Turning Data into Knowledge: Modeling and Structuring for Linked Open Data

Jian Qin
School of Information Studies
Syracuse University
Syracuse, NY, USA

NKOS Workshop at ICADL 2016, Seoul, Korea

Big data (and metadata)

In big data waves, it is difficult to see a particular section or find a particular drop.

An issue for knowledge organization

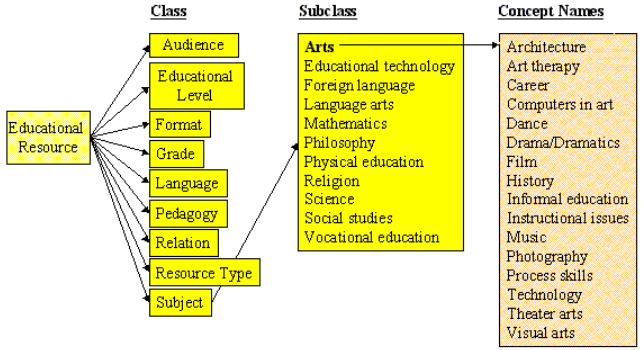
Structured data

Gaps between large data sets and knowledge that can be viewed, interacted, and acted upon

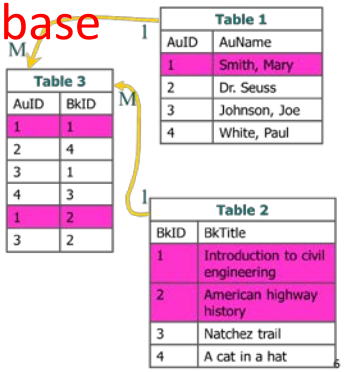
Structured knowledge

Spreadsheet

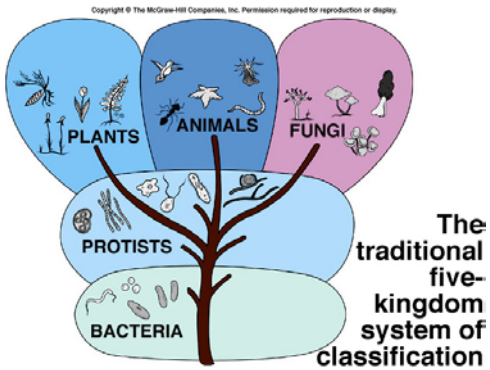
WS	MO	DA	YR	TIME	GAGE FT.	GAGE CM.	DISCH. C.F.S.	DISCH L\S.	INTERV. INCH
1	1	1	08	455	0.240	7.370	0.0700	2.0100	0.01000
1	1	1	08	1062	0.240	7.370	0.0700	2.0100	0.01000
1	1	1	08	1593	0.240	7.460	0.0700	2.0700	0.01000
1	1	1	08	2138	0.240	7.430	0.0700	2.0500	0.01000
1	1	1	08	2400	0.240	7.400	0.0700	2.0300	0.00643
1	1	2	08	645	0.230	7.280	0.0600	1.9500	0.01000
1	1	2	08	1357	0.230	7.260	0.0600	1.9300	0.01000
1	1	2	08	1844	0.230	7.310	0.0600	1.9500	0.01000
1	1	2	08	2400	0.230	7.310	0.0600	1.9700	0.00841
1	1	3	08	447	0.220	6.790	0.0500	1.6600	0.00922
1	1	3	08	1195	0.220	6.700	0.0500	1.6300	0.01000
1	1	3	08	1822	0.220	6.730	0.0500	1.6400	0.01000
1	1	3	08	2384	0.210	6.670	0.0500	1.6100	0.01000
1	1	3	08	2400	0.220	6.730	0.0500	1.6400	0.00033
1	1	4	08	573	0.220	6.790	0.0500	1.6600	0.01000
1	1	4	08	1184	0.220	6.730	0.0500	1.6400	0.01000
1	1	4	08	1725	0.220	6.880	0.0600	1.7000	0.01000
1	1	4	08	2227	0.220	6.820	0.0500	1.6700	0.01000



