Neo4j, part II

Controlling query processing

- Retrieve all nodes that the person named James Thompson directly has the FOLLOWS relationship in either direction.

```
MATCH (p1:Person)-[:FOLLOWS]-(p2:Person)
WHERE p1.name = 'James Thompson'
RETURN p1, p2
```

- Retrieve nodes that are exactly three hops away.

```
MATCH (p1:Person)-[:FOLLOWS*3]-(p2:Person)
WHERE p1.name = 'James Thompson'
RETURN p1, p2
```

- Retrieve nodes that are one and two hops away.

```
MATCH (p1:Person)-[:FOLLOWS*1..2]-(p2:Person)
WHERE p1.name = 'James Thompson'
RETURN p1, p2
```

Controlling query processing (cont.)

- People whose name begins with Tom and the movies directed.

```
MATCH (p:Person)
WHERE p.name STARTS WITH 'Tom'
OPTIONAL MATCH (p)-[:DIRECTED]->(m:Movie)
RETURN p.name, m.title
```

- Retrieve actors and the list of movies they acted in.

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name as actor, collect(m.title) AS `movie list`
```

-Retrieve all people who reviewed a movie, returning the list of reviewers and how many reviewers reviewed the movie.

```
MATCH (p:Person)-[:REVIEWED]->(m:Movie)
RETURN m.title as movie, count(p) as numReviews
        , collect(p.name) as reviewers
```

Controlling results returned

- Query with duplicate records

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE m.released >= 1990 AND m.released < 2000
RETURN DISTINCT m.released, m.title, collect(a.name)
```

- Eliminate duplicates (collect movies per year)

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE m.released >= 1990 AND m.released < 2000
RETURN  m.released, collect(m.title), collect(a.name)
```

- Eliminate duplicates in title (and sort/top 5)

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE m.released >= 1990 AND m.released < 2000
RETURN  m.released, collect(DISTINCT m.title), collect(a.name
ORDER BY m.released DESC LIMIT 5
```

Lists

- Collect

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
WITH p, collect(m) AS movies
WHERE size(movies)  > 5
RETURN p.name, movies
```

- Unwrap

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
WITH p, collect(m) AS movies
WHERE size(movies)  > 5
WITH p, movies UNWIND movies AS movie
RETURN p.name, movie.title
```

Date calculations

- Age of Tom Hanks per movie

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE a.name = 'Tom Hanks'
RETURN  m.title, m.released, date().year  - m.released as
yearsAgoReleased, m.released  - a.born AS `age of Tom`
ORDER BY yearsAgoReleased
```

Your turn

Exercises

- Retrieve all movies that Tom Cruise has acted in and the co-actors that acted in the same movie, returning the movie title and the list of co-actors that Tom Cruise worked with.
- Retrieve the actors who have acted in exactly five movies, returning the name of the actor, and the list of movies for that actor.
- Retrieve the movies that have at least 2 directors, and optionally the names of people who reviewed the movies.
- Retrieve all actors that have not appeared in more than 3 movies. Return their names and list of movies.

Solutions

- Retrieve all movies that Tom Cruise has acted in and the co-actors that acted in the same movie, returning the movie title and the list of co-actors that Tom Cruise worked with.

```
MATCH(p:Person)-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(q:Person
WHERE p.name ='Tom Cruise'
RETURN m.title as movie, collect(q.name) AS `co-actors`
```

- Retrieve the actors who have acted in exactly five movies, returning the name of the actor, and the list of movies for that actor.

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WITH  a, count(a) AS numMovies, collect(m.title) AS movies
WHERE numMovies = 5
RETURN a.name, movies
```

Solutions (cont.)

- Retrieve the movies that have at least 2 directors, and optionally the names of people who reviewed the movies.

```
MATCH (m:Movie)
WITH m, size((:Person)-[:DIRECTED]->(m)) AS directors
WHERE directors >= 2
OPTIONAL MATCH (p:Person)-[:REVIEWED]->(m)
RETURN  m.title, p.name
```

- Retrieve all actors that have not appeared in more than 3 movies. Return their names and list of movies.

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WITH  a,  count(a) AS numMovies, collect(m.title) AS movies
WHERE numMovies <= 3
RETURN a.name, movies
```