

**School of Computer Science and Engineering**

**J COMPONENT REPORT**

# **Course Title: NoSQL Databases**

**Course Code: CSE3086**

**Slot: C2**

**Title: GRAPH AI FOR BIOMEDICAL DATABASES**

|  |  |  |
| --- | --- | --- |
| **Team Member(s):** | **HARSHINI K AIYYER** | **19MIA1050** |

# **TABLE OF CONTENTS**

1. Abstract
2. Introduction
3. Literature Review
4. Software used to develop the project
5. Dataset description
6. Project Modules
7. Innovation
8. Result, Conclusion and Future Enhancements
9. References

**Abstract**

Rising advent of several diseases has taken a major toll on the society. It disrupts many families, causes enormous economic losses and perhaps forever changes our behaviour. As a result, it raises our awareness of the public health issue. It becomes immediately clear that we the human race need to invest more in the medical research to avert the next calamity. In order to dive into this field of research, we use networks or graphs. Networks or graphs describe a system of interacting elements. For example, in biomedicine and healthcare, they can represent molecular interactions, signalling pathways, disease comorbidities, or healthcare systems. In this project, there has been explored several ways in which the artificial intelligence has acted upon graphs. Graph analytics (also known as "network analysis") refers to an analysis that is based on entities or "graph nodes," which can be things like products, customers, operations, or even devices.

Global organisations and businesses are starting to use graph analytics to help uncover important insights in areas like marketing, fraud detection, supply chain management, and search engine optimization.

This project aims to explore the several use-cases that utilize a graph, through a biomedical database, the use-cases being drafting of knowledge graphs with the intention of studying the dataset through a node-entity relation concept, another use-case being making the graph act as a search engine – where there is a complete implementation of the concept referred to as a full text searching. It searches for the data using the factors such as stop words, stemming, lemmatization, etc. Graph AI also has another use-case in this project – to derive conceptual context between entities through means of developing a web network through the semantic web.

***Keywords – concept, graph, semantic web, knowledge graphs. Nodes, relations***

**Introduction**

When we realise that the natural representation of biomedical data tends to be graph-like, modelling biomedical data as a graph becomes the obvious choice. Data is typically represented in graph databases as binary nodes and edges. Consider labelled property graphs, where an edge leads from one node to another. Some examples include protein-to-protein link interaction, genome sequence identification, disease prediction and classification, etc. Graphs are useful tools for numerous practical applications. They have been applied to classify nodes and develop recommendation systems in social network mining. They have also been used in natural language processing to decipher basic queries and provide responses using relational data. Graphs have been employed in the biomedical field to perform drug repurposing, identify drug-target interactions and rank disease-relevant genes.

The graph in this context, is defined in the following manner - a tool that combines information from one or more expert-derived sources into a graph where nodes stand in for biomedical entities and edges signify connections between those entities. The project is divided into 3 modules – the first module refers to the exploration of the dataset using the several features of the graph database using a knowledge graph. The second module performs a full-text search mechanism using full text searching as well as fuzzy searching mechanisms which are an inbuilt aspect of the graph database. The graph database follows Apache’s Lucene convention for the full text searching. The third module derives contexts and the semantic significance (called ontology) behind them using SPARQL API and derives a knowledge graph.

A knowledge graph contains a domain of applications - namely, recognition of user defined speech and language patterns, also used by e-commerce organizations to analyze sentiments of the general public towards the purchase of a particular product or a dealer, this also helps these companies decide what products to be pushed forward and propagated to the consumers and stakeholders availing the services of the particular company, also giving a further insight into semantic development. A lot of researchers foresee the use of knowledge graphs and graph algorithms in big data applications along with popularly known big data analytic and mining frameworks and open-source applications such as Hadoop and Spark. Knowledge graphs, is a real-world representation of several entities involved in an activity or a concept which may change as per the demands of a given situation. A knowledge graph may also be considered as an instance of a knowledge base that stores the set of conventions and the appropriate knowledge for the intelligent agent to follow.

Large-scale text searches can be conducted using full-text search. To find keywords in all the web pages that it has indexed, a search engine might use a full-text search, for instance. Indexing is the key to this technique. Indexing can be carried out in a variety of ways, including batch indexing and incremental indexing. Then, for any documents that match, the index serves as a comprehensive glossary. The data can then be extracted using a variety of techniques. An inversed index is used by the open-source search engine Apache Lucene to locate the matching items. Each word in our menu search has a link to the appropriate menu item. An ontology refers to a structured representation of any complex idea that might exist, or to grasp the very same idea in extremely simple terms. This has risen from the need for representations in several knowledge databases that are currently that act as storehouses of information. Due to an ever-increasing amount of data that needs to be stored for future purposes, in case of a possibility for reusing the same for different purposes.

A lot of the consumers on the other side of the designing several models, need to assimilate or grasp the concepts of a complex idea - ideally a reason to convert such concepts into visualizations, either posed as a general representation or as a domain use-case, which are referred to as knowledge graphs. The key difference between an ontology and a knowledge graph is that the former refers to a bunch of frameworks defined right from all information available on that particular knowledge and the latter is more focussed on the applications required to be built by the user.   These kinds of instances are semantic in nature owing to its relation to lexical semantics, that looks forward to words, their context and the meaning behind them. Ontology graphs make their mark during the formative years of Web 3.0 development, where the ultimate aim is to develop and harness information that can make humans and machines understand it alike. Entities in an ontology graph are connected mutually to each other via some relations that are defined by vocabularies present in ontological libraries such as RDF. All ontology libraries work on the principle of the triple - the subject, predicate and the object.

Some popular instances of ontologies could include adopting the Gruninger-Fox methodology for generating an ontology related to farmers whose information constituted the local knowledge database, another example includes the Waste Water Ontology, in the domain of waste water treatment and reusing using the Uschold and King Methodology with an agricultural based ontology framework in concurrence with the policies of the Food and Agriculture Organization (FAO), a branch of the United Nations with a noble cause to defeat world hunger and issues related to food security  As simple as it can get to explain what an ontology truly is, it can always help to follow ontology libraries, so as to bring some structure and order while deciding to follow the conventions that these libraries seek to provide while forming a vocabulary for a given domain.

**Literature Review**

In the biomedical field, the need to shift toward thinking of graphs as biological knowledge repositories is particularly clear. Since many significant diseases have been studied in-depth for many years, a vast and diverse body of knowledge has been amassed. Because there is so much interconnected data, it is difficult to explore it using a network visualizer, and the heterogeneity of the data can make it challenging to manage it in a relational database system in an efficient manner. In order to give biological context to disease-related genes and proteins, we have investigated the potential of using a graph database to facilitate data management and analysis. In this use case, a prototype disease map for a complex disease was built and queried using the well-known Neo4j graph database [1][2]. Developing parsers for such semi-structured data can be particularly challenging because relational databases demand that all data be transformed to adhere to a predefined schema. Additionally, biological research is unpredictable because new kinds of information keep arising at uninidentified intervals [7]. When new analytical tools or new data resources are released, for instance, information can emerge with little advance notice. Capturing these new data types typically necessitates redesigning the database schema in a relational database management system, which is typically a time-consuming and expensive process [5]. Therefore, the quick and seamless integration of new data types would be especially well-suited to support the dynamic and constantly evolving needs of the biomedical research community [9]. Capturing these new data types in a relational database management system frequently necessitates the re-design of a database schema, which is typically a complex and costly process. However, in a research setting, this information is frequently of utmost significance and can have an impact on our comprehension of the roles of genes and pathways in disease networks. Therefore, a framework that makes it simple to quickly integrate new data types would be especially well-suited to support the constantly evolving needs of the biomedical research community. Highly connected, unevenly distributed, semi-structured, and unpredictable data, which are common in studies of biological systems, can be naturally represented by graph databases [3][4]. Additionally, they provide flexible and agile solutions that make it simple to incorporate new data types [6].

Such facilities can be necessitated by the addition of knowledge graphs. Inference techniques such as automatic reasoning, natural language processing, statistical analysis, and user crowdsourcing are used to derive new knowledge from existing data sets such as Freebase [9] or Wikidata [8] in the knowledge graph, a sizable heterogeneous knowledge base modelled through graphs and ontologies. In addition to this definition, we propose here that these knowledge bases typically rely on informal ontologies and schemas as opposed to those found in conventional expert systems or in a typical description logic-based ontology (and hence a typical OWL-based data set). On the one hand, properly formalised data enable accurate information retrieval and extraction, as well as sophisticated automatized logical reasoning. But working with extremely formal data, we can conclude from our findings that these knowledge bases frequently rely on ontologies and schemas that are less formal than those found in conventional expert systems or in a typical description logic-based ontology (and hence a typical OWL-based data set). On the one hand, properly formalised data enable accurate information extraction and retrieval as well as sophisticated automatized logical reasoning. Even when they are domain experts, dealing with extremely formal data models can be challenging for software developers and end users. This challenge is exacerbated by the web's large amount of data, which also means large amounts of data noise and varying syntax and formats [16]. Another application of knowledge graphs include the simulation of a full text search engine which is very much in use owing to the nature of the graph that is being used to detect drug-drug link interaction, drug-target antidotal interaction; there is a need for a search text engine that will scan through the many pre-existing medical documents in order to study and understand what connects with what. Text search engines play an important role in filtering information for a given query. Search engines such as Textpresso [12] divided the entire text of research articles into sentences, then annotated each sentence's terms with tags. These tags were grouped into categories, which are collections of words and expressions that have similar semantic properties. The categories were then formally defined and organised in a shallow ontology (i.e., organised in a hierarchy), with the aim of improving query precision. There are three ways to use Textpresso full text searches: 1) by typing words or phrases into a search field akin to those found on popular search engines; 2) by choosing one or more categories from menus that cascade; or 3) by combining keywords and categories. Users were shown lists of individual sentences from search results that could be sorted by relevance. Each research article in the Textpresso corpus goes through a number of processing steps before being ready for the front-end system. Additionally, the files that have been processed will be accessible to text mining and machine learning algorithms.

