# Introduction

### Knowledge graphs:

The term “knowledge graph” gained popularity in 2012 when Google launched its own *Google Knowledge Graph*. A knowledge graph is a collection of data represented as a graph. The collected data conveys knowledge of the real world, where the nodes represent entities of interest and edges the many different relations between those entities \cite{Hogan2021}. The entities are real world objects and abstract concepts. For example, “Helium has the chemical formula He”, is a statement that can be represented using a knowledge graph. Here the nodes of the graph would represent “Helium” and “He”, while the connecting edge between those nodes would represent the relation “chemical formula”.

There are many ways of modelling data as a graph. The most commonly used ones are directed edge-labelled graphs, heterogeneous graphs, property graphs and graph dataset \cite{Hogan2021}. We will see in section XYZ how we can use RDF to specify directed edge-labelled graphs.

The term knowledge base is used synonymously with knowledge graph but there is a small difference \cite{Ji2022}. Normally, a knowledge graph is viewed as a graphical structure. However, when defined semantically, it is considered to be a knowledge base for interpretation and inference from facts \cite{Bordes2011}.

Many companies such as Amazon, Facebook, Uber, Google, etc., use knowledge graphs for their applications. Depending on the organization or community there are open or enterprise knowledge graphs \cite{Hogan2021}. Open knowledge graphs include Wikidata, DBpedia, Freebase, YAGO, etc. These are available online and freely accessible to the public. Enterprise knowledge graphs are generally used internally within a company and have their commercial specific use-cases \cite{Hogan2021}.

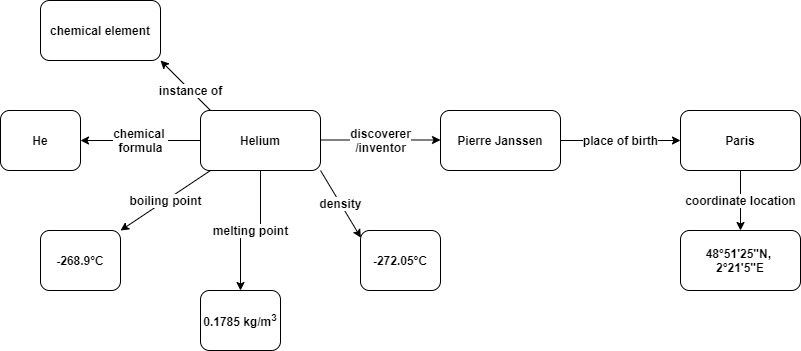
Wikidata is an open knowledge graph developed by Wikimedia Deutschland. It contains structured data and acts as a central database for Wikimedia projects like Wikipedia. The data in Wikidata is converted into RDF, and thus can be queried using SPARQL, which is a query language for RDF. However, in commercial applications the use of SPARQL remains limited. One of the main reasons is that developers are not often learned or experienced in the triples that RDF offers.

GraphQL is a query language that is more popular in commercial applications. It is easy to learn and use, providing syntax that is more human friendly than SPARQL. Our goal in this report is to provide a research on ways to query Wikidata using GraphQL, and offer two implementations for this purpose. We also provide comparisons between the implementations, and evaluate their performances along with limitations.

The remainder of the report is structured as follows. In section 2, we provide an overview of RDF, Wikidata, SPARQL and GraphQL. We also give a comparison between SPARQL and GraphQL. Section 3 gives the approaches used to query RDF graphs. The implementation of the approaches on Wikidata, along with the technicalities, and the differences in the SPARQL queries generated by the tools is provided in Section 4. The differences between the generated SPARQL queries and handwritten SPARQL queries is also shown here. This section also includes the performance and limitations of the tools. Lastly, we provide a conclusion and future work in Section 5. Also say sth about glossary and appendix.