Relational database



Data to knowledge



The traditional five-kingdom system of classification

School of Information Studies SYRACUSE UNIVERSITY

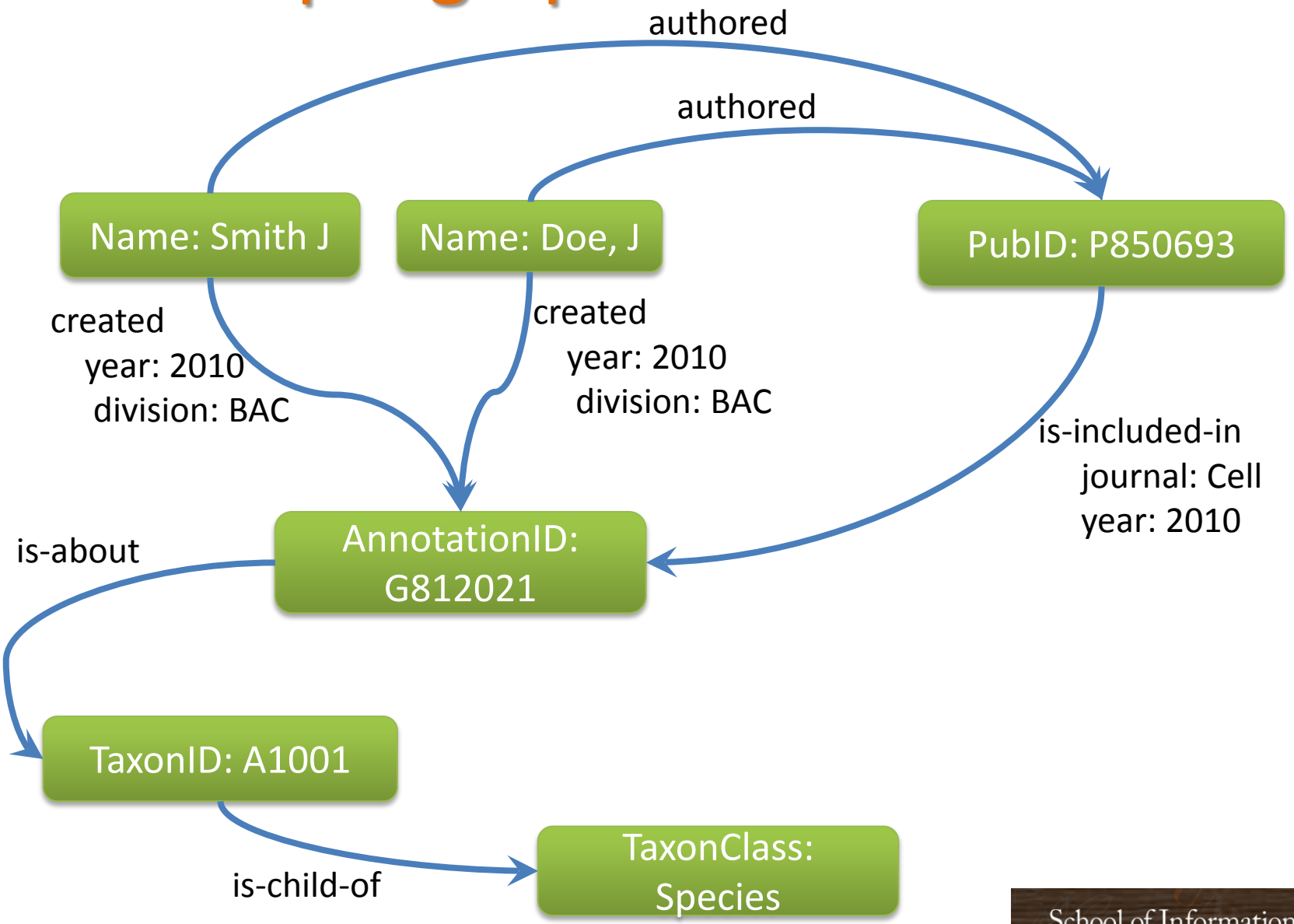
Graph database as a solution

- Relational DBs
  - Not designed to cope with the scale and agility challenges
  - Not built to take advantage of cheap storage and processing power available today
  - Require predefined data schemas
  - Usually scale vertically
- Graph database
  - One of the NoSQL DB types
  - Similar to the abstract model of subject-predicate-object
  - Built to allow the insertion of data without a predefined schema

Graph structures

- Contain:
  - Nodes represent entities such as people, organizations, or things
  - Properties are pertinent information that relate to nodes.
    - i.e. If “iSchool” was a node, it might have properties such as “college”, “Syracuse University”, and “degree program”
  - Edges are the lines that connect nodes to nodes or nodes to properties. They represent the relationship between the two. Most of the important information is stored in the edges.

