# MS4S21 CW-2

Mark Baber - 17076749

## Contents

## Section A - 20 Marks

### 1.1 – MongoDB

MongoDB describes itself as 'A general purpose, document-based, distributed database built for modern application developers and for the cloud era' (MongoDB, 2021) and is a NoSQL database. This has been described as document based and is mostly used to deal with unstructured data, which in comparison to a 'normal' database does not require as much time developing.

With NoSQL the data is stored in documents (like json objects), which allows for the use of multiple benefits when compared to a relational database. These benefits are as follows:

| | |
|---|---|
| Sharding / Partitioning | Distributing the data across multiple machines, these can be smaller, less powerful machines. |
| Flexible schemas | The data does not follow a set schema compared to a Relational Database Management System (RDMS), and each 'document' can have different schemas. |
| High availability via replication | Not only can the data be shared among other computers, but also main node can also be replicated to add even more CPU and IO at hand. |

(Table 1 – Advantages of NoSQL)

Whilst a MongoDB database does not use primary and secondary keys, they do have an ObjectId which is a hexadecimal unique value, which is made up of 4-bytes of timestamp value, 5-byte random value and 3-byte incrementing counter. These would be implemented as certain values within a document-based database would not require a primary key specified, below is an example:

```
{
   "name": "notebook",
   "qty": 50,
   "rating": [ { "score": 8 }, { "score": 9 } ],
   "size": { "height": 11, "width": 8.5, "unit": "in" },
   "status": "A",
   "tags": [ "college-ruled", "perforated"]
}
```

(Figure 1 – Example of NoSQL JSON document (MongoDB, 2021))

There are 3 different ways to store your data with a NoSQL database, Key value stores which is stores as a key and data as 1 value, Column-oriented stores which is a flat structure with keys within the columns and Document oriented stores which is an organised collection of information on one document (CoreViewSystem, 2021). For an example, if we were to store my student data as a document store, it would all be on one 'form' with all the information I have given the uni.

## 1.2 – Comparisons

There are several NoSQL databases to date, ranging from CassandraDB, CouchDB, MongoDB, DynamoDB and many more. The two which will be discussed here are MongoDB and DynamoDB.

MongoDB is a free and open source which allows for anyone to download, see, and edit the code and can deploy their database anywhere. Whereas DynamoDB is a proprietary piece of software you can only use on Amazon Web Services (AWS). Whilst the two companies have different ideas of how to develop a NoSQL database, they also have a few common concepts:

| **DynamoDB** | **MongoDB** |
|---|---|
| Table | Collection |
| Item | Document |
| Attribute | Field |
| Secondary Index | Secondary Index |

(Table 2 – Table showing common concepts)

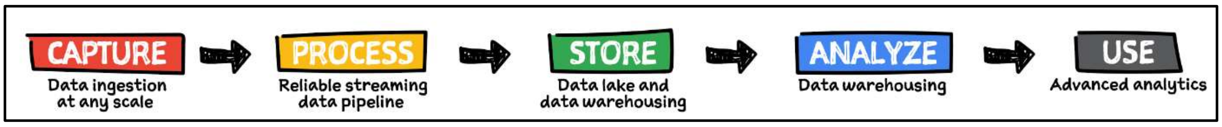
Both databases also use different data models, with MongoDB storing their data as JSON-like format, (which we saw in Figure 1) which allows for the storage of a range of datatypes, from strings, numbers, dates, timestamps and more, with a maximum size of 16 MB in size. DynamoDB uses a key-value store which adds support for JSON to provide a document-like data structure, which can store one numeric type and does not support dates, with a maximum size of 400KB.

There are also some other differences with MongoDB having a rich ad-hoc query language + IDE built into their cloud platform, whereas DynamoDB is all through CLI (provided with their own aws cli tool for local development) or programming languages such as python. With both databases having great performance, with the ability to scale horizontally and vertically with high number of read/write operations (ZenOfAI, 2021), with both providing data pipeline tools.

## 2 – Data Pipelines

A data pipeline is a process of taking data from task 1, processing the data and outputting this data as the input for the next processing step (say task 2). It follows a flow through each of the steps until the processes are complete, with some steps being able to run at parallel. With common steps within a data pipeline being data transformation, enrichment, filtering, grouping with the ability to run other algorithms on the data (Hazelcast, 2021).

