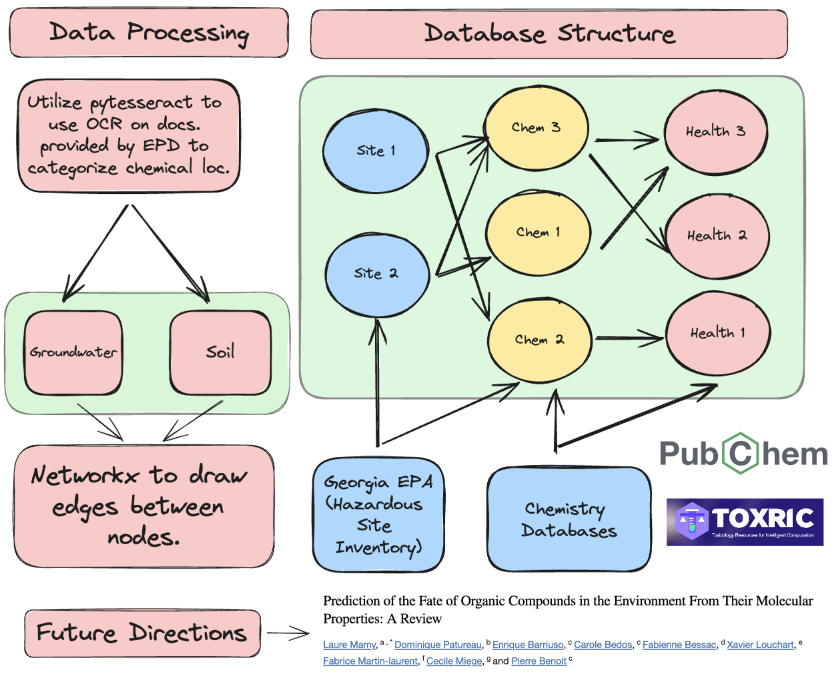
Documentation For Georgia EPD Hazardous Waste Site to Chemical Relation Neo4j Database Hosting

Ken Nakatsu

November 2023


Data Processing
Database Structure
Utilize pytesseract to use OCR on docs. provided by EPD to categorize chemical loc.
Groundwater
Soil
Networkx to draw edges between nodes.
Future Directions
Site 1
Site 2
Chem 3
Chem 1
Chem 2
Health 3
Health 2
Health 1
Georgia EPA (Hazardous Site Inventory)
Chemistry Databases
PubChem
TOXRIC
Toxicology Resources for Intelligent Computation
Prediction of the Fate of Organic Compounds in the Environment From Their Molecular Properties: A Review
Laure Mamy, a, * Dominique Patureau, b Enrique Barriuso, c Carole Bedos, c Fabienne Bessac, d Xavier Louchart, e Fabrice Martin-laurent, f Cecile Miege, g and Pierre Benoit c

1 Table of Contents

2 Data Processing and Graph Creation

2.1 Overview

Building the graph databases generally consists of two parts (a) preprocessing the data and (b) drawing edges between nodes. The advantages of a graph database is that we can quickly and easily update the database with complex information. Information on how to do this is at the end of this document. This can include metadata, chemical structures, and more. Imagine this—a collaborative team of social scientists, computational chemists, and environmental scientists all appending data to the graph database to add data such as chemical half-lives (through the properties of the structure), reactions that may interfere with biotic processes, or even geographical data to site nodes.

2.2 Data Sources

1. EPD Hazardous Site Inventory (For site data): Link
2. TOXRIC database which connects chemicals to their potential health or environmental effects: Link
3. Hazardous Substances Data Bank (HSDB) from Pubchem: Link

3 Accessing Data

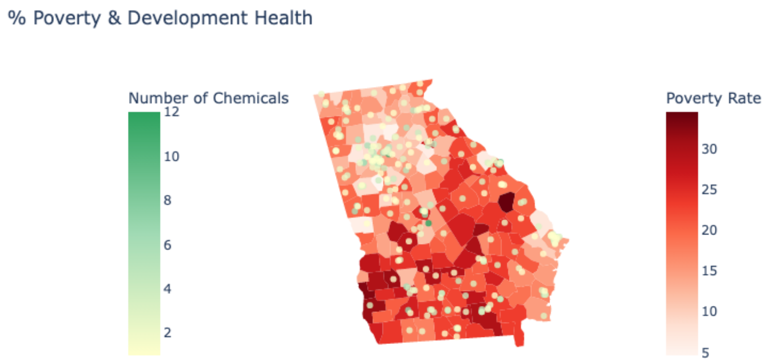
This section will provide a few different cases of useful queries. Documentation for the Cypher language which Neo4J uses to retrieve data can be found here: Link

