

Building a real-time recommendation engine with Neo4j

OSCON 2017

William Lyon
@lyonwj



neo4j

William Lyon

Developer Relations Engineer @neo4j

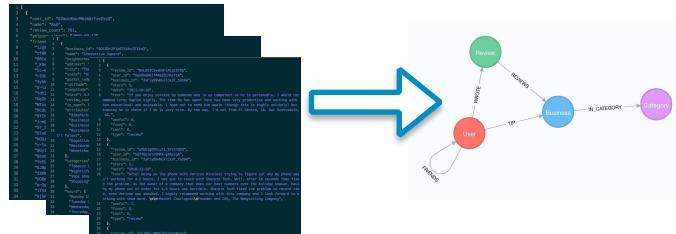
will@neo4j.com
@lyonwj
lyonwj.com



Agenda



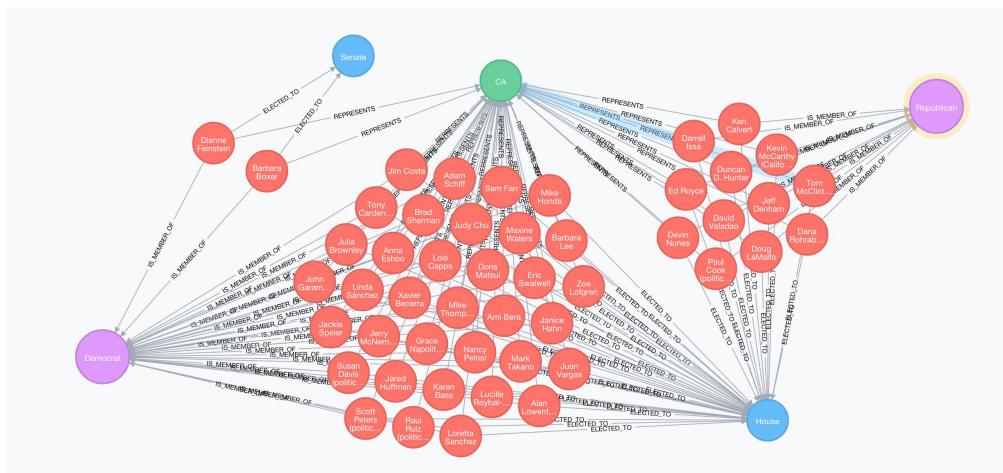
- Intro to graph databases and Neo4j
 - Graph data model
 - Cypher query language
 - Use cases
 - Recommender systems
 - Hands-on!
 - Graph data modeling
 - Cypher



Graph Database

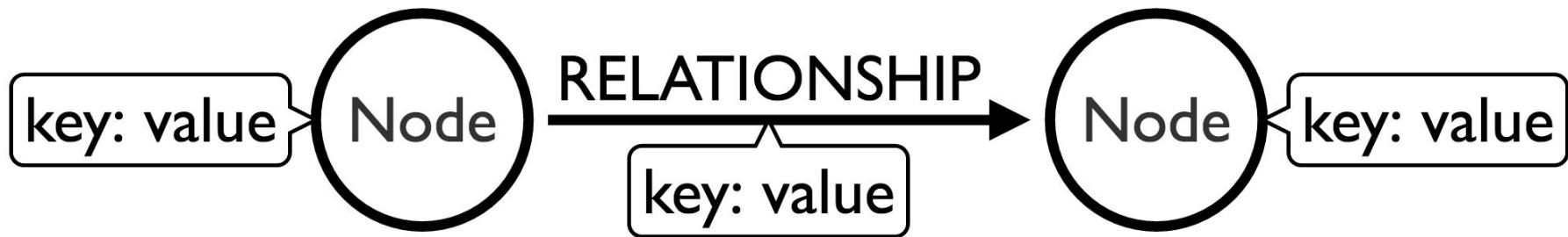
- Property graph data model
- (open)Cypher query language
- Native graph processing
- Language drivers
- Open source