### RDF:

The Resource Description Framework (RDF) is a framework used to represent information available in the Web \cite{R.Cyganiak2014}. In the context of graphs, RDF is used for describing and exchanging graphs. The graphs specified by RDF are directed edge-labelled graphs. This means that the edges connect source nodes to target nodes, and have labels. It can be the case that there are multiple edges between the same nodes. However, these edges must have different labels. Fig 1 shows how knowledge about the chemical element Helium can be represented using a directed edge-labelled graph.



Resources in RDF are divided into groups known as classes. Ever member of a class is called an instance of that class. Properties describe the relationship between subject and object resources. In Figure 1 we see that Helium is an instance of the class chemical element. The property instance of describes the relation between the subject Helium and object chemical element.

In order to exchange graphs across the web we need to identify the resources uniquely. For this we use IRIs which are basically identifiers in RDF. The graph shown in Fig 1 can be represented using an RDF graph. Formally, the building blocks of RDF graphs are IRIs, literals and blank nodes. These are defined as follows.

IRIs

A URI (Uniform Resource Identifier) is a sequence of a subset of ASCII characters that identifies any web resource by using a name, a location, or both. They have a scheme, authority, path, and query and fragment, where all parts other than scheme and path are optional. URIs are of the form ***scheme:[//authority]path[?query][#fragment].*** For example, <http://www.wikidata.org/entity/Q560> is an IRI that identifies the chemical element Helium on Wikidata. A URL (Uniform Resource Locator) is a subset of URI that is used to specify the location of a digital document.

An Internationalized Resource Identifier (IRI) is a generalized form of URI that helps to distinguish resources with Unicode. Basically, the character set in URI is extended to the Universal Coded Character Set. This enables it to contain any Latin and non-Latin characters except the reserved characters.

In RDF an IRI is used as a name (can be thought of as an ID) for [graph nodes](https://www.oxfordsemantic.tech/fundamentals/data-relationships-in-a-graph). It defines the resources that appears as nodes or edge labels in a RDF graph. There are already several pre-existing IRIs available for common use. New domain specific IRIs can be created based on the application. However, we must ensure there no conflicts with other IRIs available on the web.

RDF Literals

An RDF literal consists of three essential elements: a lexical value, a datatype IRI and an optional language tag. The lexical value is a string (footnote: RDF is based on Unicode strings) that corresponds to a particular literal value in the value space, where value space is the set of all possible values that a datatype can have. There are many datatypes in RDF some of which are string, Boolean, decimal and integer(footnote:A full list is available on the W3C’s section on RDF datatypes: [www.w3.org/TR/2014/REC-rdf11-concepts-20140225/#section-Datatypes](http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/#section-Datatypes)).

The datatype IRI refers to a datatype that defines which strings are valid (belong in the lexical space), the value space and the lexical-to-value mapping \cite{ Bonduel2019}. This mapping is essentially a function that maps each string from the lexical space to an element in the value space. The W3C standard XML Schema defines the datatypes and their IRIs. For example, decimals are identified by the IRI <http://www.w3.org/2001/XMLSchema#decimal>. W3C has a good documentation on the different XML Schema built-in datatypes \cite{ R.Cyganiak2014}.

The optional language tag helps to provide human-readable labels to RDF literals. A literal is a language-tagged string is of the form “string”@language (footnote: Here language is a well-formed

language tag (after BCP47)”. The datatype IRI of such literals is <http://www.w3.org/1999/02/22-rdf-syntax-ns#langString> (footnote: It is never used in syntax).

RDF literals are used to represent resources that have values belonging to datatypes. Each literal can have only one datatype. For example, the boiling point of Helium would be a RDF literal represented as *-268.9^^xsd:decimal* and its chemical formula as “He”@en, which is a language-tagged string. Literals are drawn as rectangular nodes in RDF graphs.

Blank nodes

A blank node in RDF, also known as a bnode, does not identify a specific resource as IRIs or literals do. It is used as a placeholder for some node, i.e., it is used to say that something with the given relationship exits at the position without specifying what the node is.

