Manipulating Data

Roland Guijt
www.rmgsolutions.nl
@rolandguijt

pluralsight
hardcore dev and IT training

Agenda

Creating, Updating and Deleting

Advanced data manipulation

Importing CSV

Indexes and Unique Constraint

Create

Creates nodes and relationships

```
CREATE (n)

CREATE (n:Actor{name: 'Peter Capaldi'})
RETURN n

MATCH (matt:Actor{name: 'Matt Smith'}),
chris:Actor{name: 'Christopher Eccleston'}

CREATE (matt) [:REGENERATED_TO] (chris)
```

Create Complete Path

```
CREATE p =(:Actor{name: 'Peter Capaldi'})-
[:APPEARED_IN] ->(:Episode{name:'The Time
of The Doctor'})

RETURN p
```

Delete

Deletes nodes and relationships

```
MATCH (matt:Actor{name: 'Matt Smith'})
DELETE matt
```

```
MATCH (matt:Actor{name: 'Matt Smith'})-[r]-()
DELETE matt, r
```

Set

Manipulates properties

```
MATCH (matt:Actor{name: 'Matt Smith'})
SET matt.salary = 100000, matt.active = true

MATCH (matt:Actor{name: 'Matt Smith'})
SET matt.salary = NULL

MATCH (matt:Actor{name: 'Matt Smith'}),
chris:Actor{name: 'Christopher Eccleston'}
SET matt = chris
```

Set

Sets labels

```
MATCH (matt:Actor{name: 'Matt Smith'})

SET matt:Doctor
```

Remove

Removes properties or labels

```
MATCH (matt:Actor{name: 'Matt Smith'})
REMOVE matt:Doctor

MATCH (matt:Actor{name: 'Matt Smith'})
REMOVE matt.salary
```

Merge

Match replacement: returns or creates (parts of) a pattern

```
MERGE (peter:Actor{name: 'Peter Capaldi'})
RETURN peter

MERGE (peter:Actor{name: 'Peter Capaldi',
salary: 100000})
RETURN peter

MATCH (peter:Actor{name: 'Peter Capaldi'}),
(doctor:Character{name: "Doctor"})
MERGE (peter –[r:PLAYED]->doctor)
RETURN r
```

Foreach

Helper to set a property or label in a path

```
MATCH p=(actors:Actor)–[r:PLAYED]->others)
WHERE actors.salary > 100000
FOREACH (n IN nodes(p)| set n.done = true)
```

Index

Performance gain when querying for a certain property value

- Keeps dictionary of values
- Watch performance issues while writing
- The use of an index is automatic

```
MATCH (matt:Actor{name: 'Matt Smith'})
RETURN matt
```

```
CREATE INDEX ON :Actor(name)
```

```
DROP INDEX ON :Actor(name)
```

Unique Constraint

Ensures uniqueness of a property value

- Currently the only constraint available
- Watch performance issues while writing
- Will also add an index

```
CREATE CONSTRAINT ON (a:Actor)
ASSERT a.name IS UNIQUE

DROP CONSTRAINT ON (a:Actor)
ASSERT a.name IS UNIQUE
```

Importing CSV

- Cypher supports importing CSV
- CSV files can be loaded from the local file system or via HTTPS, HTTP and FTP
- Use CREATE and MERGE in conjunction with LOAD CSV
- Example: actors, movies, connections

Importing CSV: Step 1

- Import actors
- CSV looks like this:

id	name
3	Michael Douglas
4	Martin Sheen
5	Morgan Freeman

```
LOAD CSV WITH HEADERS FROM
"http://docs.neo4j.org/chunked/2.1.6/csv/import/persons.csv"
AS csvLine

CREATE (p:Person {id: toInt(csvLine.id),
name: csvLine.name})
```

Importing CSV: Step 2

- Import movies, normalize countries
- CSV looks like this:

id	title	country
1	Wall Street	USA
2	The American President	USA

```
LOAD CSV WITH HEADERS FROM
"http://docs.neo4j.org/chunked/2.1.6/csv/import/movies.csv"
AS csvLine
MERGE (country: Country {name:
csvLine.country})
CREATE (movie:Movie {id: toInt(csvLine.id),
title: csvLine.title})
CREATE (movie)-[MADE_IN]->(country)
```

Importing CSV: Step 3

- Import actor -> movies relationship
- CSV looks like this:

personId	movieId	role
4	1	Carl Fox
4	2	A.J. MacInerney

```
LOAD CSV WITH HEADERS FROM
"http://docs.neo4j.org/chunked/2.1.6/csv/import/roles.csv"
AS csvLine
MATCH (actor:Person {id: toInt(csvLine.personId}),
(movie:Movie {id: toInt(csvLine.movieId})
CREATE (actor)-[:PLAYED {role: csvLine.role}]-
>(movie)
```

Summary

- Use CREATE to create nodes and relationships.
- With DELETE you can remove them.
- Set and update property values and add labels to nodes with SET.
- REMOVE deletes properties and labels.
- MERGE only creates if needed.
- An index on a property makes querying faster, but writing slower.
- Use the unique constraint to make property values unique.
- Import data from other systems with Cypher's support for reading CSV.

What's Next?

- REST API