neo4j.com

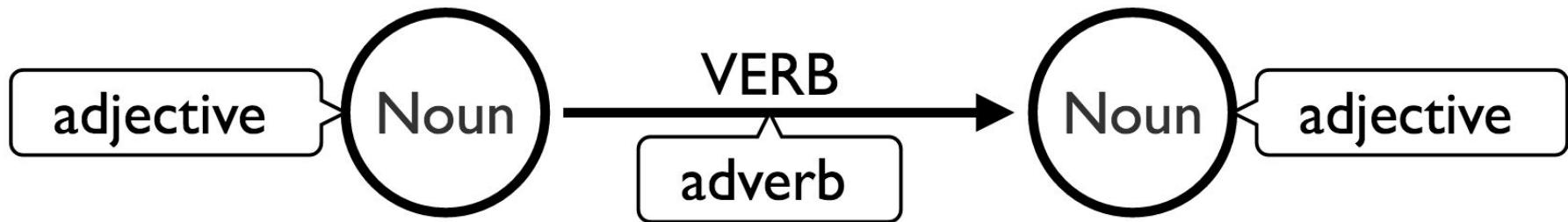


Graph Data Model

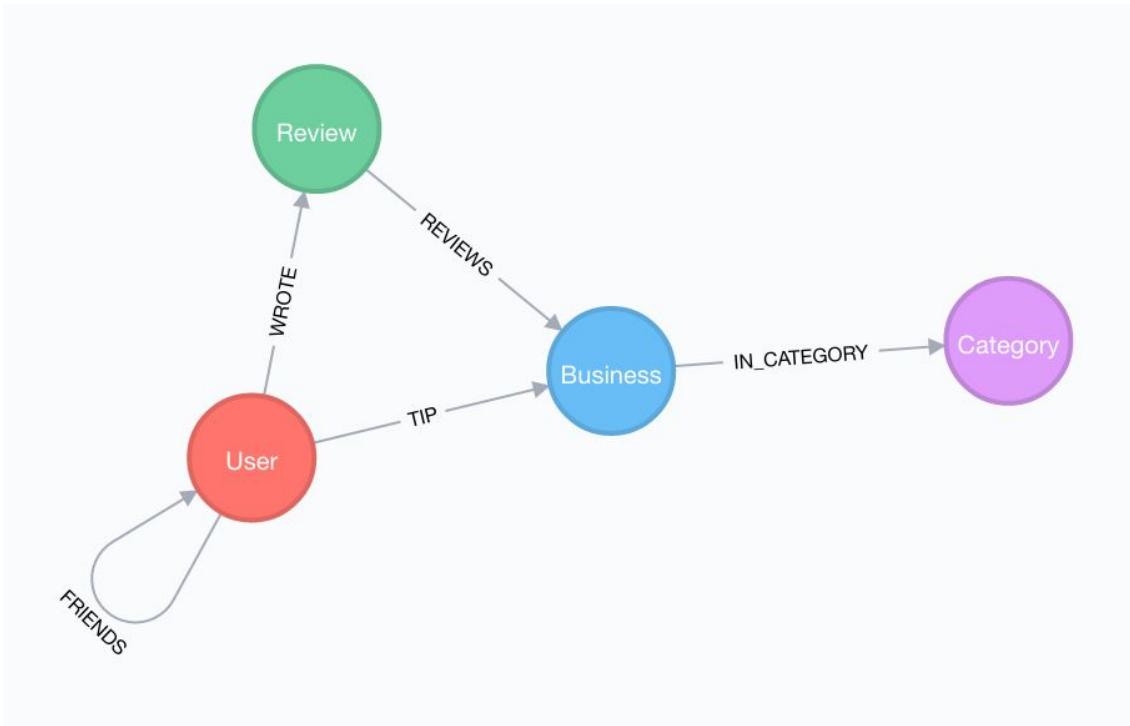
Labeled Property Graph Model



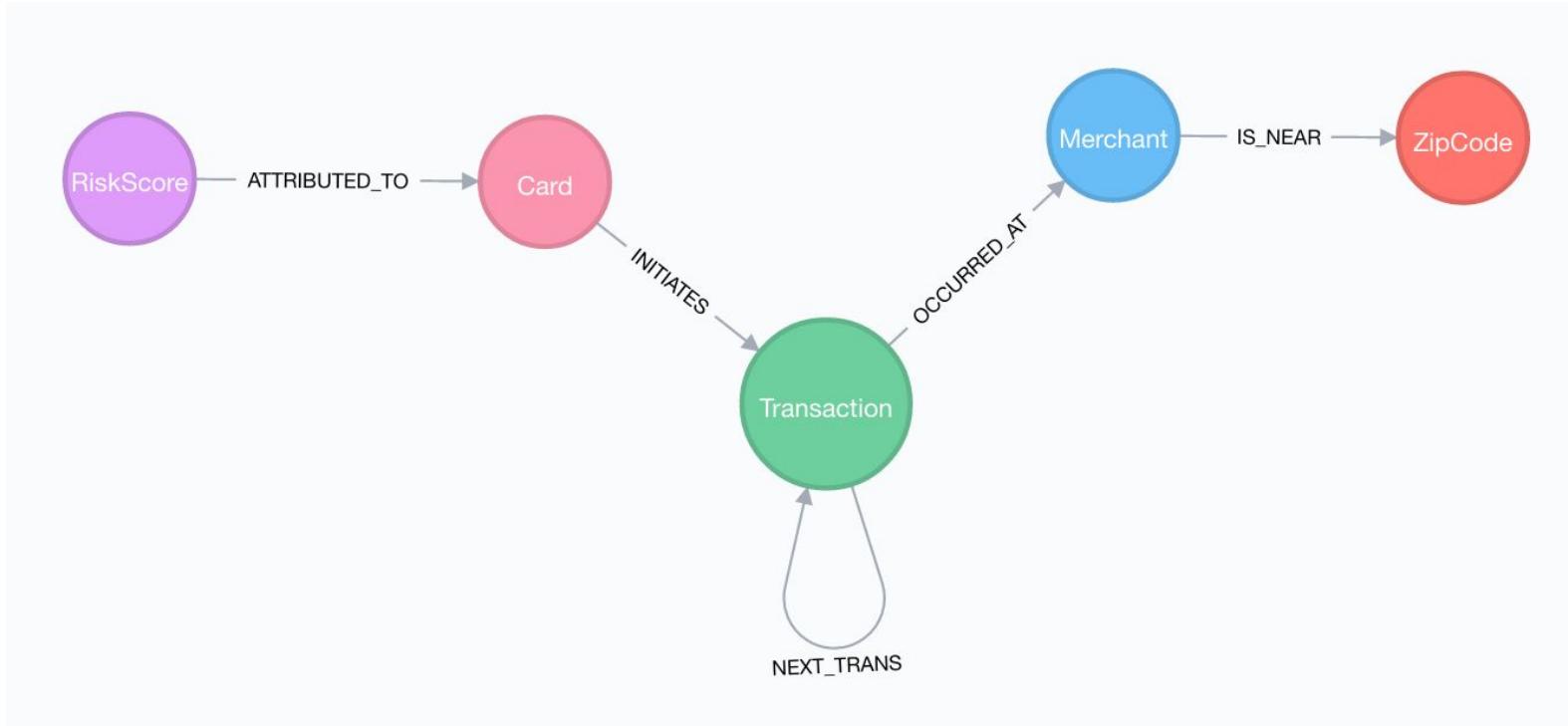
Labeled Property Graph Model



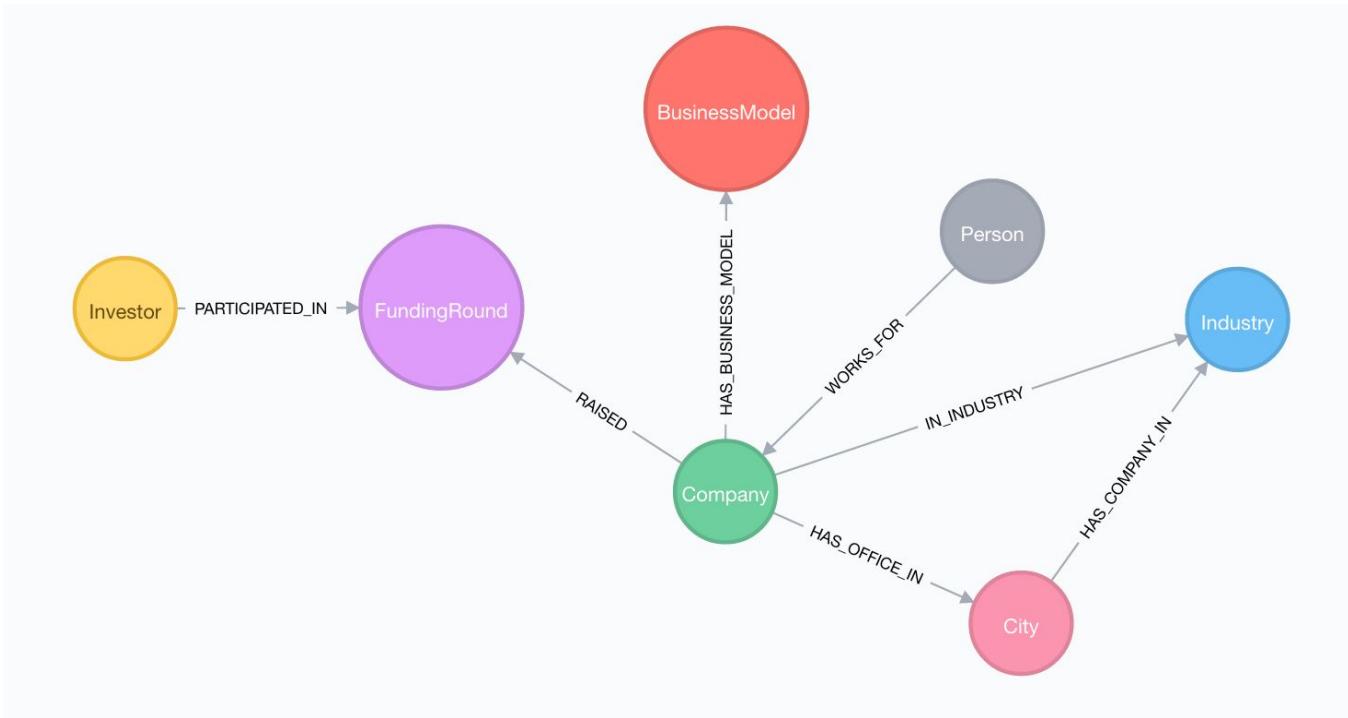
The Graph



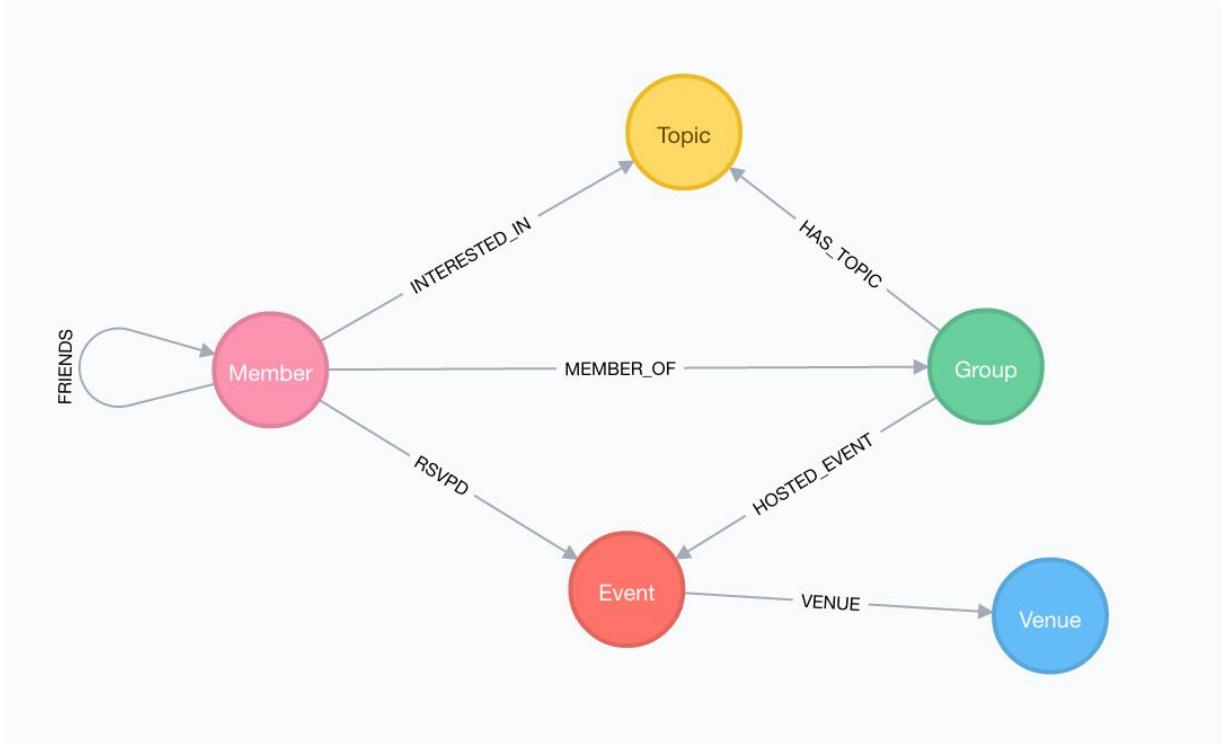
The Graph



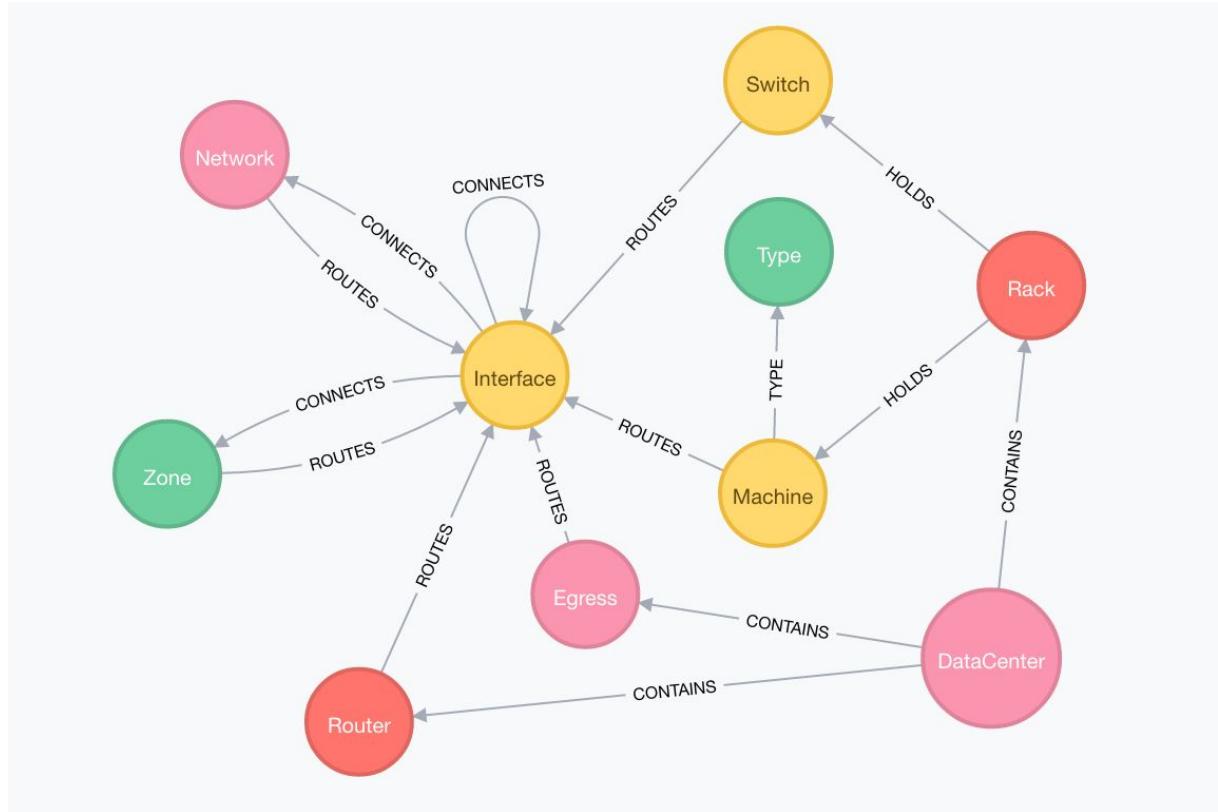
The Graph

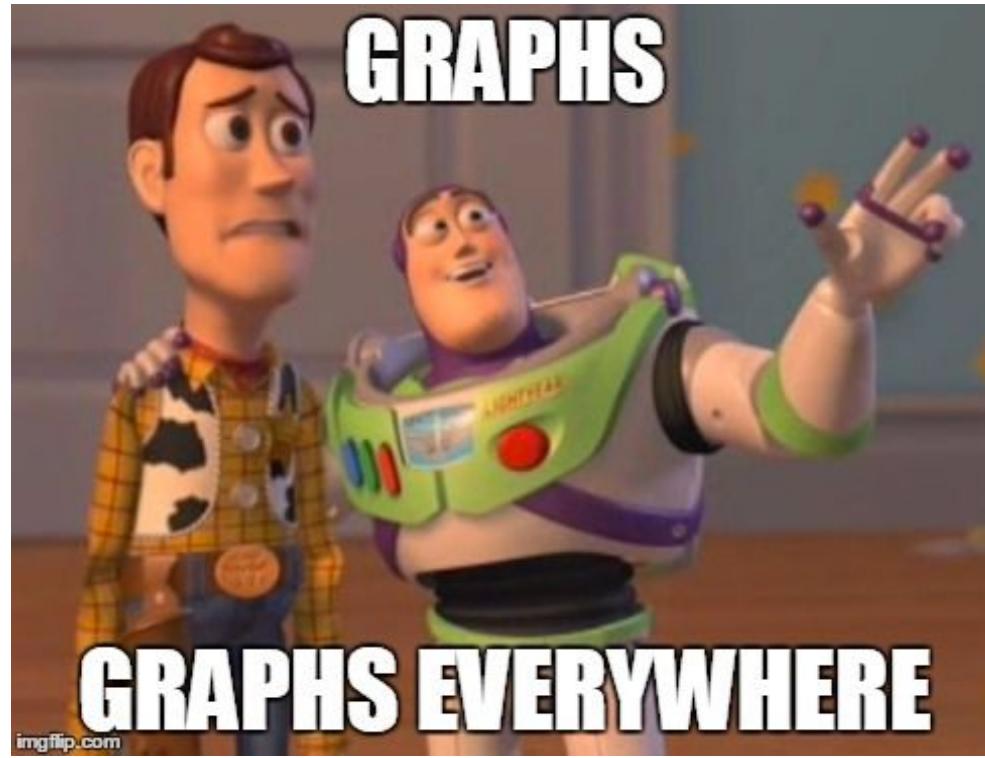


The Graph



The Graph





openCypher

(queryLanguage) - [:FOR] -> (graphs)

<http://www.opencypher.org/>



Cypher



```
1 MATCH (m:Movie){:RATED}-(u:User)
2 WHERE m.title CONTAINS "Matrix"
3 WITH m.title AS movie, COUNT(*) AS reviews
4 RETURN movie, reviews
5 ORDER BY reviews DESC
6 LIMIT 5;
```



```
1 MATCH (m:Movie)<-[ :RATED]-(u:User)
2 WHERE m.title CONTAINS "Matrix"
3 WITH m.title AS movie, COUNT(*) AS reviews
4 RETURN movie, reviews
5 ORDER BY reviews DESC
6 LIMIT 5;
```

| | | |
|-----------|--|--|
| find | MATCH (m:Movie)<-[:RATED]-(u:User) | Search for an existing graph pattern |
| filter | WHERE m.title CONTAINS "Matrix" | Filter matching paths to only those matching a predicate |
| aggregate | WITH m.title AS movie, COUNT(*) AS reviews | Count number of paths matched for each movie |
| return | RETURN movie, reviews | Specify columns to be returned by the statement |
| order | ORDER BY reviews DESC | Order by number of reviews, in descending order |
| limit | LIMIT 5; | Only return first four records |

Test-Drive Neo4j with Cypher

Social

Network Management

Fraud Detection

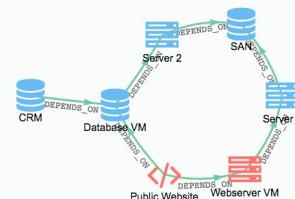
[Impact Analysis](#) | [Dependency Analysis](#) | [Statistics](#)

