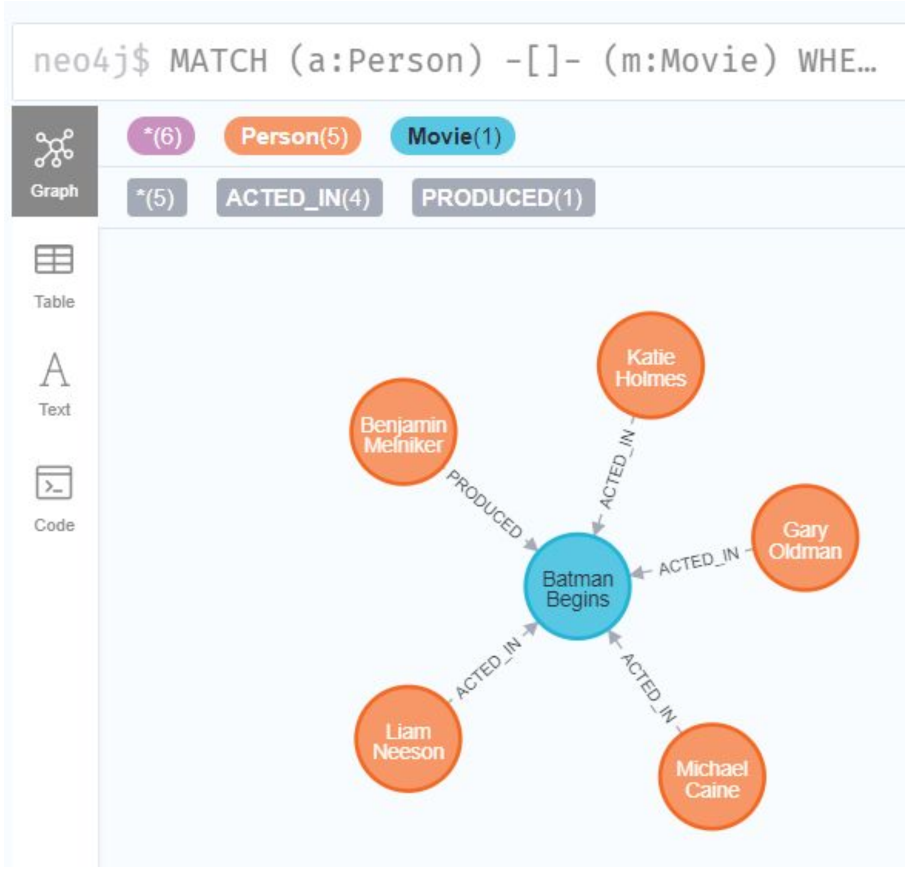
```
MATCH (a:Person) -[rel:WROTE | DIRECTED]-> (m:Movie)

WHERE a.name = 'Katie Holmes' AND m.title = 'Batman Begins'


RETURN a, m
```

Syntax: Check a relationship

```
MATCH (a:Person) -[rel]- ()

WHERE a.name = 'Katie Holmes'

RETURN count(rel) AS `Number of Katie Holmes relationships:`
```

Syntax: Delete a node and a relationship

```
MATCH (p:Person)

WHERE p.name = 'Liam Neeson'
```



```
neo4j$ MATCH (a:Person) -[]- (m:Movie) WHE…
```

Dr. Sirasit Lochanachit