## RDF Graph

An RDF graph is a directed edge-labelled graph composed of a set of triples. These triples consist of the following elements:

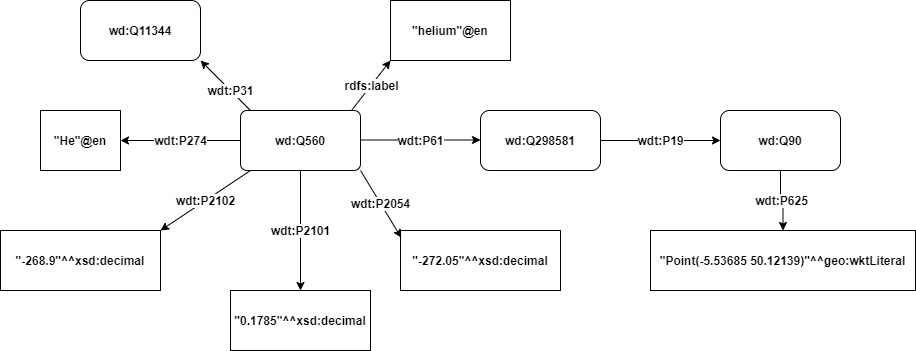
1) a subject (node) that is an IRI or a blank node;

2) a predicate (edge) that is an IRI;

3) an object (node) that is an IRI, a blank node, or a literal

Figure 2 shows an RDF graph based on our example represented in Fig 1. Our main interest is in querying the knowledge graph Wikidata, and so all the data correspond to the resources in its knowledge base. In Wikidata items are things in human knowledge that can be topics, concepts and objects. For example, Helium is an item. In Wikidata the subject and object represent items, and the predicate represents properties. All items and properties are identified as Unique IDs. For example, the item Helium has the ID of Q560, and the property chemical formula has the ID P274. These are not understood by humans and have a label property that makes them understood. Moreover, items belong to the namespace <http://www.wikidata.org/entity/> (prefixed by wd) and properties to <http://www.wikidata.org/prop/direct/> (prefixed by wdt). As a result, Helium would have the IRI <http://www.wikidata.org/entity/Q560> (wd:Q560) and chemical formula the IRI <http://www.wikidata.org/prop/direct/P274> (wdt:P274). Section XYZ gives an elaborate understanding of entities and the namespaces they belong to in Wikidata.

From Figure 2 we understand that **Helium** (Q560) is an **instance of** (P31) of **chemical element** (Q11344). It has a human understandable english label **helium** and the **chemical formula** (P274) **He**. Its **boiling point** (P2102), **melting point** (P2102) and **density** (P2054) are **-268.9°C, 0.1785°C and -272-05 kg/m3** respectively. Helium has a **discoverer/inven**tor (P61) by the name of **Pierre Janssen** (Q298581). His **place of birth** (P19) was **Paris** (Q90) that belongs to the **country** (P17) of **France** (Q142).



## Serialisations

For exchanging graphs across the web, we need a syntactical representation of RDF. There are different formats available (footnote: A full list along with explanation is available on the W3C’s section on RDF Syntax https://www.w3.org/wiki/RdfSyntax), the most common ones are N-Triples, Turtle, JSON-LD, RDF/XML and RDFa. In this report we focus on N-Triples and Turtle.

## N-Triples

N-Triples represents RDF graphs in a simple line-based format (footnote: Full specification available at: https://www.w3.org/TR/n-triples/). Every triple is encoded in a single line. The IRIs are written within pointy brackets and literals are written as lexical value^^datatype-IRI. Blank nodes are represented as \_:stringID, where stringID can be any string used to identify the blank node in the document. After every element of a triple there is a whitespace, and all the lines end with a dotWe can use comments using hash after the end of every triple in a line or in a single dedicated line, and they are treated as white spaces. The files are saved with a .nt extension.

Listing~\ref{listing:listing1} shows the representation of the RDF graph in Figure~\ref{fig:figure 2} in N-Triples format. We have given line breaks for better readability.