Impact Analysis

Find all services that depend on Server 1. These would be impacted by an outage of that server.

```
MATCH
  (n:Service)-[:DEPENDS_ON*]-(dependent:Service)
  WHERE
    n.name = "Server 1"
  RETURN
    dependent
```

Only WebsERVER VM depends on Server 1. Because we're looking at variable length paths of DEPENDS_ON relationships, we're also able to determine that Public Website would be impacted by an outage of Server 1.



See Code In:

JAVA

PYTHON

RUBY

PHP

C#

JAVASCRIPT

```
# npm install neo4j-driver
from neo4j.v1 import GraphDatabase, basic_auth
driver = GraphDatabase.driver("bolt://localhost", auth=basic_auth)
session = driver.session()

# Insert data
insert_query = """
UNWIND {pairs} AS pair
MERGE (s1:Service {name: pair[0]})
MERGE (s2:Service {name: pair[1]})
MERGE (s1)-[:DEPENDS_ON]-(s2);
"""

data = [{"CRM": "Database VM"}, {"Database VM": "Server 2"}, {"Server 2": "SAN"}, {"SAN": "Server 1"}, {"Server 1": "Public Website"}, {"Public Website": "WebsERVER VM"}, {"WebsERVER VM": "Public Website"}]
```

Downloading and Installing Python

1. [Install neo4j-python-driver](#) using pip:
\$ pip install neo4j-driver
2. Copy and paste code at left into py file and run

Use Case: Personalized Recommendations



SCOTT BAIO LEAH REMINI
A.C.O.D.
ADULT CHILDREN OF DIVORCE



THE LETTERS
a film by
Nancy Meyers



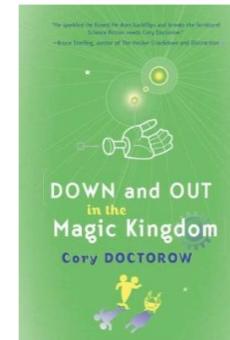
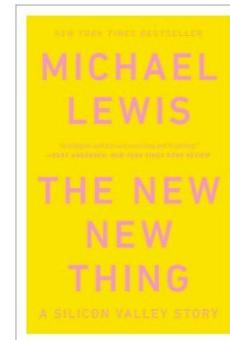
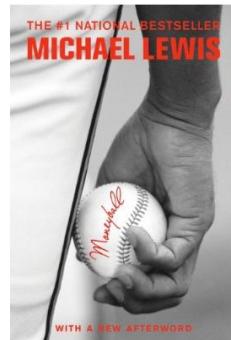
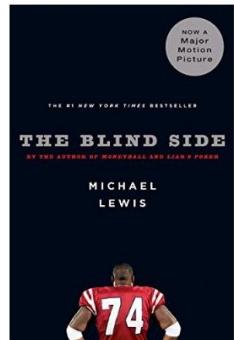
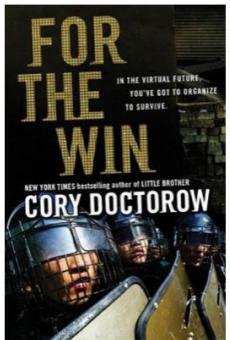
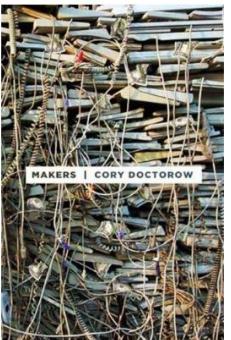
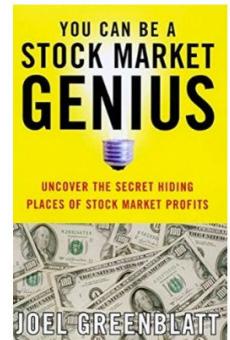
>
SUG