With the cloud becoming more popular, the use of data pipelines within the cloud is growing rapidly. If we were to explore a pipeline for data analytics, it should look something like this:


CAPTURE
Data ingestion at any scale
PROCESS
Reliable streaming data pipeline
STORE
Data lake and data warehousing
ANALYZE
Data warehousing
USE
Advanced analytics

(Figure 2 – Data Analytics Pipeline (Vergadia, 2021)

With each step being just as important as the last, this could be applied to any service from AWS, Microsoft Azure (Azure) and Google Cloud Platform (GCP). Let us apply this pipeline to AWS and see how each step can be used on that platform.

Starting with capture, this could be an application which is reading user input from a virtual machine (VM) which is up and running on an Amazon EC2 service. This could be an app which is reading tweets for example, before being moved to the next step process. The process part is the reading of the data which was previously captures.

Next would be the store, and for storing data AWS has an Amazon S3 service which is their storage buckets also known as Objective storage which they claim is used to store data for millions of applications. From the smaller storage, this data can be stored/archived in a data warehouse which is usually used for long term storage and analysing, Amazons Redshift is used for this.

Now this data has been collected and stored in a data warehouse, the next step would be to analyse the data which could be used for machine learning algorithms to predict trends. The Amazon service which is mostly used for this is the Amazon ERM which is used to run and scale Apache Spark, Hive, and other big data frameworks.

This can be seen as a positive when using a big service like AWS as everything can easily talk to each other, and data pipelines can become a real benefit to businesses and data scientists when working with such customisable tools.

## Section B - 20 Marks

### 2.1 – Introduction to Knowledge Graphs

With the ever-growing mass of data, quickly understanding and drawing insight from this data could become challenging (Bulao, 2021). This is where the company Google comes in, with developing Knowledge Graphs. Google describes Knowledge Graphs as a 'huge collection of the people, places and things in the world and how they're connected to one another' (Google, 2012).

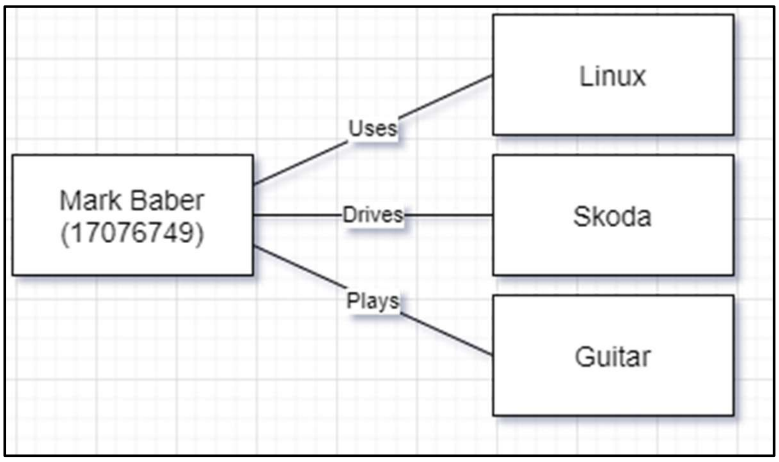
Knowledge Graphs are, at its core made up of Big Data techniques and Machine Learning. This is used for multiple areas such as data mining, natural language processing and more, to find factual knowledge about human information. This knowledge has been annotated from online sources, public sources, licensed data such as sports, stocks, weather, and other peer reviewed sources (Fensel et al, 2020; Google, 2021).

These knowledge graphs were first introduced by Google to get the most relevant information from the users search terms, instead of looking at the term as strings, Google was looking at the search term as a thing (Google, 2021). This could then be produced in a Knowledge Panel which shows the most relevant information about this search term (Appendix 1).

### 2.2 – Resource Description Framework

A resource description framework is as it states in the name, a framework. This framework is used for representing information on the web, which is linked together in a simple graph, and is the standard for modelling online data (Fullstack Academy, 2017). Whilst not really being a data model for big datasets or KGs, RDFs are often linked to KGs as that is where they are mostly developed and used (Kejriwal et al, 2021) and is also a part of the semantic web stack (Cardoso and Gouveia, 2007).

The main structure of an RDF is simple yet structured and is made up of multiple 'triples' which are nodes that represent a Subject, a Predicate and an Object. This model is used to represent facts and can be used to understand the semantics of a subject or a search term (Kejriwal et al, 2021). There are multiple data types for RDF, these are RDF / JSON and RDF / XML which are both widely used within the web, along with some other standards such as OWL, TTL / Turtle / Terse RDF Triple Language (Easy for humans to read) and can be queried with SPARQL.