Manufacturing is transitioning from mechanisation to intelligence and digitization as information technology advances. The next generation of the Web aims to address this issue by making Web resources more accessible to automated processes by including information that describes Web content in a machine-readable and manipulable format [18]. Ontologies emerged as an important tool for representing I4.0 domain knowledge in order to support integration and interoperability. Ontology is defined as the formal, explicit specification of a shared conceptualization [19][20]. A concept and its relationship, as well as axioms, are the fundamental elements of an ontology. When instances of concepts are added to an ontology, it becomes a knowledge base, also known as a knowledge graph. Reference architectures use analogous standards and face interoperability issues due to comparable standards. To enable interoperability in smart industries, several reference architectures and standards have been proposed. The resources and processes, as well as their communication, are defined, classified, aligned, and integrated by standard and reference architectures [17]. Several research studies used ontologies to communicate the mutual knowledge of standards to resolve interoperability conflicts in industry 4.0 standards. Recently, the problem of standard interoperability has been addressed by characterising the standards and proposing standard ontology (STO). This includes using certain type of ontologies for various domains, mainly healthcare, e-commerce, sports, entertainment, finance, etc. Ontologies can be used profitably in e-commerce sites (McGuinness 1998), where they can facilitate machine-based communication between buyer and seller, enable vertical integration of markets, and allow descriptions to be reused in different marketplaces; and in search engines [14], where they can help searching go beyond the current keyword-based approach, and allow pages containing syntactically different but semantically similar words/phrases to be found. The basic idea of having a formally-specified web language that can represent ontology information, in particular, will go a long way toward allowing computer programmes to interoperate without pre-existing, outside-of-the-web agreements. If this language also has an effective reasoning mechanism, computer programmes can manipulate this interoperability information and determine whether a common meaning for the information that they pass back and forth exists [16].

**Software/Tools Used**

**Neo4j**

The project was done completely in Neo4j, that is an open-source graph database that caters to the ACID properties along with belonging to the group of NoSQL databases. The query language used was Cypher – which Neo4j’s own querying database language.

***Other tools used***

**GDS in Neo4j**

The Neo4j Graph Data Science (GDS) library exposes as Cypher procedures efficiently implemented, parallel versions of common graph algorithms. GDS also includes machine learning pipelines for training predictive supervised models to solve graph problems like missing relationship prediction.

**APOC Data Science Library**

The APOC library consists of many (about 450) procedures and functions to help with many different tasks in areas like collections manipulation, graph algorithms, and data conversion.

**Neosemantics**

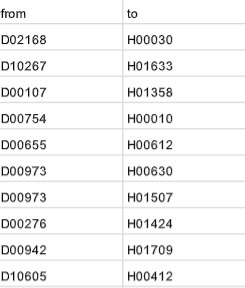
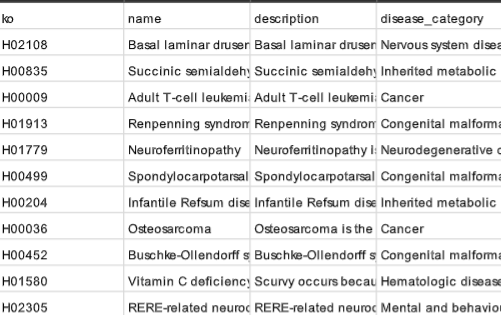
Neosemantics (n10s) is a plugin for Neo4j that allows the use of RDF and its associated vocabularies such as (OWL, RDFS, SKOS, and others). RDF is a W3C standard data interchange model.

**Wikidata API for Neo4j**

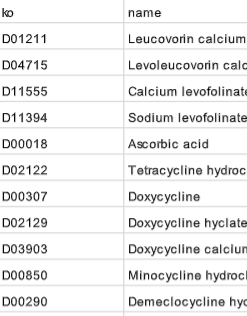
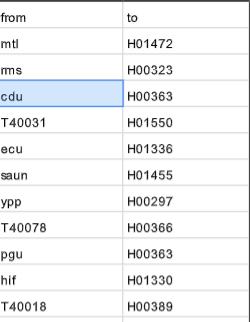
Information was fetched the information regarding the diseases from the WikiData API. the [Neosemantics](https://neo4j.com/labs/neosemantics/4.0/) library to our stack. It is used to interact with RDF data in the Neo4j environment. We can either import RDF data to Neo4j or export property graph model in RDF format.

**Dataset Description**

The datasets used here are ideal for a graph database referred to as the KEGG database or the  [Kyoto Encyclopedia of Genes and Genomes (KEGG)](https://www.genome.jp/kegg/) database. This database contains details about many known human diseases. Moreover, together with other databases such as Genome, Drug and Genes, KEGG provides us with a very extensive network of knowledge about disorders, their causes and drugs. The data from the KEGG Disease Database via [API](https://www.kegg.jp/kegg/rest/keggapi.html), was imported into Neo4j, run several analyses and have discovered some interesting information.

Diseases.csv Drug\_disease.csv

Drug.csv Pathogen- disease.csv Pathogen.csv

****

The datasets such as drug, pathogen and disease present individual node features whereas datasets such as pathogen – disease and drug- disease establish links between the nodes.

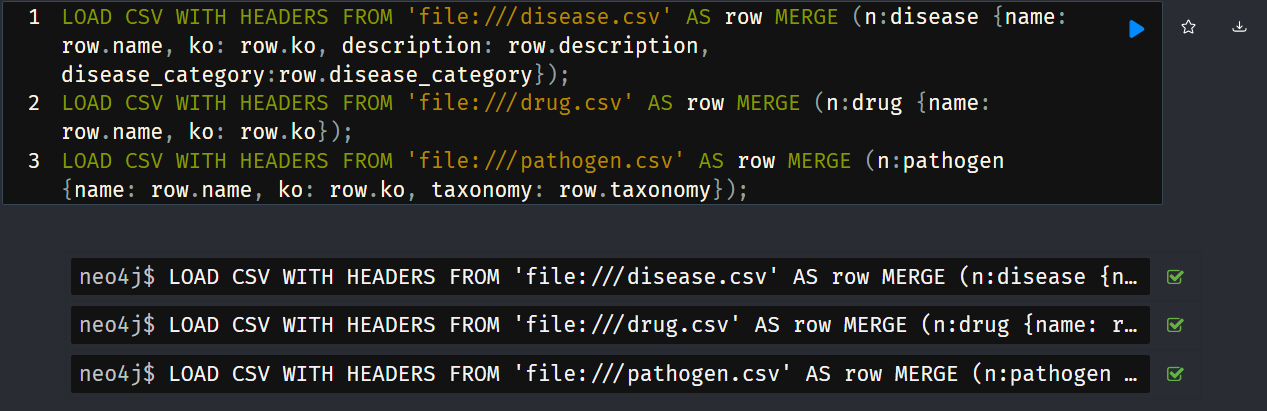
**Project Modules**

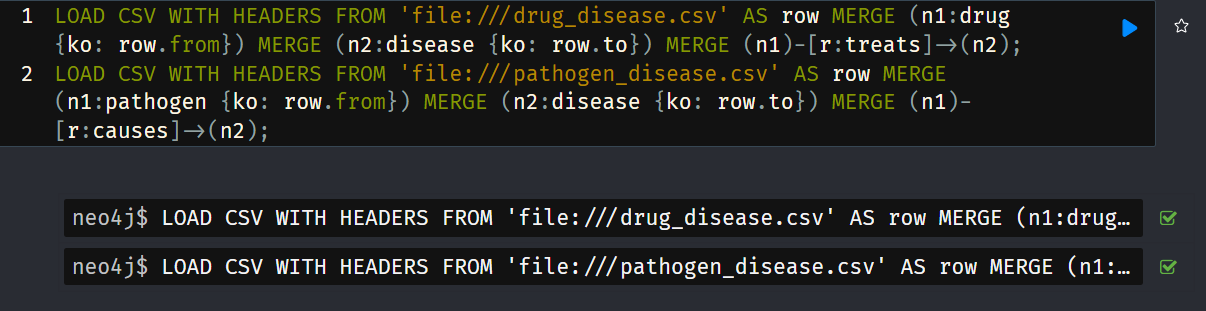
The project is primarily divided into 3 modules. The first module would include some basic visualizations and exploration of data using Neo4j. The second module tries to implement the aspects of full-text searching and fuzzy searching using an inbuilt structure that Neo4j follows which is built on top an open-source search engine known as Apache Lucene. The third and final module is the building of a knowledge graph with the Neosemantics library and some ontologies using the WikiData Open Knowledge database API for terms and entities associated with the KEGG database.

Part – 1 Basic Visualizations

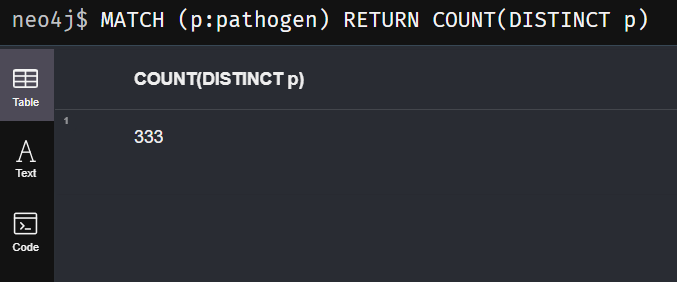
This one was done by applying simple Cypher queries that were applied on the Neo4j Desktop Shell.

