

Neo4j Database & Graph Platform

GraphTour Boston – May 8, 2018

Ryan Boyd
@ryguyrg

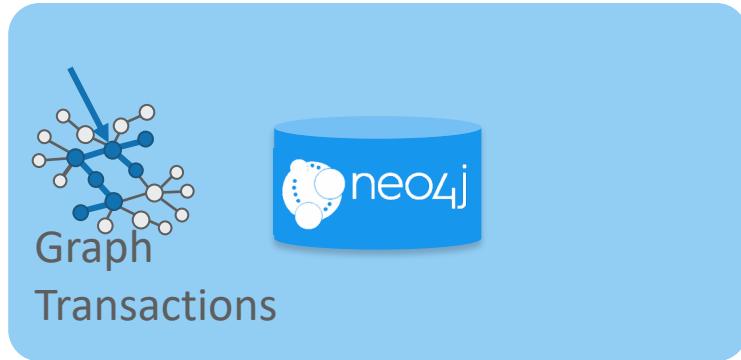


An IT Portfolio View of Graph Technologies:

The Neo4j Graph Platform Vision



The Neo4j Graph Platform Vision



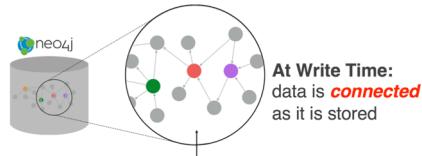
Neo4j: Key Components



Key Architecture Components

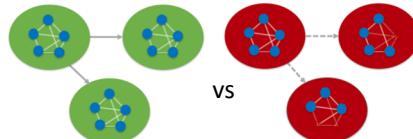
1

Index-Free Adjacency



In memory and on flash/disk

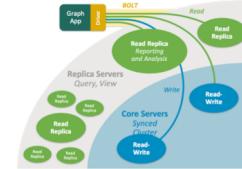
ACID Foundation



Required for safe writes

3

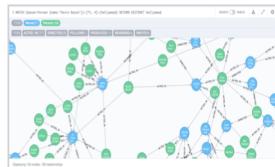
Full-Stack Clustering



Causal consistency

4

Language, Drivers, Tooling



*Developer Experience,
Graph Efficiency, Type Safety*

2

Graph Engine



*Cost-Based Optimizer, Graph
Statistics, Cypher Runtime, ...*

5

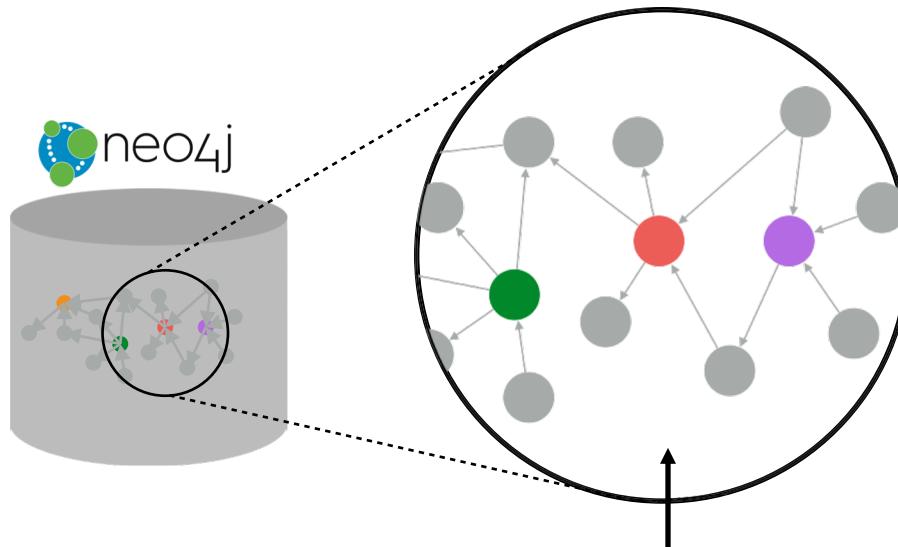
6

Hardware Optimizations



For next-gen infrastructure

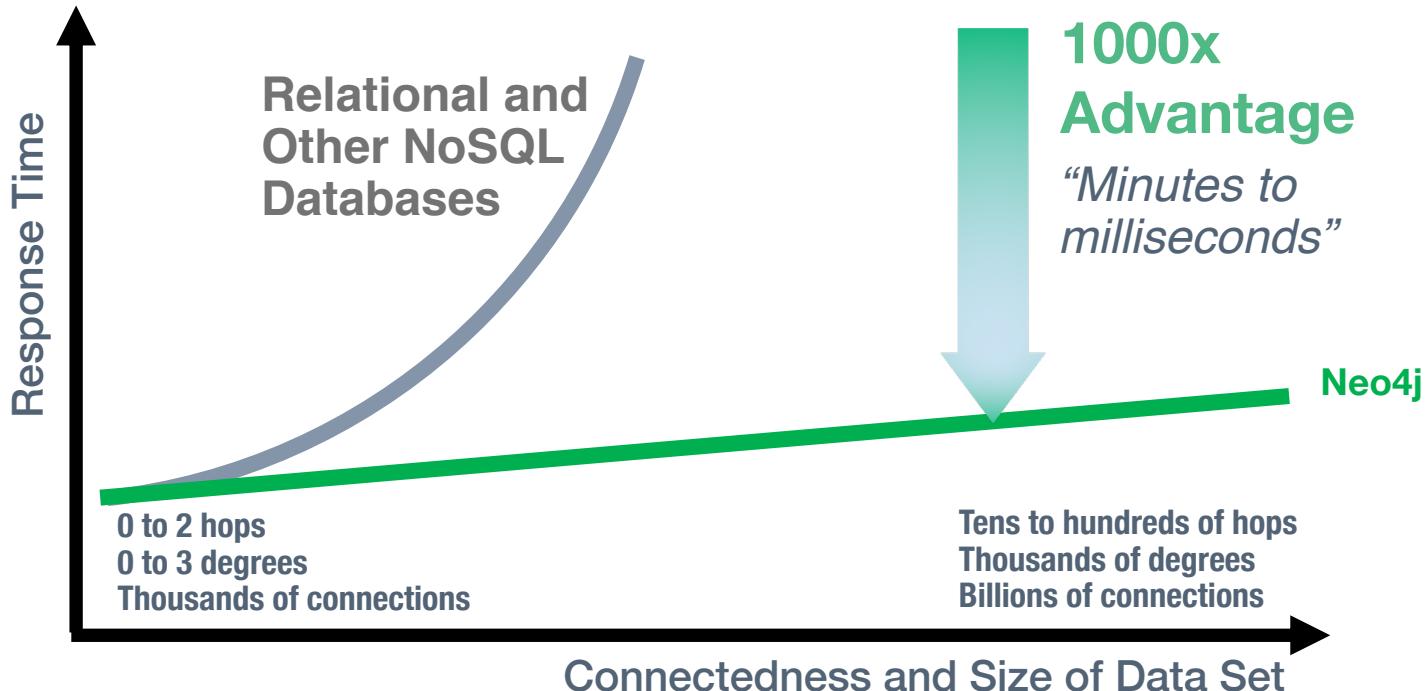
Index-Free Adjacency:



At Write Time:
data is *connected*
as it is stored

At Read Time:
Lightning-fast retrieval of data and
relationships via pointer chasing

“Minutes to Milliseconds” Real-Time Query Performance



Neo4j: Enabling the Connected Enterprise

Consumers of Connected Data

AI & Graph Analytics

- Sentiment analysis
- Customer segmentation
- Machine learning
- Cognitive computing
- Community detection



Applications

Transactional Graphs

- Fraud detection
- Real-time recommendations
- Network and IT operations management
- Knowledge Graphs
- Master Data Management



Data
Scientists

Discovery & Visualization

- Fraud detection
- Network and IT operations
- Product information management
- Risk and portfolio analysis



Business
Users

The Neo4j Graph Platform Vision



Neo4j Graph Algorithm Library

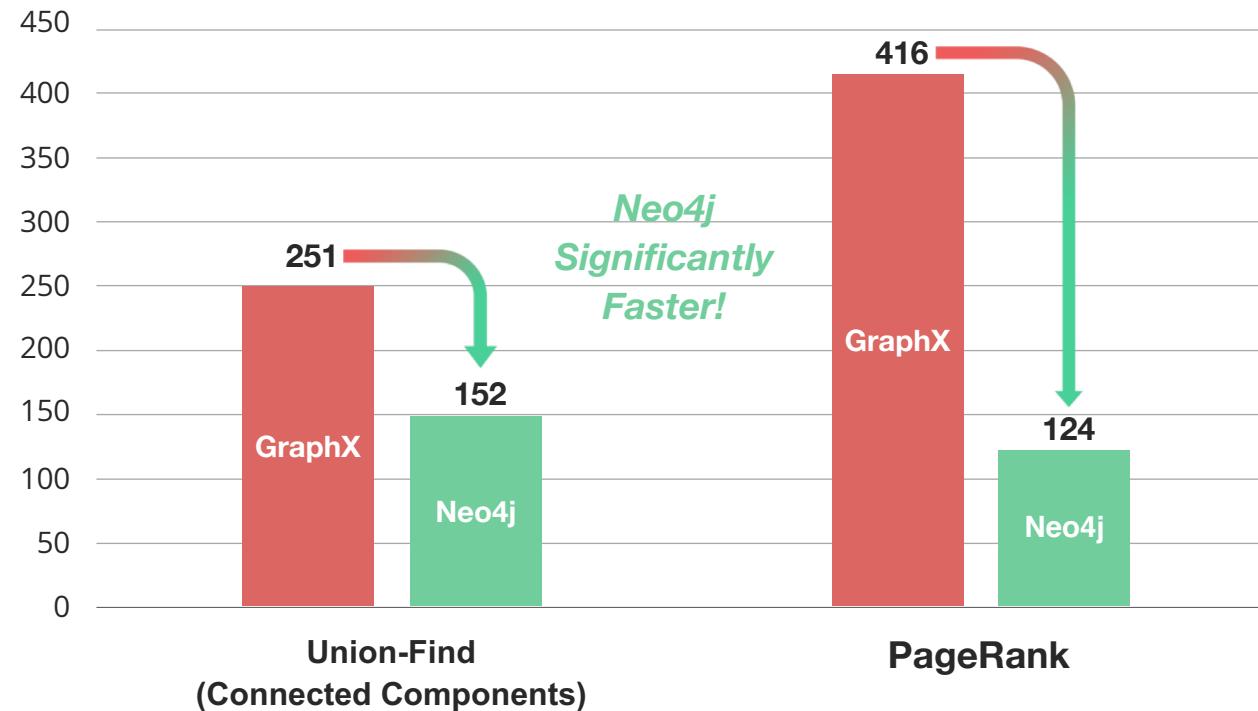


Evaluates how a graph is clustered or partitioned

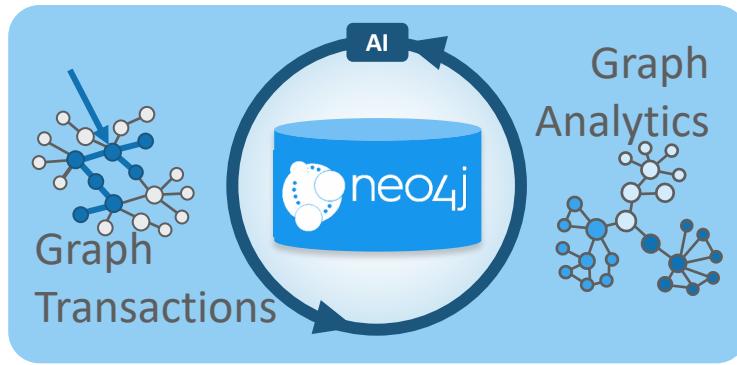
Determines the importance of distinct nodes in the network

Finds the optimal path or evaluates route availability and quality

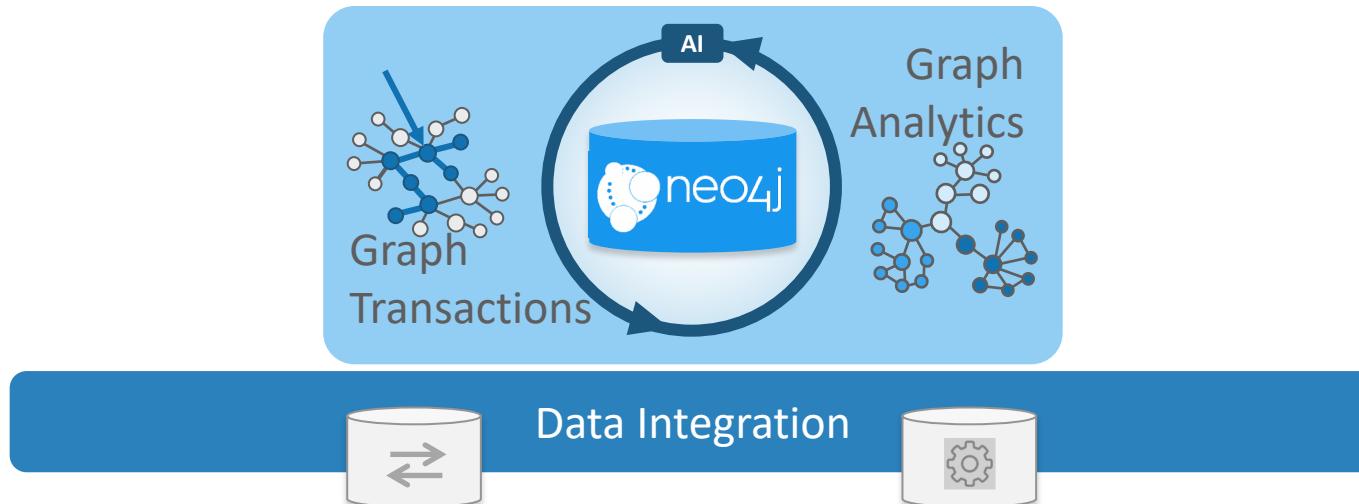
Graph Algorithm Performance



The Neo4j Graph Platform Vision



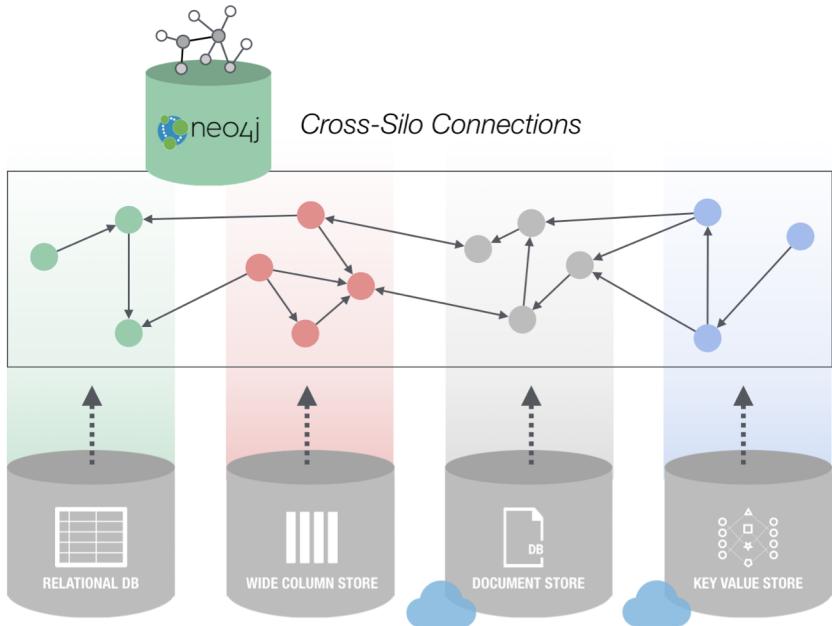
The Neo4j Graph Platform Vision



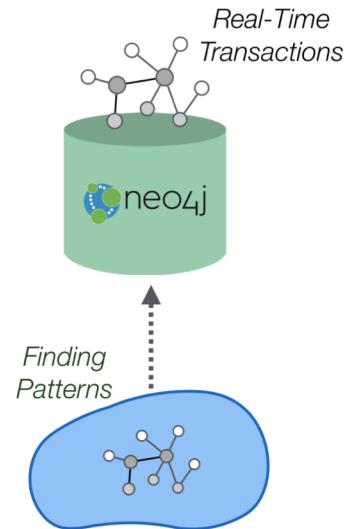
Common Integration Patterns Inside the Enterprise