<<http://www.wikidata.org/entity/>Q560> <<http://www.wikidata.org/prop/direct/>P31> <<http://www.wikidata.org/entity/>Q11344> .

<<http://www.wikidata.org/entity/>Q560> <<http://www.w3.org/2000/01/rdf-schema#label>> “helium”@en .

<<http://www.wikidata.org/entity/>Q560> <<http://www.wikidata.org/prop/direct/>P274> “He”@en .

<<http://www.wikidata.org/entity/>Q560> <<http://www.wikidata.org/prop/direct/>P2102> “-268.9”^^< <http://www.w3.org/2001/XMLSchema#decimal>> .

<<http://www.wikidata.org/entity/>Q560> <<http://www.wikidata.org/prop/direct/>P2101> “0.1785”^^< <http://www.w3.org/2001/XMLSchema#decimal>> .

<<http://www.wikidata.org/entity/>Q560> <<http://www.wikidata.org/prop/direct/>P2054> “-272.05”^^< <http://www.w3.org/2001/XMLSchema#decimal>> .

<<http://www.wikidata.org/entity/>Q560> <<http://www.wikidata.org/prop/direct/>P61> <<http://www.wikidata.org/entity/>Q298581> .

<<http://www.wikidata.org/entity/>Q298581> <<http://www.wikidata.org/prop/direct/>P19> <<http://www.wikidata.org/entity/>Q90> .

<<http://www.wikidata.org/entity/>Q90> <<http://www.wikidata.org/prop/direct/>P17 <<http://www.wikidata.org/entity/>Q142>.

## Turtle

Turtle is an easy to read representation of RDF graphs. It extends the N-Triples format by providing several simplifications (footnote: Full specification available at: https://www.w3.org/TR/turtle/.). We can use prefix declarations and base namespaces at the beginning of the file to shorten IRIs. Turtle allows us to avoid repetition. We can use a semicolon (;) at the end of a line instead of a dot (.) if we know the next line has the same subject. Consequently, the next line will only have a predicate and object omitting the subject. Also, we can use a comma (,) at the end of a line if we know the next line will start with the same subject and predicate. Analogously, the next line will only have a an object omitting the subject and the predicate. Blank nodes are represented using only square brackets ([]). Additionally, we can provide predicate-object pairs within the square brackets to give further triples keeping the blank node as the subject. Turtle also provides a shorthand syntax for writing numbers. Numbers of the datatype integer, decimal and double can be written without quotes and datatype-IRIs. Boolean values can be written directly as either 'true' or 'false'. The files are saved with a .ttl extension.

Listing~\ref{listing:listing2} shows the representation of the RDF graph in Figure~\ref{fig:figure 2} in Turtle format.

PREFIX wd: <<http://www.wikidata.org/entity/>>

PREFIX wdt: <<http://www.wikidata.org/prop/direct/>>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <[http://www.w3.org/2001/XMLSchema#](http://www.w3.org/2001/XMLSchema#decimal)>

PREFIX geo: <[http://www.opengis.net/ont/geosparql#](http://www.opengis.net/ont/geosparql#wjktLiteral)>

wd:Q560 wdt:P31 wd:Q11344 ;

rdfs:[label](http://www.w3.org/2000/01/rdf-schema#label) “helium”@en ;

wdt:P274 “He”@en ;

wdt:P2102 -268.9 ;

wdt:P2101 0.1785 ;

wdt:P2054 -272.05 ;

wdt:P61 wd:Q298581 .

wd:Q298581 wdt:P19 wd:Q90 .

wd:Q90 wdt:P17 wd:Q142.

### Wikidata:

Wikidata is a free and publicly available knowledge base of Wikipedia. It also acts as a central storage and data management platform (for structured data) for a variety of Wikimedia projects like Wikipedia, Wikitionary and Wikisource. Wikidata was created in 2012 at Wikimedia Deutschland by a community of volunteers. These volunteers edit and control all content. As of December 2022, Wikidata has more 23000 active editors /footnote(https://wikidata.wikiscan.org/). Wikidata provides a website where data can be viewed and also edited \cite{Foundationa}.