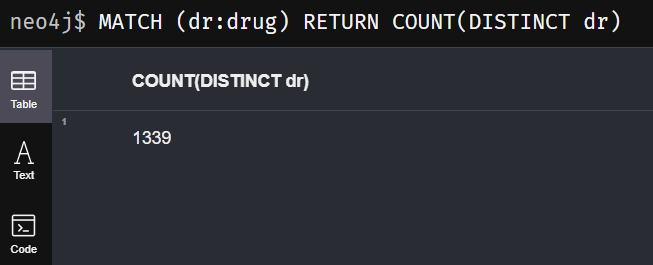
Inserting and reading into the shell

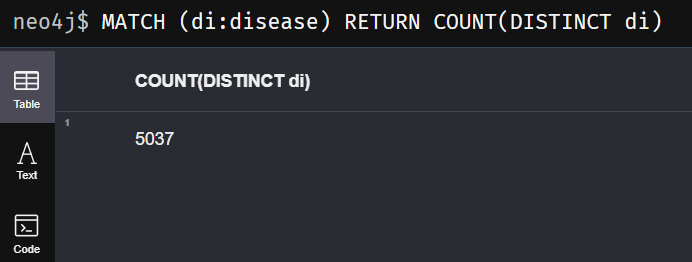




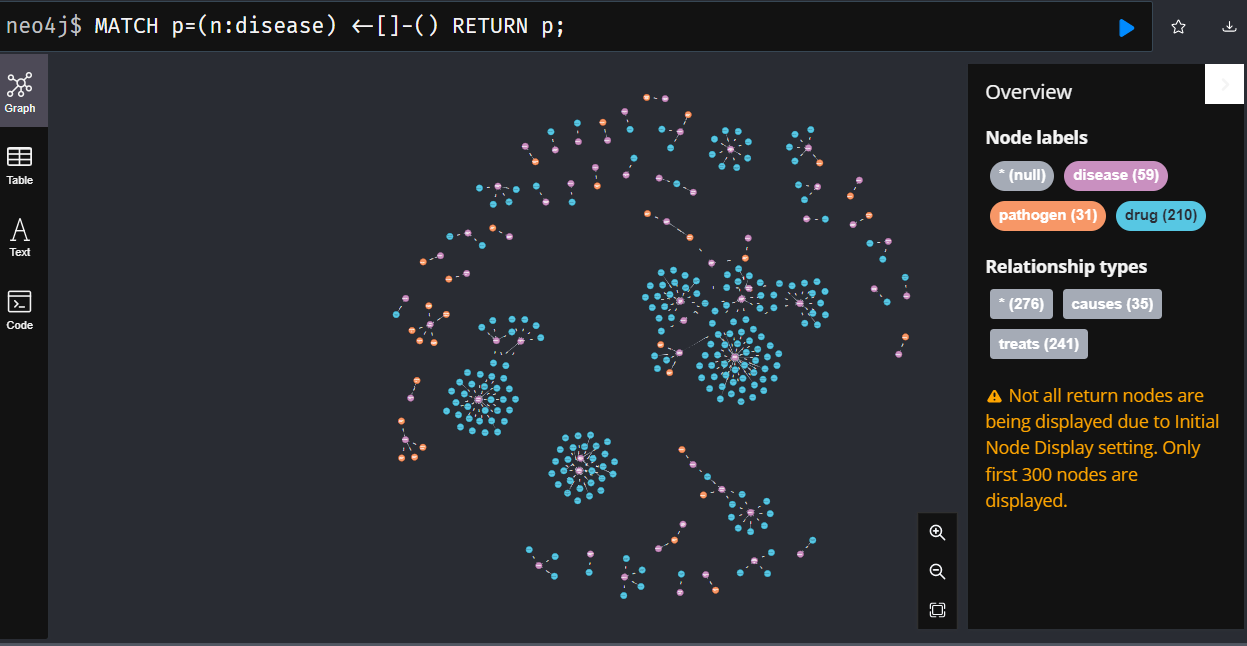
Counting the number of pathogens, diseases and drugs

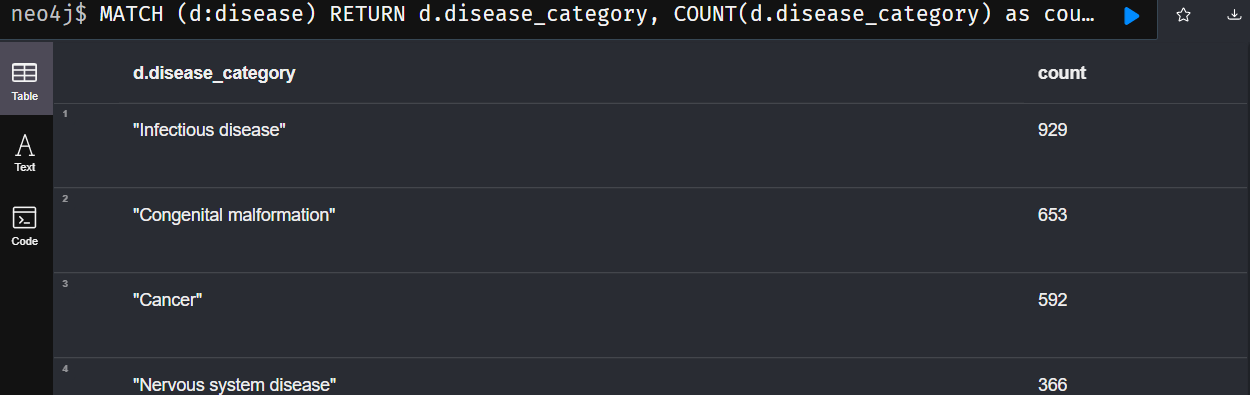




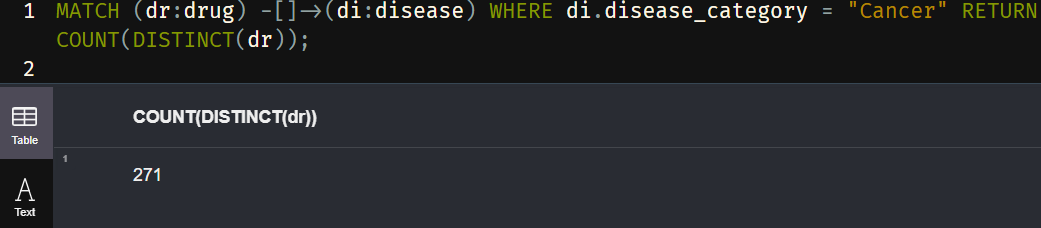


Graphing the whole database

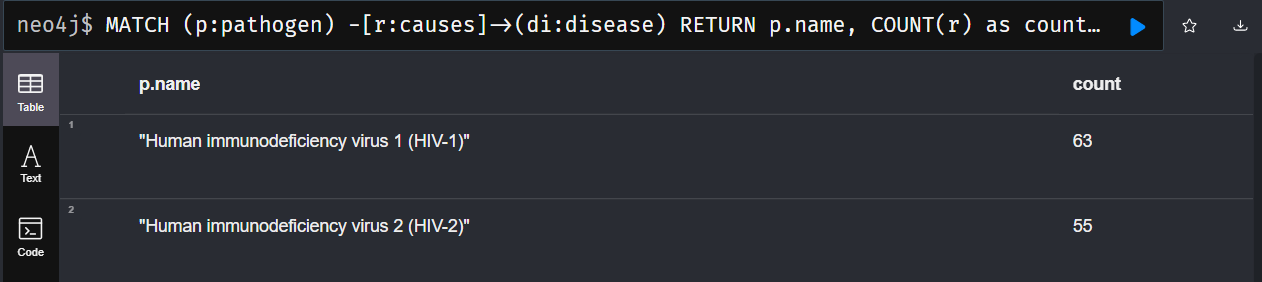
****

Return the total number of unique disease category

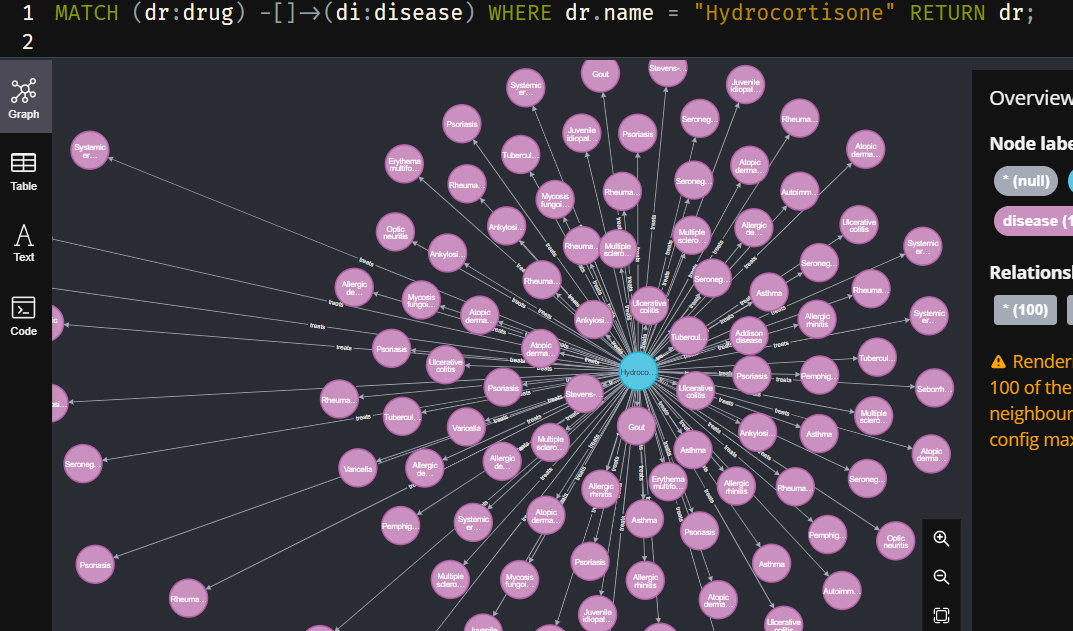
Return the number of diseases under Cancer.

****

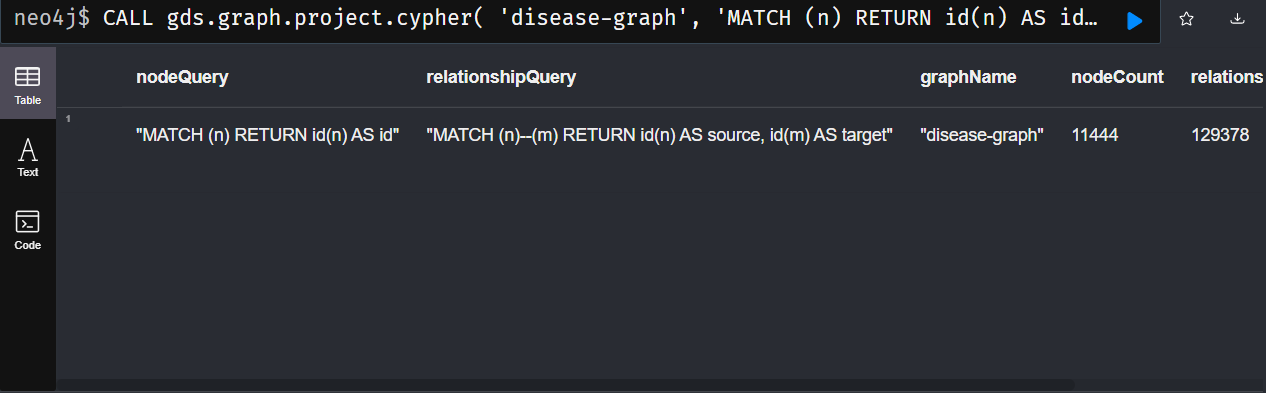
Return count of pathogen causing diseases