(Figure 3 – Example of an RDF Triple)

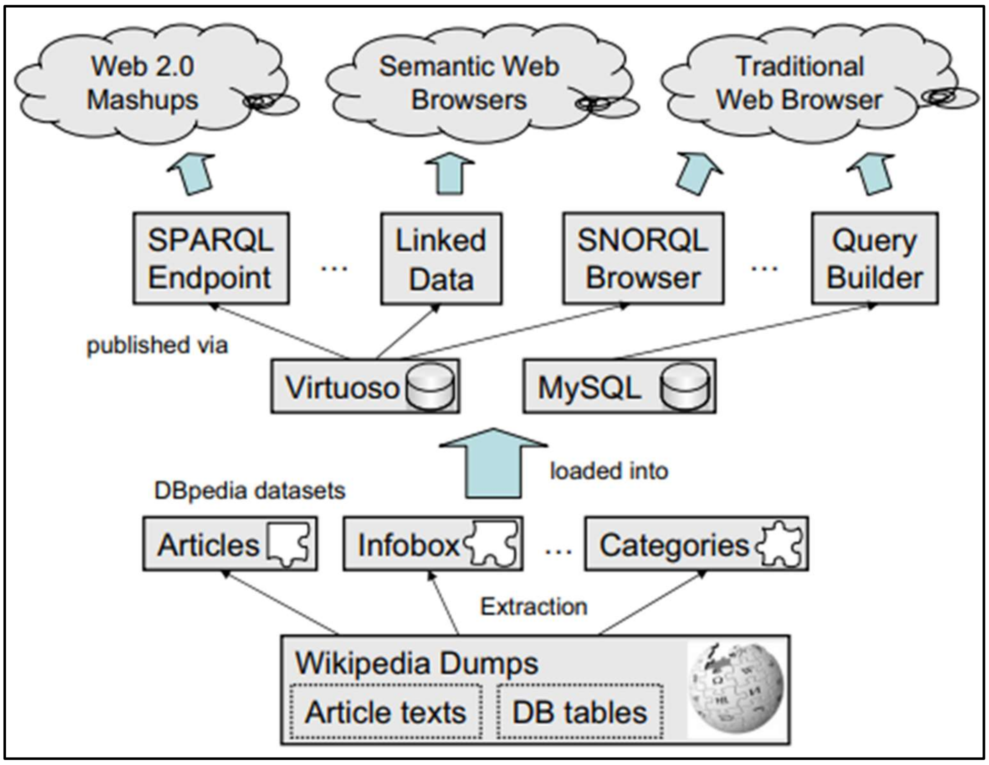
Since the creation of RDFs there have been several ontologies which have been developed, with the likes of Linked Open Data (LOD) which is the vision of an accessible and linked dataset on the web. The LOD is thought of as a linked data cloud which follows the same standard of formatting (RDFs), which allows users to easily create, connect and consume data easily online without distributing the original source (Jain et al, 2010, W3, 2010). There is also another ontology, which is the Web Ontology Language (OWL), which is described as "more expressive than XML, RDF or RDF  Schemas" (Cardoso and Gouveia, 2007) and was designed to process information, instead of simply presenting it (McGuinness and Van Harmelen, 2004).

## 2.3 – The use of Knowledge Graphs

Wikipedia is a great example of a knowledge graph in industry, whilst Wikipedia is a free website which has collected information about millions of subjects (Wikipedia, 2021). The technology being used to draw insight from these Wikipedia articles is DBpedia which describes themselves as, 'a crowd-sourced community effort to extract structured content from the information on Wikipedia' (DBpedia, 2021).

This crowd sourced effort has managed to set 27 million RDF links to 30 external datasets, which in turn encompass' that dataset into DBpedia and has added to the growth of the Web of Linked Data and been a major role to the Linked Open Data movement (Lehmann et al, 2015). Whilst being a crowd sourced initiative, DBpedia is coordinated by the DBpedia Internationalisation Committee when it comes to dealing with the translation of the many articles on Wikipedia (Lehmann et al, 2015).

Extracting unstructured data from Wikipedia and converting it into RDF triples does not sound like a big task at first but lets first break down the sort of algorithm which would be required to do such a task.



(Figure 4 – Overview of DBpedia components (Auer et al, 2007))

From looking at figure 4, the first stage would be the extraction of the data which is then sorted into articles, info boxes, hyperlinks, categories and more. To extract the semantic relationships, DBpedia create RDF links for data, which is in already stored in a RDMS, followed by extracting additional information from the articles related to the subject (Auer et al, 2007).

This data is then stored either in a traditional RDMS which allows to be extracted and deployed back to the web, but also as in an RDF compatible database which allows for semantic web browsers to access the linked data. This data can also be queried with smart queries by searching the data for things not strings (Google, 2012) by using SPARQL with the example of Virtuoso (Virtuoso, 2021).