NETFLIX

Recommendations drive user engagement



Recommendations for you in Kindle Store



“35 percent of what consumers purchase on Amazon
and 75 percent of what they watch on Netflix come from
product recommendations”



Search



Hi, login
My Account



Track Orders | Gift Cards | Store finder | Credit Card | Grocery Pickup | Help

All departments Living room | Kitchen | Hallway | Lightning | Bedroom | Garden | Home Office Space



People who bought Side Table also bought:



Coffee Table
\$235



Low Book Shelf
\$150



Bed Side Table
\$90

Similar product in from Home Office Series:



Wood Side Table
\$110



Green Side Table
\$135



Walnut Side Table
\$120

Personalized Promotions

Personalized Real-Time
Recommendations

Personalized Real-Time
Recommendations

People who bought Side Table also bought:



Coffee Table
\$235



Low Book Shelf
\$150



Bed Side Table
\$90

Similar product in from Home Office Series:



Wood Side Table
\$110



Green Side Table
\$135



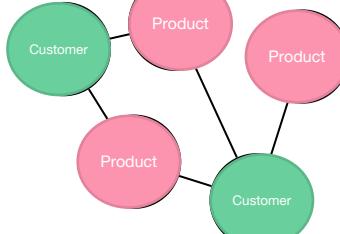
Walnut Side Table
\$120

Collaborative Filtering

An algorithm that considers users interactions with products, with the assumption that other users will behave in similar ways.

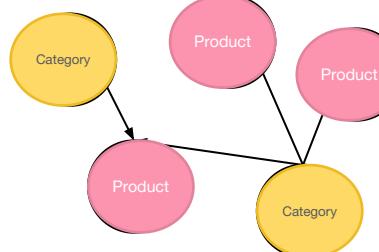
Algorithm Types

Data-Model
(Expressed as
a graph)

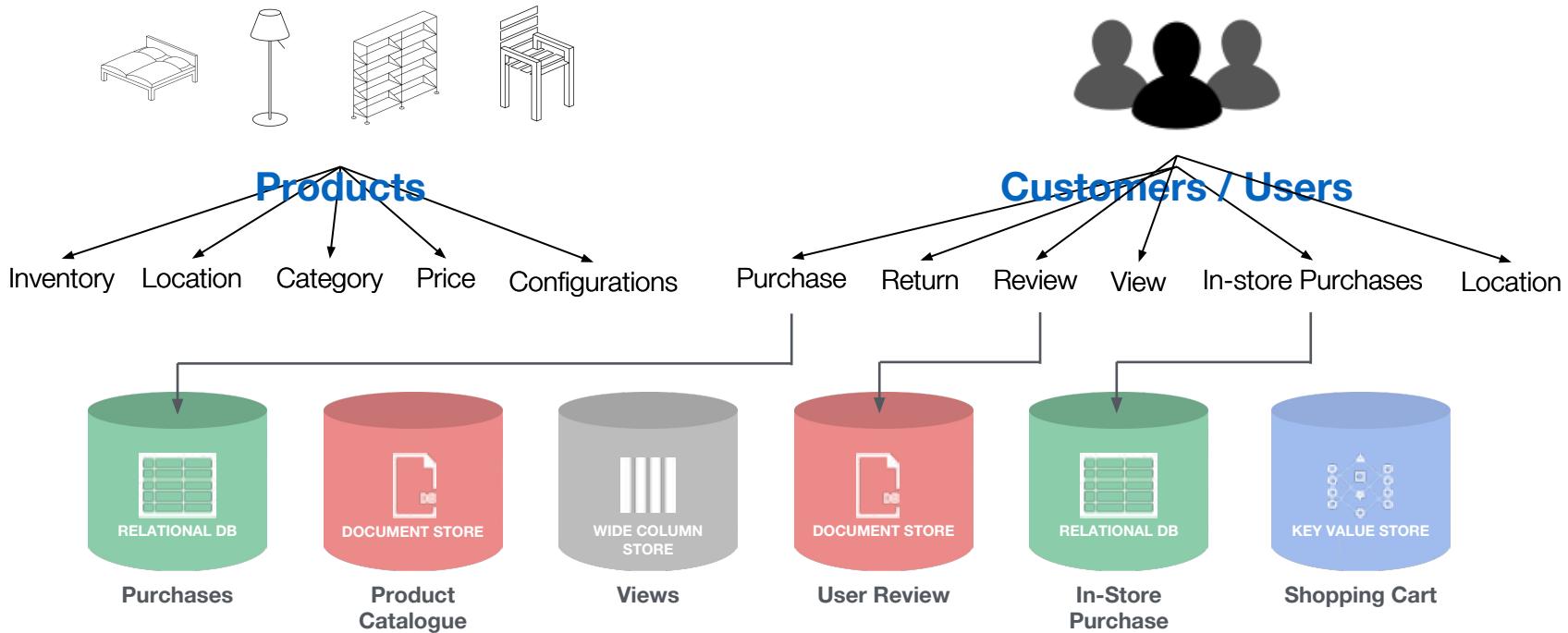


Content Based

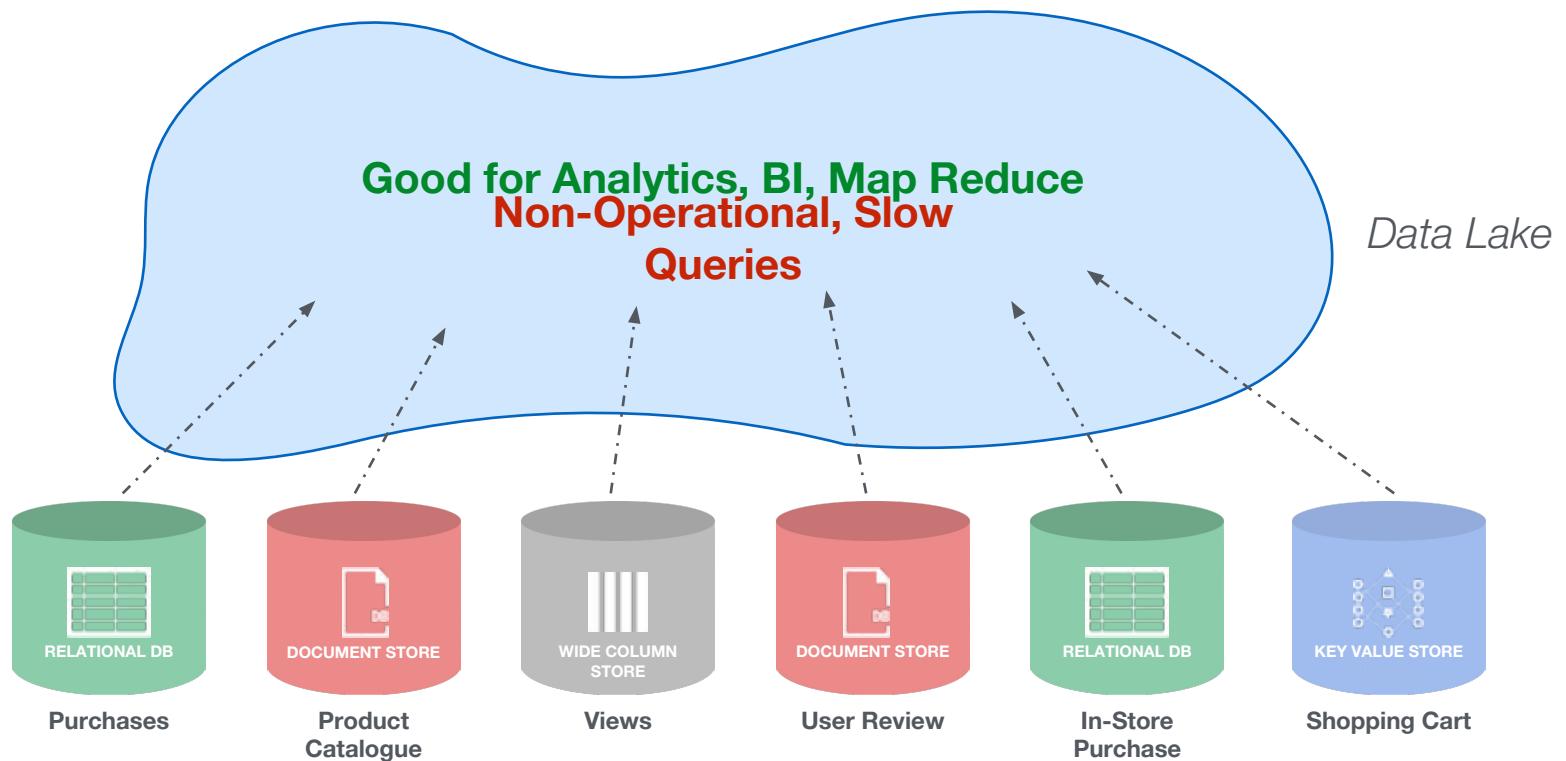
An algorithm that considers similarities between products and categories of products.