There is a wide range of applications for Wikidata. We previously mentioned that Wikidata functions as a database for Wikimedia community like Wikipedia. It also has external uses in many large organizations like Eurowings, Google, Apple and Amazon for tasks such as data integration, authority control, identity providing and data-driven journalism. In the field of research, Wikidata is used for collecting test data for knowledge graph related algorithms and training data for machine learning projects.

Wikidata is built on the RDF framework. However, it does not define its resources in terms of RDF. Instead, it has its own model known as Wikibase data model (footnote: https://www.mediawiki.org/wiki/Wikibase/DataModel). This distinction creates an abstract layer between its model and RDF. Consequently, there are similarities with RDF’s W3C standards but there are also some important differences. For instance, the Wikidata property instance of (P31) is semantically equivalent to the property rdf:type in RDF. The value of P31 is a class that is itself an item. Wikidata offers an explanation for some of the standards properties in Wikidata that correspond to the ones in RDF \cite{Foundation}. Statements (triples) in Wikidata can have annotations (qualifiers) and references.

The basic elements in Wikidata are entities, also known as resources. These are mainly items and properties. The next section describes entities in Wikidata.

It as an RDF export format used for export and indexing purposes (footnote: https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF\_Dump\_Format). Fig 3 shows the key components of the format.

## Entities

Wikidata maintains its data structure by pages. Each entity has a dedicated page for itself. Basically, every element on which Wikidata has structured data is known as an entity\cite{Erxleben2014}. Entities are identified by an unique ID and not by names or labels. Currently, Wikidata has mainly three types of entities – item, property and lexeme. Other extensions can define new entity types such as MediaInfo and subentities like Form and Sense. We will discuss about items and properties. The rest are out of the scope of this report.

Items are real-world objects, concepts or events. According to RDF terminology, items are instances and classes. Instead of a human understandable name, they are identified by a QID - an ID prefixed with the letter “Q” and followed a number. Items in Wikidata belong to the main namespace - <http://www.wikidata.org/wiki/QID>. Every item constitutes of the following main parts - labels, descriptions, aliases, sitelinks and statements\cite{Erxleben2014} .

Labels, descriptions and aliases are multilingual which help to find the respective item. Since items are identified by an ID, these are used to identify the items clearly. Sitelinks provide links about an individual item in Wikidata to external pages on other Wikimedia sites such as [Wikipedia](https://en.wikipedia.org/wiki/Wikipedia) and its other sister projects. The most important part are statements.

A statement in Wikidata consists of a claim, references and a rank. Claims are property-value pairs, which along with the item form a RDF triple, where the item is the subject. Claims can also contain optional qualifiers that provide some additional information for the claim. This information is a property-value pair that refers to the main part of the statement instead of the item itself\cite{Erxleben2014}. References point to resources that support the claim. An item can have several statements for the same property, and all of them might not necessarily be important or relevant. Ranks are used as a quality factor to distinguish between several statements. A Wikidata statement can have one of three types of ranks - “normal”, “preferred” and “deprecated”. By default, a statement has the normal rank unless changed to preferred or deprecated. A statement with preferred rank means it should be given priority over the normal ranked statements. A deprecated rank indicates that the statement is incorrect or under discussion, and it may have a reference. Deprecated statements are kept either for the sake for completion or to prevent users from constantly adding or removing them. Wikidata has around 100 million items (footnote: https://grafana.wikimedia.org/d/000000167/wikidata-datamodel) and around 1.43 billion item statements as of December 2022 (footnote: <https://grafana.wikimedia.org/d/000000175/wikidata-datamodel-statements>).