An example graph



Graph Stores

- Provide index-free adjacency, meaning that every element contains a direct pointer to its adjacent elements and no index lookups are necessary
  - Graph queries largely involve using this locality to traverse through the graph, literally chasing pointers
  - Operations can be carried out with extreme efficiency, traversing millions of nodes per second

Graph Stores

- Place a heavy emphasis on the *relationships* between data objects and are designed to store *interconnected data*

Why do we do it?

How does it work?

How is it related to knowledge organization?

An Example: Converting MySQL To Neo4j

The MySQL database

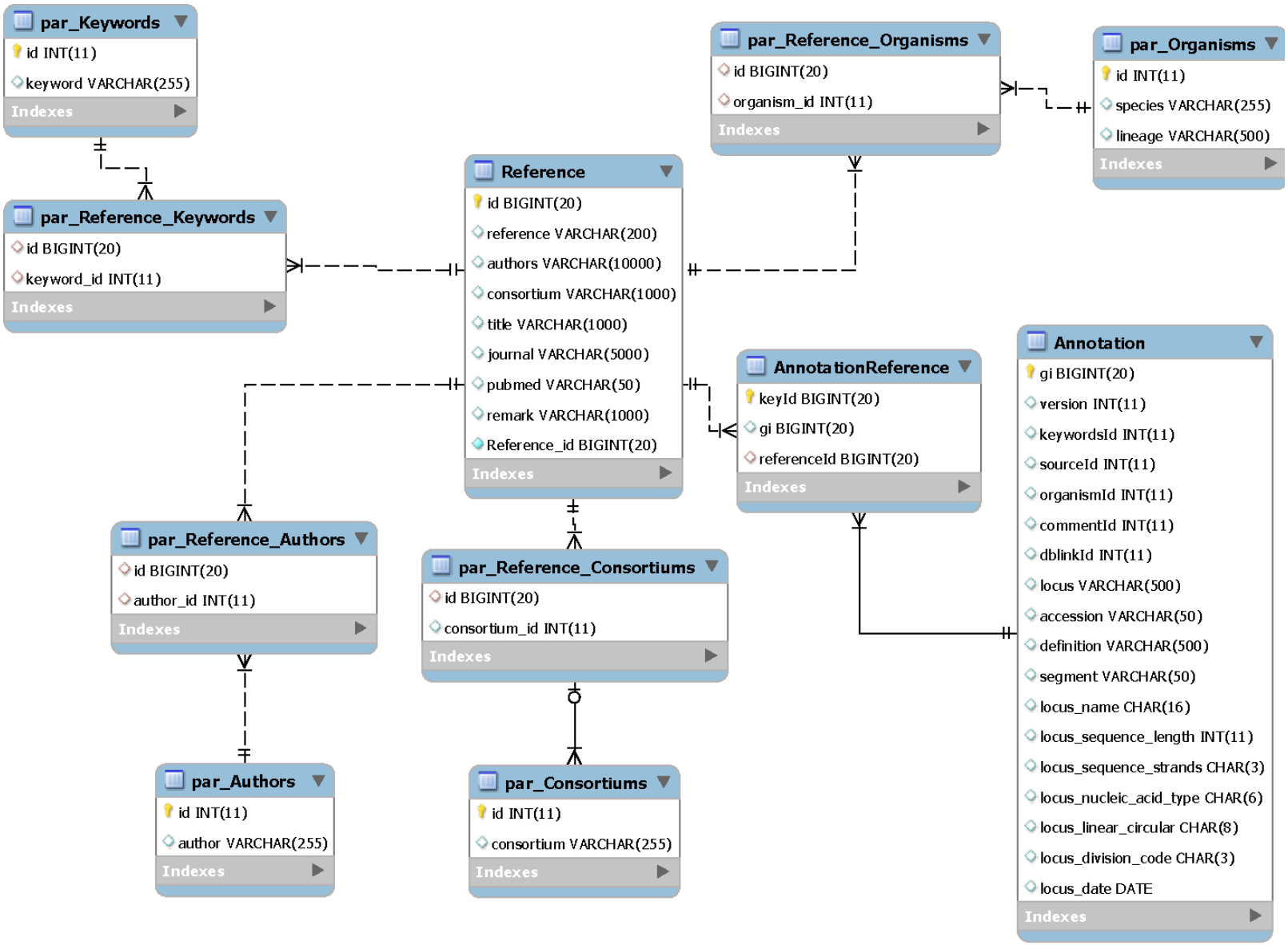
Size of the data:

Property	Overall network	Submission network	Publication network
Total references	1,360,938	1,015,697	345,241
Total vertices	531,019	386,133	404,466
Total edges	121,471,078	101,305,810	9,909,522
Clusters	2,699	4860	1487