From Tabular Data
To Connected Data

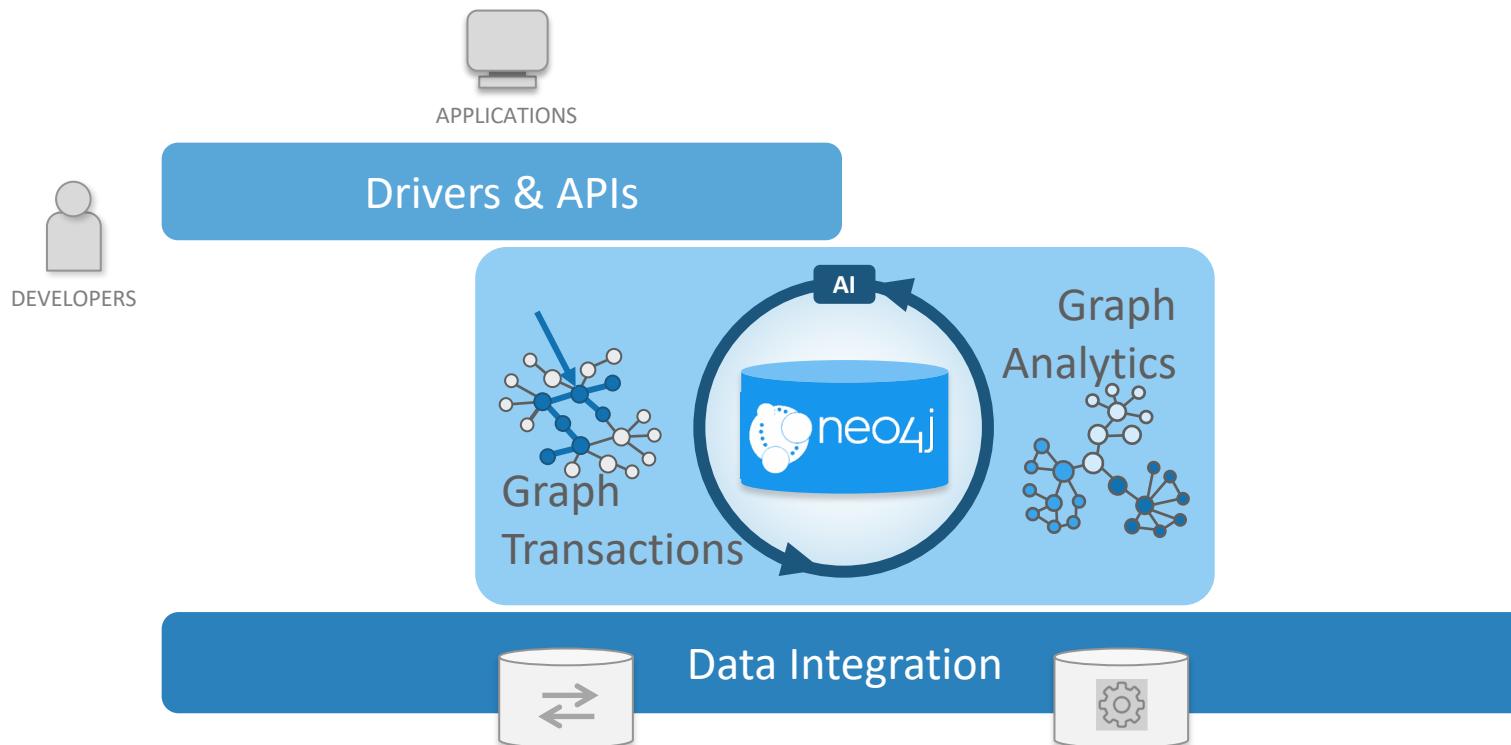


From Disparate Silos
To Cross-Silo Connections



From Data Lake Analytics
to Real-Time Operations

The Neo4j Graph Platform Vision



Drivers & APIs

- Native Language Drivers
 - Java
 - .Net
 - Python
 - JavaScript
 - more to come...
- Massive Community Support (Go, Ruby, R, Perl, Clojure, C/C++, ...)
- Partners like GraphAware (PHP Client)

See Code In: [JAVA](#) [PYTHON](#) [RUBY](#) [PHP](#) [C#](#) [JAVASCRIPT](#)

```
import org.neo4j.driver.v1.*;
import static org.neo4j.driver.v1.Values.parameters;

import java.util.List;
import static java.util.Arrays.asList;
import static java.util.Collections.singletonMap;

public class Social {

    public static void main(String...args) {

        Config noSSL = Config.build().withEncryptionLevel(Config.EncryptionLevel.NO);
        Driver driver = GraphDatabase.driver("bolt://localhost", AuthTokens.basic("n
try (Session session = driver.session()) {

    List data =
        asList(asList("Jim", "Mike"), asList("Jim", "Billy"), asList("Anna", "Jim")
        asList("Anna", "Mike"), asList("Sally", "Anna"), asList("Joe", "Sally"),
        asList("Joe", "Bob"), asList("Bob", "Sally"));

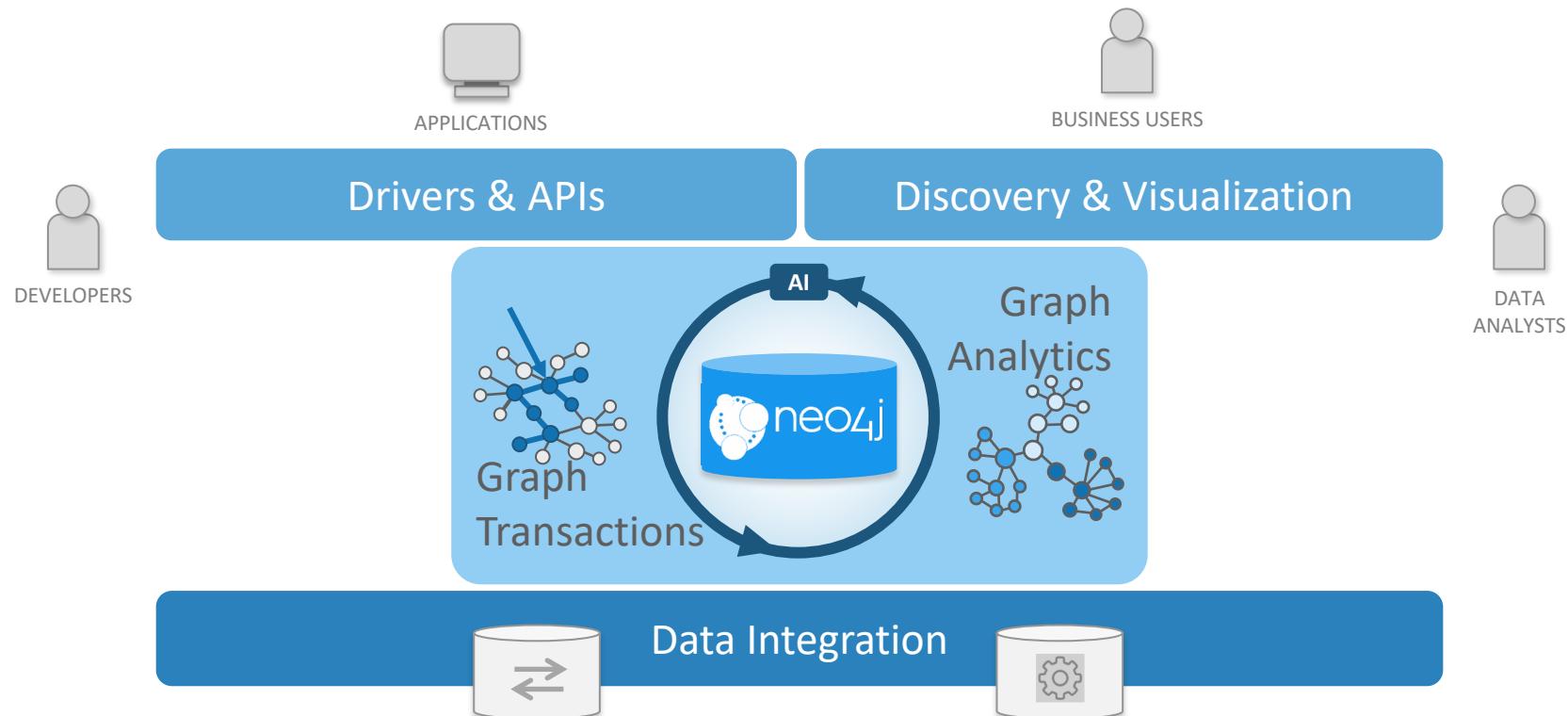
    String insertQuery = "UNWIND {pairs} as pair "
        "MERGE (p1:Person {name:pair[0]}) "
        "MERGE (p2:Person {name:pair[1]}) "
        "MERGE (p1)-[:KNOWS]-(p2)";

    session.run(insertQuery, singletonMap("pairs", data)).consume();
    StatementResult result;
```

Downloading and Installing java

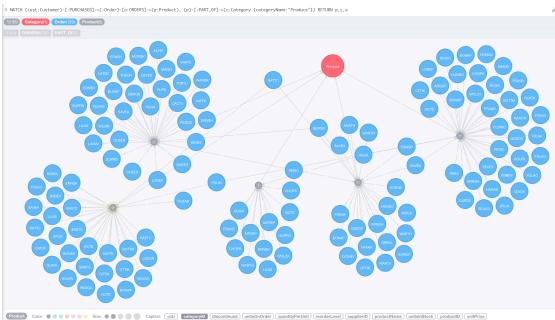
1. [Download Neo4j Driver](#)
2. Copy and paste code at left into `Social.java`
3. Run
`javac -cp neo4j-javascript-driver-1.0.jar Social.java`
4. Run
`java -cp neo4j-javascript-driver-1.0.jar:. Social`

The Neo4j Graph Platform Vision

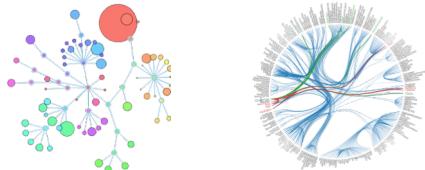


Discovery & Visualization

Neo4j Browser



Custom / JS Libraries



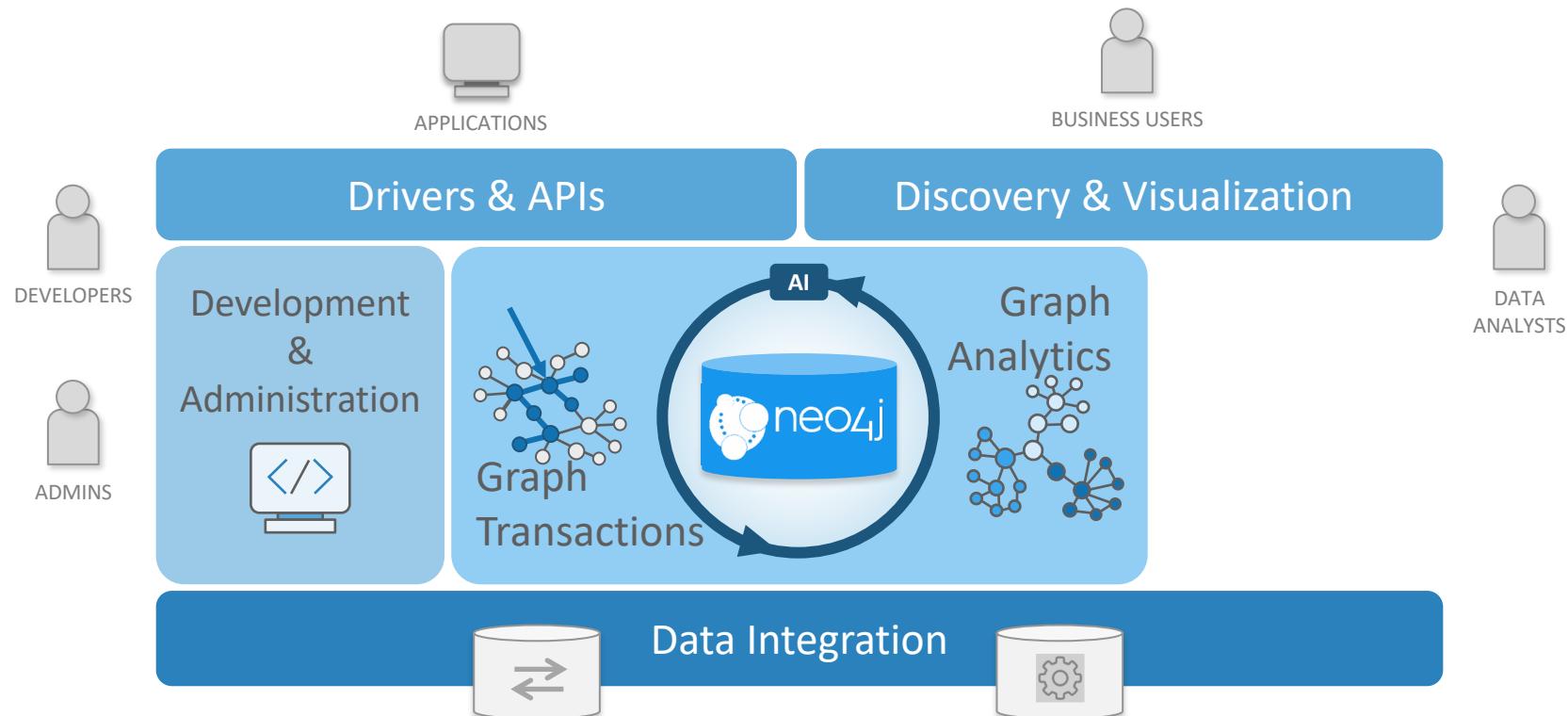
Partner Applications



Cambridge Intelligence

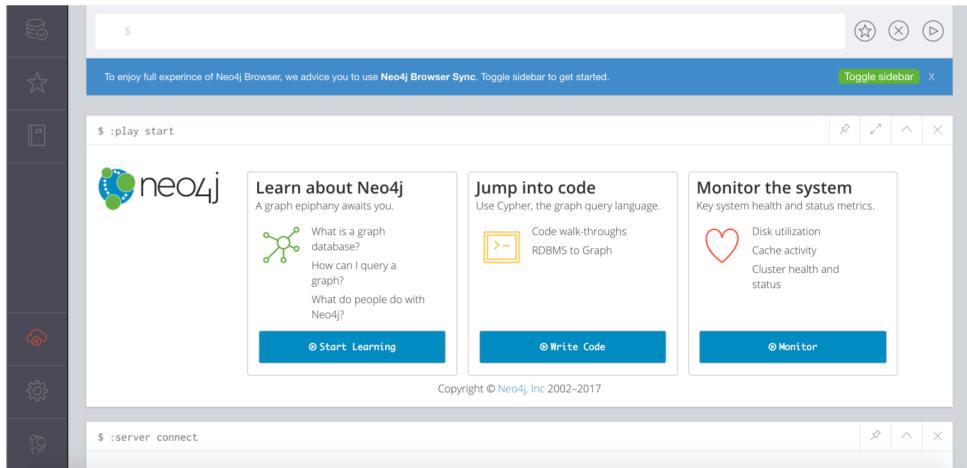


The Neo4j Graph Platform Vision

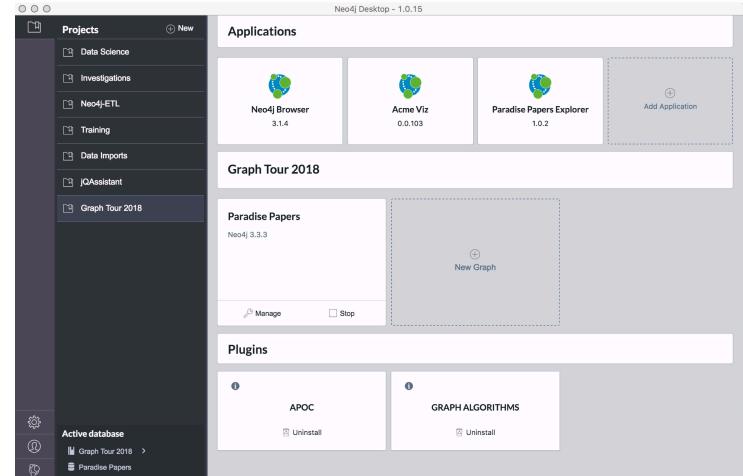


Development & Administration

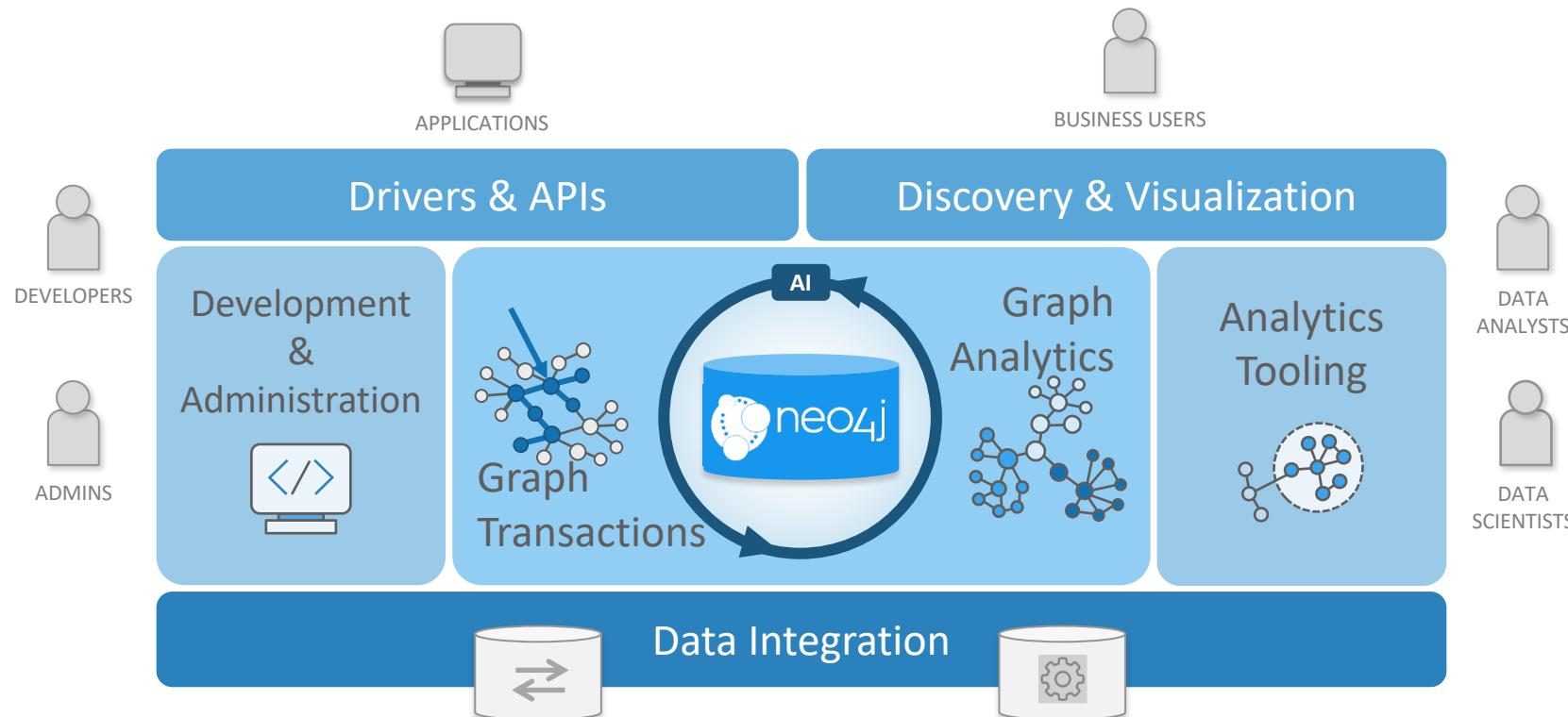
Neo4j Browser



Neo4j Desktop



The Neo4j Graph Platform Vision



Zeppelin Notebook Job Search anonymous

CAPS Multiple Graph Demo

Friendship Graph New York/SN_NA

```
val CITYFRIENDS_NA = SN_NA.cypher(
    """
    MATCH (a:Person)-[IS LOCATED_IN]->(city:City)<-[IS LOCATED_IN]->(b:Person),
    (a)-[I KNOWS*1..2]->(b)
    WHERE city.name = "New_York" OR city.name =
        "San Francisco"
    | RETURN GRAPH result OF (a)-[r:SIMILAR_CIRCLE]->(b)
    """.stripMargin).graphs("result").cache
```