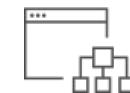


Polyglot Persistence



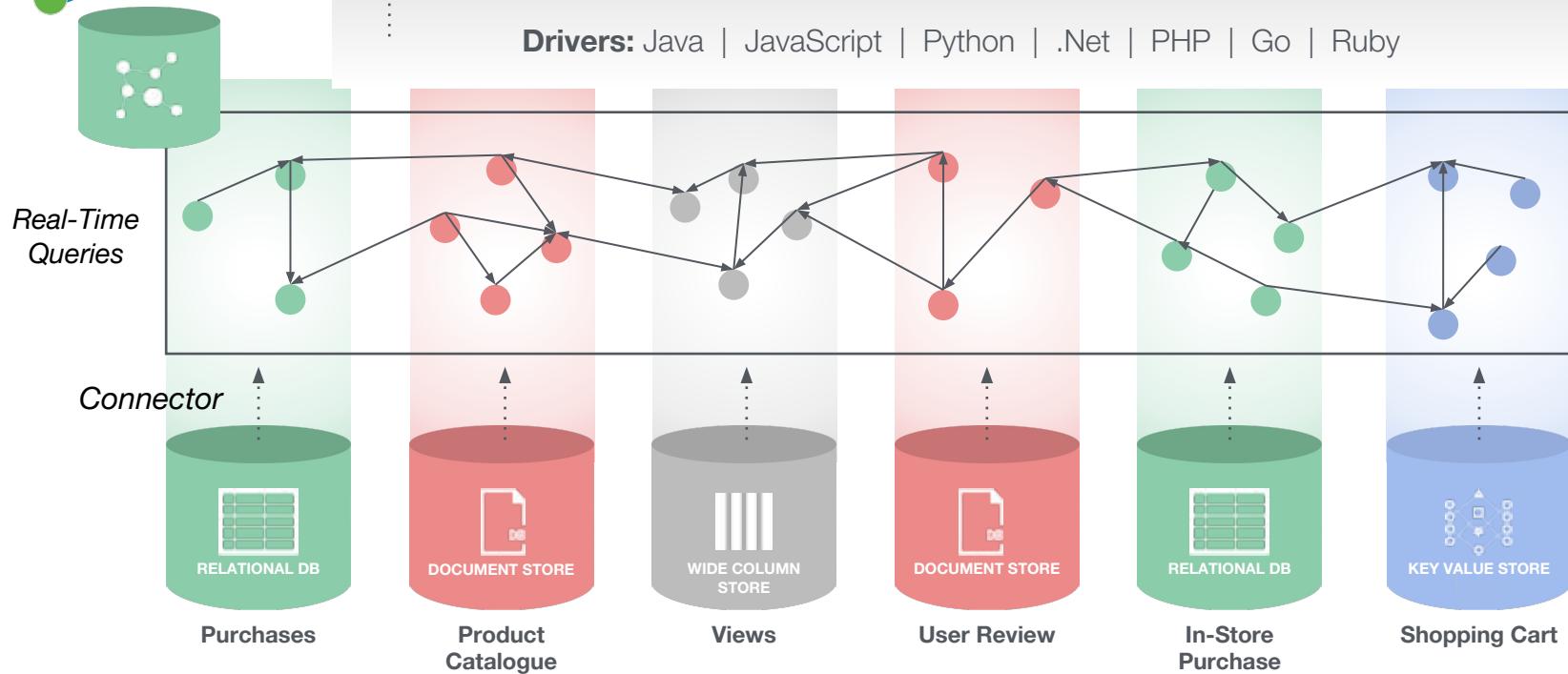
Recommendations require an operational workload — it's in the moment, real-time!





Apps and Systems

Drivers: Java | JavaScript | Python | .Net | PHP | Go | Ruby



Graph-based recommendations



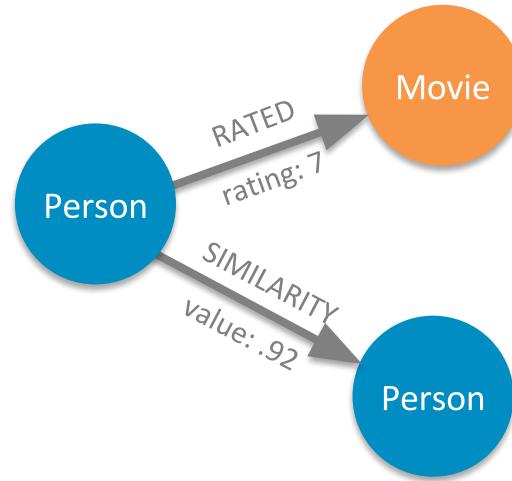
Using Data Relationships for Recommendations

Content-based filtering

Recommend items based on what users have liked in the past

Collaborative filtering

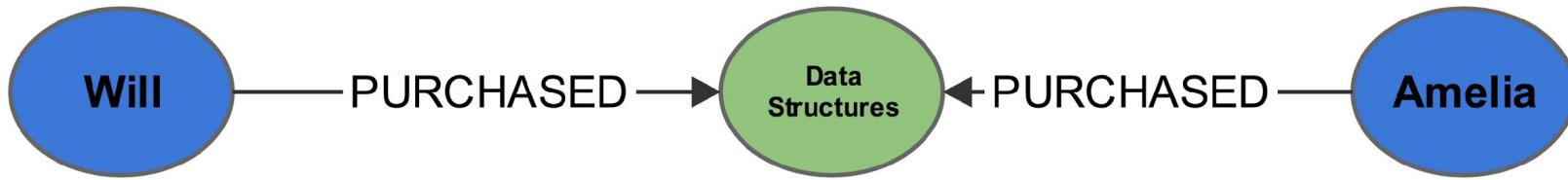
Predict what users like based on the similarity of their behaviors, activities and preferences to others