Number of annotations by organism

ID	Class name	Parent name	Count
9606	Homo sapiens	Homo	2,0398,647
10090	Mus musculus	Mus	9,789,456
408172	Marine metagenome	Ecological metagenomes	6,261,089
32630	Synthetic construct	Artificial sequences	4,475,898
4577	Zea mays	Zea	3,953,008
9823	Sus scrofa	Sus	3,304,324
77133	Uncultured bacterium	Environmental samples	3,080,129
3702	Arabidopsis thaliana	Arabidopsis	2,337,308
9913	Bos taurus	Bos	2,209,082

It is computationally expensive and slow in response time in querying the data due to the very large size of the tables



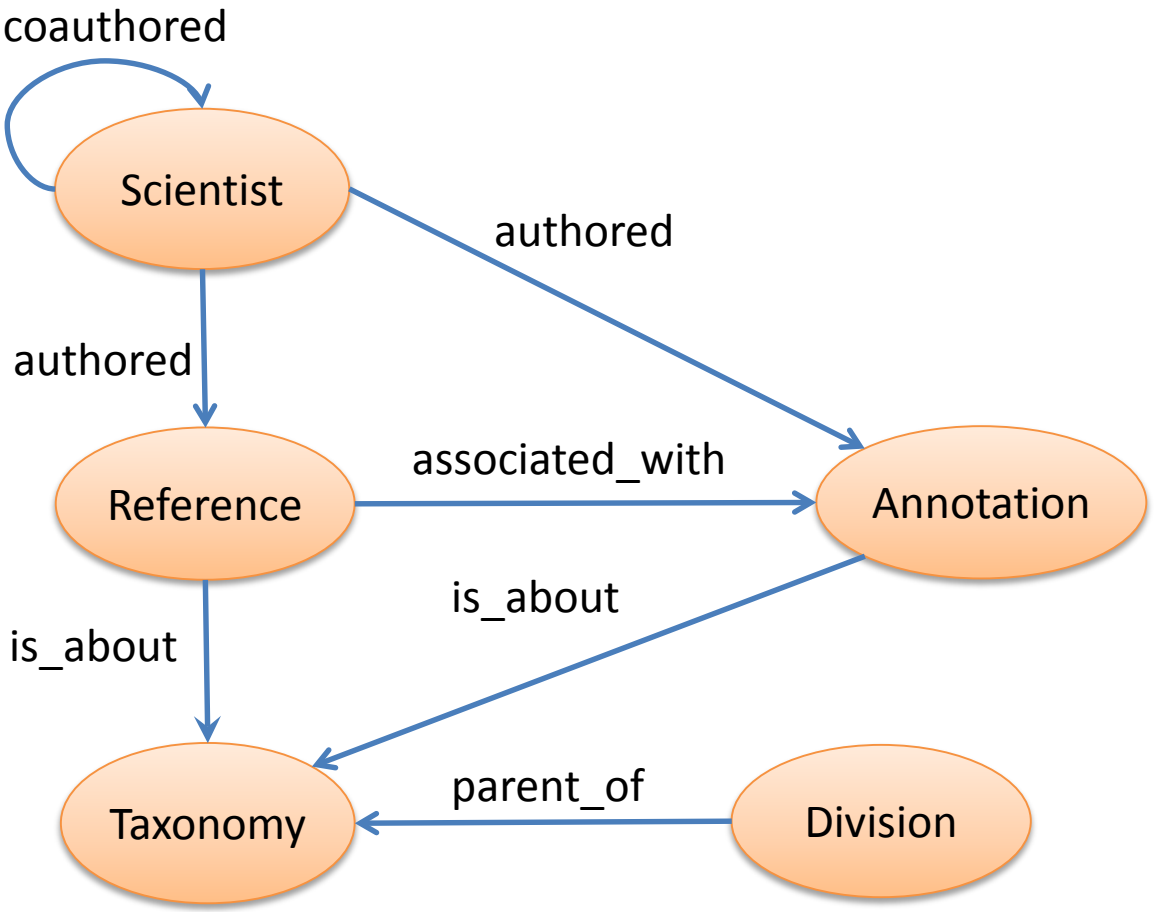
Purpose for migrating from a relational DB to a graph DB

- Avoid too many inner joins in relational database to improve query performance
- Develop data products by embedding predefined queries
- Visualize query results in real time
- Experiment with converting relational data model into a model suitable for Linked Data applications