CITYFRIENDS_NA: org.opencypher.caps.api.spark.CAPSGraph = CAPSPatternGraph

Nodes 37 : Person

Relationships 99 : SIMILAR_CIRCLE

node_id: 1526 node_type: Person
email: 'David.Jones.627428834534086045@yahoo.com'
firstName: 'David' lastName: 'Jones' personId: '19639'

Took 6 sec. Last updated by anonymous at October 23 2017, 8:06:23 PM. (updated)

Friendship Graph Berlin/STN_EU

```
val CITYFRIENDS_EU = SN_EU.cypher(
    """
    MATCH (a:Person)-[IS LOCATED_IN]->(city:City)<-[IS LOCATED_IN]->(b:Person),
    (a)-[I KNOWS*1..2]->(b)
    WHERE city.name = "Berlin" OR city.name =
        "San Francisco"
    | RETURN GRAPH result OF (a)-[r:SIMILAR_CIRCLE]->(b)
    """.stripMargin).graphs("result").cache
```

CITYFRIENDS_EU: org.opencypher.caps.api.spark.CAPSGraph = CAPSPatternGraph

Nodes 52 : Person

Relationships 290 : SIMILAR_CIRCLE

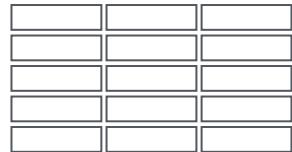
node_id: 3061 node_type: Person
email: 'Johan3298534892493@gmx.com' firstName: 'Johan'
lastName: 'Svensson' personId: '15692'

Took 3 sec. Last updated by anonymous at October 23 2017, 8:18:24 PM.

Cypher for Apache® Spark™

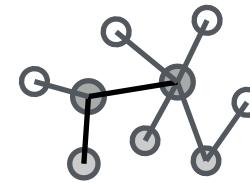


openCypher / cypher-for-apache-spark



Relational

Spark SQL

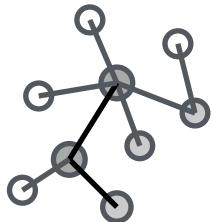


Graph

Cypher for
Apache® Spark™

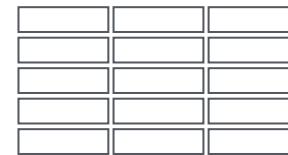


Composable Queries

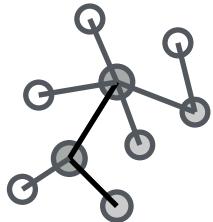


Graph

Cypher

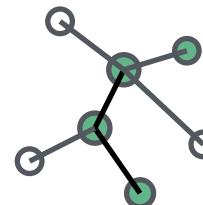


Table



Graph

Cypher



Graph

Platform Means:

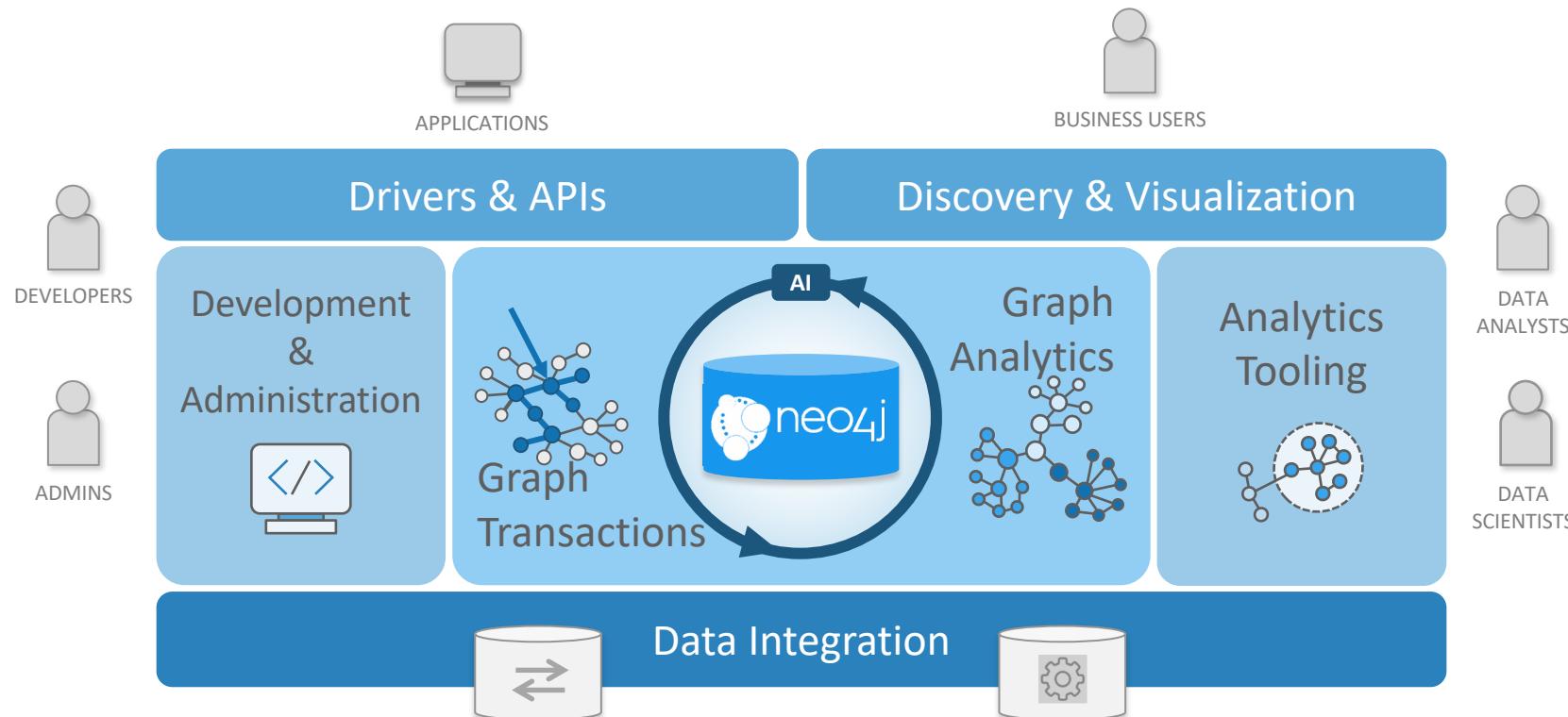
Better Access to Technology Partner Software

- Improved visibility, provisioning, and integration of partners' software

New Neo4j-Provided Capabilities

- Products and add-ons that satisfy basic Neo4j project needs across the software lifecycle

The Neo4j Graph Platform Vision



(Cypher)

-[:IS]->

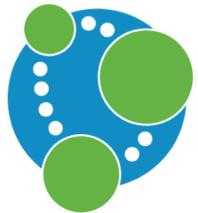
(Everywhere)

openCypher.org

Neo4j Product Update

Latest Innovations





neo4j 3.3

Latest GA Release

Recap

(Oct '17)

Neo4j 3.3 Performance Improvements

“Least Connected”
load balancing



Drivers & Bolt Protocol

Faster & more memory efficient
runtime



Cypher Engine

Batch generation of IDs
Schema operations now take local
locks



Kernel & Transactions

Page cache metadata moved off
heap
Native GB+ Tree numeric indexes



Memory Management

Bulk importer paging & memory
improvements



Storage & Indexing

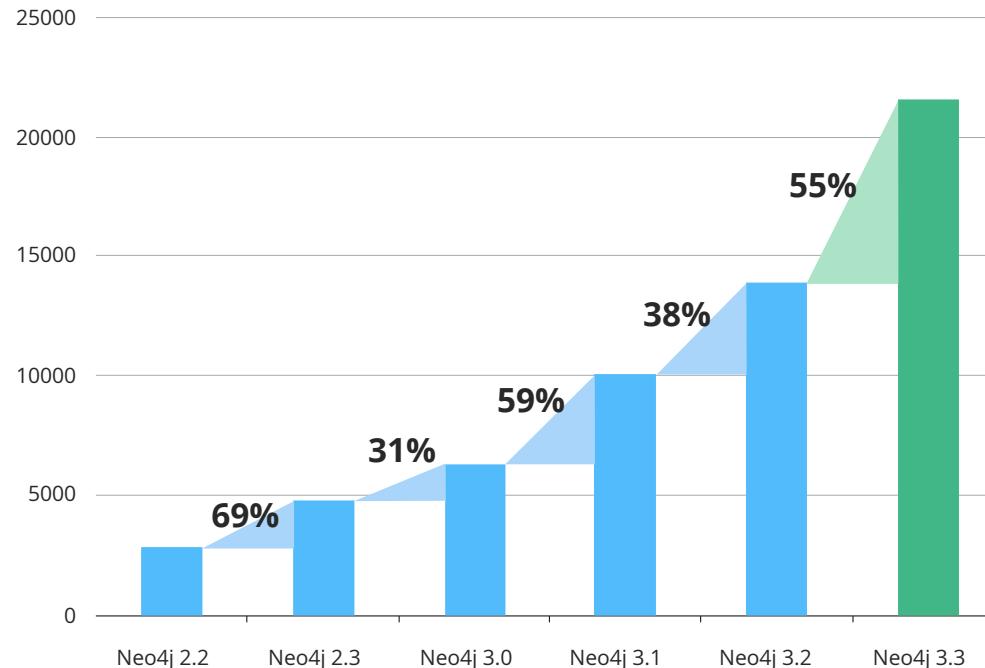
Dynamically reload config settings
without restarting Neo4j



Admin & Config

Concurrent/Transactional Write Performance

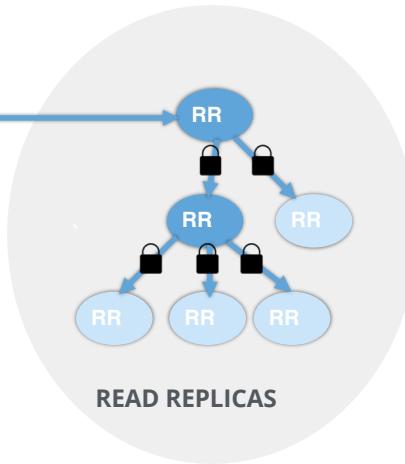
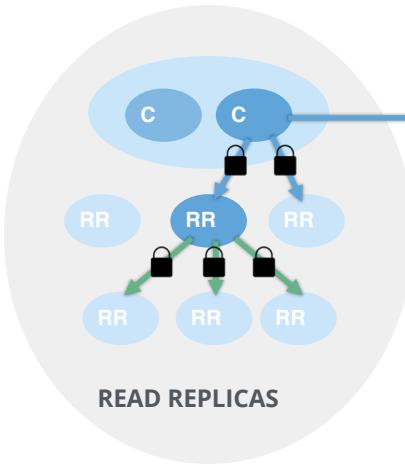
(Simulates Real-World Workloads)



Multi-Data Center Clustering



New York



London

Neo4j Security Foundation



TLS Wire
Encryption



User & Roles



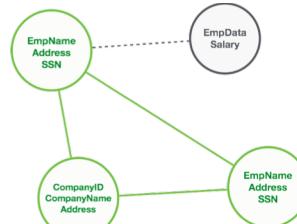
LDAP & Active Directory



Kerberos
Strong Security



Security
Event Logging



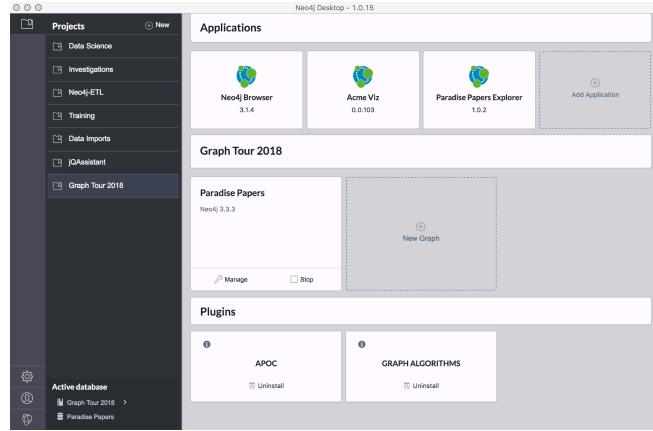
Procedure
Access Controls



Intra-Cluster
Encryption

Neo4j Desktop 1.0

- Mission control for developers
- Connect to both local and remote Neo4j servers
- Free with registration
- Includes development license for Neo4j Enterprise Edition
- Keeps you up to date with latest versions, plugins, etc.
- <https://neo4j.com/download>

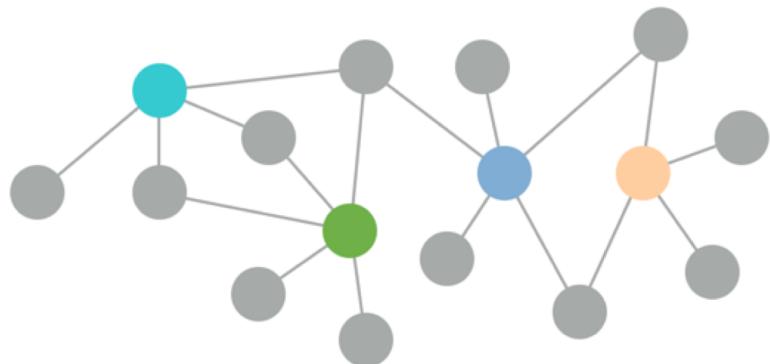


What's Next?



neo4j 3.4

Graphs



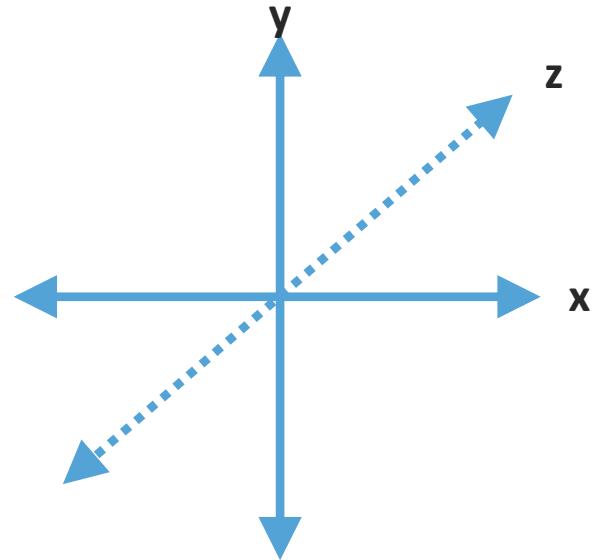
+



Geospatial Graph Queries

Query Example: “Find all coffee shops within 100m”

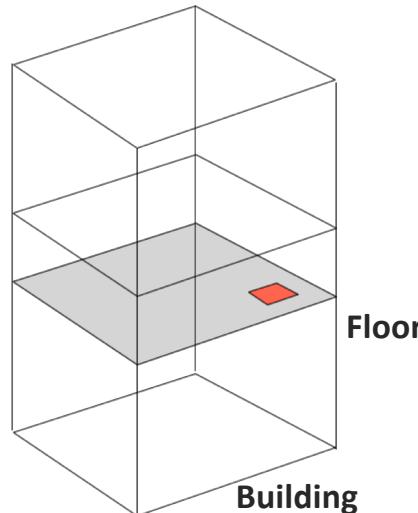
```
WITH point({latitude: 55.612149, longitude: 12.995090}) AS poi
MATCH (l:Location)<-[ :AT ]-(b:Business)-[ :OF ]->(c:Category)
WHERE c.name = "coffee"
AND distance(l.location, poi) < 1000
RETURN distance(l.location, poi) as distance, b.name as coffee_shop
ORDER BY distance DESC
```



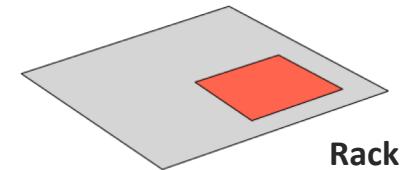
“Recommend a shirt available in a store close by in the men’s department”