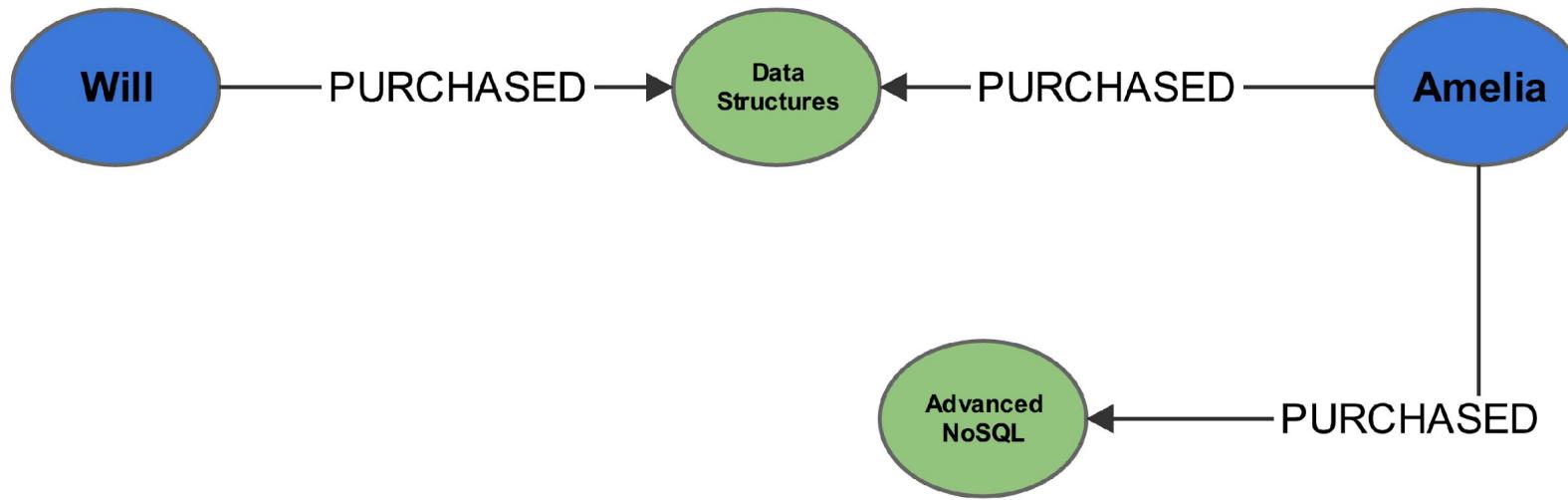
Collaborative Filtering



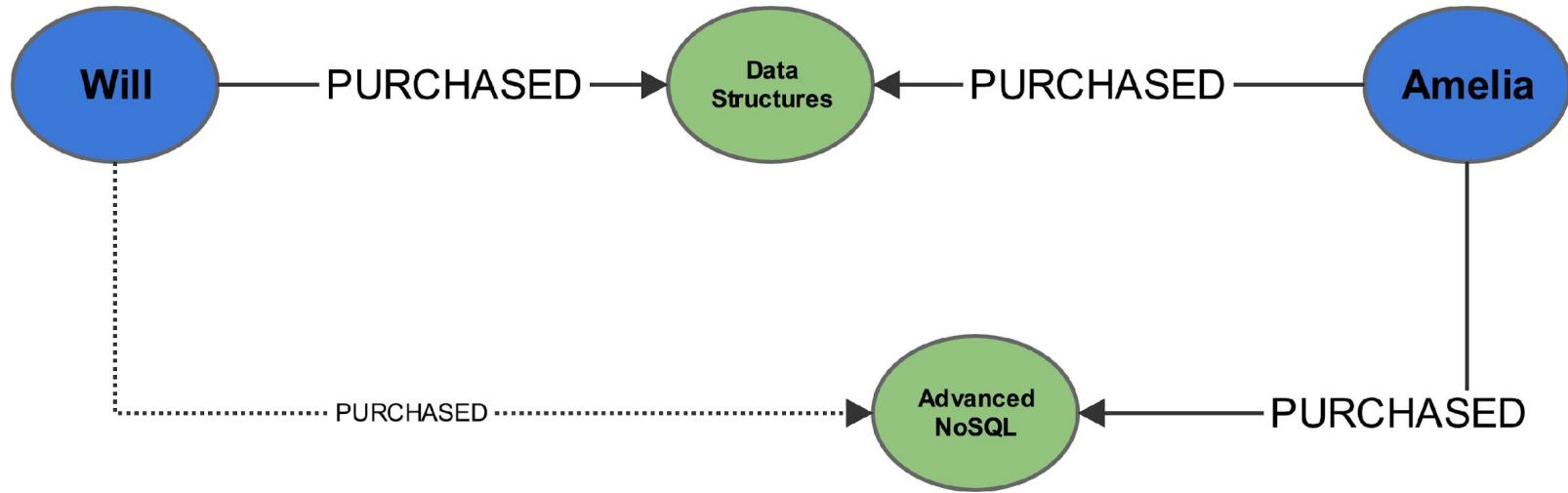
Collaborative Filtering



Collaborative Filtering



Collaborative Filtering



Link prediction

In Cypher

```
1 MATCH (will:Person {name: "Will"})-[:PURCHASED]->(b:Book)<-[:PURCHASED]-(o:Person)
2 MATCH (o)-[:PURCHASED]->(rec:Book) WHERE NOT exists((will)-[:PURCHASED]->(rec))
3 RETURN rec
```

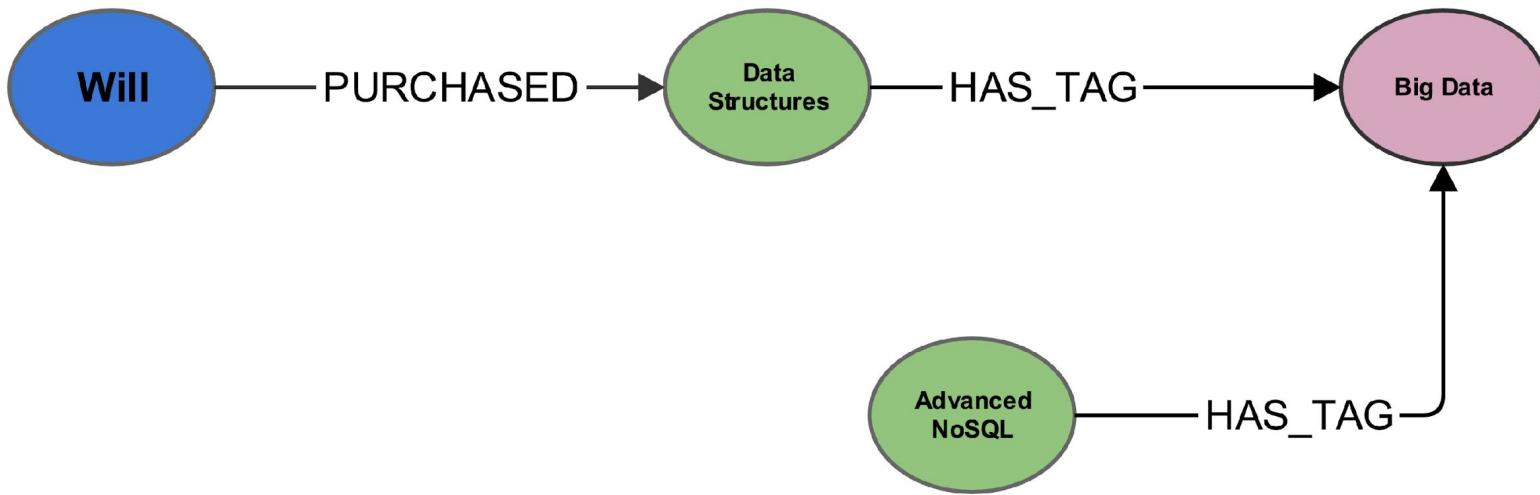
In Cypher

```
1 MATCH (will:Person {name: "Will"})-[:PURCHASED]->(b:Book)<-[:PURCHASED]-(o:Person)
2 MATCH (o)-[:PURCHASED]->(rec:Book) WHERE NOT exists((will)-[:PURCHASED]->(rec))
3 RETURN rec
```

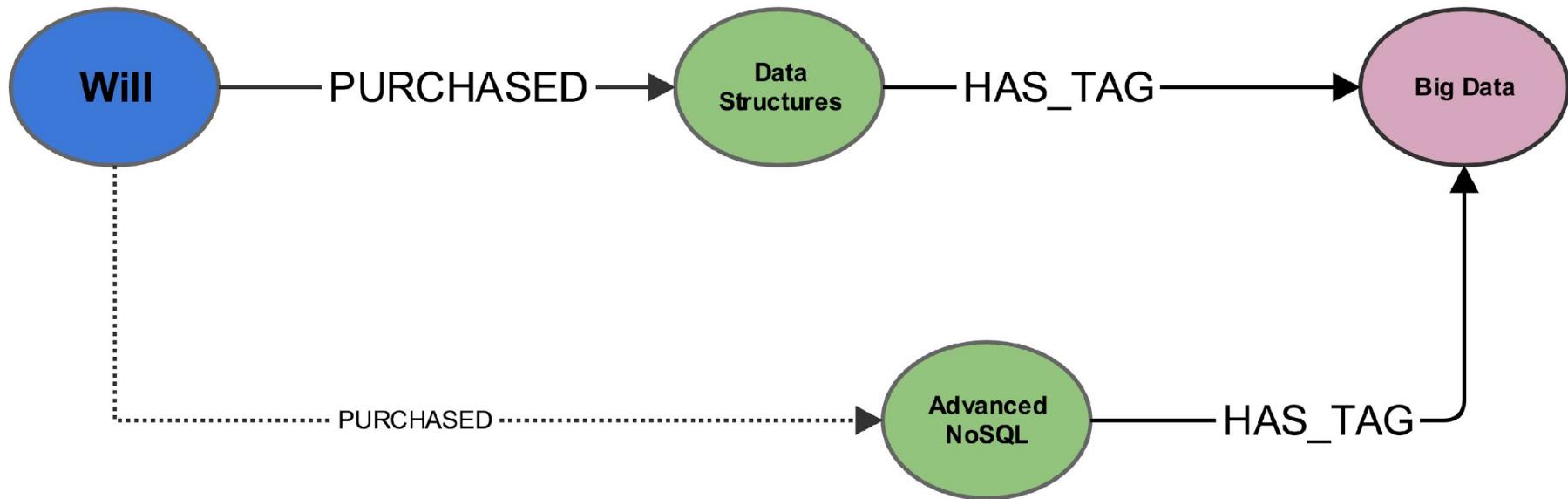
Basic initial approach. Improvements:

- aggregate across all purchases
- scoring / normalize
- compute similarity metrics

