

DBT ASSIGNMENT 4

Sneha Hegde

PES1201801157

3. Neo4j-

- Establishing a new Movie node (Coco) and setting timestamp value-
create (m:Movie {title:"Coco",tagline:"That is not dead which can eternal lie",released:2017})
return m

set timestamp value-
merge (m:Movie {title:"Coco"})
on match set m.lastUpdatedAt = timestamp()
on create set m.lastUpdatedAt = timestamp()
return m

The top screenshot shows the Neo4j Cypher Shell interface. The command entered is `$ create (m:Movie {title:"Coco",tagline:"That is not dead which can et...'})`. The interface displays a graph with a single orange node labeled "Coco". The status bar at the bottom indicates "Displaying 1 nodes, 0 relationships."

The bottom screenshot shows the Neo4j Cypher Shell interface. The command entered is `$ create (m:Movie {title:"Coco",tagline:"That is not dead which can et...'})`. The interface displays the JSON output for the node `m`. The status bar at the bottom indicates "Added 1 label, created 1 node, set 3 properties, started streaming 1 records after 195 ms and completed after 195 ms."

```
{
  "identity": 171,
  "labels": [
    "Movie"
  ],
  "properties": {
    "title": "Coco",
    "tagline": "That is not dead which can eternal lie",
    "released": 2017
  }
}
```



- Establishing new Person nodes-

```
create (p:Person
{name:"Anthony Gonzalez"})
return p
```

```
set timestamp value-
merge (p:Person {name:"Anthony Gonzalez"})
on match set p.lastUpdatedAt = timestamp()
on create set p.lastUpdatedAt = timestamp()
return p
```

```
create (p:Person
{name:"Benjamin Bratt"})
return p
```

```
set timestamp value-
merge (p:Person {name:"Benjamin Bratt"})
on match set p.lastUpdatedAt = timestamp()
on create set p.lastUpdatedAt = timestamp()
return p
```

```
create (p:Person
{name:"Adrian Molina"})
```

return p

set timestamp value-

merge (p:Person {name:"Adrian Molina"})

on match set p.lastUpdatedAt = timestamp()

on create set p.lastUpdatedAt = timestamp()

return p

\$ create (p:Person {na...

Graph

Table

Text

Code

```
{
  "identity": 172,
  "labels": [
    "Person"
  ],
  "properties": {
    "name": "Anthony Gonzalez"
  }
}
```

Added 1 label, created 1 node, set 1 property, started streaming 1 records af...

\$ create (p:Person {na...

Graph

Table

Text

Code

```
{
  "identity": 173,
  "labels": [
    "Person"
  ],
  "properties": {
    "name": "Benjamin Bratt"
  }
}
```

Added 1 label, created 1 node, set 1 property, started streaming 1 records af...

\$ create (p:Person {na...

Graph

Table

Text

Code

```
{
  "identity": 191,
  "labels": [
    "Person"
  ],
  "properties": {
    "name": "Adrian Molina"
  }
}
```

Added 1 label, created 1 node, set 1 property, started streaming 1 records af...

- Associating these two new Person nodes (Anthony Gonzalez, Benjamin Bratt) with the Movie node (Coco) through the relationship ACTED_IN-

```
match (p:Person),(m:Movie)
where p.name="Anthony Gonzalez" and m.title="Coco"
create (p)-[r:ACTED_IN]->(m)
return type(r)
```

```
match (p:Person),(m:Movie)
where p.name="Benjamin Bratt" and m.title="Coco"
create (p)-[r:ACTED_IN]->(m)
return type(r)
```

\$ match (p:Person),(m:Movie)

type(r)
"ACTED_IN"

Created 1 relationship, started streaming 1 records after 19 ms and completed...

- Associating the third Person node (Adrian Molina) with the Movie node (Coco) through the relationship DIRECTED-

```
match (p:Person),(m:Movie)
where p.name="Adrian Molina" and m.title="Coco"
create (p)-[r:DIRECTED]->(m)
return type(r)
```

\$ match (p:Person),(m:Movie)

type(r)
"DIRECTED"

Created 1 relationship, started streaming 1 records after 3 ms and completed...

- Return the name of the movie which Adrian Molina directed and Anthony Gonzalez acted in (It should give Coco)-