Location



Floor

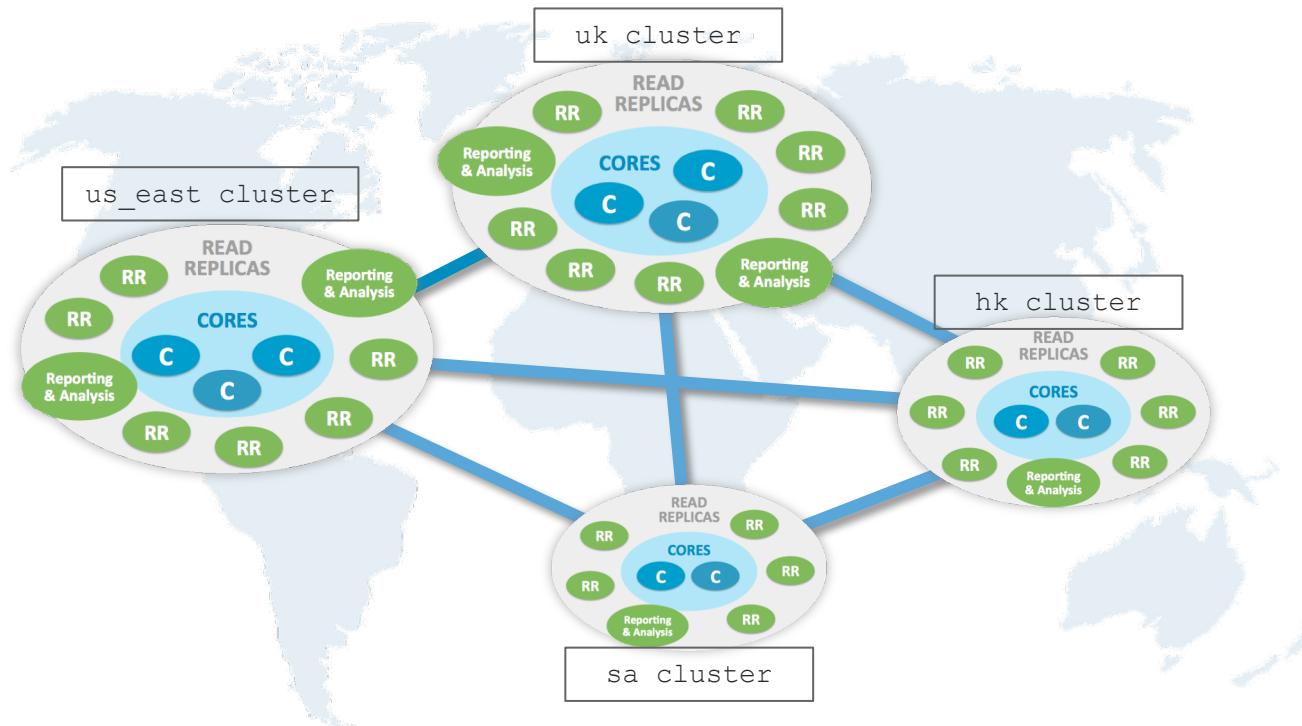


Rack

Scalability

Multi-Clustering Support for Global Internet Apps

Horizontally partition graph by domain (country, product, customer, data center)



Multi-tenancy

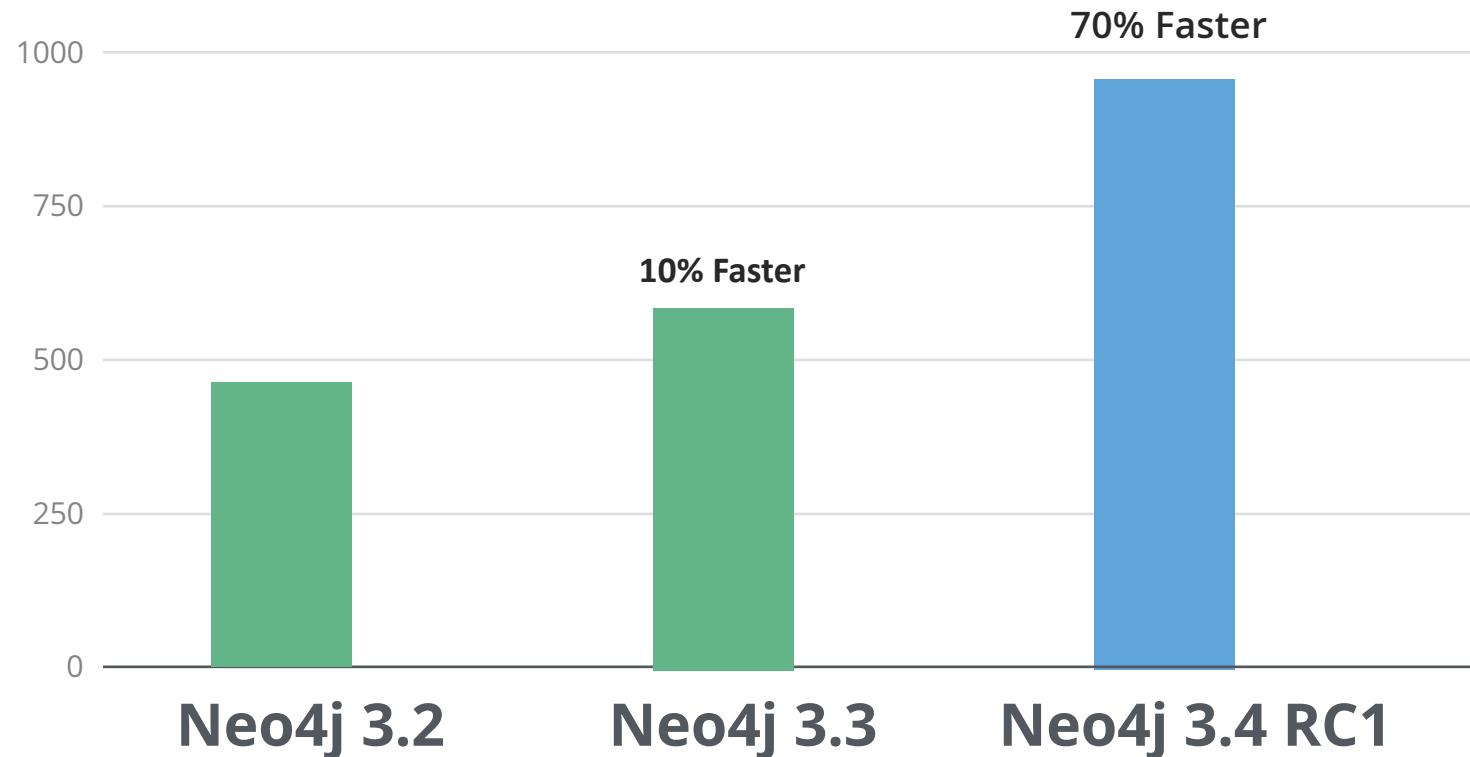
Geo Partitioning

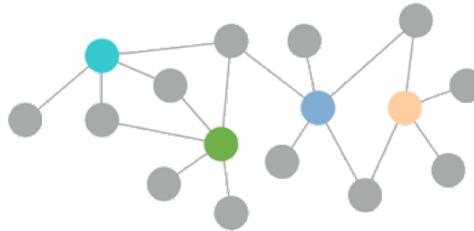
Write Scaling

Performance

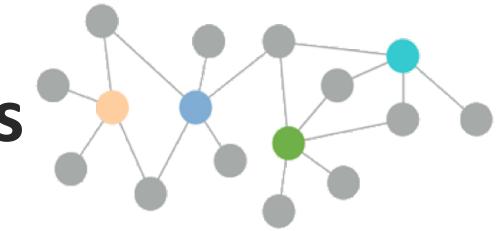
Reads – Neo4j Enterprise Cypher Runtime

Mixed Workload Read Benchmark

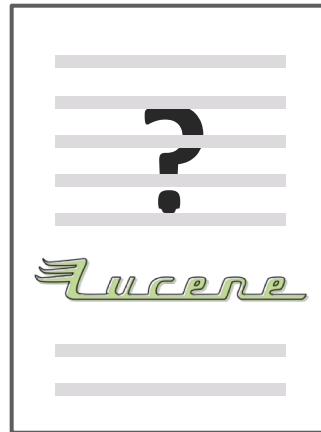




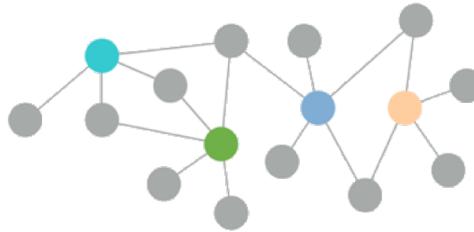
Supercharging Graph Writes



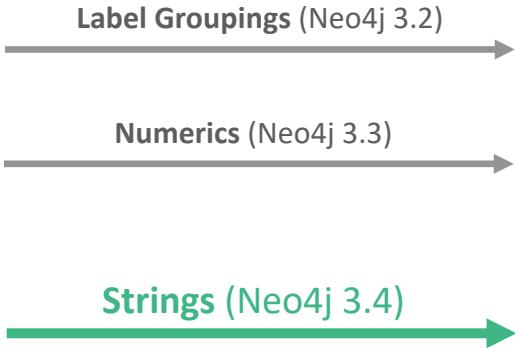
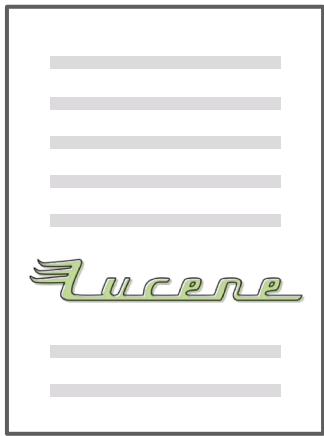
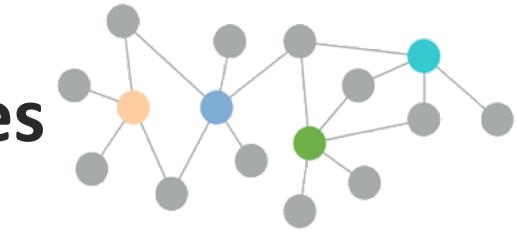
One Component:
80% of transactional write overhead (!!!)



Index Insertion



Supercharging Graph Writes



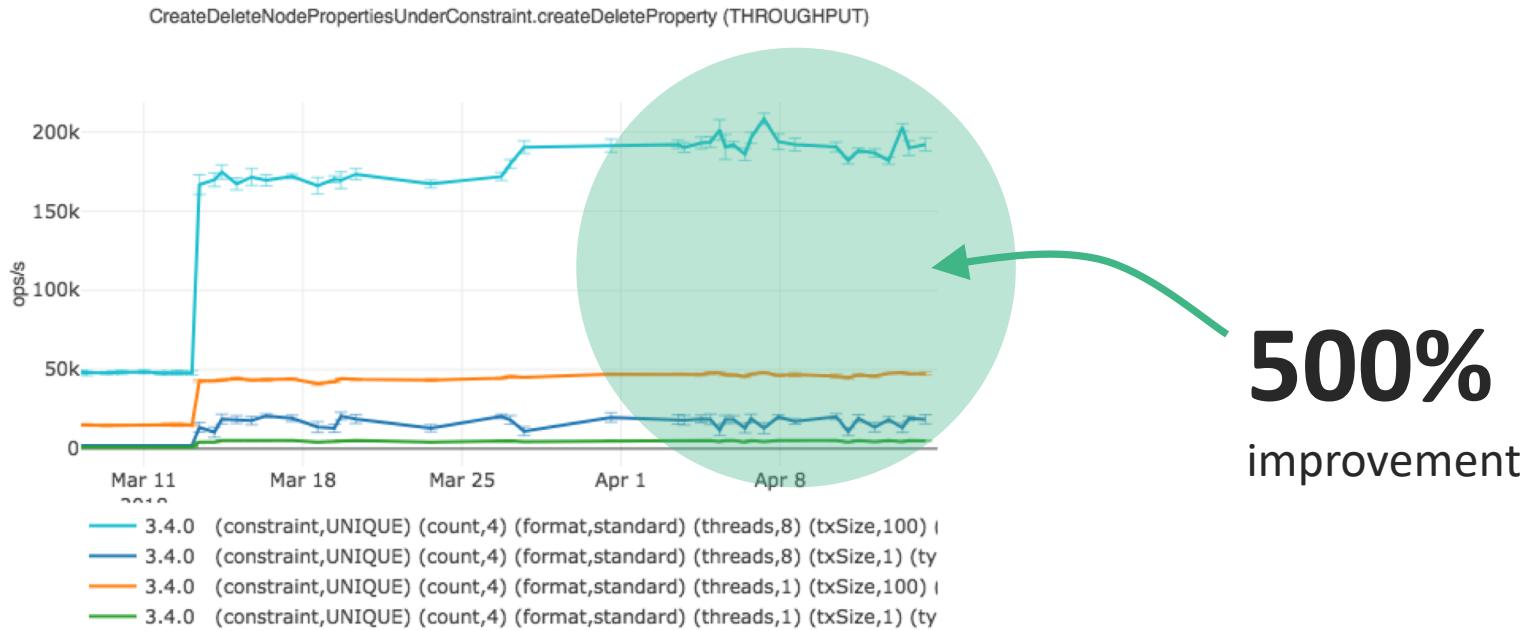
- ✓ ACID
- ✓ Optimized for graph
- ✓ Fast Reads
- ✓ ~10x Faster Writes



Writes with Native String Indexes

Neo4j 3.4 RC1

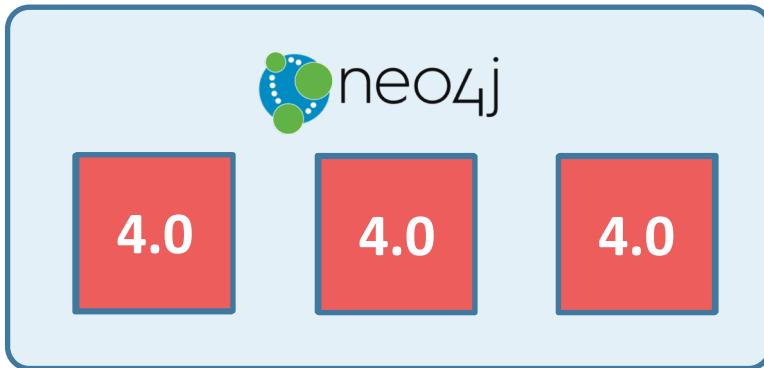
Performance



Ops & Admin

Rolling Upgrades

Available in Neo4j 3.4!

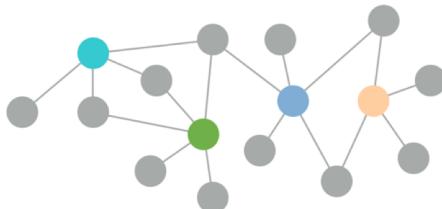


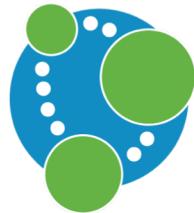
Upgrade to new versions
with zero downtime

Store upgrades
may require downtime but
can be done subsequently

Auto Cache Reheating

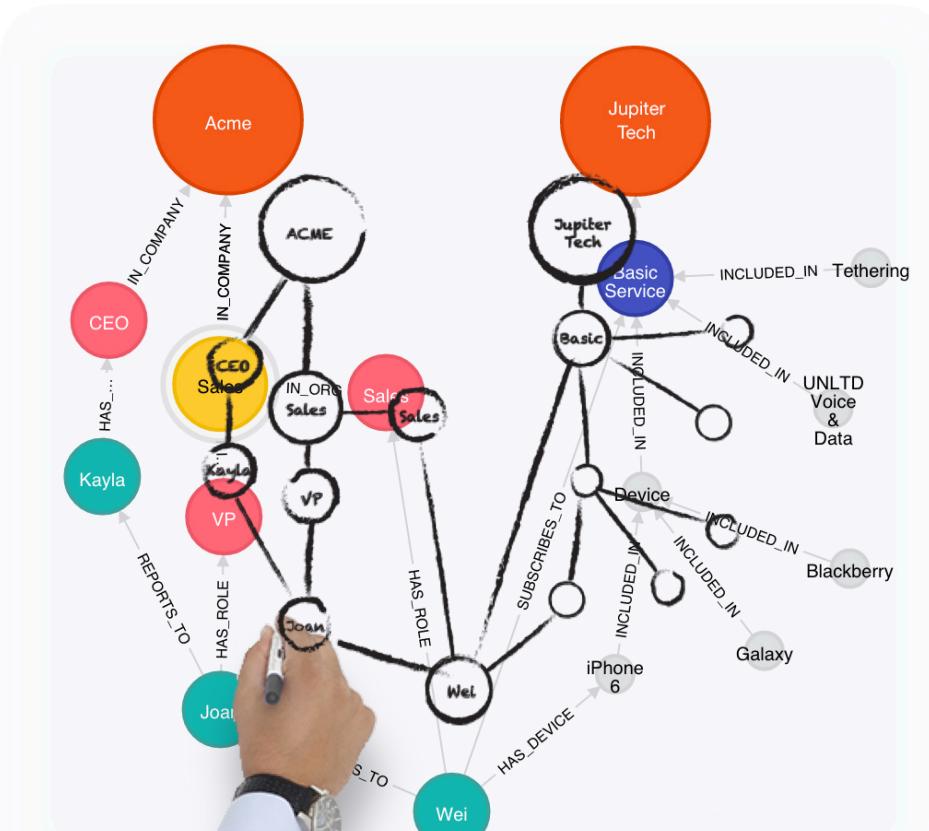
For Restarts, Restores, and Cluster Expansion

