Neo4j commands

LOAD CSV WITH HEADERS FROM

"file:////home/training/training\_materials/data/product/product\_name.csv"

as row

CREATE (p:product {name:row.product\_name, id:row.id })

LOAD CSV WITH HEADERS FROM

"file:////home/training/training\_materials/data/product/category\_name.csv"

as row

CREATE (c:category {name:row.category\_name, id:row.id })

LOAD CSV WITH HEADERS FROM

" file:////home/training/training\_materials/data/product/product\_category\_relationship.csv"

as row

MATCH (p:product {id:row.id2})

MATCH (c:category {id:row.id1})

CREATE UNIQUE ((p)-[:refered {frequency: row.count}]->(c))

# Analysis commands

### Which are the top 5 most popular referrer?

MATCH (p:product)-[r:refered]-(c:category)

RETURN p,r,c order by r.frequency desc limit 5

### Looking for a path of 2 hops between products – 33 and 36:

MATCH (p1:product {id:"33"})-[:refered]-(c:category)-[:refered]-(p2:product{id:"36"})

RETURN p1,c,p2

### Node with most incoming relationships – Most influential node?

MATCH (p:product)-[r:refered]-(c:category)

RETURN count(rels(c)) as count

order by count desc limit 1

### Matching all categories for a product:

MATCH (p:product{id:"135"})-[:refered]->(c:category)

RETURN p,c

### Returning all nodes and relationships

MATCH (n)

DETACH DELETE n

## Shortest Path commands – to be modified as required:

### Looking for all unique actors with a path of 4 hops from Bacon:

MATCH (b:actor {fullname:"Kevin Bacon"})-[\*4]-(a:actor)

RETURN distinct a.fullname

### Looking for the shortest path from Bacon to other actors ("0.." is added to include Kevin Bacon in the list):

MATCH p=shortestpath(

((b:actor {fullname:"Kevin Bacon"})-[\*0..]-(a:actor))

)

RETURN a.fullname, length(p)

### Looking for the shortest path from Bacon to other actors and outputting the Bacon Number ("0.." is added to include Kevin Bacon in the list):

MATCH p=shortestpath(

((b:actor {fullname:"Kevin Bacon"})-[\*0..]-(a:actor))

)

RETURN a.fullname, 1+length(p)/2