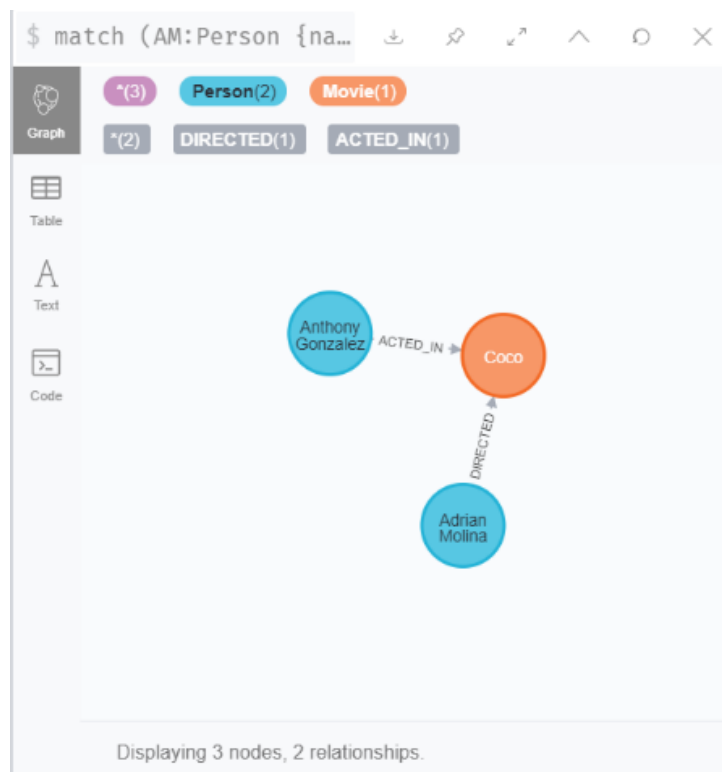
```
match (AM:Person {name:"Adrian Molina"})-[d:DIRECTED]->(m:Movie)-[a:ACTED_IN]-(p:Person {name:"Anthony Gonzalez"}) return m.title
```

\$ match (AM:Person {na...

m.title
"Coco"

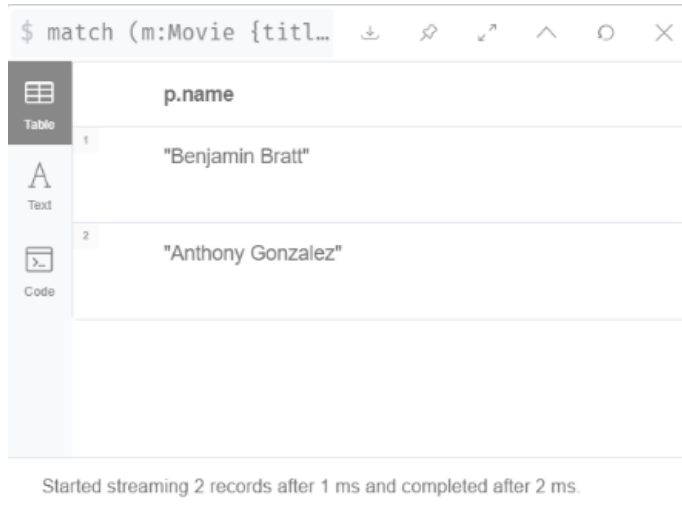
Started streaming 1 records after 1 ms and completed after 2 ms.

```
match (AM:Person {name:"Adrian Molina"})-[d:DIRECTED]->(m:Movie)-[a:ACTED_IN]-(p:Person {name:"Anthony Gonzalez"}) return AM,d,m,a,p
```



- Return the names of the people who acted in Coco (based on what we've entered)-

```
match (m:Movie {title:"Coco"})<-[a:ACTED_IN]-(p:Person)
return p.name
```

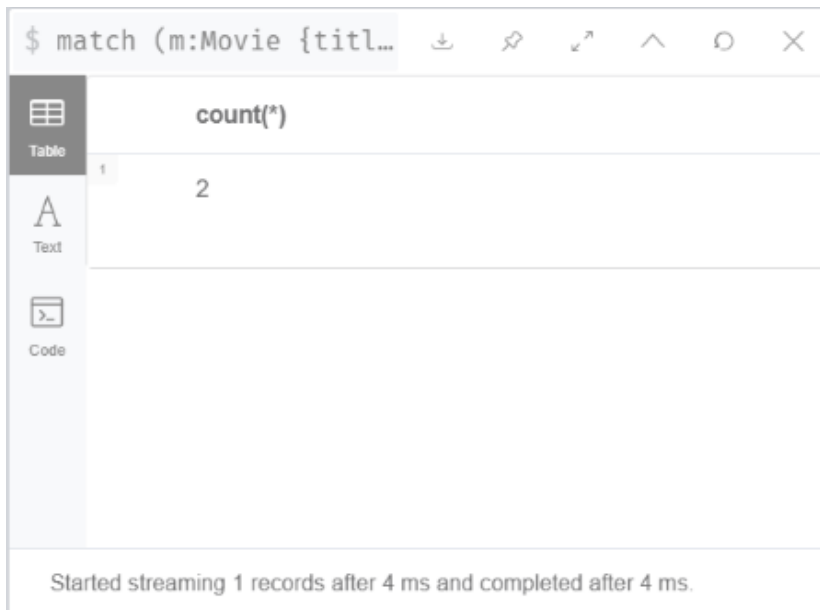


	p.name
1	"Benjamin Bratt"
2	"Anthony Gonzalez"

Started streaming 2 records after 1 ms and completed after 2 ms.

- Return the number of people who acted in Coco (based on what we've entered)-

```
match (m:Movie {title:"Coco"})<-[a:ACTED_IN]-(p:Person)
return count(*)
```



	count(*)
1	2

Started streaming 1 records after 4 ms and completed after 4 ms.