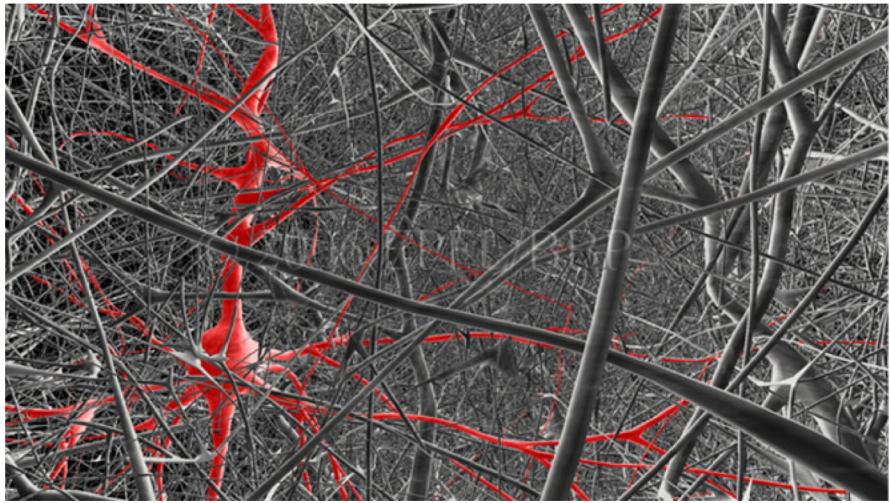
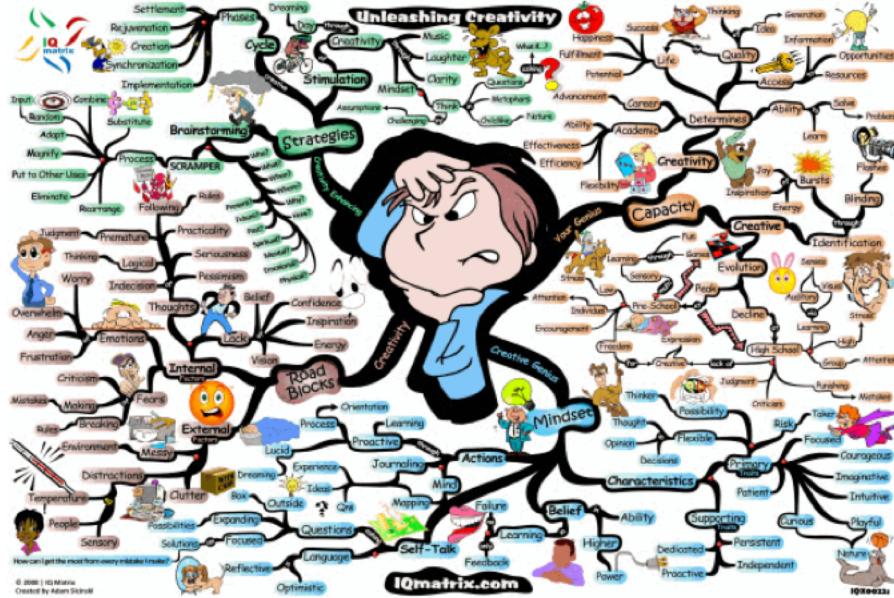




neo4j 3.4

Available This Month!





How we think

How we're wired

bolt://127.0.0.1:7687 - Neo4j Browser

```
$ MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
```

\$:play start



Learn about Neo4j

A graph epiphany awaits you.

-  What is a graph database?
- How can I query a graph?
- What do people do with Neo4j?

[Start Learning](#)

Jump into code

Use Cypher, the graph query language.

-  Code walk-throughs
- RDBMS to Graph

Monitor the system

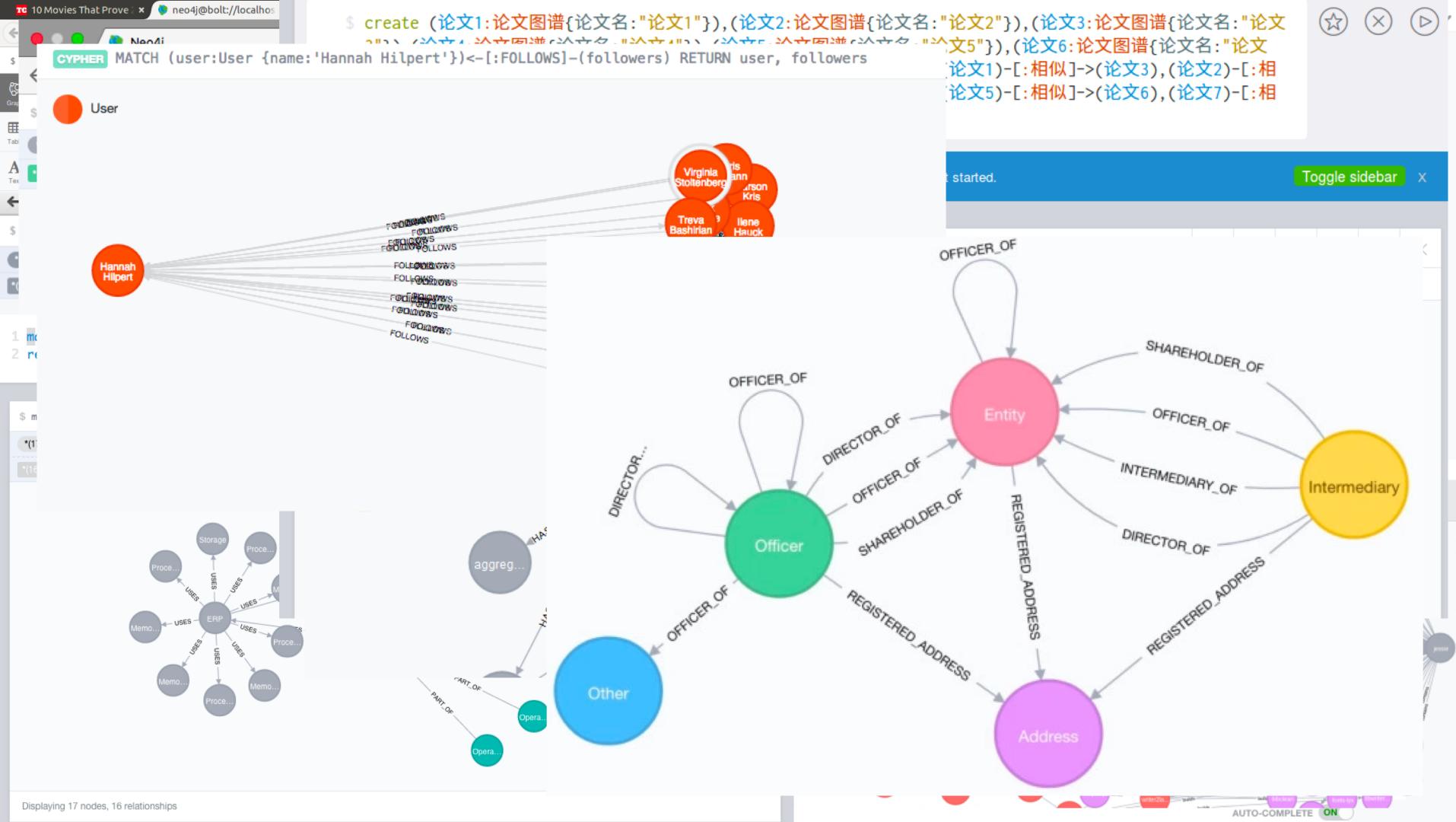
Key system health and status metrics.

-  Disk utilization
- Cache activity
- Cluster health and status

[Monitor](#)

Copyright © Neo4j, Inc 2002– 2018



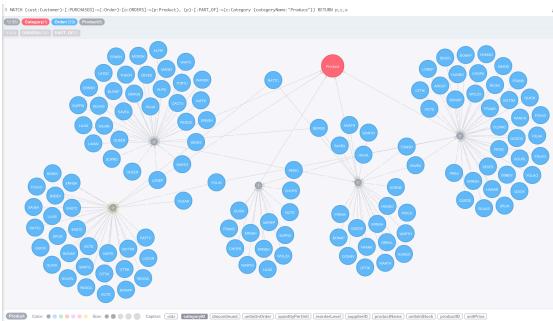


Neo4j Bloom

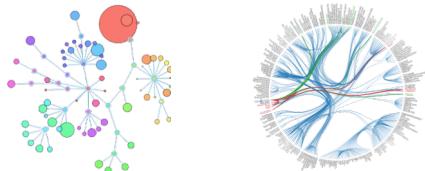
neo4j bloom Early Access Program Preview

Discovery & Visualization

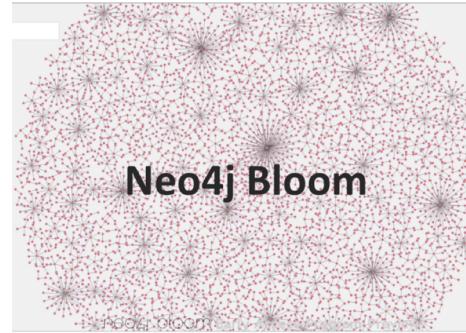
Neo4j Browser



Custom / JS Libraries



Partner Applications





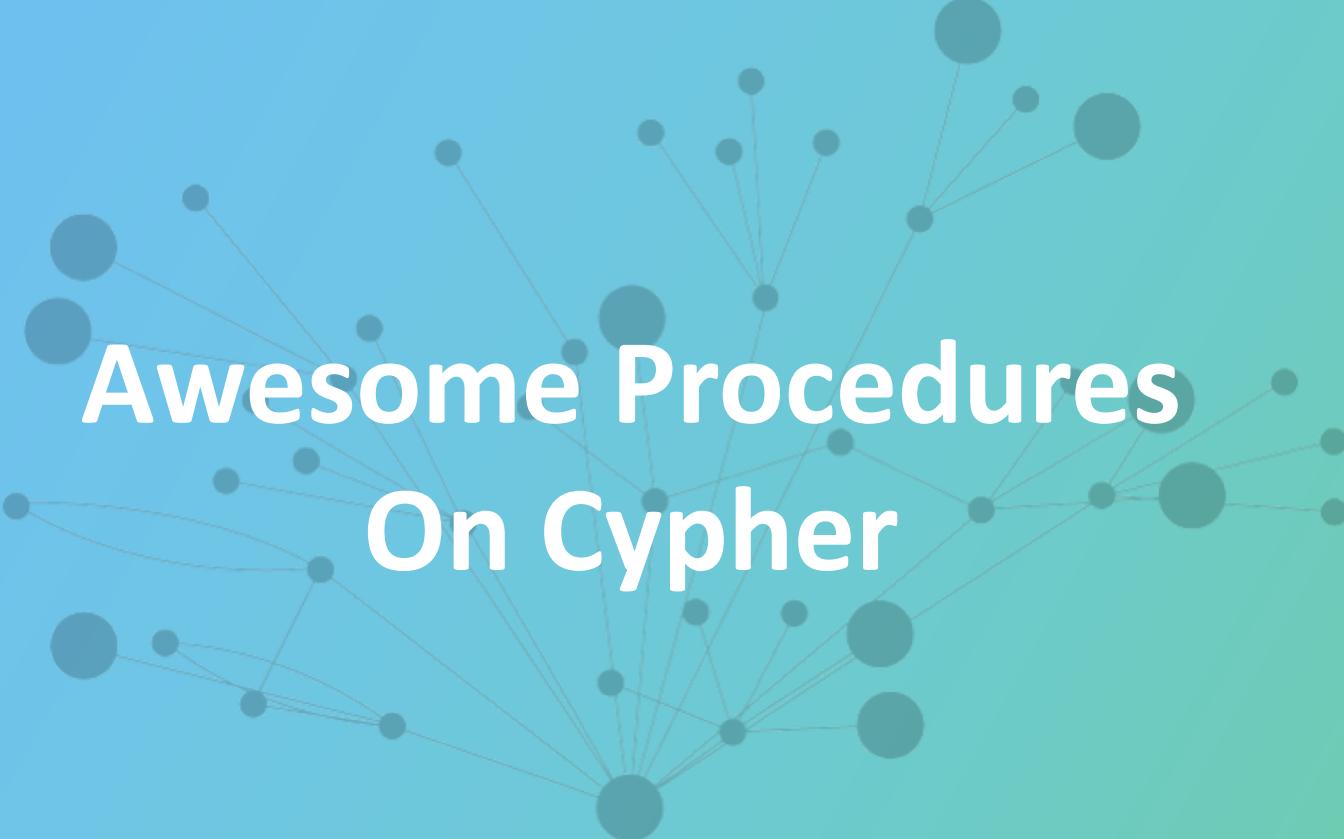
Demo Time!











Awesome Procedures On Cypher



neo4j-contrib / neo4j-apoc-procedures

 Unwatch ▾

58

 Unstar

494

 Fork

132

 Code

 Issues 195

 Pull requests 3

 Projects 1

 Insights

Awesome Procedures On Cypher for Neo4j 3.x - codenamed "apoc"
contrib.github.io/neo4j...

If you like it, please  above ↑

[https://neo4j-](https://neo4j-contrib.github.io/neo4j/)

[graph-database](#)

[graph-algorithms](#)

[stored-procedures](#)

[neo4j](#)

[neo4j-plugin](#)

 704 commits

 8 branches

 23 releases

 59 contributors

 Apache-2.0





> 400
procedures and functions

Projects

- Acme Payment Network
- Paradise Papers
- IAM
- Russian Twitter Trolls
- Credit Card Fraud
- Project

Active database
Russian T... > Database

Applications

Neo4j Browser
3.1.7

Neo4j Viz
Not installed

Cypher Fraud Detector
1.0.0

Neo4j ETL
0.0.7

Payments Dashboard
2.4.0

Acme Payment Network

Transaction Database

Neo4j 3.2.6

Manage Start

New Graph

Plugins

APOC

Install

GRAPH ALGORITHMS

Install

GraphQL

Install

Projects

- Acme Payment Network
- Paradise Papers
- IAM
- Russian Twitter Trolls
- Credit Card Fraud
- Project

Active database

Russian T... > Database

Applications



Neo4j Browser

3.1.7



Neo4j Viz

Not installed



Cypher Fraud Detector

1.0.0



Neo4j ETL

0.0.7



Payments Dashboard

2.4.0

Acme Payment Network

Transaction Database

Neo4j 3.2.6

New Graph

Manage

Start

Plugins



APOC

Install



GRAPH ALGORITHMS

Install



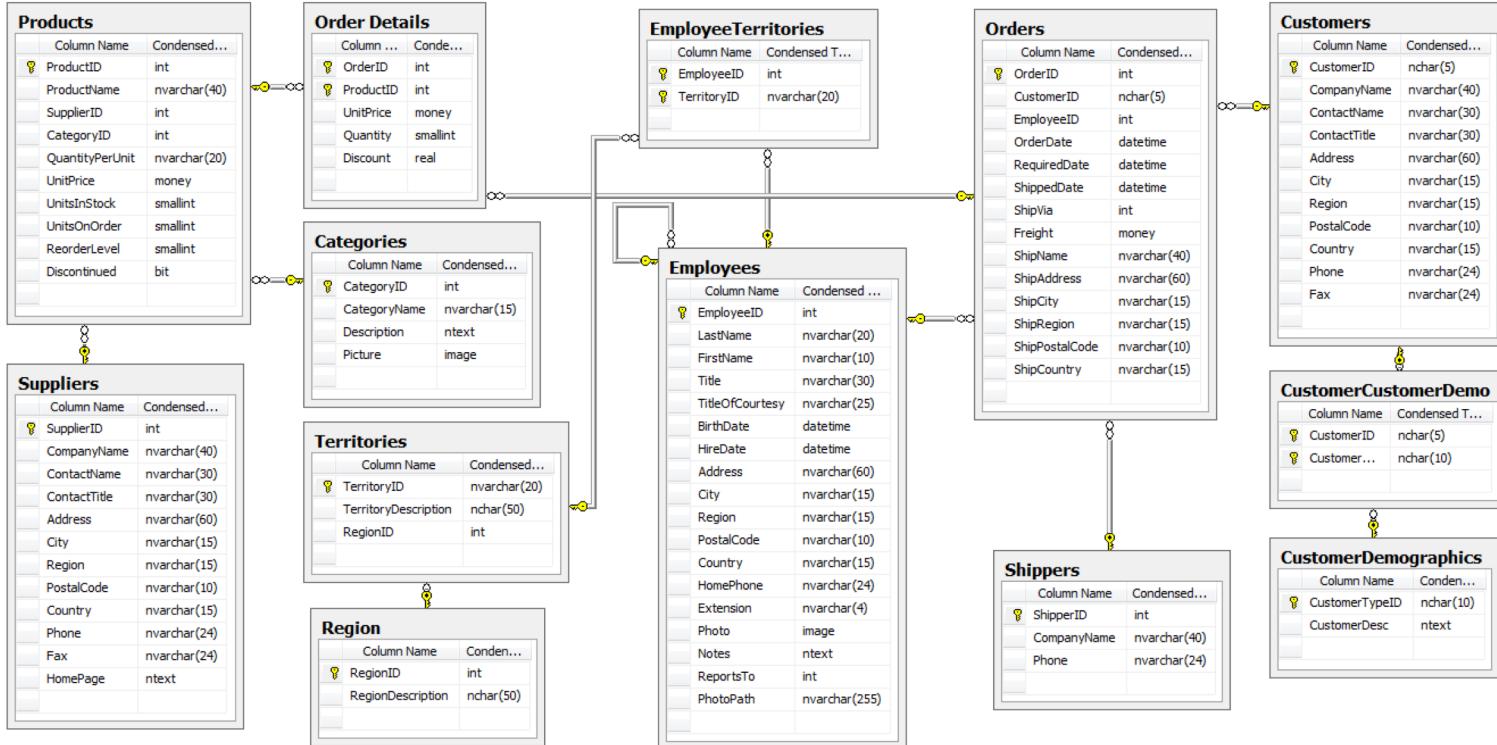
GraphQL

Install



Data Integration

Load from a relational database



apoc.load.jdbc

```
WITH "jdbc:mysql://localhost:3306/northwind?user=root" AS url
CALL apoc.load.jdbc(url, "products")
YIELD row
MERGE (p:Product {id: row.ProductID})
SET p.name = row.ProductName, p.unitPrice = row.UnitPrice
```