## 2.4 – Overview of Knowledge Graphs

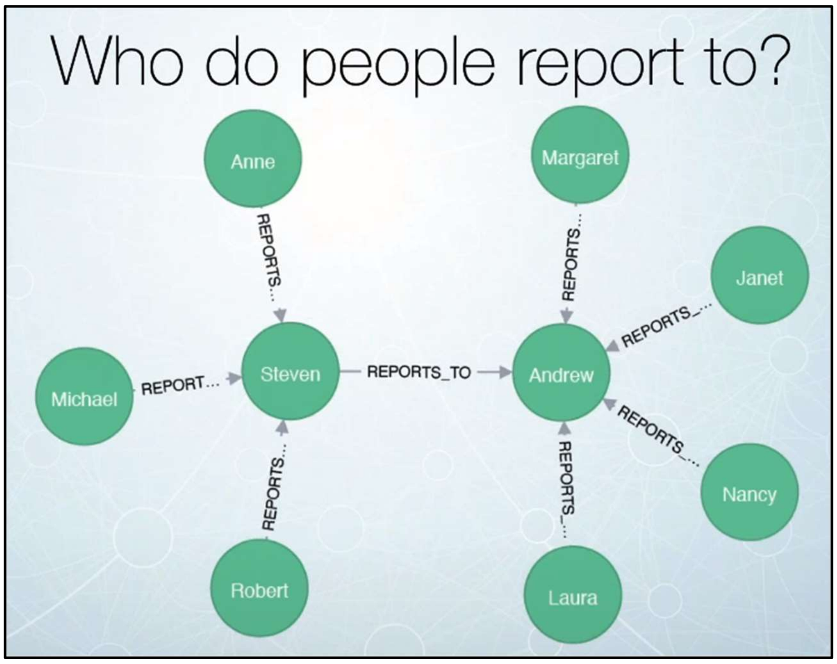
After exploring KGs and their use cases, this section will look at the different types of software packages that are available. When first researching KGs there were a few which stood out, these are as follows:

| | |
|---|---|
| Cayley | Open-source piece of software written in the programming language, GO. Developed by Google and is a Graph Stack database (Appendix 1). |
| Grakn | Grakn and Graql are another open-source project, which allows the users to query their graph databases with Java, Python or NodeJS (Appendix 2) |
| Titan | Titan graph database is an open-source project, which supports different storage backends such as (CassandraDB, Apache HBase and Oracle Berkeley DB) (Appendix 3). |
| Neo4j | Neo4j is a native graph database platform which is used to look at data graphically and data relationships (Appendix 4). |

(Table 3 – List of 4 KG software packages)

From here we will look to explore the software Neo4j which will also be referred to as Neo. These knowledge graph packages are used to query graph databases and allow to easily visualise not just your data, but also the data relationships (Neo4j, 2021). Whilst this report has been exploring KGs and RDFs with data on the internet and benefiting search results, Neo has been used for map air transportation networks and food web of grassland species (Neo4j, 2021, Webber, 2012).

Neo4j can be used for real time recommendations, master data management, fraud detection, graph-based search, network and IT operations and identity and access management (Neo4j, 2016). This is done by graphing your data with nodes which are connected via triples which displays the flow of the relationships and can be used to easily find the shortest path (Neo4j, 2020) whilst also being faster than a RDMS.



(Figure 5 – Example of Neo4j (Neo4j, 2016))

Here is an example of Neo4j which represents the relationships as triples, this clearly shows who reports to who with a 1 to many relationships. Even though you do not need to design your schemas like traditional databases (RDMS), it is still important to have a general idea of how many nodes you would have within your database. For example, if you model a person, name, country as entities to a node, querying these nodes could become less performance. Especially when compared to having these entities as separate nodes with a relationship, and any speed increase on a smaller dataset could be beneficial in a larger dataset (Neo4j, 2016).

Below there are a few advantages and disadvantages to Neo4j, whilst there are only a few mentioned here, most of their advantages come from being a NoSQL document type db.

| **Advantages** | **Disadvantages** |
|---|---|
| Performance: As graph databases are based on NoSql formats which has the benefits of speed and scalability. | New Methodology: A different way of thinking about how to store your data compared to RDMS' which have been widely used for years. |
| Easy to read: As the graph data is visualised, it can be easy to read and understand the relationships. | Learning curve: As the database is not structured, it can be difficult to learn at the start. |

## 2.5 – Conclusion

Whilst I have personally never used a graph-based database or heard about knowledge graphs, since doing this report I have been interested in checking out a project from GitHub which looks to explore Cayley with a Pokémon dataset. This could be a fun project to explore to get a better understanding and get my hands stuck in (which is the best way for me to learn) src.