Strategies

- Popular method: export data as a CSV and import into Neo4j using independent Batch Importer
- To change from relational to graph, data structure may need to be slightly altered
  - Need to create a model to clarify what are entities, relationships and properties
  - Focused on the specific use case
  - Get data from rows and columns to nodes and edges

Developing models

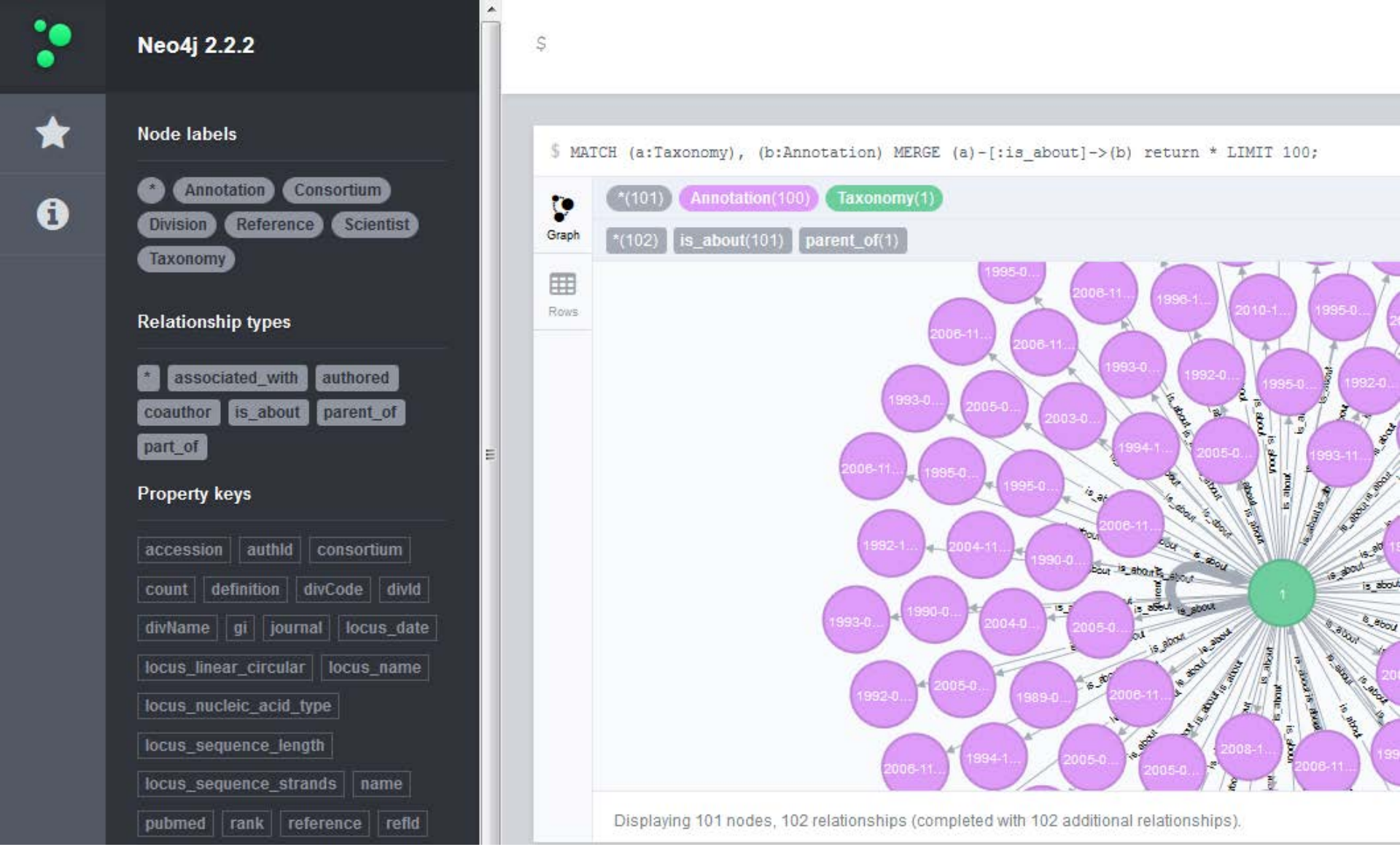


Steps in converting data

- Converting data in MySQL tables into CSV files
- Write Cypher queries to import data into Neo4j