Execute procedure

```
WITH "jdbc:mysql://localhost:3306/northwind?user=root" AS url  
CALL apoc.load.jdbc(url,"products")  
YIELD row  
  
MERGE (p:Product {id: row.ProductID})  
SET p.name = row.ProductName, p.unitPrice = row.UnitPrice
```



Apply Cypher transformation

```
WITH "jdbc:mysql://localhost:3306/northwind?user=root" AS url  
CALL apoc.load.jdbc(url,"products")  
YIELD row  
MERGE (p:Product {id: row.ProductID})  
SET p.name = row.ProductName, p.unitPrice = row.UnitPrice
```



Load XML

```
▼<osm version="0.6" generator="Overpass API 0.7.54.12 054bb0bb">
  <meta osm_base="2018-02-07T10:05:03Z"/>
  ▼<node id="398692" lat="48.1452196" lon="11.5414971" version="20" timestamp="2015-10-15T10:53:28Z" changeset="34651972" uid="2290263" user="soemisch">
    <tag k="tmc" v="DE:35375"/>
  </node>
  ▼<node id="398694" lat="48.1451404" lon="11.5411141" version="19" timestamp="2016-01-04T08:57:34Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="TMC:cid_58:abcd_1:Class" v="Point"/>
    <tag k="TMC:cid_58:abcd_1:Direction" v="positive"/>
    <tag k="TMC:cid_58:abcd_1:LCLversion" v="9.00"/>
    <tag k="TMC:cid_58:abcd_1:LocationCode" v="35355"/>
    <tag k="TMC:cid_58:abcd_1:NextLocationCode" v="35356"/>
    <tag k="TMC:cid_58:abcd_1:PrevLocationCode" v="27974"/>
    <tag k="highway" v="traffic_signals"/>
  </node>
  ▼<node id="1956100" lat="48.1434822" lon="11.5487963" version="43" timestamp="2015-10-27T14:01:37Z" changeset="34904180" uid="2385132" user="MENTZ_TU">
    <tag k="TMC:cid_58:abcd_1:Class" v="Point"/>
    <tag k="TMC:cid_58:abcd_1:Direction" v="positive"/>
    <tag k="TMC:cid_58:abcd_1:LCLversion" v="9.00"/>
    <tag k="TMC:cid_58:abcd_1:LocationCode" v="35356"/>
    <tag k="TMC:cid_58:abcd_1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_58:abcd_1:PrevLocationCode" v="35355"/>
    <tag k="tmc" v="DE:61453"/>
  </node>
  <node id="19404292" lat="48.1442786" lon="11.5454386" version="16" timestamp="2015-10-15T11:05:47Z" changeset="34652188" uid="2290263" user="soemisch"/>
  ▼<node id="21324005" lat="48.1436171" lon="11.5488362" version="51" timestamp="2015-10-27T14:01:38Z" changeset="34904180" uid="2385132" user="MENTZ_TU">
    <tag k="TMC:cid_58:abcd_1:Class" v="Point"/>
    <tag k="TMC:cid_58:abcd_1:Direction" v="negative"/>
    <tag k="TMC:cid_58:abcd_1:LCLversion" v="9.00"/>
    <tag k="TMC:cid_58:abcd_1:LocationCode" v="35356"/>
    <tag k="TMC:cid_58:abcd_1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_58:abcd_1:PrevLocationCode" v="35355"/>
  </node>
  ▼<node id="21324376" lat="48.1451843" lon="11.5416781" version="54" timestamp="2016-01-04T08:57:33Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="bicycle" v="yes"/>
    <tag k="crossing" v="traffic_signals"/>
    <tag k="crossing_ref" v="toucan"/>
    <tag k="highway" v="crossing"/>
  </node>
```

apoc.load.xml

```
CALL apoc.load.xml('http://overpass.osm.rambler.ru/cgi/xapi_meta?*[bbox=11.54,48.14,11.543,48.145]')
YIELD value
UNWIND value["_children"] AS child

WITH child WHERE child["_type"] = "node"
WITH child.id AS id,
    child.lat AS latitude,
    child.lon AS longitude,
    child["user"] AS userName

MERGE (point:Point {id: id})
SET point.latitude = latitude,
    point.longitude = longitude
MERGE (user:User {name: userName})
MERGE (user)-[:EDITED]->(point)

<osm version="0.6" generator="Overpass API 0.7.54.12 054bb0bb">
  <meta osm_base="2018-02-07T10:05:03Z"/>
  <node id="398692" lat="48.1452196" lon="11.5414971" version="20" timestamp="2015-10-15T10:53:28Z" changeset="34651972" uid="2290263" user="soemisch">
    <tag k="tmc" v="DE:35375"/>
  </node>
  <node id="398694" lat="48.1451404" lon="11.5411141" version="19" timestamp="2016-01-04T08:57:34Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="TMC:cid_58:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_58:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_58:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_58:tabcd_l1:LocationCode" v="35355"/>
    <tag k="TMC:cid_58:tabcd_l1:NextLocationCode" v="35356"/>
    <tag k="TMC:cid_58:tabcd_l1:PrevLocationCode" v="27974"/>
    <tag k="highway" v="traffic_signals"/>
  </node>
  <node id="1956100" lat="48.1434822" lon="11.5487963" version="43" timestamp="2015-10-27T14:01:37Z" changeset="34904180" uid="2385132" user="MENTE_TU">
    <tag k="TMC:cid_58:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_58:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_58:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_58:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_58:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_58:tabcd_l1:PrevLocationCode" v="35355"/>
    <tag k="tmc" v="DE:61453"/>
  </node>
  <node id="19404292" lat="48.1442786" lon="11.5454386" version="16" timestamp="2015-10-15T11:05:47Z" changeset="34652188" uid="2290263" user="soemisch">
    <tag k="TMC:cid_59:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_59:tabcd_l1:Direction" v="negative"/>
    <tag k="TMC:cid_59:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_59:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_59:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_59:tabcd_l1:PrevLocationCode" v="35355"/>
  </node>
  <node id="21324005" lat="48.1436171" lon="11.5488362" version="51" timestamp="2015-10-27T14:01:38Z" changeset="34904180" uid="2385132" user="MENTE_TU">
    <tag k="TMC:cid_59:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_59:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_59:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_59:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_59:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_59:tabcd_l1:PrevLocationCode" v="35355"/>
  </node>
  <node id="21324376" lat="48.1451843" lon="11.5416781" version="54" timestamp="2016-01-04T08:57:33Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="bicycle" v="yes"/>
    <tag k="crossing" v="traffic_signals"/>
    <tag k="crossing_ref" v="toucan"/>
    <tag k="highway" v="crossing"/>
  </node>
```



Execute procedure

```
CALL apoc.load.xml('http://overpass.osm.rambler.ru/cgi/xapi_meta?*[bbox=11.54,48.14,11.543,48.145]')
YIELD value

UNWIND value["_children"] AS child

WITH child WHERE child["_type"] = "node"
WITH child.id AS id,
    child.lat AS latitude,
    child.lon AS longitude,
    child["user"] AS userName

MERGE (point:Point {id: id})
SET point.latitude = latitude,
    point.longitude = longitude
MERGE (user:User {name: userName})
MERGE (user)-[:EDITED]->(point)

<osm version="0.6" generator="Overpass API 0.7.54.12 054bb0bb">
  <meta osm_base="2018-02-07T10:05:03Z"/>
  <node id="398692" lat="48.1452196" lon="11.5414971" version="20" timestamp="2015-10-15T10:53:28Z" changeset="34651972" uid="2290263" user="soemisch">
    <tag k="tmc" v="DE:35375"/>
  </node>
  <node id="398694" lat="48.1451404" lon="11.5411141" version="19" timestamp="2016-01-04T08:57:34Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="TMC:cid_58:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_58:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_58:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_58:tabcd_l1:LocationCode" v="35355"/>
    <tag k="TMC:cid_58:tabcd_l1:NextLocationCode" v="35356"/>
    <tag k="TMC:cid_58:tabcd_l1:PrevLocationCode" v="35294"/>
    <tag k="highway" v="traffic_signals"/>
  </node>
  <node id="1956100" lat="48.1434822" lon="11.5487963" version="43" timestamp="2015-10-27T14:01:37Z" changeset="34904180" uid="2385132" user="MENTE_TU">
    <tag k="TMC:cid_58:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_58:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_58:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_58:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_58:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_58:tabcd_l1:PrevLocationCode" v="35355"/>
    <tag k="tmc" v="DE:614537"/>
  </node>
  <node id="19404292" lat="48.1442786" lon="11.5454386" version="16" timestamp="2015-10-15T11:05:47Z" changeset="34652188" uid="2290263" user="soemisch"/>
  <node id="21324005" lat="48.1436171" lon="11.5488362" version="51" timestamp="2015-10-27T14:01:38Z" changeset="34904180" uid="2385132" user="MENTE_TU">
    <tag k="TMC:cid_59:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_59:tabcd_l1:Direction" v="negative"/>
    <tag k="TMC:cid_59:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_59:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_59:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_59:tabcd_l1:PrevLocationCode" v="35355"/>
  </node>
  <node id="21324376" lat="48.1451843" lon="11.5416781" version="54" timestamp="2016-01-04T08:57:33Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="bicycle" v="yes"/>
    <tag k="crossing" v="traffic_signals"/>
    <tag k="crossing_ref" v="toucan"/>
    <tag k="highway" v="crossing"/>
  </node>
```



UNWIND the array of elements

```
CALL apoc.load.xml('http://overpass.osm.rambler.ru/cgi/xapi_meta?*[bbox=11.54,48.14,11.543,48.145]')
YIELD value
UNWIND value[_children] AS child

WITH child WHERE child["_type"] = "node"
WITH child.id AS id,
    child.lat AS latitude,
    child.lon AS longitude,
    child["user"] AS userName

MERGE (point:Point {id: id})
SET point.latitude = latitude,
    point.longitude = longitude
MERGE (user:User {name: userName})
MERGE (user)-[:EDITED]->(point)

<osm version="0.6" generator="Overpass API 0.7.54.12 054bb0bb">
  <meta osm_base="2018-02-07T10:05:03Z"/>
  <node id="398692" lat="48.1452196" lon="11.5414971" version="20" timestamp="2015-10-15T10:53:28Z" changeset="34651972" uid="2290263" user="soemisch">
    <tag k="tmc" v="DE:35375"/>
  </node>
  <node id="398694" lat="48.1451404" lon="11.5411141" version="19" timestamp="2016-01-04T08:57:34Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="TMC:cid_58:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_58:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_58:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_58:tabcd_l1:LocationCode" v="35355"/>
    <tag k="TMC:cid_58:tabcd_l1:NextLocationCode" v="35356"/>
    <tag k="TMC:cid_58:tabcd_l1:PrevLocationCode" v="27974"/>
    <tag k="highway" v="traffic_signals"/>
  </node>
  <node id="1956100" lat="48.1434822" lon="11.5487963" version="43" timestamp="2015-10-27T14:01:37Z" changeset="34904180" uid="2385132" user="MENTE_TU">
    <tag k="TMC:cid_58:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_58:tabcd_l1:Direction" v="positive"/>
    <tag k="TMC:cid_58:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_58:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_58:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_58:tabcd_l1:PrevLocationCode" v="35355"/>
    <tag k="tmc" v="DE:614537"/>
  </node>
  <node id="19404292" lat="48.1442786" lon="11.5454386" version="16" timestamp="2015-10-15T11:05:47Z" changeset="34652188" uid="2290263" user="soemisch"/>
  <node id="21324005" lat="48.1436171" lon="11.5488362" version="51" timestamp="2015-10-27T14:01:38Z" changeset="34904180" uid="2385132" user="MENTE_TU">
    <tag k="TMC:cid_59:tabcd_l1:Class" v="Point"/>
    <tag k="TMC:cid_59:tabcd_l1:Direction" v="negative"/>
    <tag k="TMC:cid_59:tabcd_l1:ClVersion" v="9.00"/>
    <tag k="TMC:cid_59:tabcd_l1:LocationCode" v="35356"/>
    <tag k="TMC:cid_59:tabcd_l1:NextLocationCode" v="35357"/>
    <tag k="TMC:cid_59:tabcd_l1:PrevLocationCode" v="35355"/>
  </node>
  <node id="21324376" lat="48.1451843" lon="11.5416781" version="54" timestamp="2016-01-04T08:57:33Z" changeset="36354525" uid="354141" user="Anoniman">
    <tag k="bicycle" v="yes"/>
    <tag k="crossing" v="traffic_signals"/>
    <tag k="crossing_ref" v="toucan"/>
    <tag k="highway" v="crossing"/>
  </node>
```



Filter rows

```
CALL apoc.load.xml('http://overpass.osm.rambler.ru/cgi/xapi_meta?*[bbox=11.54,48.14,11.543,48.145]')
YIELD value
UNWIND value[_children] AS child

WITH child WHERE child[_type] = "node"
WITH child.id AS id,
     child.lat AS latitude,
     child.lon AS longitude,
     child["user"] AS userName

MERGE (point:Point {id: id})
SET point.latitude = latitude,
    point.longitude = longitude
MERGE (user:User {name: userName})
MERGE (user)-[:EDITED]->(point)

WITH child WHERE child[_type] = "node"
WITH child.id AS id,
     child.lat AS latitude,
     child.lon AS longitude,
     child["user"] AS userName

MERGE (point:Point {id: id})
SET point.latitude = latitude,
    point.longitude = longitude
MERGE (user:User {name: userName})
MERGE (user)-[:EDITED]->(point)
```

Apply Cypher transformation

```
CALL apoc.load.xml('http://overpass.osm.rambler.ru/cgi/xapi_meta?*[bbox=11.54,48.14,11.543,48.145]')
YIELD value
UNWIND value["_children"] AS child

WITH child WHERE child["_type"] = "node"
WITH child.id AS id,
    child.lat AS latitude,
    child.lon AS longitude,
    child["user"] AS userName

    <osm version="0.6" generator="Overpass API 0.7.54.12 054bb0bb">
        <meta osm_base="2018-02-07T10:05:03Z"/>
        <node id="398692" lat="48.1452196" lon="11.5414971" version="20" timestamp="2015-10-15T10:53:28Z" changeset="34651972" uid="2290263" user="soemisch">
            <tag k="tmc" v="DE:35375"/>
        </node>
        <node id="398694" lat="48.1451404" lon="11.5411141" version="19" timestamp="2016-01-04T08:57:34Z" changeset="36354525" uid="354141" user="Anoniman">
            <tag k="TMC:cid_58:tabcd_l1Class" v="Point"/>
            <tag k="TMC:cid_58:tabcd_l1Direction" v="positive"/>
            <tag k="TMC:cid_58:tabcd_l1LCversion" v="9.00"/>
            <tag k="TMC:cid_58:tabcd_l1LocationCode" v="35355"/>
            <tag k="TMC:cid_58:tabcd_l1NextLocationCode" v="35356"/>
            <tag k="TMC:cid_58:tabcd_l1PrevLocationCode" v="27974"/>
            <tag k="highway" v="traffic_signals"/>
        </node>
        <node id="1956100" lat="48.1434822" lon="11.5487963" version="43" timestamp="2015-10-27T14:01:37Z" changeset="34904180" uid="2385132" user="MENTE_TU">
            <tag k="TMC:cid_58:tabcd_l1Class" v="Point"/>
            <tag k="TMC:cid_58:tabcd_l1Direction" v="positive"/>
            <tag k="TMC:cid_58:tabcd_l1LCversion" v="9.00"/>
            <tag k="TMC:cid_58:tabcd_l1LocationCode" v="35356"/>
            <tag k="TMC:cid_58:tabcd_l1NextLocationCode" v="35357"/>
            <tag k="TMC:cid_58:tabcd_l1PrevLocationCode" v="35355"/>
            <tag k="highway" v="traffic_signals"/>
        </node>
        <node id="4537" lat="48.1442786" lon="11.5454386" version="16" timestamp="2015-10-15T11:05:47Z" changeset="34652188" uid="2290263" user="soemisch"/>
        <node id="48.1436171" lon="11.5488362" version="51" timestamp="2015-10-27T14:01:38Z" changeset="34904180" uid="2385132" user="MENTE_TU">
            <tag k="TMC:cid_58:tabcd_l1Class" v="Point"/>
            <tag k="TMC:cid_58:tabcd_l1Direction" v="negative"/>
            <tag k="TMC:cid_58:tabcd_l1LCversion" v="9.00"/>
            <tag k="TMC:cid_58:tabcd_l1LocationCode" v="35356"/>
            <tag k="TMC:cid_58:tabcd_l1NextLocationCode" v="35357"/>
            <tag k="TMC:cid_58:tabcd_l1PrevLocationCode" v="35355"/>
            <tag k="highway" v="traffic_signals"/>
        </node>
        <node id="48.1451843" lon="11.5416781" version="54" timestamp="2016-01-04T08:57:33Z" changeset="36354525" uid="354141" user="Anoniman">
            <tag k="TMC:cid_58:tabcd_l1Class" v="Point"/>
            <tag k="TMC:cid_58:tabcd_l1Direction" v="positive"/>
            <tag k="TMC:cid_58:tabcd_l1LCversion" v="9.00"/>
            <tag k="TMC:cid_58:tabcd_l1LocationCode" v="35356"/>
            <tag k="TMC:cid_58:tabcd_l1NextLocationCode" v="35357"/>
            <tag k="TMC:cid_58:tabcd_l1PrevLocationCode" v="35355"/>
            <tag k="highway" v="traffic_signals"/>
        </node>
    
```