## 3 - References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z., 2007. Dbpedia: A nucleus for a web of open data. In The semantic web (pp. 722-735). Springer, Berlin, Heidelberg.

Bulao, J., 2021 *How Much Data Is Created Every Day in 2021?* Available at: How Much Data Is Created Every Day in 2021? [You'll be shocked!] (techjury.net) (Accessed 11/05/2021)

CoreViewSystem, 2021 *NoSql Databases – Document Stores* Available at: https://coreviewsystems.com/nosql-databases-document-stores/ (Accessed 04/05/2021)

DBpedia, 2021 *About DBpedia* Available at: https://www.dbpedia.org/about/ (Accessed 12/05/2021)

Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J. and Wahler, A., 2020. Knowledge Graphs. Springer International Publishing.

Fullstack Academy (2017) *RDF Tutorial - An Introduction to the Resource Description Framework* Available at: https://www.youtube.com/watch?v=zeYfT1cNKQg (Accessed 11/05/2021)

Google (2012) *Introducing the Knowledge Graph* Available at: https://www.youtube.com/watch?v=mmQl6VGvX-c (Accessed 11/05/2021)

Gouveia, A. and Cardoso, J., 2009. OWL: Web Ontology Language. In Encyclopedia of Information Science and Technology, Second Edition (pp. 3009-3017). IGI Global.

Grakn, 2021 *Building a Text Mined Knowledge Graph is Easy* Available at: https://grakn.ai/text-mining (Accessed 11/05/2021)

Guillaume, R., 2014 *Visualizing the Cayley Graph Database* Available at: https://cambridge-intelligence.com/visualizing-cayley-graph-database-keylines/ (Accessed 11/05/2021)

Hazelcast, 2021 *What is a Data Pipeline* Available at: https://hazelcast.com/glossary/data-pipeline/ (Accessed 07/05/2021)

Jain, P., Hitzler, P., Sheth, A.P., Verma, K. and Yeh, P.Z., 2010, November. Ontology alignment for linked open data. In International semantic web conference (pp. 402-417). Springer, Berlin, Heidelberg.

Kejriwal, M., Knoblock, C.A. and Szekely, P., 2021. Knowledge Graphs: Fundamentals, Techniques, and Applications. MIT Press.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S. and Bizer, C., 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6(2), pp.167-195.

McGuinness, D.L. and Van Harmelen, F., 2004. OWL web ontology language overview. W3C recommendation, 10(10), p.2004.

MongoDB, 2021 *Structure your Data for MongoDB* Available at: https://docs.mongodb.com/guides/server/introduction/ (Accessed 03/05/2021)

MongoDB, 2021 *The database for modern applications* Available at: https://www.mongodb.com/ (Accessed 30/04/2021)

Neo4j (2016) *Intro to Neo4j* Available at: https://www.youtube.com/watch?v=U8ZGVx1NmQg (Accessed 12/05/2021)

Neo4j, 2013 *Importing Data into Neo4j – the Spreadsheet Way* Available at: https://neo4j.com/blog/importing-data-into-neo4j-the-spreadsheet-way/ (Accessed 13/05/2021)

Neo4j, 2016 *Welcome to the Dark Side: Neo4j Worst Practices (& How to Avoid Them)* Available at: https://neo4j.com/blog/dark-side-neo4j-worst-practices/ (Accessed 13/05/2021)

Vergadia, P., 2020 *How to Build a Scalable Data Analytics Pipeline* Available at: https://www.freecodecamp.org/news/scalable-data-analytics-pipeline/ (Accessed 07/05/2021)

Viaboxx, 2021 *Introduction to The Titan Graph Database* Available at: https://www.viaboxx.com/uncategorized/introduction-to-the-titan-graph-database/ (Accessed 11/05/2021)