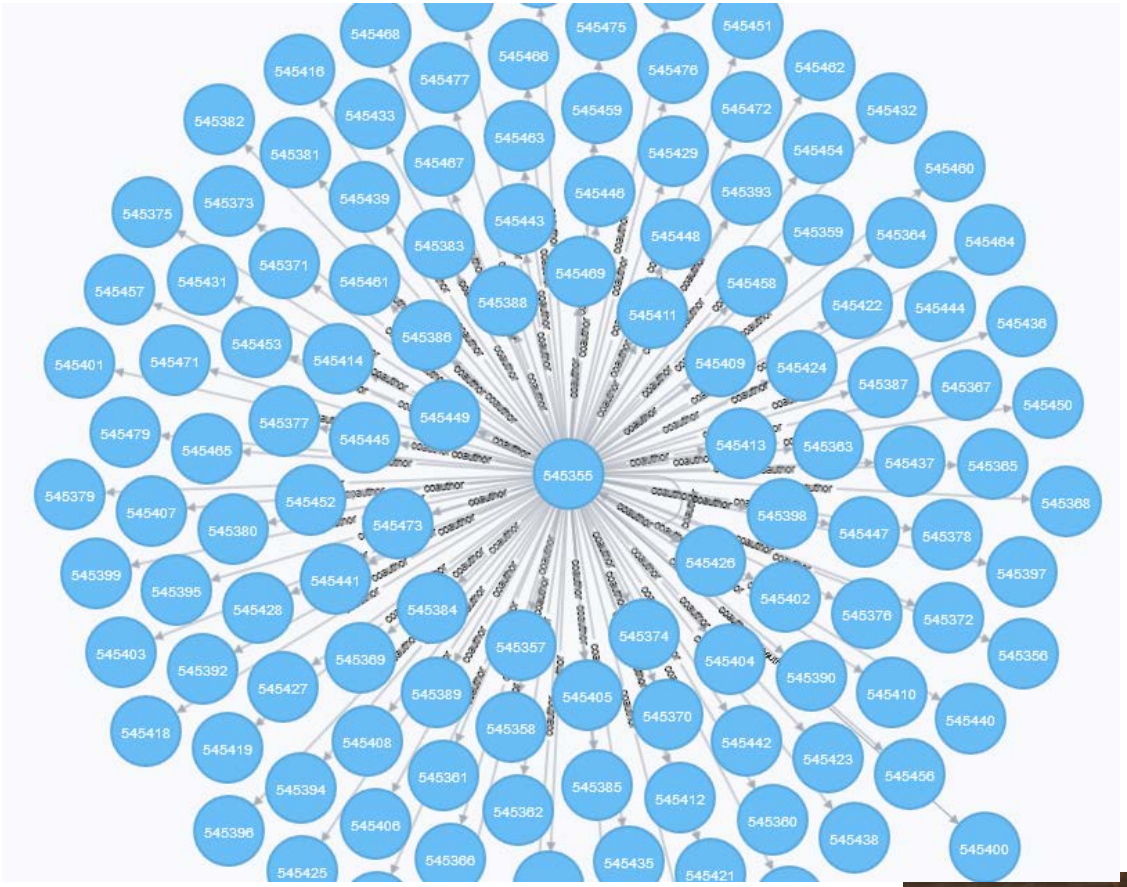
Sample code for setting up the Neo4j server

```
#Annotation
using periodic commit 1000
load csv from
'file:/home/neo4j/Annotation_1.csv' as row
fieldterminator ';'
create (:Annotation {gi: toInt(row[0]),
version: toInt(row[1]),
accession: row[2],
definition: row[3],
segment: row[4],
locus_name: row[5],
locus_sequence_length: row[6],
locus_sequence_strands: row[7],
locus_nucleic_acid_type: row[8],
locus_linear_circular: row[9],
locus_date: row[10]});
```


Neo4j 2.2.2
Node labels
* Annotation Consortium Division Reference Scientist Taxonomy
Relationship types
* associated_with authored coauthor is_about parent_of part_of
Property keys
accession authId consortium count definition divCode divId divName gi journal locus_date locus_linear_circular locus_name locus_nucleic_acid_type locus_sequence_length locus_sequence_strands name pubmed rank reference refId
$ MATCH (a:Taxonomy), (b:Annotation) MERGE (a)-[:is_about]->(b) return * LIMIT 100;
Graph
Rows
*(101) Annotation(100) Taxonomy(1)
*(102) is_about(101) parent_of(1)
Displaying 101 nodes, 102 relationships (completed with 102 additional relationships).