MERGE (point:Point {id: id})
SET point.latitude = latitude,
 point.longitude = longitude
MERGE (user:User {name: userName})
MERGE (user)-[:EDITED]->(point)

Load JSON

```
{  
  - items: [  
    - {  
      - tags: [  
        "neo4j",  
        "remote-access"  
      ],  
      - comments: [  
        - {  
          - owner: {  
            reputation: 2332,  
            user_id: 2286415,  
            user_type: "registered",  
            profile_image: "https://i.stack.imgur.com/tbRoS.png?s=128&g=1",  
            display_name: "logisima",  
            link: "https://stackoverflow.com/users/2286415/logisima"  
          },  
          edited: false,  
          score: 0,  
          creation_date: 1518014325,  
          post_id: 48665959,  
          comment_id: 84330430,  
          link: "https://stackoverflow.com/questions/48665959/running-neo4j-from-a-remote-machine#comment84330430\_48665959"  
        },  
        - {  
          - owner: {  
            reputation: 100,  
            user_id: 1268630,  
            user_type: "registered",  
            accept_rate: 50,  
            profile_image: "https://www.gravatar.com/avatar/6e53bbcecb67621195a85441df97f8d9?s=128&d=identicon&r=PG",  
            display_name: "ErEcTuS",  
            link: "https://stackoverflow.com/users/1268630/erectus"  
          },  
          reply_to_user: {  
            reputation: 2332,  
            user_id: 2286415  
          }  
        }  
      ]  
    }  
  ]  
}
```



apoc.load.json

```
WITH "https://api.stackexchange.com/2.2/questions?pagesize=100&order=desc&sort=creation&tagged=neo4j&site=stackoverflow&filter=!5-i6Zw8Y)4W7vpy91PMYsKM-k9yzEsSC1_Ux1f" AS url  
CALL apoc.load.json(url) YIELD value
```

```
UNWIND value.items AS q
```

```
MERGE (question:Question {id:q.question_id})  
ON CREATE SET question.title = q.title,  
question.share_link = q.share_link,  
question.favorite_count = q.favorite_count
```

```
MERGE (owner:User {id:q.owner.user_id})  
ON CREATE SET owner.display_name = q.owner.display_name
```

```
MERGE (owner)-[:ASKED]->(question)
```

```
FOREACH (tagName IN q.tags |  
MERGE (tag:Tag {name:tagName}) MERGE (question)-[:TAGGED]->(tag))
```

```
FOREACH (a IN q.answers |  
MERGE (question)<-[:ANSWERS]-(answer:Answer {id:a.answer_id})  
MERGE (answerer:User {id:a.owner.user_id})  
ON CREATE SET answerer.display_name = a.owner.display_name  
MERGE (answer)<-[: PROVIDED]-(answerer)  
)
```

```
{  
  - items: [  
    - {  
      - tags: [  
        "neo4j",  
        "remote-access"  
      ],  
      - comments: [  
        - {  
          - owner: {  
              reputation: 2332,  
              user_id: 2286415,  
              user_type: "registered",  
              profile_image: "https://i.stack.imgur.com/tbRo8.png?e=128&q=1",  
              display_name: "logisima",  
              link: "https://stackoverflow.com/users/2286415/logisima"  
            },  
          edited: false,  
          score: 0,  
          creation_date: 1518014325,  
          post_id: 48665959,  
          comment_id: 94330430,  
          link: "https://stackoverflow.com/questions/48665959/running-neo4j-from-a-remote-machine#comment84330430_48665959"  
        },  
        - {  
          - owner: {  
              reputation: 100,  
              user_id: 1268630,  
              user_type: "registered",  
              accept_rate: 50,  
              profile_image: "https://www.gravatar.com/avatar/6e53bbcecb67621195a85441df97f8d97a=128&d=identicon&r=PG",  
              display_name: "Ercetin",  
              link: "https://stackoverflow.com/users/1268630/ercetin"  
            },  
          reply_to_user: {  
            reputation: 2332,  
            user_id: 2286415,  
            user_type: "registered",  
            profile_image: "https://i.stack.imgur.com/tbRo8.png?e=128&q=1",  
            display_name: "logisima",  
            link: "https://stackoverflow.com/users/2286415/logisima"  
          },  
          edited: false,  
          score: 0,  
          creation_date: 1518017234,  
          post_id: 48665959,  
          comment_id: 94332534,  
          link: "https://stackoverflow.com/questions/48665959/running-neo4j-from-a-remote-machine#comment84332534_48665959"  
        },  
        - {  
          - owner: {  
              reputation: 100,  
              user_id: 1268630,  
              user_type: "registered",  
              accept_rate: 50,  
              profile_image: "https://www.gravatar.com/avatar/6e53bbcecb67621195a85441df97f8d97a=128&d=identicon&r=PG",  
              display_name: "Ercetin",  
              link: "https://stackoverflow.com/users/1268630/ercetin"  
            },  
          last_editor: {  
            reputation: 100,  
            user_id: 1268630,  
            link: "https://stackoverflow.com/users/1268630/ercetin"  
          }  
      ]  
    }  
  ]  
}
```



Use FOREACH for arrays within a row

```
WITH "https://api.stackexchange.com/2.2/questions?pagesize=100&order=desc&sort=creation&tagged=neo4j&site=stackoverflow&filter=!5-i6Zw8Y)4W7vpy91PMYsKM-k9yzEsSC1_Uxlf" AS url
CALL apoc.load.json(url) YIELD value

UNWIND value.items AS q

MERGE (question:Question {id:q.question_id})
ON CREATE SET question.title = q.title,
question.share_link = q.share_link
question.favorite_count = q.favorite_count
FOREACH (tagName IN q.tags |
    MERGE (tag:Tag {name:tagName}) MERGE (question)-
    [:TAGGED]->(tag))

[ :ANSWER ] -> (answer)
    ON CREATE SET answer.id = answer_id
    MERGE (questioner:User {id:answer.owner_id})
    ON CREATE SET questioner.display_name =
    answer.owner.display_name
```

```
FOREACH (a IN q.answers |
    MERGE (question)->[:ANSWER]->(answer:Answer
{id:a.answer_id})
    MERGE (answerer:User {id:a.owner.user_id})
    ON CREATE SET answerer.display_name =
    a.owner.display_name
```

```
23b0eeb67621195a85441df97f8d97e=128&d=identicon&r=PO",
ptus"

q7e=128&q=_",
isima"

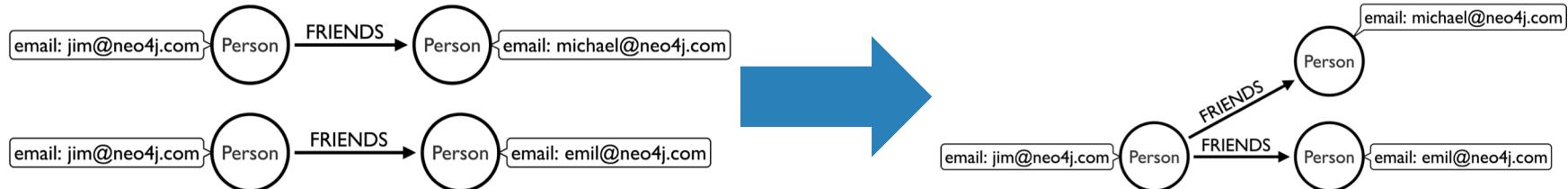
unning-neo4j-from-a-remote-machine#comment84332534_48665959"
867621195a85441df97f8d97e=128&d=identicon&r=PO",
ptus"
```



Graph Refactorings

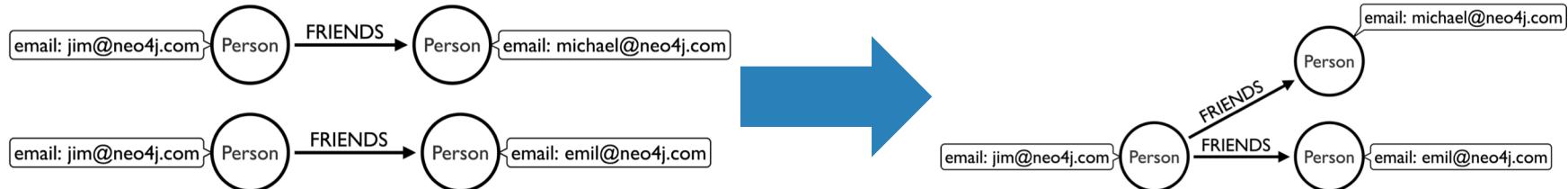
apoc.refactor.mergeNodes

```
MATCH (n:Person)
WITH n.email AS email, collect(n) as people
WHERE size(people) > 1
CALL apoc.refactor.mergeNodes(people)
YIELD node
RETURN node
```



apoc.refactor.mergeNodes

```
MATCH (n:Person)
WITH n.email AS email, collect(n) as people
WHERE size(people) > 1
CALL apoc.refactor.mergeNodes(people)
YIELD node
RETURN node
```



apoc.create.addLabels

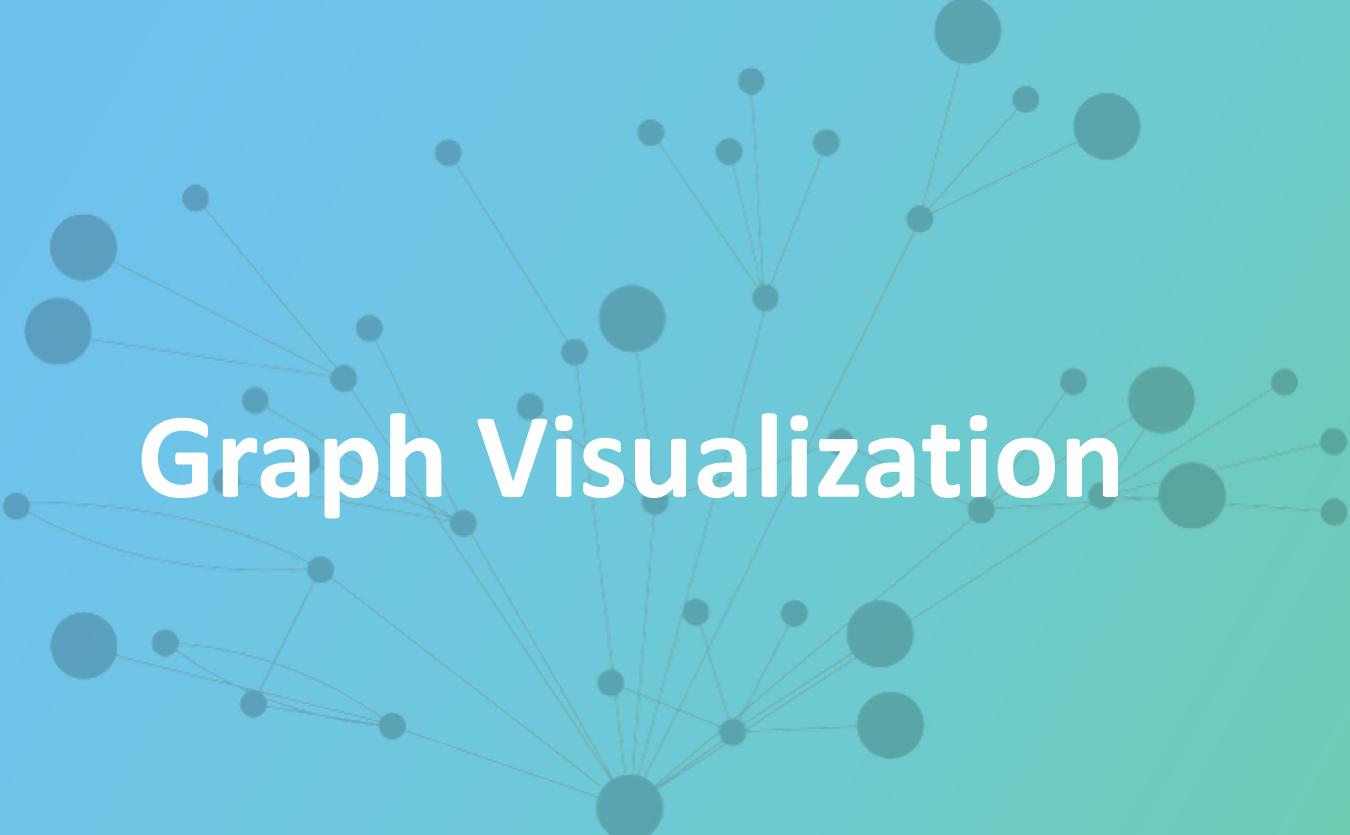
```
MATCH (n:Movie)
CALL apoc.create.addLabels( id(n), [ n.genre ] ) YIELD node
REMOVE node.genre
RETURN node
```



Cypher Execution

Run large scale updates

```
CALL apoc.periodic.iterate(  
  'MATCH (n:Person) RETURN n',  
  'SET n.name = n.firstName + " " + n.lastName',  
  {batchSize:10000, parallel:true})
```



Graph Visualization



johnymontana / neovis.js

 Watch ▾ 12

 Star 83

 Fork 16

 Code

 Issues 2

 Pull requests 0

 Projects 0

 Wiki

 Insights

Neo4j + vis.js = neovis.js. Graph visualizations in the browser with data from Neo4j.

 25 commits

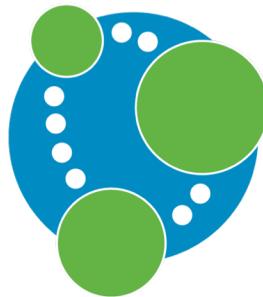
 1 branch

 0 releases

 2 contributors

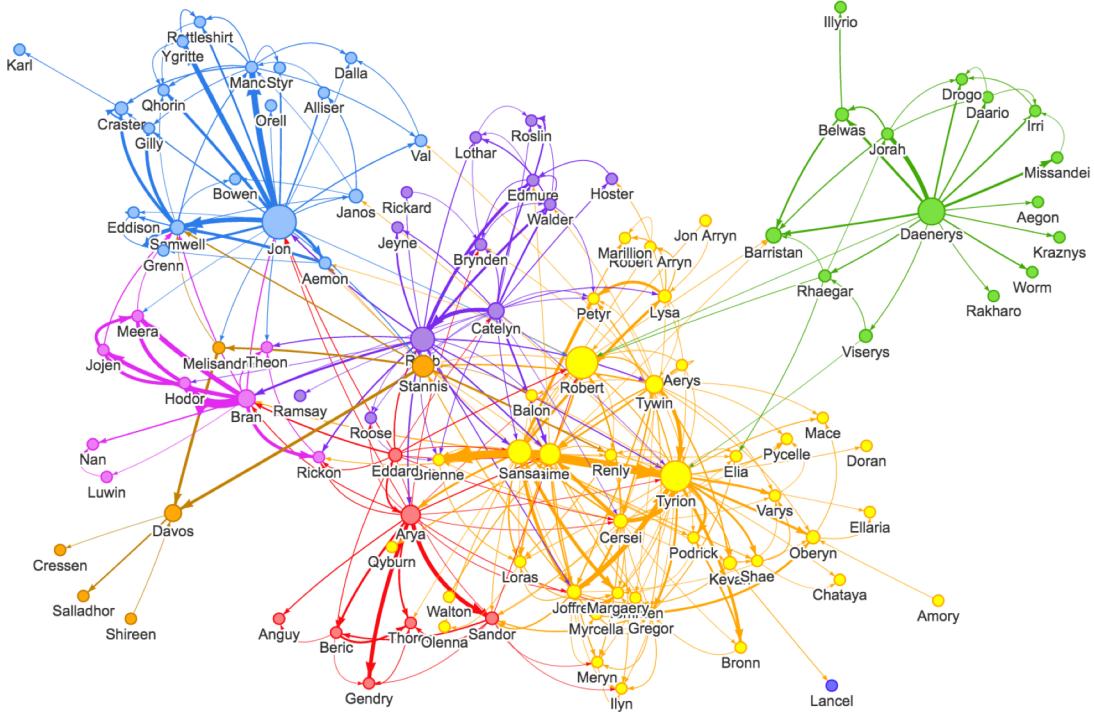
 Apache-2.0





neo4j





```
var viz;  
  
function draw() {  
    var config = {  
        container_id: "viz",  
        server_url: "bolt://localhost:7687",  
        server_user: "neo4j",  
        server_password: "sorts-swims-burglaries",  
        labels: {  
            "Character": {  
                "caption": "name",  
                "size": "pagerank",  
                "community": "community"  
            }  
        },  
        relationships: {  
            "INTERACTS": {  
                "thickness": "weight",  
                "caption": false  
            }  
        },  
        initial_cypher: "MATCH (n)-[r:INTERACTS]->(m) RETURN *"  
    };  
  
    viz = new NeoVis.default(config);  
    viz.render();  
}
```