Figure 3 shows an excerpt of the Wikidata page on the item Helium (footnote: <https://www.wikidata.org/wiki/Q560>). We can see Helium has the QID of Q560, and has labels, descriptions and aliases in different languages. It has a sitelink for the item in Wikipedia offered in 186 languages. Helium has a statement that indicates that Helium has the *instance of* property with *chemical element* as the values. The property-value pairs *follows*, *hydrogen* and *followed by*, *lithium* are qualifiers. This statement hence gives us the information that Helium is an instance of chemical element, and it comes after the element hydrogen and if followed by the element lithium. This statement has no references and has the normal rank (indicated by the middle portion greyed).

Properties in Wikidata resemble RDF properties and are essentially attributes for describing entities. They are identified by a PID - an ID prefixed with the letter “P” and followed a number. They belong to the property namespace in wikidata - <http://www.wikidata.org/wiki/Property:PID.> Like items, they also have labels, descriptions, aliases and statements but no sitelinks. However, they has an additional part called datatype that determines which values they accept, such as string, quantity and time (footnote: Wikidata provides a list of all the datatypes: <https://www.wikidata.org/wiki/Special:ListDatatypes>).

Figure 4 shows an excerpt of the property *instance of* page on Wikidata (footnote: <https://www.wikidata.org/wiki/P31>). The property has a PID of P31 along and with multilingual labels, descriptions and aliases. From the statement we understand that it this property also has an instance of property with Wikidata property being the value. The statement has no qualifiers or references, and has the normal rank. The instance of property accepts the datatype item as a value.

Wikidata contains different data types like text, images, dates, etc., that can be queried using SPARQL. Section XYZ gives an introduction on SPARQL.

**a dataset** that can be downloaded and freely used and distributed - 14

## Querying Wikidata

The easiest and most popular way to query Wikidata is through the Wikidata Query Service (WDQS). This is Wikidata’s SPARQL endpoint. We can use this service two ways. Firstly, we can write queries in SPARQL directly on the web user interface of the service (footnote https://query.wikidata.org/) and obtain the results in different formats like table, tree, graph, etc. Secondly, the service can also be used pragmatically by submitting GET or POST requests (footnote <https://query.wikidata.org/sparql>).

Another popular way to query Wikidata is by using the Wikidata API (footnote https://www.wikidata.org/wiki/Special:ApiSandbox). However, this API should mainly be used when we want to edit the contents of Wikidata or get data about entities like revision history.

Wikidata dumps is useful when we know our result set will be significantly large or if we want to set up our own local query service. These dumps are full exports of all the available entities in Wikidata (footnote <https://dumps.wikimedia.org/>). To get started we should download the latest complete dump (footnote <https://dumps.wikimedia.org/wikidatawiki/latest/>). Wikidata also mentions some other ways to accessing Wikidata’s data like Search and Linked Data Fragments endpoint, the complete list and usage of which can be found on Wikidata’s Data Access webpage(footnote: https://www.wikidata.org/wiki/Wikidata:Data\_access).

### SPARQL:

SPARQL Protocol and RDF Query Language (SPARQL) is a W3C recommended query language for RDF. This means it allows to query any data source that can be mapped to RDF. It is also a HTTP-based protocol for linked open data on the web. This enables the transmission of SPARQL queries and results between a client and a SPARQL engine. The first working draft for SPARQL was released in 2004 and it became a W3C Recommendation in 2008 \cite{Perez2009}.

W3C provides a comprehensive document on SPARQL \cite{C.B.Aranda2013}.

Queries in SPARQL are based on matching graph patterns and can be used to retrieve, add or delete data in the RDF based dataset. In section XYZ we saw that RDF data is based on triples - subject, object and predicate. Consequently, a query in SPARQL consists of a set of triple patterns. Each of the elements of the triple can be a variable (a string beginning with ? or $) that needs to be queried. The solution to the variables is obtained by matching the query patterns to the triples in the dataset.

There are four forms of queries - SELECT, ASK, CONSTRUCT and DESCRIBE.

