

# openCypher Graph addressing

## Draft graph URI scheme

Peter Furniss, Alastair Green  
26 July 2017

# Addressing multiple graphs

OLTP graph database

```
FROM GRAPH social-network
  AT "graph:bolt:(bolt+routing://social-network/transactional)"

MATCH (a:Person)-[:KNOWS]->(b:Person)-[:KNOWS]->(c:Person) WHERE NOT (a)--(c)

INTO NEW GRAPH recommendations      -snap
  AT "graph:hdfs+parquet+NE:(hdfs://social-network/2018-04-12/recommendations)"

CREATE (a)-[:POSSIBLE_FRIEND]->(c)
```

Hadoop snapshot collection

# Graph URI - draft scheme

graph:<locator-scheme>:(<escaped-locator-uri>)

registered  
scheme name

Defines **how** the graph  
is stored/accessed

Defines **where** the  
graph is  
stored/accessed

<locator-uri>

parentheses or  
single-quotes

%-encode/decode to hide  
URI meta-characters from  
standard parsers

# Examples

## RDF

*graph:rdf+http:'http://chucknorris.com/data /chuck/foaf\_based\_near'*

## Bolt named graph

*graph:bolt:(bolt+routing://west-coast/orders-snap-2016-04-12)*

## HDFS in Parquet files

*graph:hdfs+parquet+NE:(hdfs://production/west-coast/orders/snap/2016-04-12)*

Password is  
sec#ret

## SQL tables, with escaped character in the locator uri

*graph:sql+NE+jdbc:(jdbc:derby://pluto.paleo.com/g3;password=sec%23ret)*

Graph is accessible via Node and Edge views (Neo4j SQL representation proposal)

# Relative URIs

Base locator URI

```
STORE IS "graph:hdfs+parquet+NE:(hdfs://social-network/snapshot/2018-04-12)"
```

```
FROM GRAPH social-network
```

```
AT "(whole-social-network)"
```

Relative locator URI

```
MATCH (a:Person)-[:KNOWS]->(b:Person)-[:KNOWS]->(c:Person) WHERE NOT (a)--(c)
```

```
INTO NEW GRAPH recommendations
```

```
AT "(recommendations)"
```

```
CREATE (a)-[:POSSIBLE_FRIEND]->(c)
```