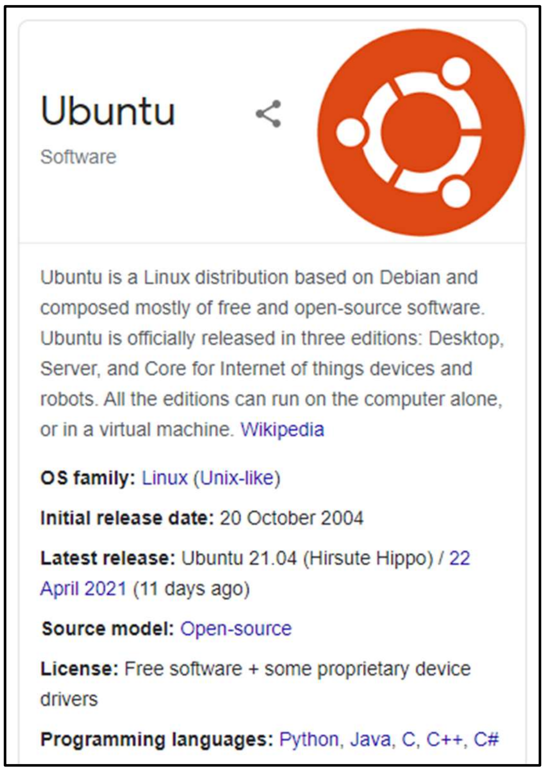
Virtuoso, 2021 *Data-driven agility without compromise* Available at: https://virtuoso.openlinksw.com/ (Accessed 12/05/2021)

Wikipedia, 2021 *Wikipedia: Size of Wikipedia* Available at: https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia (Accessed 12/05/2021)

ZenOfAI, 2021 *Scaling DynamoDB for Big Data using Parallel Scan* Available at: https://medium.com/zenofai/scaling-dynamodb-for-big-data-using-parallel-scan-1b3baa3df0d8 (Accessed 07/05/2021)

# 4 – Appendices

## 4.1 – Example of Knowledge Panel



Ubuntu

Software

Ubuntu is a Linux distribution based on Debian and composed mostly of free and open-source software. Ubuntu is officially released in three editions: Desktop, Server, and Core for Internet of things devices and robots. All the editions can run on the computer alone, or in a virtual machine. Wikipedia

**OS family:** Linux (Unix-like)

**Initial release date:** 20 October 2004

**Latest release:** Ubuntu 21.04 (Hirsute Hippo) / 22 April 2021 (11 days ago)

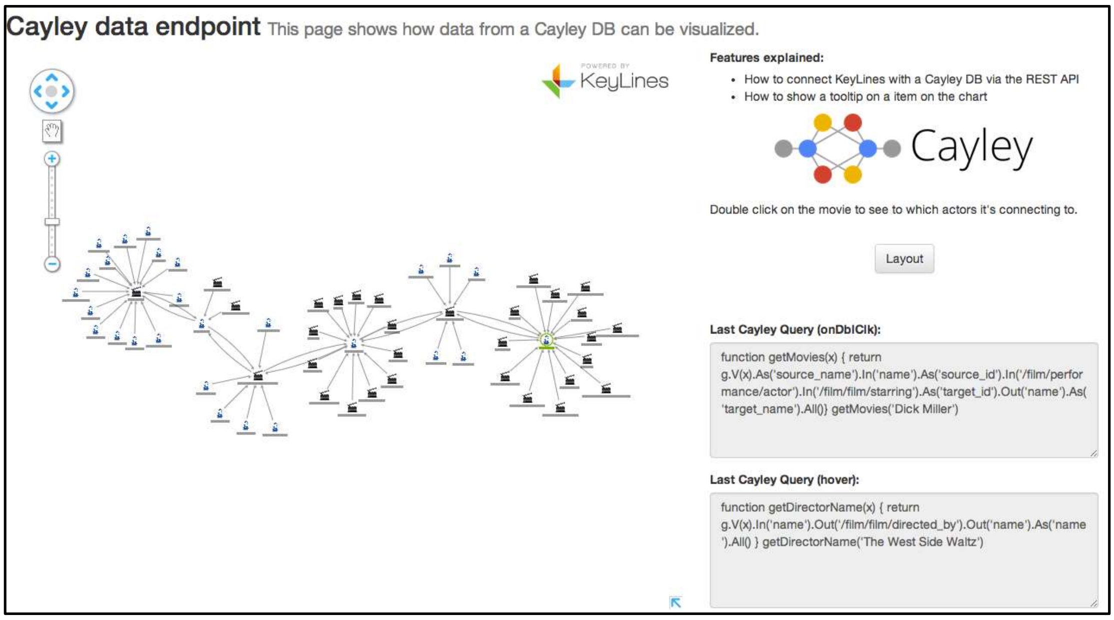
**Source model:** Open-source

**License:** Free software + some proprietary device drivers

**Programming languages:** Python, Java, C, C++, C#

(Google search of Ubuntu, 2021)

## 4.2 – Example of a Cayley Project



**Cayley data endpoint** This page shows how data from a Cayley DB can be visualized.

**Features explained:**

- How to connect KeyLines with a Cayley DB via the REST API
- How to show a tooltip on a item on the chart

Double click on the movie to see to which actors it's connecting to.