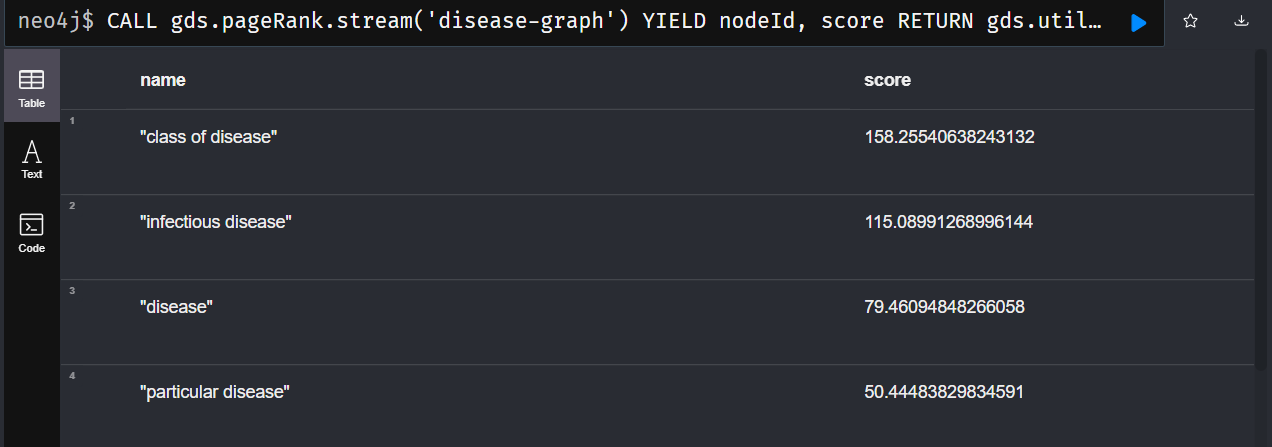
****

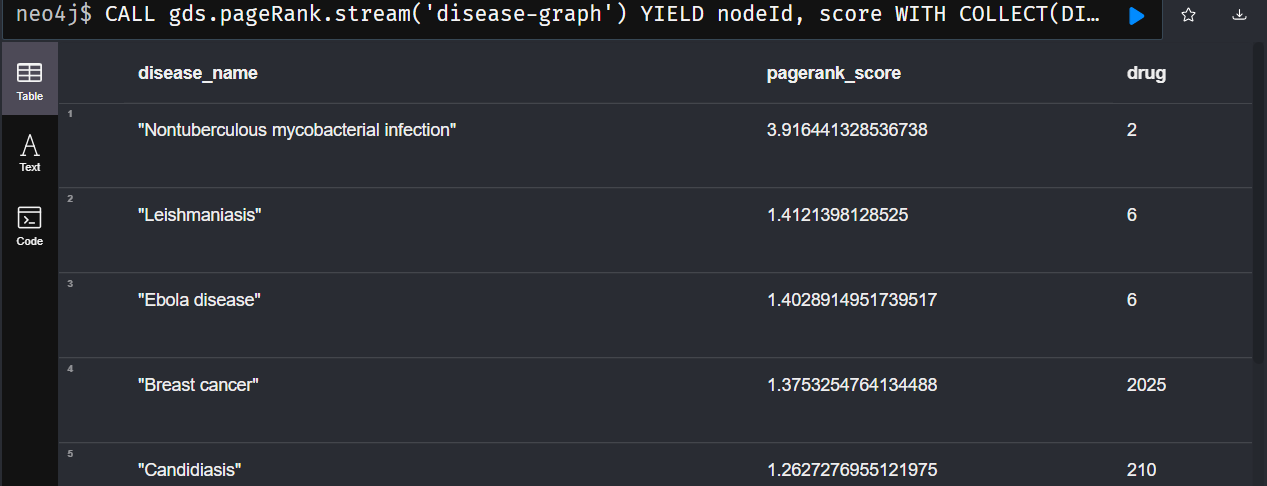
Finding out the diseases cured by a given drug

****

Page Rank

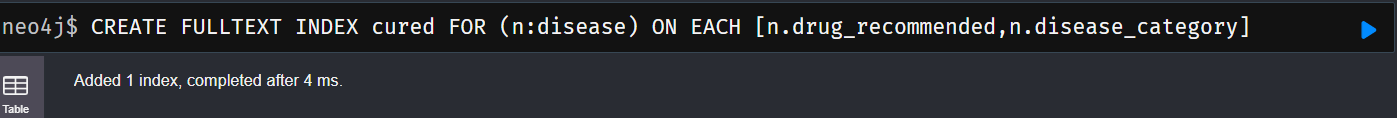
****

****

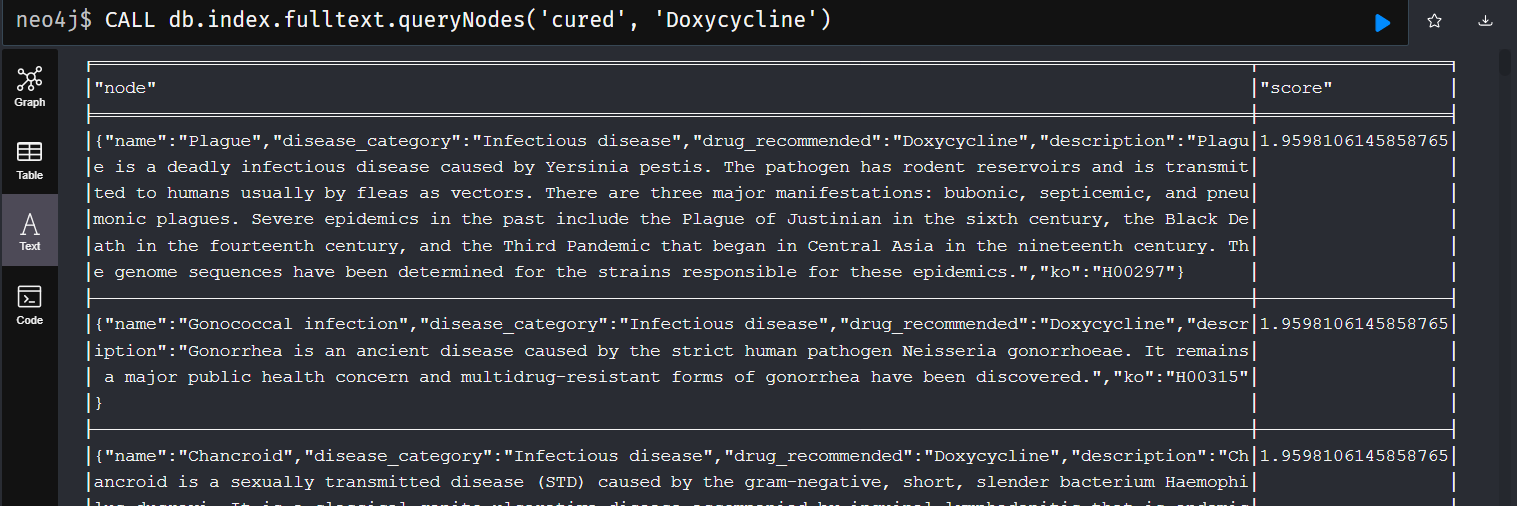
****

Part – 2 Fulltext and Fuzzy Searching

A search engine’s purpose is to store, find and retrieve content. The underlying engine used by Neo4j is Apache Lucene, a free and open-source information retrieval software library. Pls note the dataset used here is known as disease\_consol which is a consolidated version of all the other datasets used previously. The first operation to do is to create a fulltext search index, with the help of the following procedure.

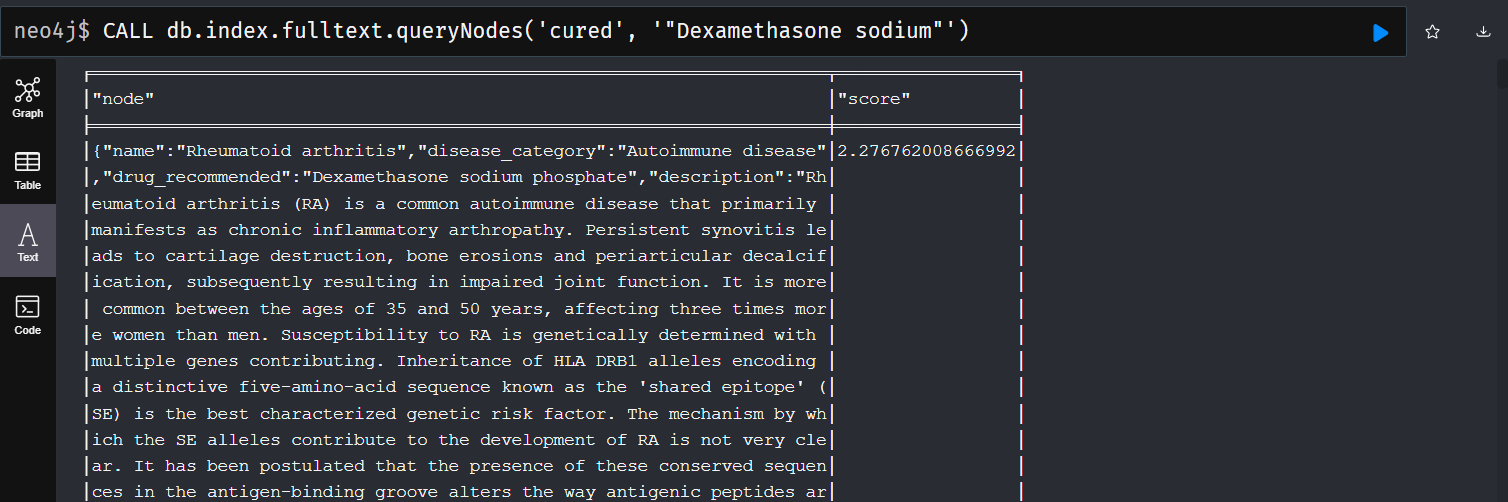


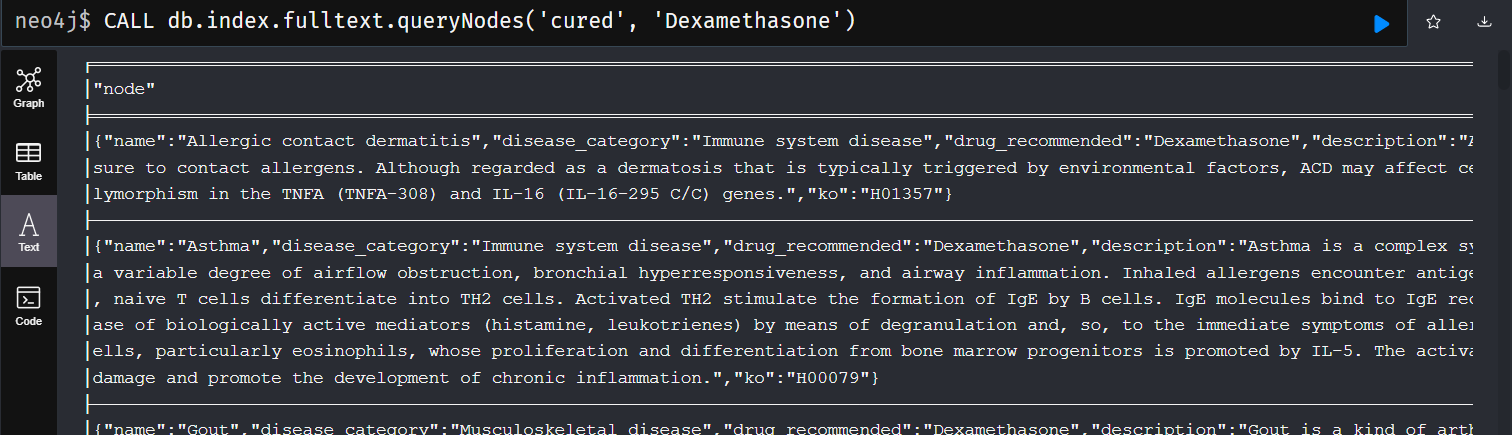
The index is created for the columns drug recommended and disease category of the disease\_consol dataset. Now that our index is created, we can query it and test our full text search queries. Let’s find all nodes containing the word “Doxycycline” in their title :