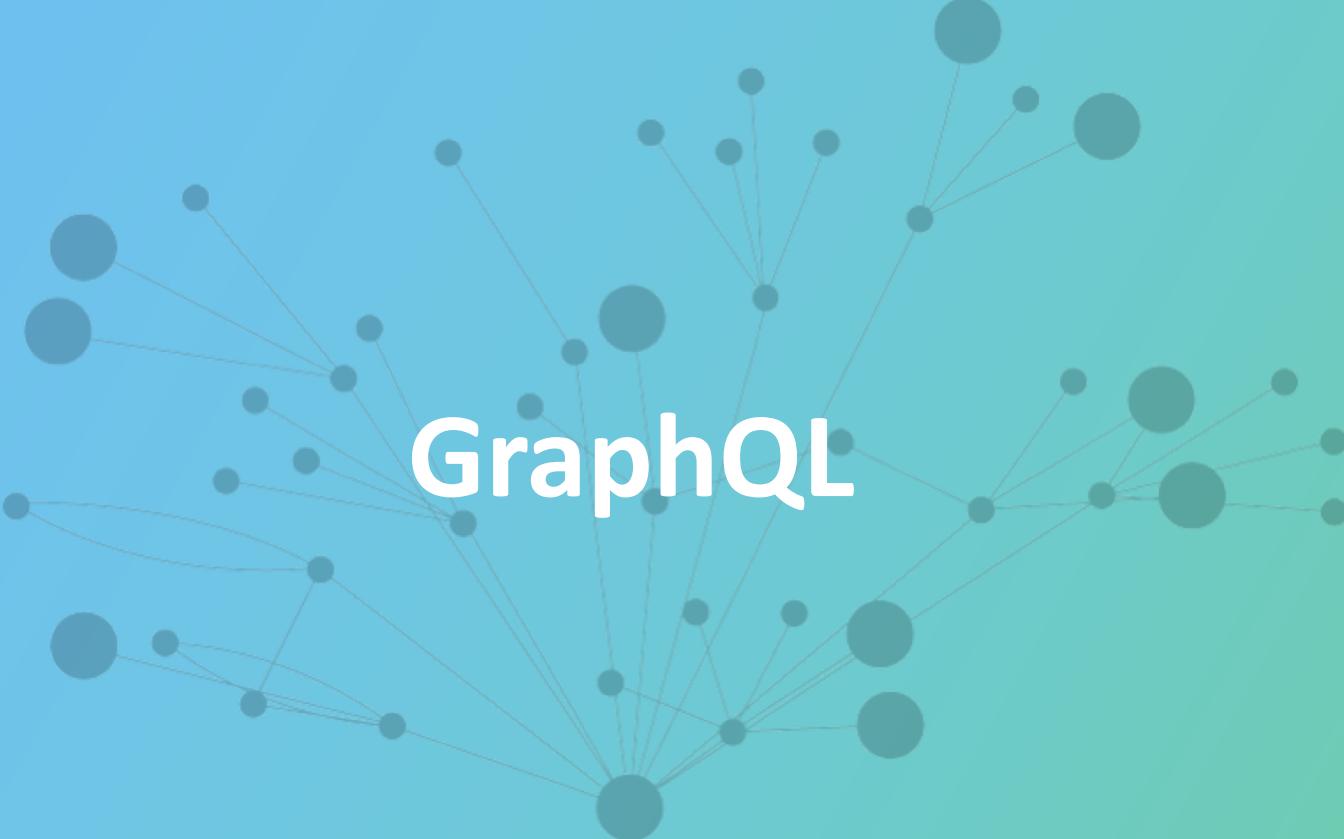


```
var viz;  
  
function draw() {  
    var config = {  
        container_id: "viz",  
        server_url: "bolt://localhost:7687",  
        server_user: "neo4j",  
        server_password: "sorts-swims-burglaries",  
        labels: {  
            "Character": {  
                "caption": "name",  
                "size": "pagerank",  
                "community": "community"  
            }  
        },  
        relationships: {  
            "INTERACTS": {  
                "thickness": "weight",  
                "caption": false  
            }  
        },  
        initial_cypher: "MATCH (n)-[r:INTERACTS]->(m) RETURN *"  
    };  
  
    viz = new NeoVis.default(config);  
    viz.render();  
}
```



```
var viz;  
  
function draw() {  
    var config = {  
        container_id: "viz",  
        server_url: "bolt://localhost:7687",  
        server_user: "neo4j",  
        server_password: "sorts-swims-burglaries",  
        labels: {  
            "Character": {  
                "caption": "name",  
                "size": "pagerank",  
                "community": "community"  
            }  
        },  
        relationships: {  
            "INTERACTS": {  
                "thickness": "weight",  
                "caption": false  
            }  
        },  
        initial_cypher: "MATCH (n)-[r:INTERACTS]->(m) RETURN *"  
    };  
  
    viz = new NeoVis.default(config);  
    viz.render();  
}
```





GraphQL



What is it?

GraphQL is a query language for your API, and a server-side runtime for executing queries by using a type system you define for your data.

What is it?

GraphQL is a **query language for your API**, and a server-side runtime for executing queries by using a type system you define for your data.



Samer Buna [Follow](#)

Reinventing coding education at <https://jsComplete.com>
Jul 24 · 17 min read

REST APIs are REST-in-Peace APIs. Long Live GraphQL.

After years of dealing with REST APIs, when I first learned about GraphQL and the problems it's attempting to solve, I could not resist tweeting the exact title of this article.

Samer Buna
@samerbuna

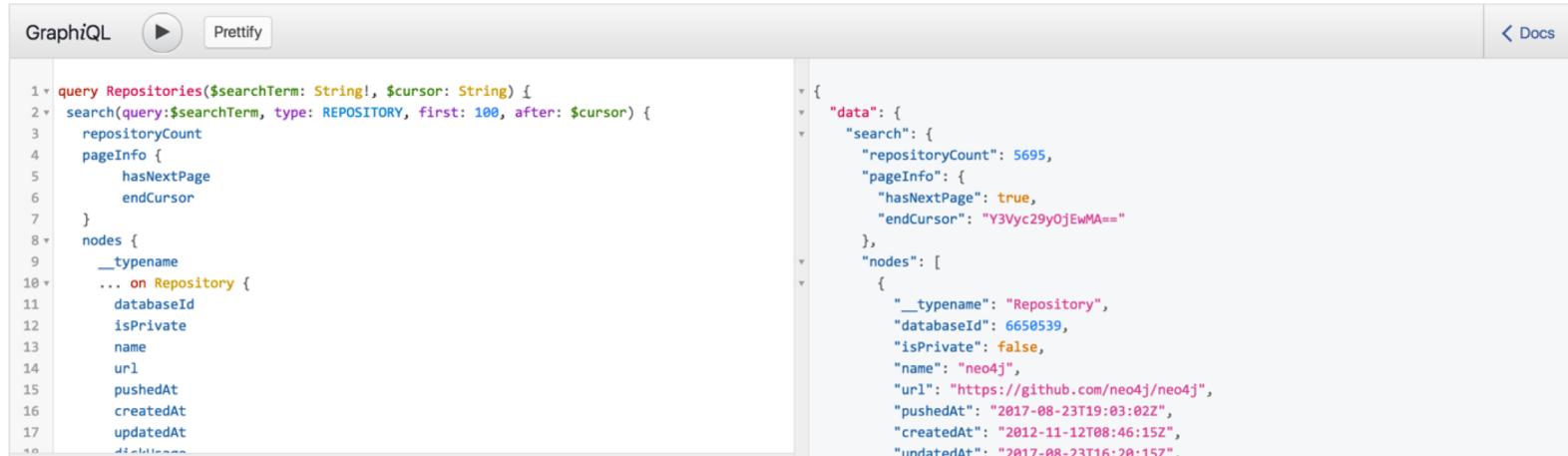
#REST APIs are now #RestInPeace APIs | Long live #GraphQL

5:30 PM - Sep 17, 2015

3 replies 48 retweets 57 likes

What is it?

GraphQL is a query language for your API, and a **server-side runtime for executing queries** by using a type system you define for your data.



The screenshot shows a GraphQL playground interface. On the left, a code editor displays a GraphQL query:

```
1 * query Repositories($searchTerm: String!, $cursor: String) {
2 *   search(query:$searchTerm, type: REPOSITORY, first: 100, after: $cursor) {
3 *     repositoryCount
4 *     pageInfo {
5 *       hasNextPage
6 *       endCursor
7 *     }
8 *     nodes {
9 *       __typename
10 *       ... on Repository {
11 *         databaseId
12 *         isPrivate
13 *         name
14 *         url
15 *         pushedAt
16 *         createdAt
17 *         updatedAt
18 *       }
19 *     }
20 *   }
21 * }
```

On the right, the results of the query are shown in JSON format:

```
{
  "data": {
    "search": {
      "repositoryCount": 5695,
      "pageInfo": {
        "hasNextPage": true,
        "endCursor": "Y3Vyc29yObjEwMA=="
      },
      "nodes": [
        {
          "__typename": "Repository",
          "databaseId": 6650539,
          "isPrivate": false,
          "name": "neo4j",
          "url": "https://github.com/neo4j/neo4j",
          "pushedAt": "2017-08-23T19:03:02Z",
          "createdAt": "2012-11-12T08:46:15Z",
          "updatedAt": "2017-08-23T16:20:15Z"
        }
      ]
    }
  }
}
```

At the bottom right of the interface is the Neo4j logo.

What is it?

GraphQL is a query language for your API, and a server-side runtime for executing queries by **using a type system** you define for your data.

```
type Character {  
    name: String  
    friends: [Character]  
    homeWorld: Planet  
    species: Species  
}
```

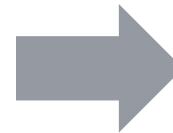
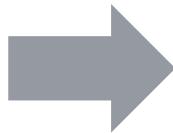
```
type Planet {  
    name: String  
    climate: String  
}
```

```
type Species {  
    name: String  
    lifespan: Int  
    origin: Planet  
}
```

Manual mapping code



GraphQL



```
Query: {  
  moviesByTitle: (root, args, context) => {  
    let session = context.driver.session();  
    let query = "MATCH (movie:Movie) WHERE movie.title CONTAINS $subString RETURN movie LIMIT $first;"  
    return session.run(query, args)  
      .then( result => { return result.records.map(record => { return record.get("movie").properties }))}  
  },  
},
```



Auto translating GraphQL → Cypher

Neo4j-GraphQL Extension

Note This branch is for supporting Neo4j 3.2.

build passing

This implementation provides a GraphQL API to Neo4j, it comes as library but can also be installed as Neo4j server extension to act as a GraphQL endpoint. It turns GraphQL queries and mutations into Cypher statements and executes them on Neo4j.

We want to explore three approaches:

1. read schema / metadata from the database provide GraphQL DataFetcher that generate and run Cypher (WIP) ✓
2. make the same work with externally configured schema information (using IDL) ✓

github.com/neo4j/graphql/neo4j-graphql

circleci passing

neo4j-graphql-js

A GraphQL to Cypher query execution layer for Neo4j and JavaScript GraphQL implementations.

neo4j-graphql-js is in early development. There are rough edges and APIs may change. Please file issues for any bugs that you find or feature requests.

Installation and usage

Install

```
npm install --save neo4j-graphql-js
```

Then call `neo4jgraphql()` in your GraphQL resolver. Your GraphQL query will be translated to Cypher and the query passed to Neo4j.

```
import {neo4jgraphql} from 'neo4j-graphql-js';

const resolvers = {
  Query: {
    Movie(object, params, ctx, resolveInfo) {
      return neo4jgraphql(object, params, ctx, resolveInfo);
    }
  }
};
```

github.com/neo4j/graphql/neo4j-graphql-js



Server Side Extension

neo4j-graphql / neo4j-graphql

 Watch ▾ 31

 Star 161

 Fork 28

 Code

 Issues 39

 Pull requests 2

 Projects 0

 Wiki

 Insights

 Settings

GraphQL bindings for Neo4j, generates and runs Cypher

Edit

 graphql

 graphql-server

 neo4j

 graph-database

 neo4j-plugin

 neo4j-procedures

Manage topics

 115 commits

 3 branches

 2 releases

 5 contributors

 Apache-2.0



conf/neo4j.conf

```
dbms.unmanaged_extension_classes=org.neo4j.graphql=/graphql  
dbms.security.procedures.whitelist=graphql.*
```

```
CALL graphql.idl('
  type Movie {
    title: String!
    released: Int
    actors: [Person] @relation(name:"ACTED_IN",direction:IN)
  }
  type Person {
    name: String!
    born: Int
    movies: [Movie] @relation(name:"ACTED_IN")
  }
)')
```

```
WITH '{ Person(born: 1961) { name, born } }' as query, {} as params  
CALL graphql.execute(query,params)  
YIELD result  
  
UNWIND result.Person as p  
RETURN p.name, p.born
```

JavaScript Library

neo4j-graphql / neo4j-graphql-js

 Watch ▾ 14

 Star 76

 Fork 11

 Code

 Issues 11

 Pull requests 0

 Projects 0

 Wiki

 Insights

 Settings

A GraphQL to Cypher query execution layer for Neo4j and JavaScript GraphQL implementations.

 Edit

<https://www.npmjs.com/package/neo4j-g...>

graphql-query

graphql-server

graphql-server-framework

graphql-resolver

neo4j

cypher

cypher-query-language

graph-database

javascript-graphql-implementations

graphql-schema

cypher-query

Manage topics

 41 commits

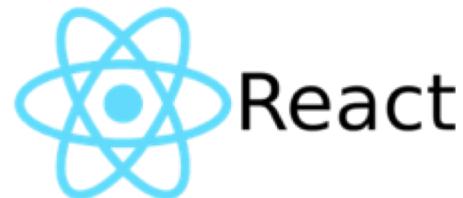
 1 branch

 0 releases

 3 contributors



The GRAND stack



The GRAND stack

GraphQL

React

Apollo

Neo4j **D**atabase

```
import {neo4jgraphql} from 'neo4j-graphql-js';

const resolvers = {
  Query: {
    Movie(object, params, ctx, resolveInfo) {
      return neo4jgraphql(object, params, ctx, resolveInfo);
    }
  }
};
```

An Overview of GraphQL

A New Paradigm for Building APIs

BY WILLIAM LYON

INTRODUCTION TO GRAPHQL

GraphQL is a powerful new tool for building APIs that allows clients (typically web and mobile apps) to ask for only the data they need. Originally designed at Facebook to minimize data sent over the wire and reduce the number of roundtrip API requests for rendering views in native mobile apps, GraphQL has since been open sourced to a healthy community that is building developer-friendly tools. GraphQL has quickly been adopted by several notable companies, including [Github](#), [Yelp](#), [Coursera](#), and [Shopify](#), replacing existing REST APIs. The goal of this Refcard is to introduce you to the concepts of GraphQL, from writing GraphQL queries to building a GraphQL server.

```
type Movie {  
    id: ID!  
    title: String  
    year: Int  
    plot: String  
    genres: [String]  
}  
  
type Query {  
    moviesByTitle(title: String!, limit: Int!): [Movie]  
}
```

Here we define a `Movie` type, the fields that exist for `Movie`, and a query, `moviesByTitle`, that takes two arguments: `title` and `limit`, both of which are required (indicated by the exclamation points). The

dzone.com/refcardz/an-overview-of-graphql

**How do I find out
about more cool stuff?**



+



Join **neo4j-users** on Slack.

7657 users are registered so far.

you@yourdomain.com

GET MY INVITE

This Week in Neo4j - Developer Update

This Week in Neo4j – Data Lineage, Google Cloud, Thomson Reuters' OpenPermid

By [Mark Needham](#) | February 3, 2018

Welcome to this week in Neo4j where we round up what's been happening in the world of graph databases in the last 7 days. This week we have a graph of Thomson Reuters' OpenPermid dataset, running Neo4j on Google Cloud,... [Learn More →](#)

Keywords: [digital humanities](#) • [django](#) • [excel](#) • [google cloud platform](#) • [jqassistant](#) • [Kubernetes](#) • [MySQL](#) • [RDF](#) • [Thomson Reuters](#) • [twin4j](#)

This Week in Neo4j – Knowledge Graph Search, Lil Jon, JIRA

By [Mark Needham](#) | January 27, 2018

Welcome to this week in Neo4j where we round up what's been happening in the world of graph databases in the last 7 days. This week we have a graph of Dave Chappelle's Lil Jon Skit, Knowledge Graph Search, an... [Learn More →](#)

Keywords: [bill of materials](#) • [elastic search](#) • [Graph Search](#) • [Lil Jon](#) • [Pentaho](#) • [twin4j](#) • [wine](#)

neo4j.com/tag/twin4j

This Week in Neo4j – Kotlin, Dynamic Decision Trees, Categorical PageRank

By [Mark Needham](#) | January 20, 2018

Welcome to this week in Neo4j where we round up what's been happening in the world of graph databases in the last 7 days. This week we have categorical PageRank using graph algorithms, more on



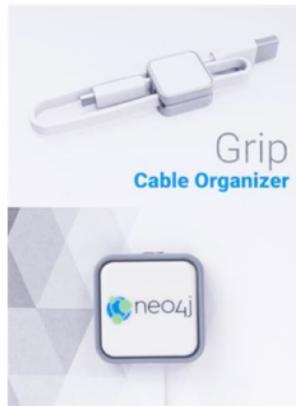
Get Dressed: Neo4j Graph Gear Store



APPAREL



WOMEN'S APPAREL



HARD GOODS



STICKERS

graphgearstore.com







Thanks!