3.1 Obtain All Sites that Had At Least One Chemical with Developmental Toxicity and Count the Number of Chemicals in Ground Water

Syntax is as follows: MATCH (tempname:Node-Name Filtering Conditions-[edge:Edge-Name]-(tempname2:Node-Name)

```
MATCH (env:Node {id: 'Dev_Toxic'})-[edge:RELATES_TO]
-(pointed:Node)-[tosite:RELATES_TO{category:"Water"}]
-(sitenode:Node {node_class: "Site"})

RETURN sitenode.label as Site_Name, count(*) as counts,

ORDER BY counts DESC
```

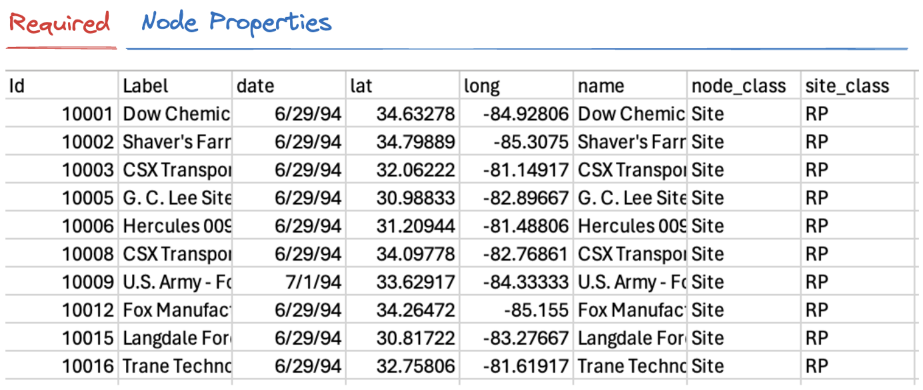
What this command is doing is essentially obtaining finding all nodes that are one degree away from the developmental toxicity node. This returns all of the chemicals. Then the second "query" in the command is getting all of the nodes one more degree away from that one. We then filter for edges that are "Water" edges which indicates that the chemical has come from a water source. Finally, we need to ensure that only Site is returned. We then count the number of times it appears, and thus we obtain the number of chemicals associated with each site.



It is now trivial to plot information about the number of chemicals. The database is easily adaptable to answer many questions once a few commands are learned!

4 Updating Data

4.1 Node File Structure


Required
Node Properties

Id	Label	date	lat	long	name	node_class	site_class
10001	Dow Chemic	6/29/94	34.63278	-84.92806	Dow Chemic	Site	RP
10002	Shaver's Farr	6/29/94	34.79889	-85.3075	Shaver's Farr	Site	RP
10003	CSX Transpoı	6/29/94	32.06222	-81.14917	CSX Transpoı	Site	RP
10005	G. C. Lee Site	6/29/94	30.98833	-82.89667	G. C. Lee Site	Site	RP
10006	Hercules 009	6/29/94	31.20944	-81.48806	Hercules 009	Site	RP
10008	CSX Transpoı	6/29/94	34.09778	-82.76861	CSX Transpoı	Site	RP
10009	U.S. Army - Fc	7/1/94	33.62917	-84.33333	U.S. Army - Fc	Site	RP
10012	Fox Manufac	6/29/94	34.26472	-85.155	Fox Manufac	Site	RP
10015	Langdale For	6/29/94	30.81722	-83.27667	Langdale For	Site	RP
10016	Trane Technc	6/29/94	32.75806	-81.61917	Trane Technc	Site	RP