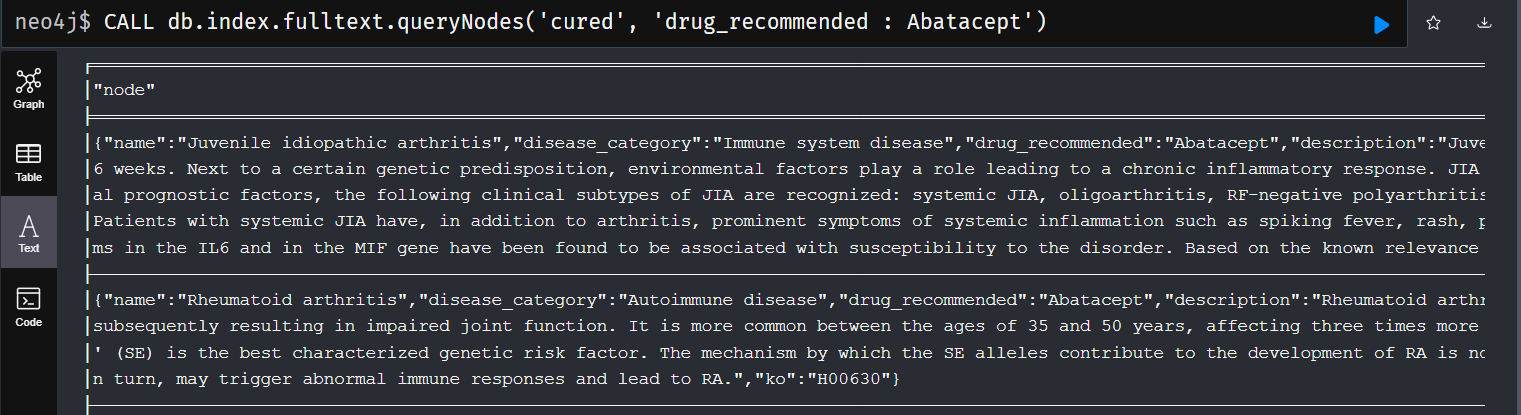


This type of querying is understood as a Term Query where the searching is only for the matching term of the word – “Doxycycline”.

