

Manipulating Data with Cypher



Roland Guijt

MVP | CONSULTANT | TRAINER | AUTHOR

@rolandguijt rolandguijt.com



Module Overview



Creating

Updating

Deleting

Advanced statements

Indexes

Unique constraint

Importing CSV



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 a Pattern in One Go

```
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 Also Sets Labels

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

SET matt:Doctor



REMOVE: Removes Properties and Labels

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

REMOVE matt:Doctor

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

REMOVE matt.salary



MERGE: Returns or Creates

```
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

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



Use Indexes to Speed Up Reads

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

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

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



Unique Constraint Ensures Uniqueness

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

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



Importing CSV

Supported by Cypher

Local file system, HTTP, FTP

CREATE and **MERGE** together with
LOAD CSV

Example: actors, movies, connections



Importing Actors from CSV

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 Countries from CSV

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 Relationships

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



Create, delete, update

Merge creates if needed

Indexes speed up reads

Use unique constraint to make property values unique

Cypher supports CSV reads