4.2 Creating new nodes

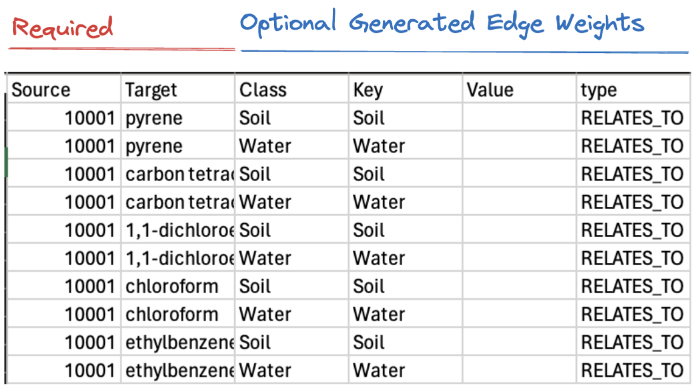
Loading in nodes is quite easy as new nodes can always be created.

1. Load in csv file
2. Assign node properties

```
LOAD CSV WITH HEADERS FROM <nodes.csv> AS row

CREATE (:Node { id: row.Id, label: row.Label, prop1: row.prop1})
```

4.3 Edge File Structure

Ensure to generate a file that has the targets and the sources. Make sure to keep a file with all of the nodes, keeping backups, to ensure that all source-¿target relationships are valid. Edge weights are arbitary and can be assigned as needed.


Required
Optional Generated Edge Weights

Source	Target	Class	Key	Value	type
10001	pyrene	Soil	Soil		RELATES_TO
10001	pyrene	Water	Water		RELATES_TO
10001	carbon tetra	Soil	Soil		RELATES_TO
10001	carbon tetra	Water	Water		RELATES_TO
10001	1,1-dichloro	Soil	Soil		RELATES_TO
10001	1,1-dichloro	Water	Water		RELATES_TO
10001	chloroform	Soil	Soil		RELATES_TO
10001	chloroform	Water	Water		RELATES_TO
10001	ethylbenzene	Soil	Soil		RELATES_TO
10001	ethylbenzene	Water	Water		RELATES_TO

4.4 Creating new edges

The process of creating edges involves:

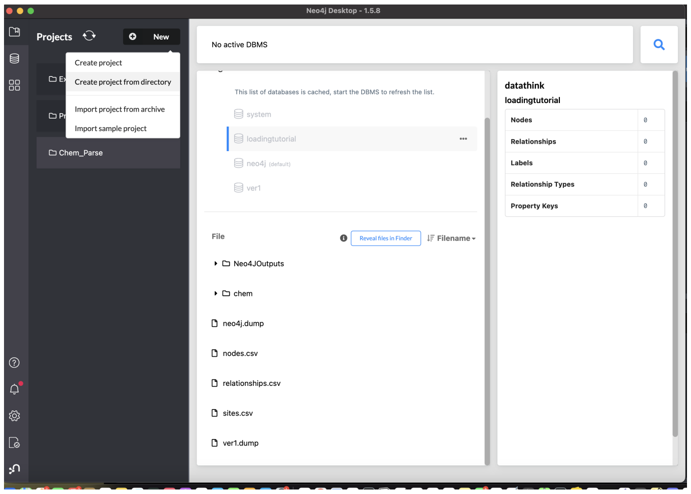
1. Loading your csv file
2. Assigning them to variables
3. Matching source and targets to the Node object
4. Finally, creating edges and assigning them weights

```
LOAD CSV WITH HEADERS FROM <edges.csv> AS row

WITH row.Key as class, row.Value as value,
row.Source as Source, row.Target as Target

MATCH (source:Node {id: Source}), (target:Node {id: Target})

CREATE (source)-[:RELATES_TO {continuous:value, category:class}]->(target)
```

5 Loading in the Graph Database and Setting up The Environment

This section is about loading the graph dump that has been created. Please download and follow the instructions for your respective platform. Download

5.1 Creating the project



This list of databases is cached, start the DBMS to refresh the list.