* SELECT queries select some or all the pattern matches and provides the results in a tabular format
* ASK queries check whether there is at least one match and the result is true or false
* CONSTRUCT queries create an RDF graph based on the query results
* DESCRIBE queries return a RDF graph providing additional information on each results

In our work, we only consider SELECT queries. These consist of the following major blocks:

* Prologue: PREFIX and BASE keywords that function similarly to those in RDF turtle format
* Select clause: SELECT keyword followed by either a list of variables and variable assignments, or by \*;
* Where clause: WHERE keyword followed by a query graph pattern to be matched
* Solution set modifiers: Change the set of solutions using modifiers such as LIMIT and OFFSET

The select and where clauses are mandatory, the rest being optional. Other optional features are filters, groups, query operators such as UNION, OPTIONAL and BIND, and aggregates. A full specification for the query language can be found on the official W3C documentation\cite{Seaborn}.

Listing 2 shows an example of querying Wikidata using SPARQL. We want to get a a list of all chemical elements, along with their English labels, that have a chemical formula, boiling point, melting point, density, an inventor/discoverer, birth place of that inventor/discoverer and the country that the place belongs to. Since there might be several results, we are limiting them to 5 using the LIMIT keyword. The namespace <http://www.wikidata.org/entity/> is used for items when querying. We are interested in the truthy values of the properties and so the namespace <http://www.wikidata.org/prop/direct/> is used for properties. Truthy values are essentially the values for which the statement has the best non-deprecated rank. This means that if a statement has preferred rank then that statement is considered to be truthy. Otherwise, the normal ranked statement is taken to be truthy. The PREFIX is optional since Wikidata recognizes the short forms wd and wdt automatically. Turtle syntax that we saw in section XYZ can be applied in SPARQL. The query can be run in Wikidata’s [query service](https://query.wikidata.org/#%23Here%20is%20an%20example%20of%20a%20SPARQL%20query%20that%20consists%20of%202%20depths%3A%0A%23Chemical%20elements%20with%20their%20properties%20and%20details%20about%20discoverer%0A%23Basically%20the%20element%20has%20a%20discoverer%2C%20the%20discoverer%20then%20has%20a%20place%20of%20birth%2C%20and%20then%20place%20of%20birth%20has%20coordinates.%0A%0ASELECT%20%3Felement%20%3Felement_formula%20%3Fboiling_point%20%3Fmelting_point%20%3Fdensity%20%3Fdiscoverer%20%3Fplace_birth%20%3Fplace_coordinate%0AWHERE%20%7B%0A%20%20%3Felement%20wdt%3AP31%20wd%3AQ11344.%0A%20%0A%20%0A%20%20%3Felement%20wdt%3AP274%20%3Felement_formula.%20%0A%20%0A%20%0A%20%20%3Felement%20wdt%3AP2102%20%3Fboiling_point.%0A%20%0A%20%0A%20%20%3Felement%20wdt%3AP2101%20%3Fmelting_point.%0A%20%0A%20%0A%20%20%3Felement%20wdt%3AP2054%20%3Fdensity.%0A%20%0A%20%0A%20%20%3Felement%20wdt%3AP61%20%3Fdiscoverer.%0A%20%0A%20%0A%20%20%3Fdiscoverer%20wdt%3AP19%20%3Fplace_birth.%0A%20%0A%20%0A%20%20%3Fplace_birth%20wdt%3AP625%20%3Fplace_coordinate.%0A%7D)\footnote{https://query.wikidata.org/}.

PREFIX wd: <<http://www.wikidata.org/entity/>>

PREFIX wdt: <<http://www.wikidata.org/prop/direct/>>

SELECT \*

WHERE {

?element wdt:P31 wd:Q11344 ;

wdt:P274 ?element\_formula ;

wdt:P2102 ?boiling\_point ;

wdt:P2101 ?melting\_point ;

wdt:P2054 ?density ;

wdt:P61 ?discoverer .