Sample query

MATCH (a:Scientist), (b:Scientist) MERGE a-[r:coauthor]->(b) RETURN * LIMIT 125

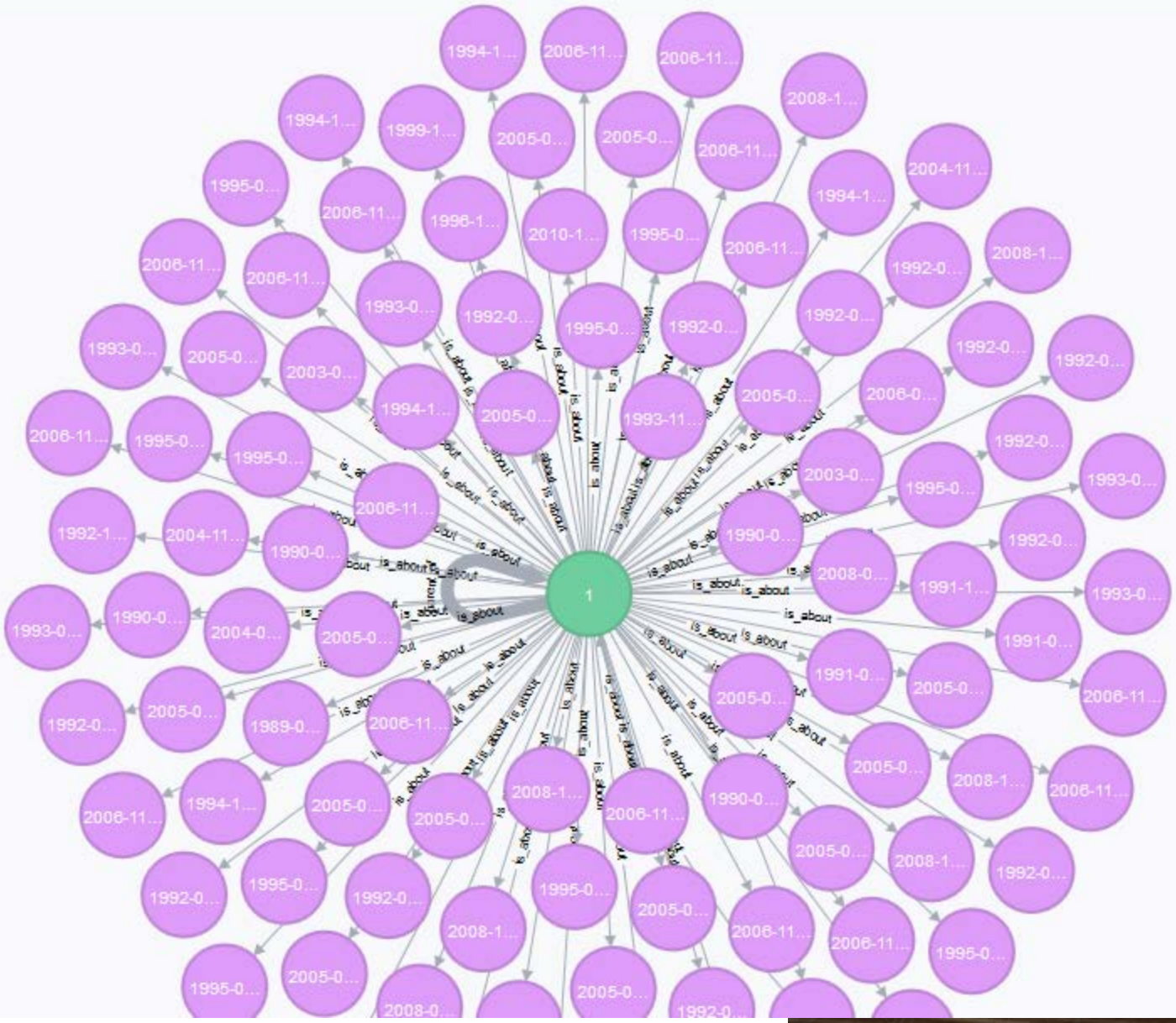


Sample query

```
$ MATCH (a:Taxonomy), (b:Annotation) MERGE (a)-[:is_about]->(b) return * LIMIT 100;
```