**Last Cayley Query (onDblClk):**

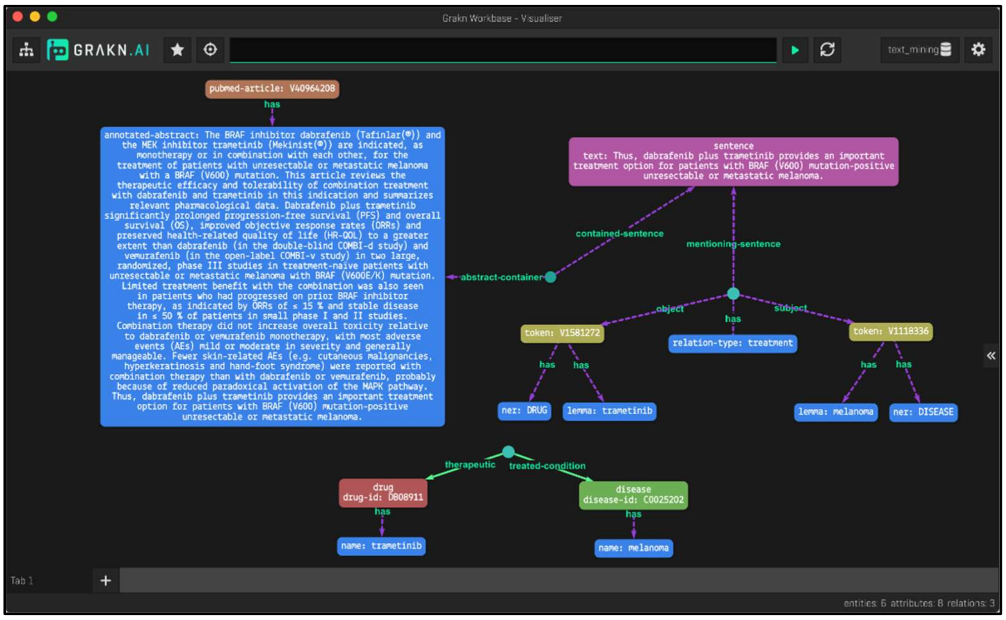
```
function getMovies(x) { return
g.V(x).As('source_name').In('name').As('source_id').In('/film/perfor
mance/actor').In('/film/film/starring').As('target_id').Out('name').As(
'target_name').All()} getMovies('Dick Miller')
```

**Last Cayley Query (hover):**

```
function getDirectorName(x) { return
g.V(x).In('name').Out('/film/film/directed_by').Out('name').As('name
').All() } getDirectorName('The West Side Waltz')
```

(Guillaume, R., 2014)