?discoverer wdt:P19 ?place\_birth .

?place\_birth wdt:P17 ?country.

FILTER(LANG(?element\_label)="en")

} LIMIT 5

Table 1 shows the results obtained in a tabular form. Among the results, there is the element Helium (Q560) that we have used as an example in Fig 1 and Fig 2.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| element | elemen  \_formula | element\_  label | boiling\_  point | melting\_  point | density | discoverer | place\_  birth | country |
| [wd:Q560](http://www.wikidata.org/entity/Q560) | He | helium | -268.9 | -272.05 | 0.1785 | [wd:Q298581](http://www.wikidata.org/entity/Q298581) | [wd:Q90](http://www.wikidata.org/entity/Q90) | [wd:Q142](http://www.wikidata.org/entity/Q142) |
| [wd:Q560](http://www.wikidata.org/entity/Q560) | He | helium | -268.9 | -272.05 | 0.1785 | w[d:Q950726](http://www.wikidata.org/entity/Q950726) | [wd:Q4093](http://www.wikidata.org/entity/Q4093) | [wd:Q145](http://www.wikidata.org/entity/Q145) |
| [wd:Q560](http://www.wikidata.org/entity/Q560) | He | helium | -268.9 | -272.05 | 0.1785 | [wd:Q127959](http://www.wikidata.org/entity/Q127959) | [wd:Q623765](http://www.wikidata.org/entity/Q623765) | [wd:Q145](http://www.wikidata.org/entity/Q145) |
| [wd:Q670](http://www.wikidata.org/entity/Q560) | Si | silicon | 4271 | 2570 | 2.329 | [wd:Q151911](http://www.wikidata.org/entity/Q151911) | [wd:Q1451001](http://www.wikidata.org/entity/Q1451001) | [wd:Q34](http://www.wikidata.org/entity/Q34) |
| wd:Q568 | Li | lithium | 1317 | 180.5 | 0.535 | w[d:Q313568](http://www.wikidata.org/entity/Q313568" \o "" \t "_blank) | [wd:Q10495519](http://www.wikidata.org/entity/Q10495519) | [wd:Q34](http://www.wikidata.org/entity/Q34) |

# Acronyms

RDF: Resource Description Framework

URI: Uniform Resource Identifier

IRI: Internationalized Resource identifier

URL: Uniform Resource Locator

BCP47: Best Current Practice 47

SPARQL: SPARQL Protocol and RDF Query Language

W3C: World Wide Web Consortium

# Glossary

wd: Is the prefix for the namespace - http://www.wikidata.org/entity used for items in Wikidata

wdt: Is the prefix for the namespace - http://www.wikidata.org/prop/direct/ used for properties in Wikidata

rdf: Is the prefix for the namespace - http://www.w3.org/1999/02/22-rdf-syntax-ns# used in RDF

rdfs: Is the for the namespace - http://www.w3.org/2000/01/rdf-schema# where the core vocabulary of RDF is defined

xsd: Is the prefix for the namespace - [http://www.w3.org/2001/XMLSchema#](http://www.w3.org/2001/XMLSchema) which is the XML Schema that defines the datatypes

Q560: Is the ID for the item “Helium” in Wikidata

Q11344: Is the ID for the item “chemical element” in Wikidata

Q298581: Is the ID for the item “Pierre Janssen” in Wikidata

Q90: Is the ID for the item “Paris” in Wikidata

Q142: Is the ID for the item “France” in Wikidata

P31: Is the ID for the property “instance of” in Wikidata

P274: Is the ID for the property “chemical formula” in Wikidata

P2102: Is the ID for the property “boiling point” in Wikidata

P2101: Is the ID for the property “melting point” in Wikidata

P2054: Is the ID for the property “density” in Wikidata

P61: Is the ID for the property “discoverer/inventor” in Wikidata

P19: Is the ID for the property “place of birth” in Wikidata

P625: Is the ID for the property “country” in Wikidata