Lecture 7 - Aggregating Data
Sample Solution

CS343 Graph Data Science
Habib University

Spring 2024

1 Lecture 6 - Updating Data

1. Create a node with name "Daniel Kaluuya"

Solution:

```
        CREATE (d:User {name: 'Daniel Kaluuya'})

```

2. Daniel Kaluuya acted in movie "Get Out": make this relationship

Solution:

```
        MATCH (d:User {name: 'Daniel Kaluuya'})
        MATCH (m:Movie {title: 'Get Out'})
        MERGE (d)-[:ACTED_IN]->(m)

```

3. Add the tag line "Gripping, scary, witty and timely!" for the movie "Get Out" and set released year to 2017

Solution:

```
        MATCH (m:Movie {title: 'Get Out'})
        SET m.tagline = 'Gripping, scary, witty and timely!', m.released = 2017

```

4. Delete the actor "Emil Eifrem" from the database

Solution:

```
        MATCH (e:User {name: 'Emil Eifrem'})
        DETACH DELETE e

```

5. Assign :Actor label to all actors

Solution:

```
MATCH (a:User)-[:ACTED_IN]->()
SET a:Actor
```

6. Assign :Director label to all directors

Solution:

```
MATCH (d:User)-[:DIRECTED]->()
SET d:Director
```

2 Lecture 7 - Aggregating Data

1. Returns the number of movies Tom Hanks acted in

Solution:

```
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(movie)
RETURN count(movie)
```

2. Returns the number of movies released in 2000

Solution:

```
MATCH (movie:Movie)
WHERE movie.released = 2000
RETURN count(movie)
```

3. Returns the number of movies for each actor

Solution:

```
MATCH (actor:Person)-[:ACTED_IN]->(movie)
RETURN actor.name, count(movie)
```

4. Returns the number of actors who have acted in more than 5 movies.

Solution:

```
        MATCH (actor:Person)-[:ACTED_IN]->(movie)
        WITH actor, count(movie) as numMovies
        WHERE numMovies > 5
        RETURN count(actor)

```

5. List of roles played by Tom Hanks

Solution:

```
        MATCH (tom:Person {name: "Tom Hanks"})-[r:ACTED_IN]->(movie)
        RETURN tom.name, collect(r.roles)

```

3 Lecture 8 - Paths and Shortest Paths

1. Returns the shortest path between the Eminem node and the Charlton Heston node

Solution:

```
        MATCH p=shortestPath((eminem:Person {name: "Eminem"})-[*]-(charlton:Person {
    name: "Charlton Heston"}))
        RETURN p

```

2. Returns the shortest path between the Eminem node and the Charlton Heston node considering only ACTED_IN relationship

Solution:

```
        MATCH p=shortestPath((eminem:Person {name: "Eminem"})-[:ACTED_IN*]-(charlton
    :Person {name: "Charlton Heston"}))
        RETURN p

```

3. Retrieve all Person nodes that are exactly two hops away from Eminem using the ACTED_IN relationship.

Solution:

```
        MATCH (eminem:Person {name: "Eminem"})-[:ACTED_IN*2]-(coActor)
        RETURN coActor

```

4. Retrieve all Person nodes that are up to four hops away from Eminem using the ACTED_IN relationship.

Solution:

```
        MATCH (eminem:Person {name: "Eminem"})-[:ACTED_IN*1..4]-(coActor)
        RETURN coActor

```

5. Return a list of actors that are up to 6 hops away from Tom Hanks

Solution:

```
        MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN*1..6]-(coActor)
        RETURN coActor

```