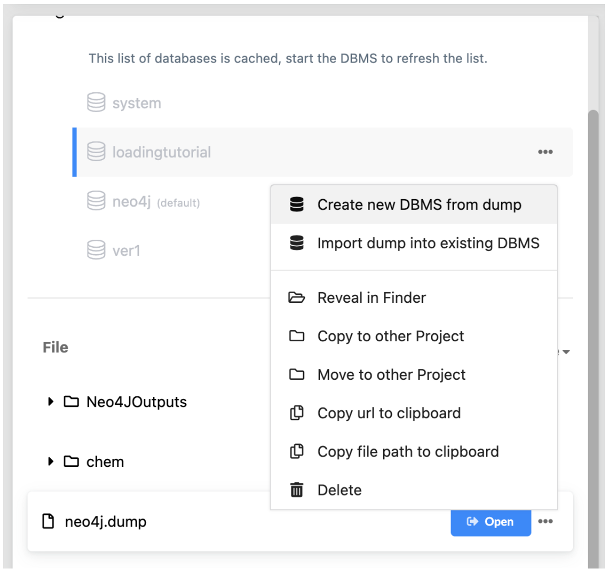
datathink
loadingtutorial

Nodes	0
Relationships	0
Labels	0
Relationship Types	0
Property Keys	0

```
Neo4JOutputs
chem
neo4j.dump
nodes.csv
relationships.csv
sites.csv
ver1.dump
```

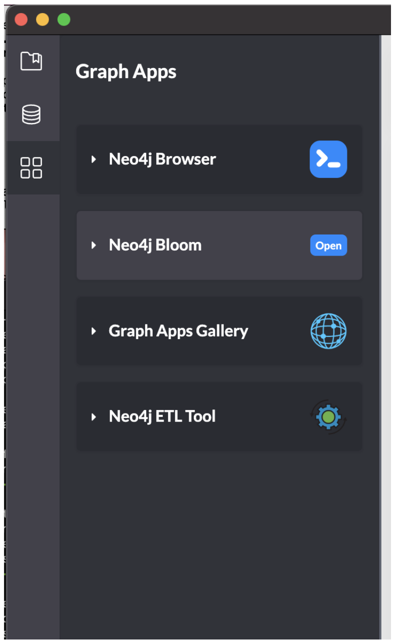
Associate it with the directory that has the dump file. The dump file is included in the github repository associated with this project.

5.2 Loading the dump

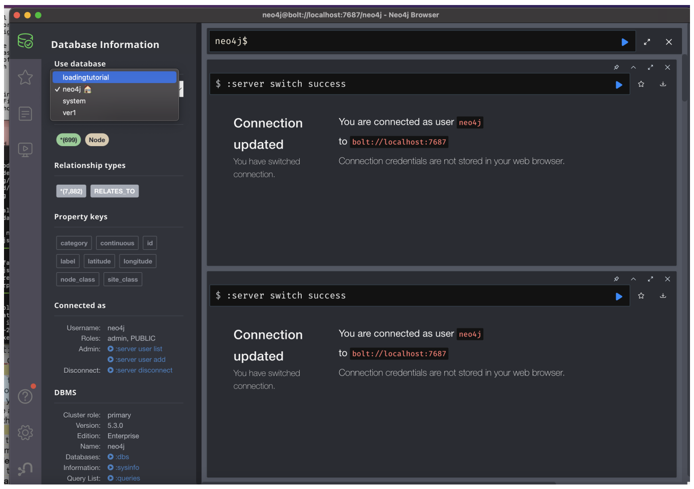

This list of databases is cached, start the DBMS to refresh the list.

Load in the dump by creating new DBMS or importing it into an existing one. Name it and give it a password if prompted.

5.3 Opening the browser and Loading Database IN



Click on the four boxes on the left. Click on the browser.



Username: neo4j
Roles: admin, PUBLIC
Admin: :server user list
:server user add
Disconnect: :server disconnect

Cluster role: primary
Version: 5.3.0
Edition: Enterprise
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries

```
neo4j$
```

```
$ :server switch success
```

Connection updated

You have switched connection.

You are connected as user `neo4j`

to `bolt://localhost:7687`

Connection credentials are not stored in your web browser.

```
$ :server switch success
```

Connection updated

You have switched connection.

You are connected as user `neo4j`

to `bolt://localhost:7687`

Connection credentials are not stored in your web browser.

Then when the browser is open (ensure that the database is also started!), click on the drop down menu under Use Database, then click on the name of the dump.