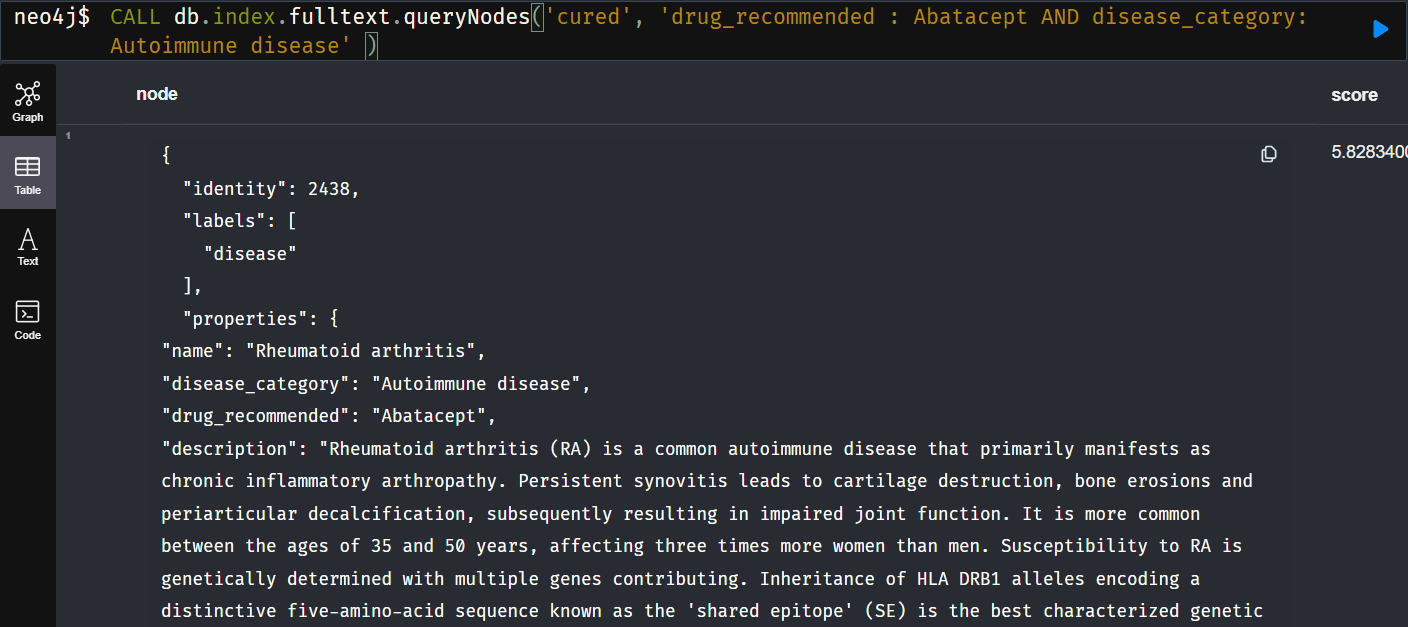
*Phrase Query*





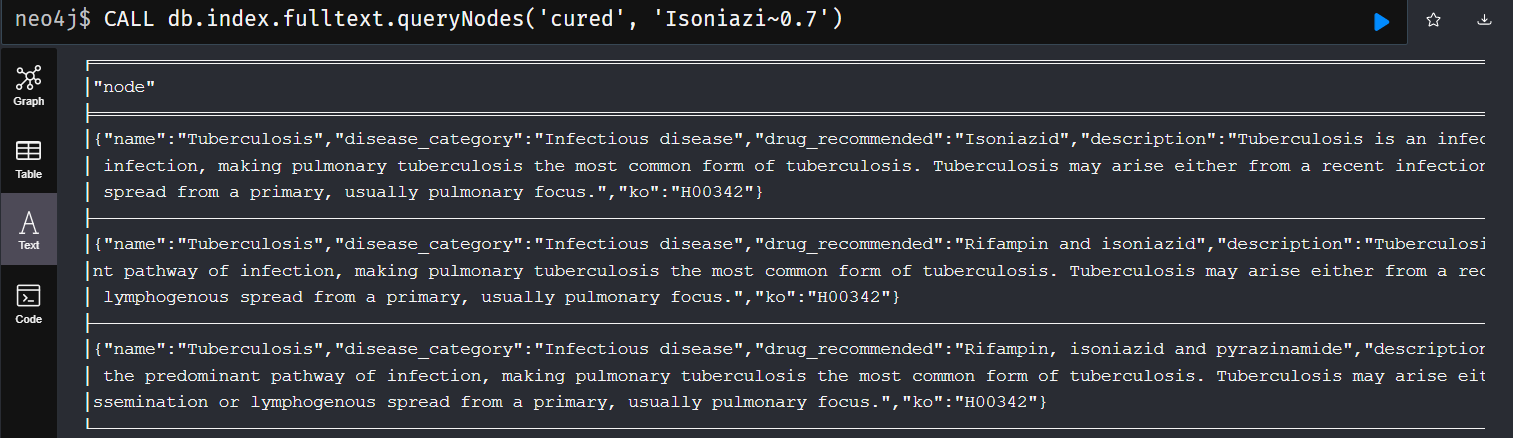


*Searching with a specific field*

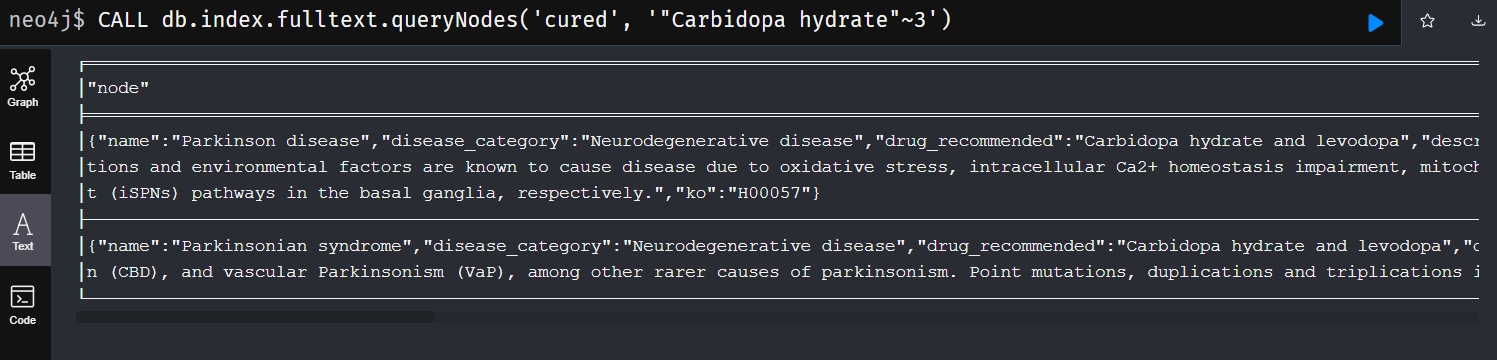


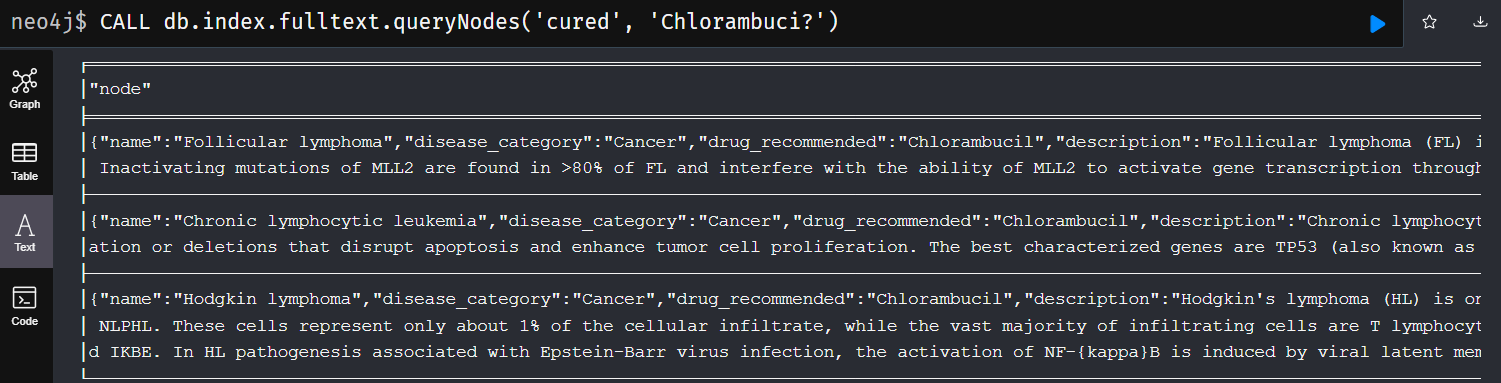
*Fuzzy searching*

The power of Full Text Search is also the ability to retrieve results even if the search query does not exactly match text in the original corpus.



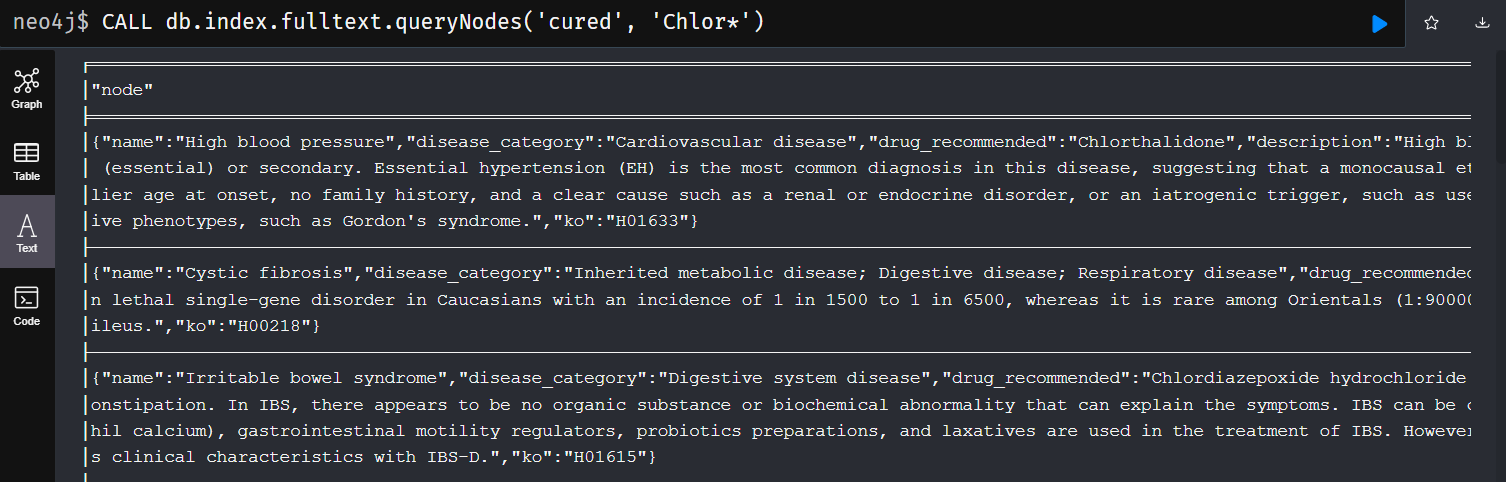
The tilde (~) allows a FuzzySearch using the Damarau-Levenshtein distance algorithm.





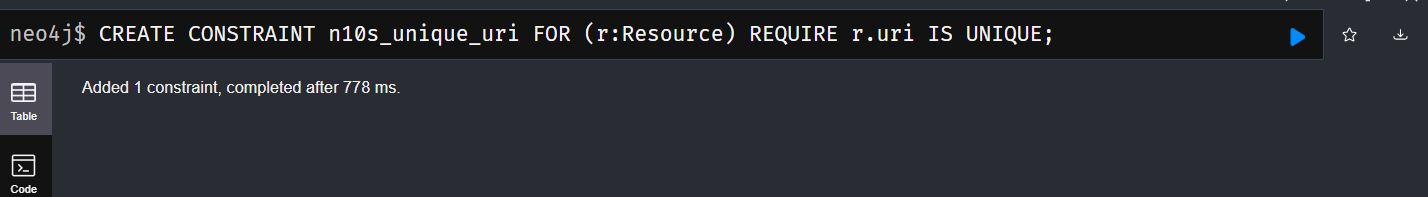
*Wildcard query*

The last implementation is the [WildcardQuery](https://lucene.apache.org/core/5_5_5/core/org/apache/lucene/search/WildcardQuery.html), where you can provide wildcards for your searches. Use ’?’ for a single character wildcard search, use \* for multiple characters wildcard search.



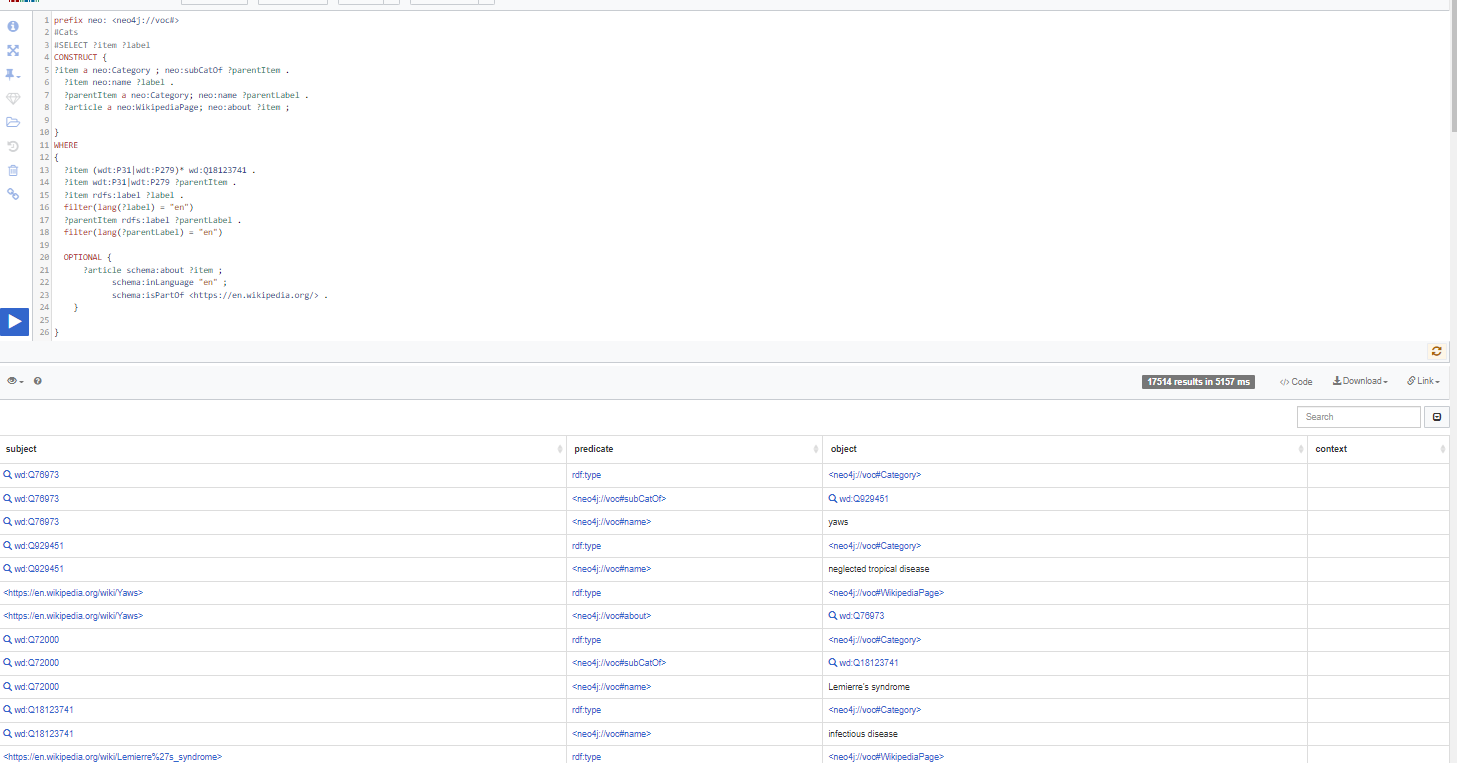
**Part -3 WikiData, SPARQL, Neosemantics**

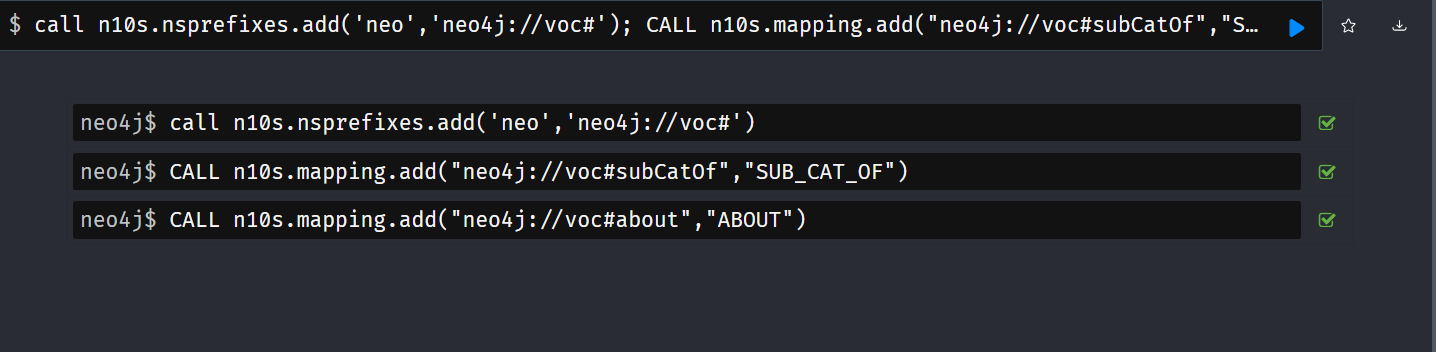
Wikidata provides a [SPARQL API](https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service) that lets users query the data directly. The screenshot below shows an example of a SPARQL query along with the results from running that query. We first import the SPARQL API into the graph.