## 4.3 – Example of a Grakn Project

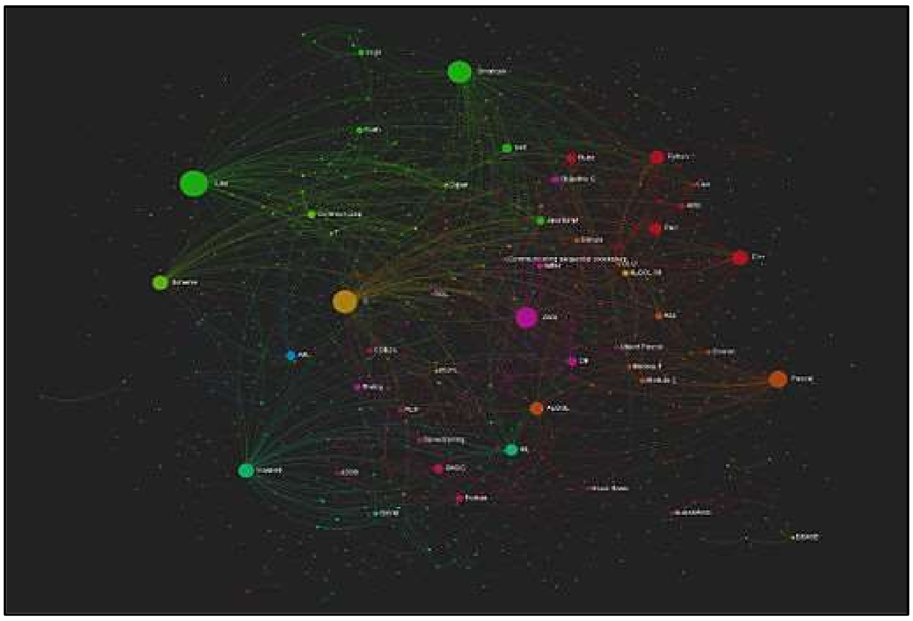


annotated-abstract: The BRAF inhibitor dabrafenib (Tafinlar(®)) and the MEK inhibitor trametinib (Mekinist(®)) are indicated, as monotherapy or in combination with each other, for the treatment of patients with unresectable or metastatic melanoma with a BRAF (V600) mutation. This article reviews the therapeutic efficacy and tolerability of combination treatment with dabrafenib and trametinib in this indication and summarizes relevant pharmacological data. Dabrafenib plus trametinib significantly prolonged progression-free survival (PFS) and overall survival (OS), improved objective response rates (ORRs) and preserved health-related quality of life (HR-QOL) to a greater extent than dabrafenib (in the double-blind COMBI-d study) and vemurafenib (in the open-label COMBI-v study) in two large, randomized, phase III studies in treatment-naïve patients with unresectable or metastatic melanoma with BRAF (V600E/K) mutation. Limited treatment benefit with the combination was also seen in patients who had progressed on prior BRAF inhibitor therapy, as indicated by ORRs of ≤ 15 % and stable disease in ≤ 50 % of patients in small phase I and II studies. Combination therapy did not increase overall toxicity relative to dabrafenib or vemurafenib monotherapy, with most adverse events (AEs) mild or moderate in severity and generally manageable. Fewer skin-related AEs (e.g. cutaneous malignancies, hyperkeratinosis and hand-foot syndrome) were reported with combination therapy than with dabrafenib or vemurafenib, probably because of reduced paradoxical activation of the MAPK pathway. Thus, dabrafenib plus trametinib provides an important treatment option for patients with BRAF (V600) mutation-positive unresectable or metastatic melanoma.

sentence
text: Thus, dabrafenib plus trametinib provides an important treatment option for patients with BRAF (V600) mutation-positive unresectable or metastatic melanoma.

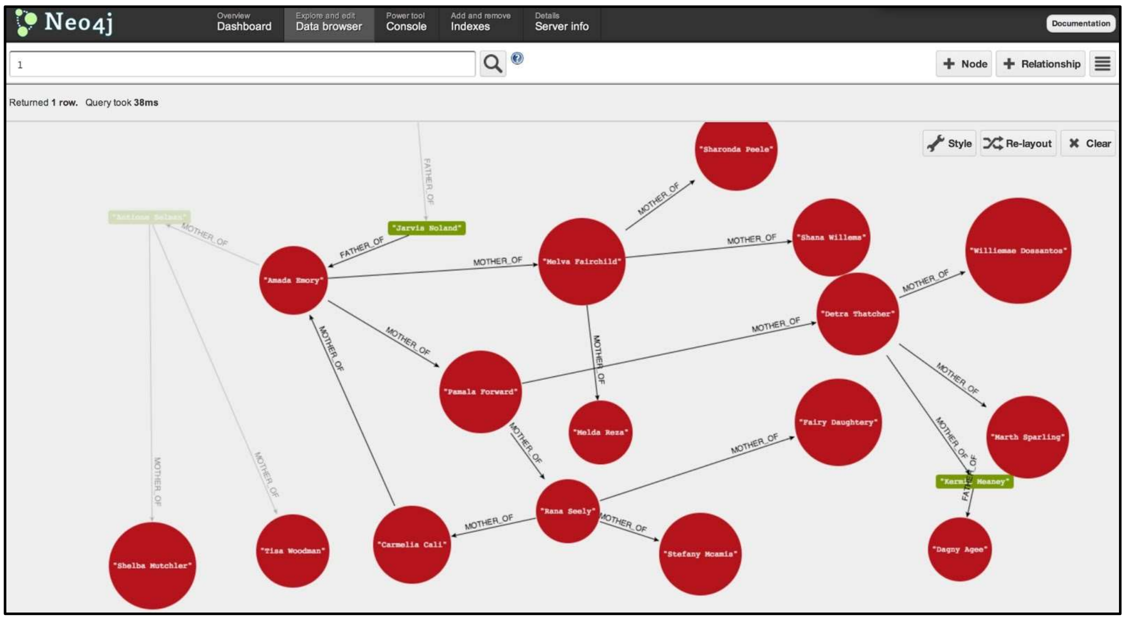
(Grakn, 2021)

## 4.4 – Example of a Titan Project



(Viaboxx, 2021)

## 4.5 – Example of a Neo4j Project



(Neo4j, 2013)