Neo4j basics

Retrieving nodes

- Get all nodes from the database

```
MATCH (n) RETURN (n)
```

- Look at the schema

```
CALL db.schema.visualization()
```

- Match all the `Person` nodes

```
MATCH (p:Person) RETURN p
```

Filter by property value

- Retrieve all Movie nodes that have a released property value of 2003.

```
MATCH (m:Movie {released:2003}) RETURN m
```

- Retrieve all Movies released in 2006, returning their titles.

```
MATCH (m:Movie {released: 2006}) RETURN m.title
```

Filter by property value: anonymous nodes

- Retrieve all people who wrote the movie 'Speed Racer'.

```
MATCH (p:Person)-[:WROTE]->(:Movie {title: 'Speed Racer'})
RETURN p.name
```

Filtering queries using relationships

- Retrieve all movies that are connected to Tom Hanks.

```
MATCH (m:Movie)<--(:Person {name: 'Tom Hanks'}) RETURN m.titl
```

- Return the type information about the relationships between Tom Hanks and the movies.

```
MATCH (m:Movie)-[rel]-(:Person {name: 'Tom Hanks'})
RETURN m.title, type(rel)
```

Filtering queries using the WHERE clause

- Movies where Tom Cruise acted in and return their titles.

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE a.name = 'Tom Cruise'
RETURN m.title as Movie
```

- People born in the 70â€™s and return their names and year born.

```
MATCH (a:Person)
WHERE a.born >= 1970 AND a.born < 1980
RETURN a.name as Name, a.born as `Year Born`
```

Filtering queries using the WHERE clause (cont.)

- Retrieve all movies released in 2000 by testing the node label and the released property, returning the movie titles.

```
MATCH (m)
WHERE m:Movie AND m.released = 2000
RETURN m.title
```

- Retrieve all people in the graph that do not have a born property, returning their names.

```
MATCH (a:Person)
WHERE NOT exists(a.born)
RETURN a.name as Name
```

Filtering queries using the WHERE clause (cont.)

- Retrieve all people related to movies where the relationship has the rating property, then return their name, movie title, and the rating.

```
MATCH (a:Person)-[rel]->(m:Movie)
WHERE exists(rel.rating)
RETURN a.name as Name, m.title as Movie, rel.rating as Rating
```

- Retrieve all actors whose name begins with James, returning their names.

```
MATCH (a:Person)-[:ACTED_IN]->(:Movie)
WHERE a.name STARTS WITH 'James'
RETURN a.name
```

Filtering queries using the WHERE clause (cont.)

- Retrieve the movies and their actors where one of the actors also directed the movie, returning the actors names, the directorâ€™s name, and the movie title.

```
MATCH
(a1:Person)-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(a2:Person)
WHERE exists( (a2)-[:DIRECTED]->(m) )
RETURN  a1.name as Actor, a2.name as `Actor/Director`,
m.title as Movie
```

Filtering queries using the WHERE clause (cont.)

Your turn!

Exercises

- Retrieve all Movie nodes from the database and return the title, released, and tagline values.
- Retrieve all of the roles that Tom Hanks acted in.
- Retrieve the actors who acted in the movie The Matrix who were born after 1960, and return their names and year born.
- Retrieve all people that wrote movies by testing the relationship between two nodes, returning the names of the people and the titles of the movies.
- Retrieve all people who have produced a movie, but have not directed a movie, returning their names and the movie titles.
- Retrieve the movies that have an actorâ€™s role that is the name of the movie, return the movie title and the role.

Solutions

- Retrieve all Movie nodes from the database and return the title, released, and tagline values.

```
MATCH (m:Movie) RETURN m.title, m.released, m.tagline
```

- Retrieve all of the roles that Tom Hanks acted in.

```
MATCH (m:Movie)-[rel:ACTED_IN]-(:Person {name: 'Tom Hanks'})
RETURN m.title, rel.roles
```

- Retrieve the actors who acted in the movie The Matrix who were born after 1960, and return their names and year born.

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE a.born > 1960 AND m.title = 'The Matrix'
RETURN a.name as Name, a.born as `Year Born`
```

Solutions (cont.)

- Retrieve all people that wrote movies by testing the relationship between two nodes, returning the names of the people and the titles of the movies.

```
MATCH (a)-[rel]->(m)
WHERE a:Person AND type(rel) = 'WROTE' AND m:Movie
RETURN a.name as Name, m.title as Movie
```

- Retrieve all people who have produced a movie, but have not directed a movie, returning their names and the movie titles.

```
MATCH (a:Person)-[:PRODUCED]->(m:Movie)
WHERE NOT ((a)-[:DIRECTED]->(:Movie))
RETURN a.name, m.title
```

Solutions (cont.)

- Retrieve the movies that have an actorâ€™s role that is the name of the movie, return the movie title and the role.

```
MATCH (a:Person)-[r:ACTED_IN]->(m:Movie)
WHERE m.title in r.roles
RETURN  m.title as Movie, a.name as Actor
```