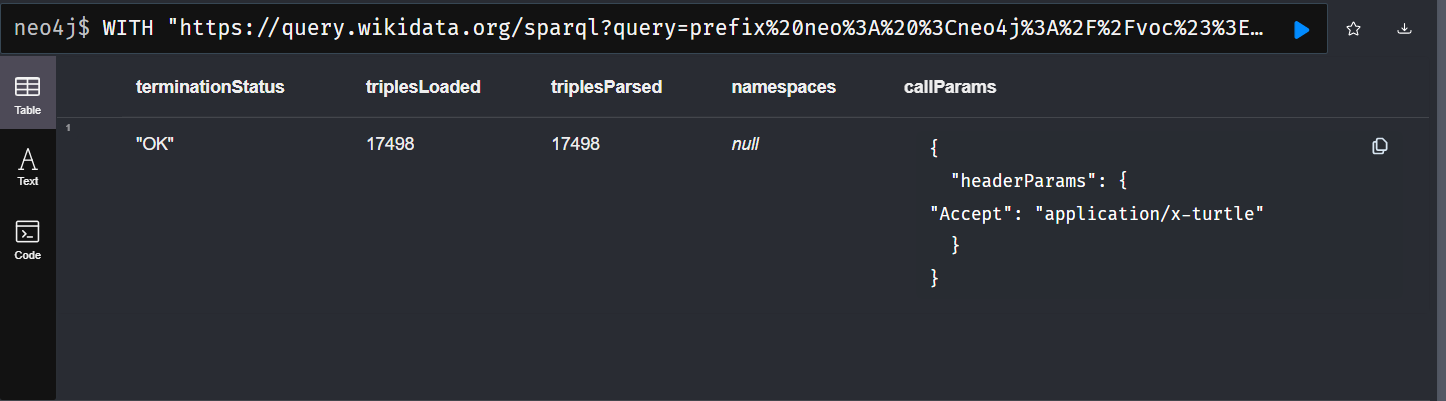
Entities in the Wikidata knowledge graph are identified in the form of the subject, predicate and the object, each connected by a unique ID that can be found within the URL of Wikidata. Here we need to find out the pages that are linked to Q18123741 – which is the Wikidata identifier ID for the entity known as ‘infectious disease’. If closely looked, we can find the links to other webpages as well as its Wikidata identifier ID, on the subject, predicate and the object part of the query processing. On clicking the link, we find out that the link will take us to the wikidata link of the connected entity.

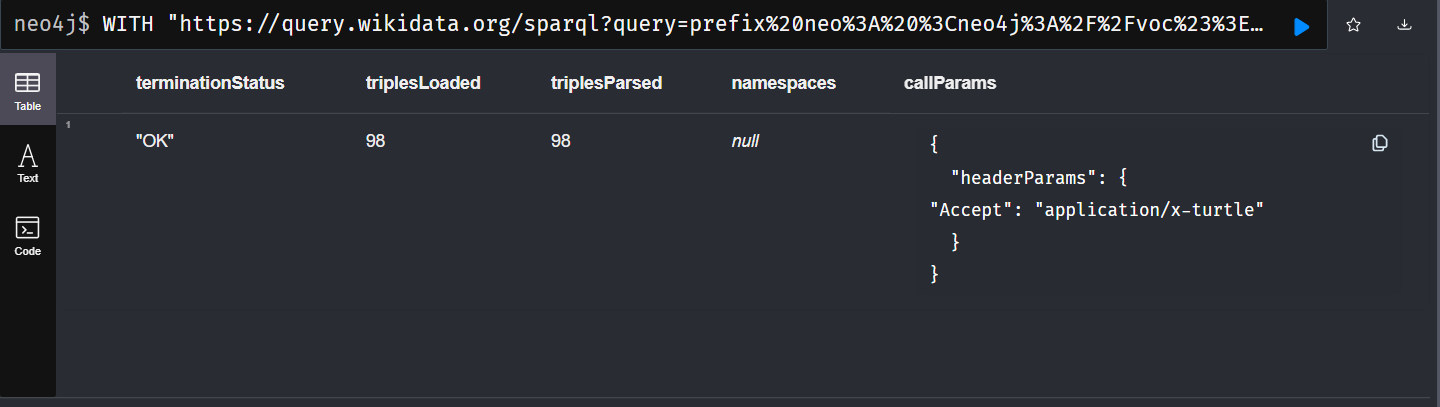
A good example is wd:720000, which is the identifier ID for Lemierre’s Syndrome that comes under the subset of the Infectious disease category.





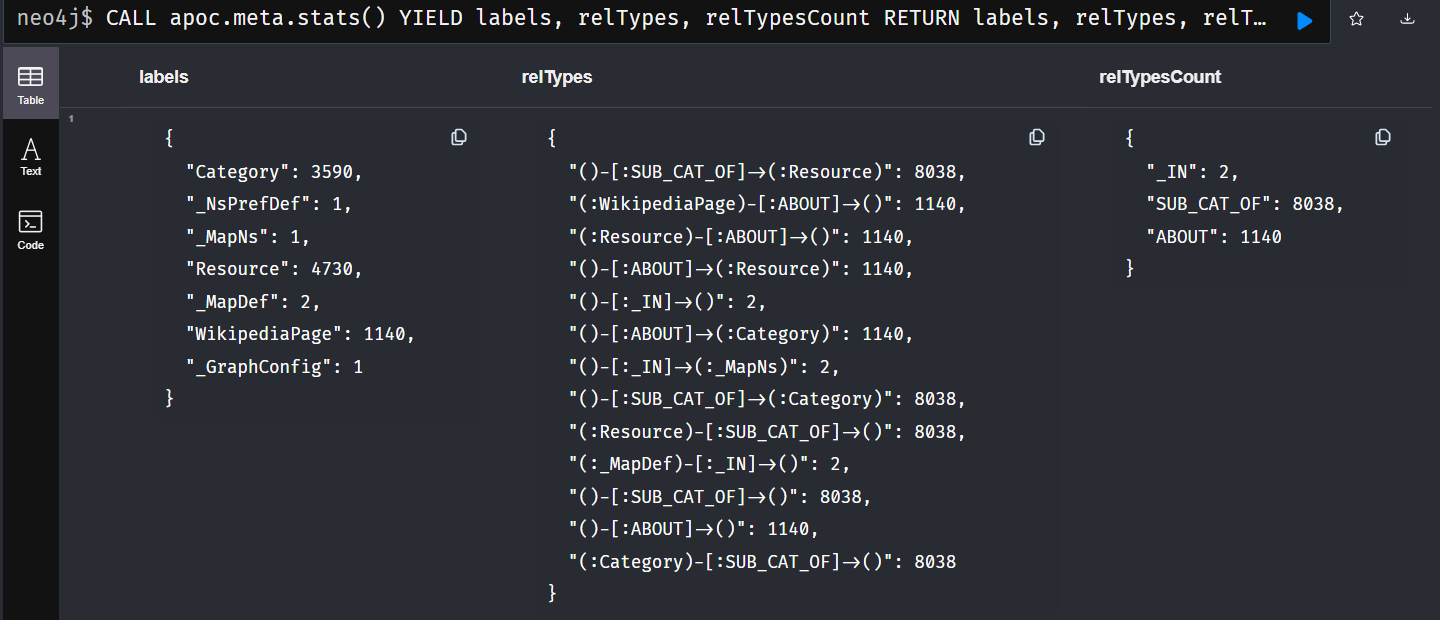
We obtain a URL at the end of writing the query in SPARQL query which we then import to the Neo4j server to connect with Neosemantics. This process is repeated for the entity known as “Infectious diseases” as well as the entity “Malaria”, a subset of Infectious disease for both which we would importing from WikiData.



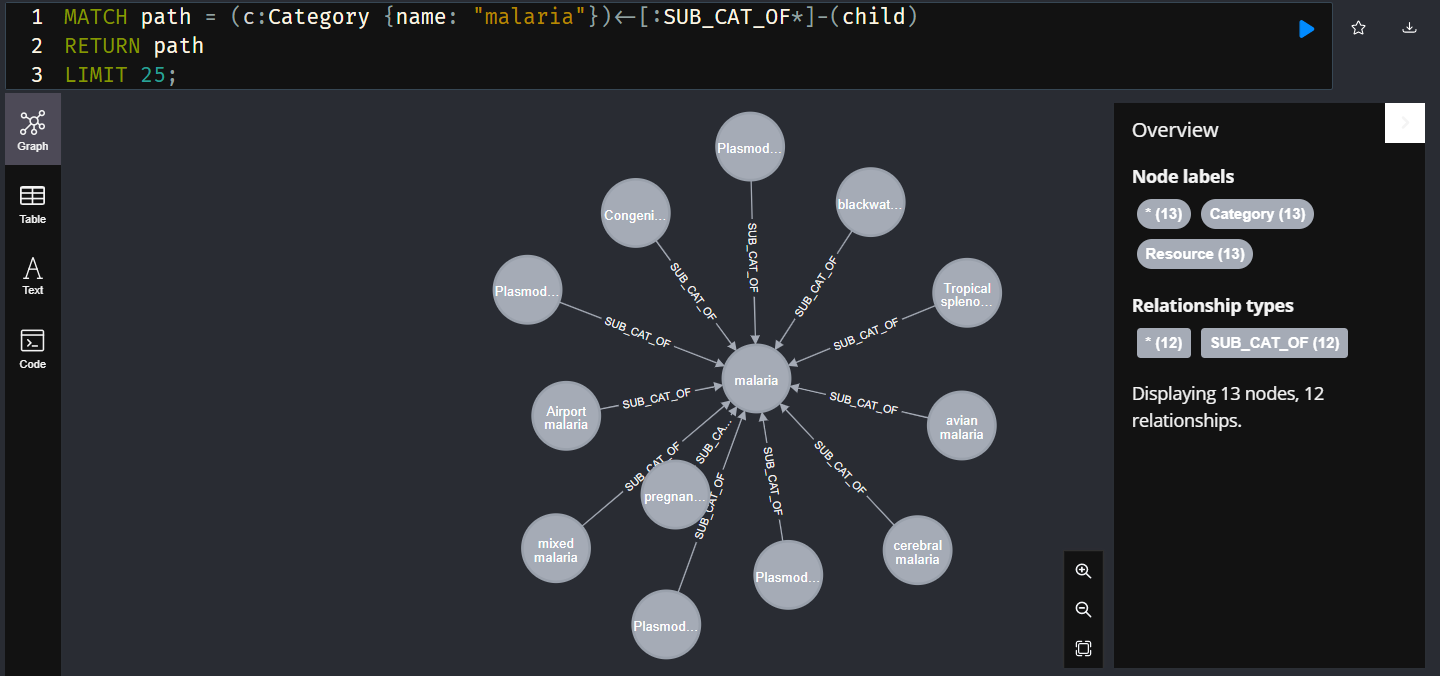


We then pass that URL to the n10s.rdf.import.fetch procedure, which will import the stream of triples into Neo4j.