Content Filtering



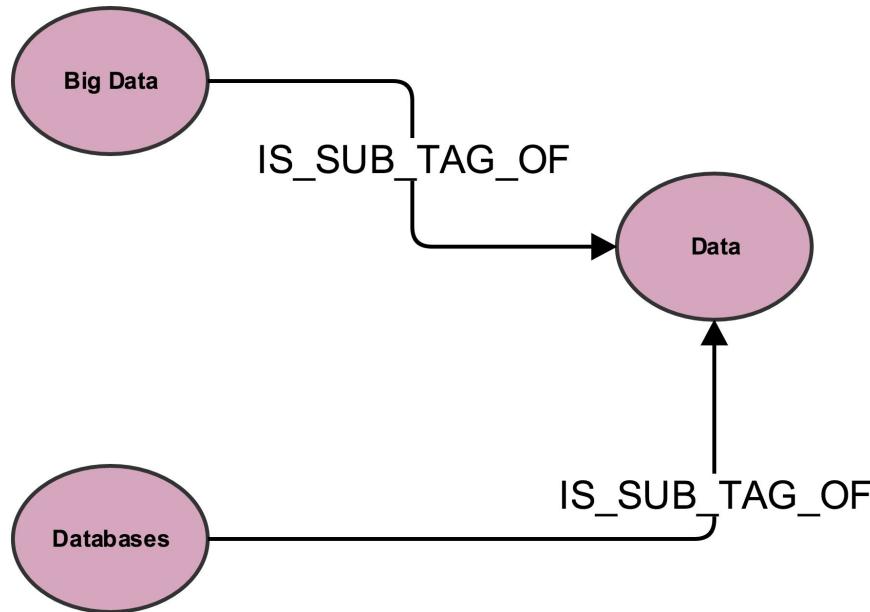
Content Filtering



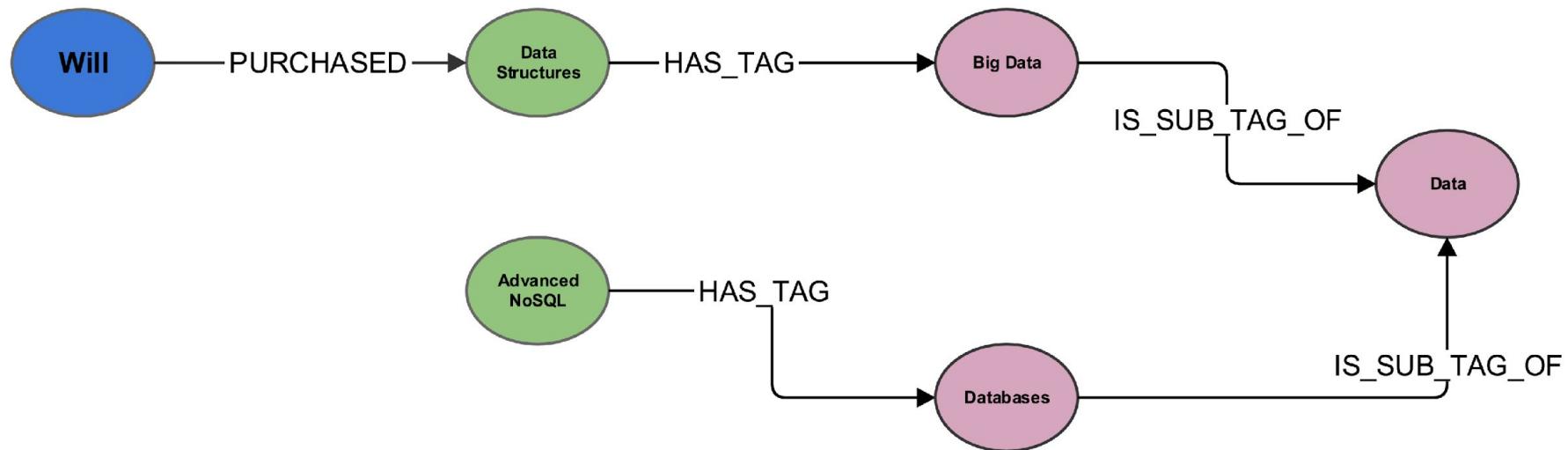
Content Filtering w/ Cypher

```
1 MATCH (will:Person {name: "Will"})-[:PURCHASED]->(b:Book)<- [:HAS_TAG]-(t:Tag)
2 MATCH (t)<- [:HAS_TAG]-(other:Book) WHERE NOT exists((will)-[:PURCHASED]->(other))
3 RETURN other
```

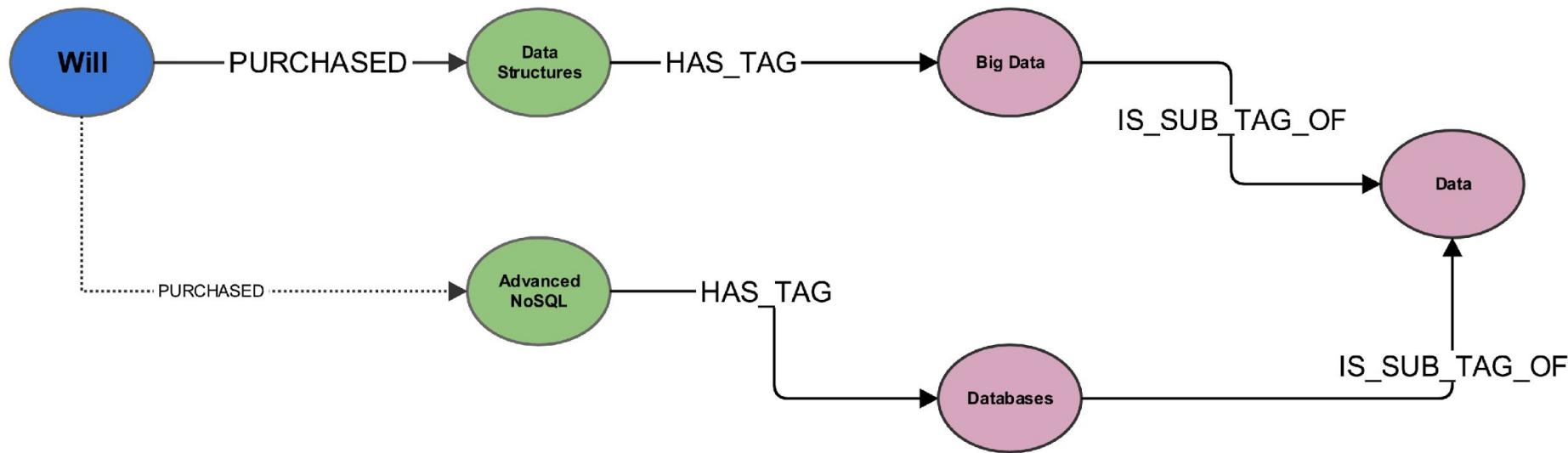
Content Filtering - Concept Hierarchy



Content Filtering - Concept Hierarchy



Content Filtering - Concept Hierarchy



Content Filtering - Concept Hierarchy w/ Cypher

```
1 MATCH (will:Person {name: "Will"})-[:PURCHASED]->(b:Book)<-[:HAS_TAG]-(t:Tag)
2 MATCH (t)-[:IS_SUB_TAG_OF]->(:Tag)<-[:IS_SUB_TAG_OF]-(ot:Tag)
3 MATCH (ot)<-[:HAS_TAG]-(other:Book) WHERE NOT exists((will)-[:PURCHASED]->(other))
4 RETURN other
```

Content Filtering - Concept Hierarchy w/ Cypher

```
1 MATCH (will:Person {name: "Will"})-[:PURCHASED]->(b:Book)<-[:HAS_TAG]-(t:Tag)
2 MATCH (t)-[:IS_SUB_TAG_OF]->(:Tag)<-[:IS_SUB_TAG_OF]-(ot:Tag)
3 MATCH (ot)<-[:HAS_TAG]-(other:Book) WHERE NOT exists((will)-[:PURCHASED]->(other))
4 RETURN other
```

Basic initial approach. Improvements:

- aggregate across all purchases
- cold start
- variable length concept hierarchy
- tag similarity / clusters

Neo4j Sandbox



Greetings William Lyon

Welcome to the Neo4j Sandbox. If you have any questions or problems, feel free to reach out to us at devrel@neo4j.com.

Your Current Sandboxes

Get Started with your Neo4j Sandbox

- 1 Visit the Neo4j Browser. Login with the credentials found under the "Details" tab above. A tutorial will guide you through the datamodel and example data, while teaching you how property graphs work in real-world use cases.
- 2 Start building your application backed by Neo4j. Write your own code, in PHP, Java, JavaScript, Python, or one of any number of other languages, using [templates provided](#).
- 3 Download Neo4j to your own computer, or start a long-living Neo4j instance in the cloud on AWS or other hosting platforms.

Launch a New Sandbox

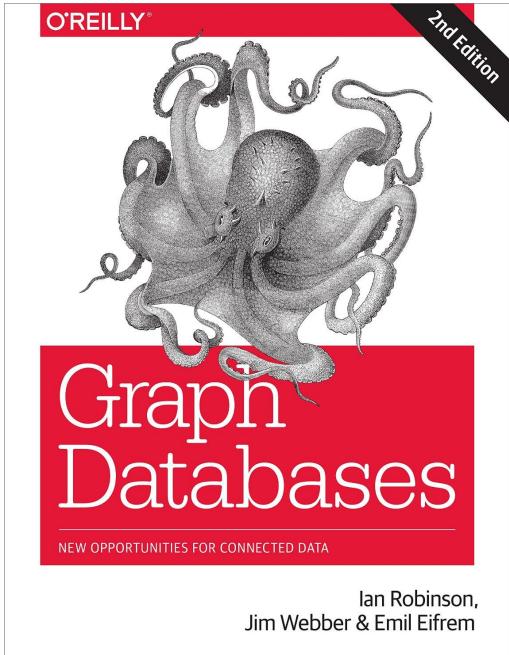
Each sandbox includes data, interactive guides with example queries, and sample code.

| | |
|---|--|
| <p>Fundamentals Training</p>  <p>Neo4j Fundamentals classroom training with instructor-led guides</p> <p>Launch Sandbox</p> | <p>NICAR 2017 Workshop</p>  <p>NICAR 2017 graph database workshop. Analyze campaign finance and US Congress data as a graph in Neo4j.</p> <p>Launch Sandbox</p> |
| <p>Trumpworld</p>  <p>Explore connections in and around the Trump Administration using this dataset from BuzzFeed.</p> <p>Launch Sandbox</p> | <p>Twitter</p>  <p>If signed into Neo4j Sandbox using Twitter, this Sandbox will allow you to Graph your Twitter network.</p> <p>Launch Sandbox</p> |
| <p>Legis-Graph</p>  <p>US Congress modeled as a Graph - bills, votes, members, and more.</p> <p>Launch Sandbox</p> | <p>Blank Sandbox</p>  <p>Blank Sandbox. Load your own data with LOAD CSV or create data from scratch.</p> <p>Launch Sandbox</p> |



neo4jsandbox.com

Want to learn more?



graphdatabases.com

A screenshot of the Neo4j Sandbox v2 interface. At the top, there's a navigation bar with links for PRODUCTS, SOLUTIONS, PARTNERS, CUSTOMERS, LEARN, DEVELOPERS, and a search bar. Below that is a header with the text "Neo4j Sandbox v2" and a "Logout" button. The main content area has a section titled "Greetings William Lyon" with a welcome message and an email address. It also includes a "Your Current Sandboxes" section with a "Get Started" tab selected, showing a list of available sandboxes with icons and "Launch Sandbox" buttons.