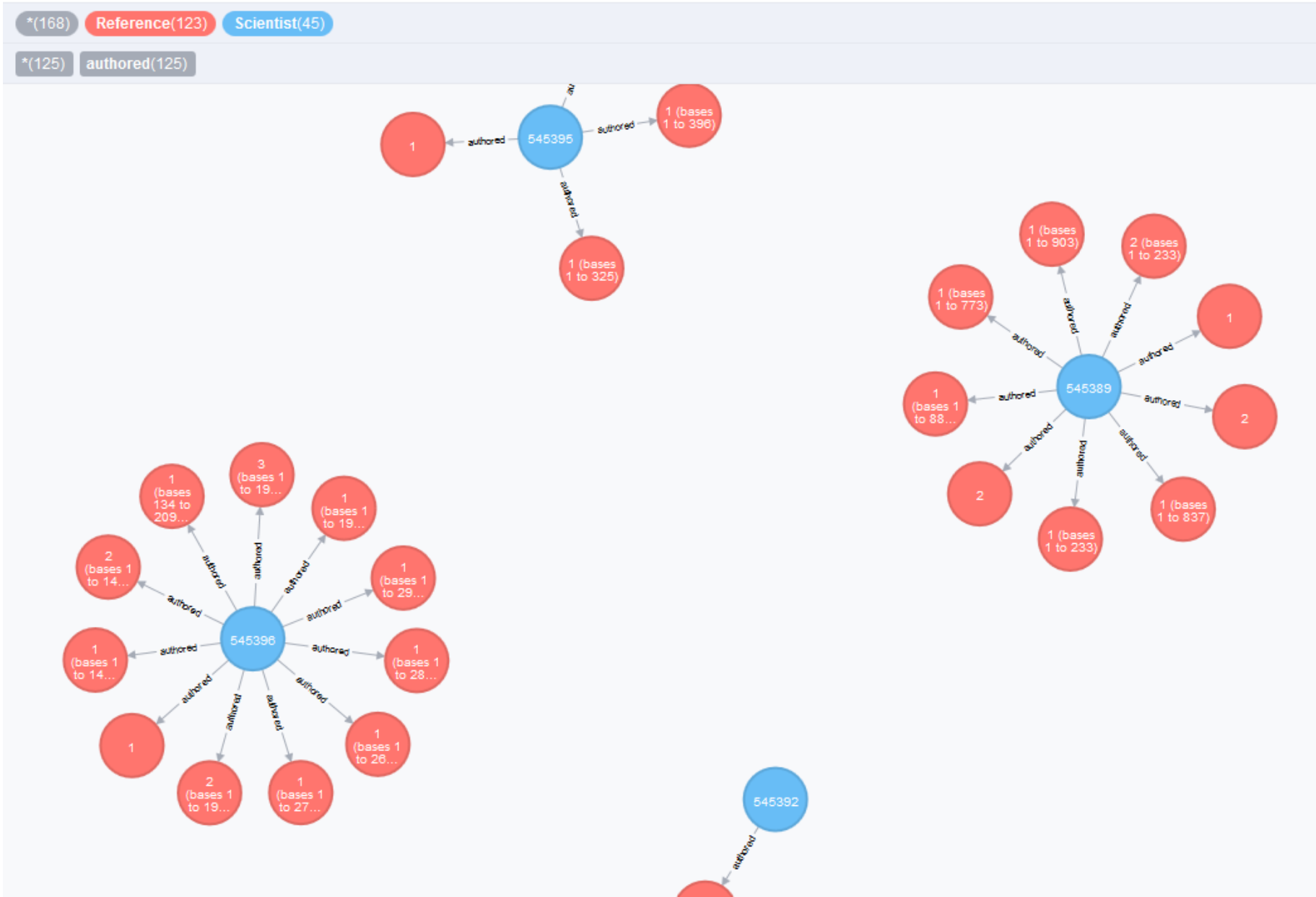
*(101) Annotation(100) Taxonomy(1)

*(102) is_about(101) parent_of(1)



Sample query

MATCH ()-[r]->() RETURN r LIMIT 125



Summary

- Graph databases
  - Useful for presenting relationships/networks
  - Scalable
  - Allow for use together with R, Python, and other languages
- Data structure resembles RDF triples

Conclusion

- Knowledge organization in data-driven environment
  - Goes beyond controlled vocabularies and classification schemes
  - Fills the gap between data and knowledge
  - Models data into the structures for linkable data sets and real-time interaction between users and data as well as between computers

Sample queries

MATCH (a:Scientist), (b:Scientist) MERGE a-[r:coauthor]->(b) RETURN * LIMIT 25

MATCH ()-[r]->() RETURN r LIMIT 125

MATCH (a:Taxonomy), (b:Annotation) MERGE (a)-[:is_about]->(b) return * LIMIT 50

http://neo4j-genbank.syr.edu:7474/browser/