We next look at the contents of the database in the given image. 8038 sub categories of infectious disease exist, as per Wikidata and Wikipedia.



We then pass to query to look at the subgroups of a popular subset of Infectious disease, to design the knowledge graph. Subset Malaria has multiple inherent subsets as well, which may have its sub branches as well.



**Innovation**

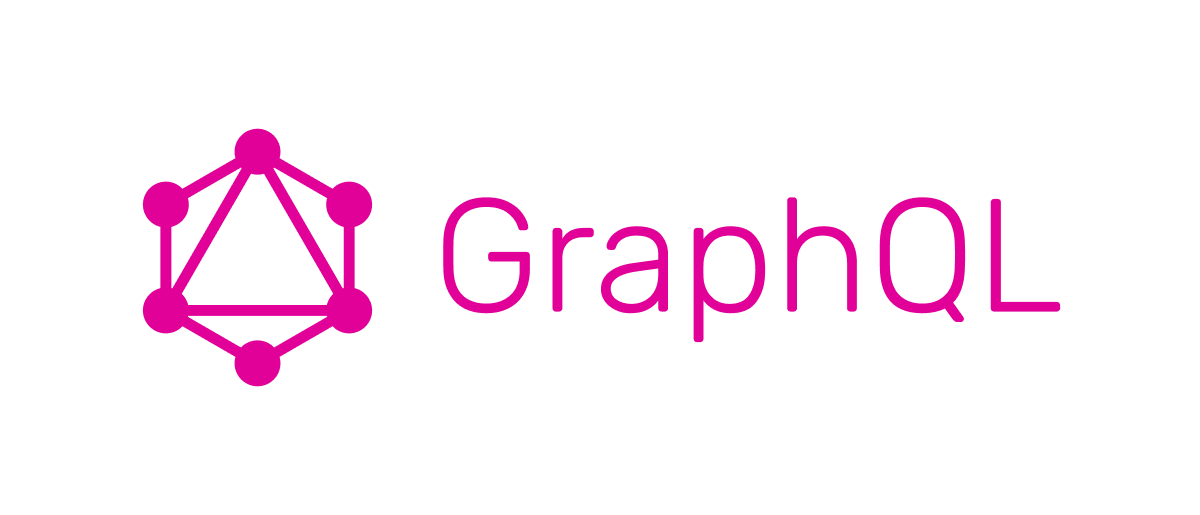
* Learning of a new query language Cypher and implementation of a completely new database architecture, Neo4j.
* Implement the RDF with other possible ontologies and Neosemantics library for a Biomedical dataset.
* Pagerank algorithm to rank maximum connected pathogens to disease and finding out what disease is caused by the maximum number of pathogens. This algorithm is used for websites and SEOs however this also proves useful for any graph data.
* Made a drug search and retrieval API in GraphQL using React.js, Neo4j, Apollo Server and Express.js which somehow couldn’t make it to the final cut of the project.
* Implementing a knowledge graph using WikiData and SPARQL API.
* Implemented a mini-full text search engine built using Apache Lucene for biomedical drug and disease descriptions.

**Results and Conclusion**

Research shows that the brain likes to work based on association and has a tendency to connect every idea or skill that it encounters in its lifetime, which is enabled by the mind-map-like the KGs. Such visualization tools involve a unique combination of imagery, color, and visual-spatial arrangement which is proven to significantly improve recall when compared to conventional methods of documentation. This leads to individuals of different learning speeds and abilities to understand the links between entities for this very reason [21]. There exist some nodes that have more than one relation to another node. There can also exist an isolated, individual node with no relation/edge weight to any other node whatsoever. A node without any relation can also exist when the information database is familiar with the existence of the given term, but there isn't any sufficient information on the same topic. Sometimes. we may also come across a case where there is a self-looping node present in the graph.

*Future enhancements*

The potential for this project is endless. Apart from using graph databases as a potential storage for knowledge systems, the same graph can be used to store information obtained in any kind of domain. Another development that can be put forward is the utilization of growth of Graph machine learning algorithms. Graph ML algorithms that study node embeddings can put be to use in order to predict specific communities or even predict the growth of a specific community. A good enhancement that can be added to this would be the inclusion of a GraphQL API that can perform CRUD operations as well as act as a search and text engine that can automate work and find the required records as a faster rate. Introduced by Facebook, GraphQL acts as the solution to all the problems that is faced by the REST API architecture. GraphQL can also be connected and worked efficiently with the Neo4j database as well as Python framework Flask to create a fully functional working application.



**References**

[1] Lysenko, A., Roznovăţ, I.A., Saqi, M. *et al.* Representing and querying disease networks using graph databases. *BioData Mining* **9**, 23 (2016). <https://doi.org/10.1186/s13040-016-0102-8>

[2] Shoshi, A., Hofestädt, R., Zolotareva, O., Friedrichs, M., Maier, A., Ivanisenko, V. A., ... & Bragina, E. Y. (2018). GenCoNet–a graph database for the analysis of comorbidities by gene networks. *Journal of integrative bioinformatics*, *15*(4).

[3] Brandizi, M., Singh, A., Rawlings, C., & Hassani-Pak, K. (2018). Towards FAIRer biological knowledge networks using a hybrid linked data and graph database approach. *Journal of integrative bioinformatics*, *15*(3).

[4] Ren, Z. H., Yu, C. Q., Li, L. P., You, Z. H., Guan, Y. J., Wang, X. F., & Pan, J. (2022). BioDKG–DDI: predicting drug–drug interactions based on drug knowledge graph fusing biochemical information. *Briefings in Functional Genomics*, *21*(3), 216-229.

[5] Gao, Z., Ding, P., & Xu, R. (2022). KG-Predict: A knowledge graph computational framework for drug repurposing. *Journal of biomedical informatics*, *132*, 104133.

[6] Sequeda, J., & Lassila, O. (2021). Designing and building enterprise knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge*, *11*(1), 1-165.

[7] Waagmeester, A., Stupp, G., Burgstaller-Muehlbacher, S., Good, B. M., Griffith, M., Griffith, O. L., ... & Su, A. I. (2020). Science Forum: Wikidata as a knowledge graph for the life sciences. *Elife*, *9*, e52614.

[8] Ren, Z. H., You, Z. H., Yu, C. Q., Li, L. P., Guan, Y. J., Guo, L. X., & Pan, J. (2022). A biomedical knowledge graph-based method for drug–drug interactions prediction through combining local and global features with deep neural networks. *Briefings in Bioinformatics*, *23*(5).

[9] Galluzzo, Y. (2022). A Review: Biological Insights on Knowledge Graphs. In *European Conference on Advances in Databases and Information Systems* (pp. 388-399). Springer, Cham.

[10] Lelong, R., Soualmia, L. F., Grosjean, J., Taalba, M., & Darmoni, S. J. (2019). Building a semantic health data warehouse in the context of clinical trials: development and usability study. *JMIR Medical Informatics*, *7*(4), e13917.

[11] Wei, C. H., Allot, A., Leaman, R., & Lu, Z. (2019). PubTator central: automated concept annotation for biomedical full text articles. *Nucleic acids research*, *47*(W1), W587-W593.

[12] Müller, H. M., Van Auken, K. M., Li, Y., & Sternberg, P. W. (2018). Textpresso Central: a customizable platform for searching, text mining, viewing, and curating biomedical literature. *BMC bioinformatics*, *19*(1), 1-16.

[13] Mishra, R., & Tripathi, S. P. (2021). Deep learning - based search engine for biomedical images using convolutional neural networks. *Multimedia Tools and Applications*, *80*(10), 15057-15065.

[14] Lee, K., Wei, C. H., & Lu, Z. (2021). Recent advances of automated methods for searching and extracting genomic variant information from biomedical literature. *Briefings in bioinformatics*, *22*(3), bbaa142.

[15] Guntupally, K., Dumas, K., Darnell, W., Crow, M., Devarakonda, R., & Giri, P. (2020, December). Automated indexing of structured scientific metadata using apache solr. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 5685-5687). IEEE.

[16] Aminu, E. F., Oyefolahan, I. O., Abdullahi, M. B., & Salaudeen, M. T. (2020). A review on ontology development methodologies for developing ontological knowledge representation systems for various domains.

[17] Bonatti, P. A., Decker, S., Polleres, A., & Presutti, V. (2019). Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371). In *Dagstuhl reports* (Vol. 8, No. 9). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[18] Hitzler, P. (2021). A review of the semantic web field. *Communications of the ACM*, *64*(2), 76-83.

[19] Ristoski, P. (2019). *Exploiting semantic web knowledge graphs in data mining* (Vol. 38). IOS Press.

[20] Yahya, M., Breslin, J. G., & Ali, M. I. (2021). Semantic web and knowledge graphs for industry 4.0. *Applied Sciences*, *11*(11), 5110.

[21] Shridevi, S., Aiyyer, Harshini. K., & Manthripragada, A. (2021, November). Cryptocurrency and Associated Bourse Analysis using Machine Learning and Knowledge Graphs. In *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)* (pp. 1-8). IEEE.

**Individual Contribution**

This project was a solo project done by me. Hence there would be no requirement of documenting the same.