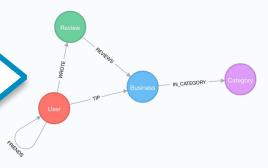
A detailed view of the "Your Current Sandboxes" section. It shows a numbered list of three items: 1. "Get Started with your Neo4j Sandbox" (with an icon of a shopping cart and heart), 2. "Start building your application backed by Neo4j. Write your own code, in PHP, Java, JavaScript, Python, or one of any number of other languages, using templates provided.", and 3. "Download Neo4j to your own computer, or start a long-living Neo4j instance in the cloud on AWS or other hosting platforms." Each item has a "Launch Sandbox" button.

A grid of six Neo4j Sandboxes: 1. Fundamentals Training (Neo4j Fundamentals classroom training with instructor-led guides, Launch Sandbox). 2. NCAR 2017 Workshop (NCAR 2017 graph database workshop. Analyze campaign finance and US Congress data as a graph in Neo4j, Launch Sandbox). 3. Trumpworld (Explore connections in and around the Trump Administration using this dataset from Buzzfeed, Launch Sandbox). 4. Twitter (If signed into Neo4j Sandbox using Twitter, this Sandbox will allow you to Graph your Twitter network, Launch Sandbox). 5. Legis-Graph (US Congress modeled as a Graph - bills, votes, members, and more, Launch Sandbox). 6. Blank Sandbox (Blank Sandbox. Load your own data with LOAD CSV or create data from scratch, Launch Sandbox).

neo4j.com/sandbox

HANDS ON TIME!

```
[1] "name": "APL_PersonalDetailsIndex",
[2] "type": "index",
[3] "fields": [
[4]     "id"
[5] ],
[6] "options": {
[7]     "analyze": true,
[8]     "boost": 1.0,
[9]     "cache": true,
[10]    "comment": "Index for Personal Details",
[11]    "create_in_background": true,
[12]    "max_shard_size": "100mb",
[13]    "min_shard_size": "100mb",
[14]    "norms": false,
[15]    "order": 1,
[16]    "score_mode": "sum",
[17]    "similarity": "classic",
[18]    "store": true,
[19]    "update": "normal"
[20] }
[21]
[22] "name": "APL_SocialIndex",
[23] "type": "index",
[24] "fields": [
[25]     "id"
[26] ],
[27] "options": {
[28]     "analyze": true,
[29]     "boost": 1.0,
[30]     "cache": true,
[31]     "comment": "Index for Social Data",
[32]     "create_in_background": true,
[33]     "max_shard_size": "100mb",
[34]     "min_shard_size": "100mb",
[35]     "norms": false,
[36]     "order": 2,
[37]     "score_mode": "sum",
[38]     "similarity": "classic",
[39]     "store": true,
[40]     "update": "normal"
[41] }
[42]
[43] "name": "APL_BusinessIndex",
[44] "type": "index",
[45] "fields": [
[46]     "id"
[47] ],
[48] "options": {
[49]     "analyze": true,
[50]     "boost": 1.0,
[51]     "cache": true,
[52]     "comment": "Index for Business Data",
[53]     "create_in_background": true,
[54]     "max_shard_size": "100mb",
[55]     "min_shard_size": "100mb",
[56]     "norms": false,
[57]     "order": 3,
[58]     "score_mode": "sum",
[59]     "similarity": "classic",
[60]     "store": true,
[61]     "update": "normal"
[62] }
[63]
[64] "name": "APL_CategoriesIndex",
[65] "type": "index",
[66] "fields": [
[67]     "id"
[68] ],
[69] "options": {
[70]     "analyze": true,
[71]     "boost": 1.0,
[72]     "cache": true,
[73]     "comment": "Index for Categories Data",
[74]     "create_in_background": true,
[75]     "max_shard_size": "100mb",
[76]     "min_shard_size": "100mb",
[77]     "norms": false,
[78]     "order": 4,
[79]     "score_mode": "sum",
[80]     "similarity": "classic",
[81]     "store": true,
[82]     "update": "normal"
[83] }
```



- No need to install Neo4j → **bit.ly/neo4joscon**
 - Neo4j Sandbox. Private hosted Neo4j instances
 - **Start with this!**
- Jupyter notebook → **bit.ly/neo4jnotebook**
 - Topic clustering, and NLP
 - *Will be used later*

Neo4j Sandbox

 Login

1) GOTO:

bit.ly/neo4joscon

Get Started with Graphs

No Download Required

The Neo4j Sandbox enables you to get started with Neo4j, with built-in guides and sample datasets for popular use cases. No Neo4j experience necessary. Great for RDBMS experts.

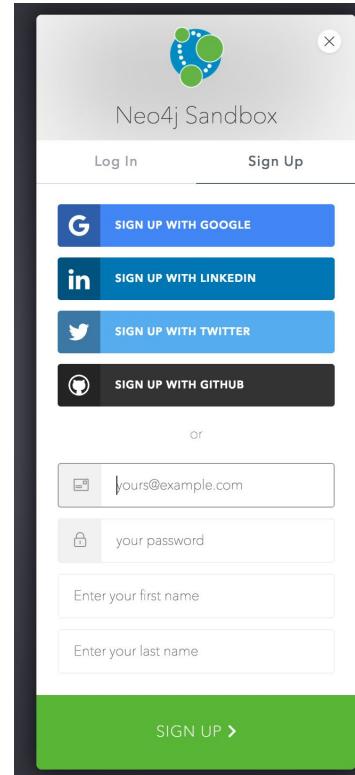
 Start Now



1) GOTO:

bit.ly/neo4joscon

2) Sign Up for Neo4j
Sandbox



The image shows the sign-up page for the Neo4j Sandbox. At the top, there is a logo and tabs for "Log In" and "Sign Up". Below these are four social media sign-up buttons: "SIGN UP WITH GOOGLE" (blue), "SIGN UP WITH LINKEDIN" (dark blue), "SIGN UP WITH TWITTER" (light blue), and "SIGN UP WITH GITHUB" (black). A "or" link leads to a standard sign-up form with fields for email ("yours@example.com"), password ("your password"), first name, and last name. A large green "SIGN UP >" button is at the bottom.

Neo4j Sandbox

Log In Sign Up

G SIGN UP WITH GOOGLE

in SIGN UP WITH LINKEDIN

Twitter SIGN UP WITH TWITTER

Github SIGN UP WITH GITHUB

or

yours@example.com

your password

Enter your first name

Enter your last name

SIGN UP >

Launch a New Sandbox

Each sandbox includes data, interactive guides with example queries, and sample code.



Recommendations



Generate personalized real-time recommendations using a dataset of movie reviews.

[Launch Sandbox](#)

OSCON 2017 Neo4j Workshop



OSCON 2017 Neo4j workshop. Learn how to build a real time recommendation system with Neo4j

[Launch Sandbox](#)

Launch me!



Trumpworld



Explore connections in and around the Trump Administration using this dataset from BuzzFeed.

[Launch Sandbox](#)

Twitter



If signed into Neo4j Sandbox using Twitter, this Sandbox will allow you to Graph your Twitter network.

[Launch Sandbox](#)

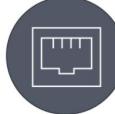
Legis-Graph



US Congress modeled as a Graph - bills, votes, members, and more.

[Launch Sandbox](#)

Network and IT Management



Dependency and root cause analysis + more for network and IT management

[Launch Sandbox](#)

bit.ly/neo4joscon

Your Current Sandboxes

OSCON 2017 Neo4j Workshop

[Get Started](#)[Details](#)[Data Model](#)[Code](#)[Advanced▼](#)

Get Started with your Neo4j Sandbox

- ① Visit the [Neo4j Browser](#). Login with the credentials found under the "Details" tab above. A tutorial will guide you through the datamodel and example data, while teaching you how property graphs work in real-world use cases.
- ② Start building your application backed by Neo4j. Write your own code, in PHP, Java, JavaScript, Python, or one of any number of other languages, using [templates provided](#).
- ③ [Download Neo4j](#) to your own computer, or start a long-living Neo4j instance in the cloud on [AWS or other hosting platforms](#).

Click “Neo4j Browser”

bit.ly/neo4joscon

You should see this:

Neo4j Browser

- Query workbench / visualization for Neo4j
- Interactive “guides” for our tutorial today
- Embed content, queries

The screenshot shows the Neo4j Browser interface. At the top, there's a toolbar with various icons and a URL bar pointing to <https://10-0-1-153-32894.neo4jsandbox.com/browser/>. Below the toolbar, there's a sidebar with icons for database, star, and folder. The main area has two tabs: one with the query `:play https://guides.neo4j.com/reco-oscon` and another with `:play start`. The first tab displays a "Our schedule for the day" section with a numbered list of items, each with a blue circular icon and a link: 1. [Recommend Groups by Topic](#), 2. [Groups similar to mine](#), 3. [My Interests](#), 4. [Event Recommendations](#), 5. [Venues](#), 6. [RSVPs](#), 7. [Procedures](#), 8. [Latent Social Graph](#), 9. [Scoring recommendations](#). A large red "meetup" logo is overlaid on this section. The second tab shows a "neo4j ENTERPRISE EDITION 3.1.1" logo, followed by four cards: "Learn about Neo4j" (with sub-links for what is a graph database, how to query, and what people do), "Jump into code" (with sub-links for code walk-throughs, RDBMS to Graph, and Query templates), and "Monitor the system" (with sub-links for disk utilization, cache activity, and cluster health).

bit.ly/neo4joscon

(you) - [:HAVE] -> (?)
 (?) <- [:ANSWERS] - (will)

