

SPARQL Query Tasks Report

Introduction

The SPARQL Protocol and RDF Query Language (SPARQL) is a powerful and versatile query language used to retrieve and manipulate data stored in Resource Description Framework (RDF) format. It is a standard query language for databases, able to retrieve and manipulate data stored in RDF format, which is widely used in semantic web and linked data projects.

Wikidata is a collaboratively edited knowledge base hosted by the Wikimedia Foundation. It serves as a common source of structured data for Wikimedia projects such as Wikipedia, and for everyone on the web. The data stored in Wikidata is accessible through a variety of interfaces, including a SPARQL endpoint, which allows users to query the data using the SPARQL query language.

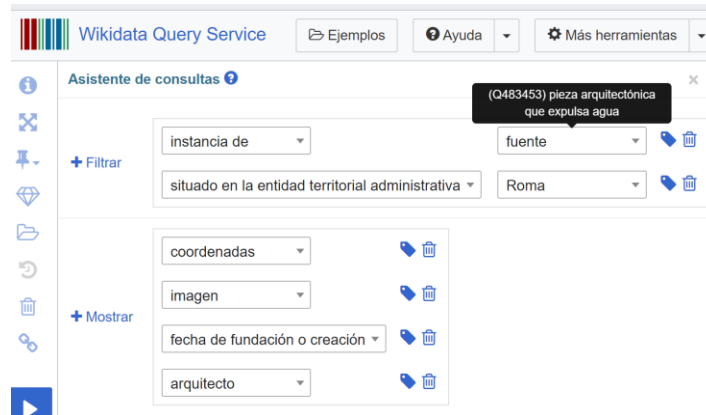
This report aims to construct and execute SPARQL queries to extract specific information from the Wikidata knowledge graph and present the results in a clear and informative manner.

Task 1: Fountains in Rome

Step-by-Step Query Construction

To construct the SPARQL query for listing the names of fountains in Rome documented on Wikidata, along with their geographical coordinates and images, the following steps were followed:

1. **Select Variables:** The query needs to select the fountain's name, coordinates, and image.
 - `?fountain`: Represents the fountain entity.
 - `?fountainLabel`: Represents the name of the fountain.
 - `?coordinates`: Represents the geographical coordinates of the fountain.
 - `?image`: Represents the image of the fountain.
2. **Define the Entity Type and Location:** The query needs to filter entities that are fountains located in Rome. The IDs can be found by looking for the entities in Wikidata or with the query assistant.



- Using wd:Q483453: The Wikidata entity for a fountain.
- Using wd:Q220: The Wikidata entity for Rome.

3. Include Language Filter: The names should be in Italian.

- Adding a language filter for ?fountainLabel to ensure names are in Italian.

4. Construct the Query:

```
SELECT ?fountain ?fountainLabel ?coordinates ?image WHERE {
  ?fountain wdt:P31 wd:Q483453.
  ?fountain wdt:P131 wd:Q220.
  ?fountain wdt:P625 ?coordinates.
  OPTIONAL { ?fountain wdt:P18 ?image. }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "it". }
}
```

Parts of the query:

1. **SELECT ?fountain ?fountainLabel ?coordinates ?image**: Specifies the variables to be returned in the results.
2. **WHERE { ... }**: Defines the pattern to match in the graph.
 - **?fountain wdt:P31 wd:Q483453**: Matches entities that are instances of the class "fountain" (Q483453).
 - **?fountain wdt:P131 wd:Q220**: Matches entities that are located in Rome (Q220).
 - **?fountain wdt:P625 ?coordinates**: Matches the geographical coordinates of the fountain.
3. **OPTIONAL { ?fountain wdt:P18 ?image }**: Optionally matches the image of the fountain, if available.
4. **SERVICE wikibase:label { bd:serviceParam wikibase:language "it" }**: Ensures that labels are returned in Italian.

Results

The query returned a list of 42 fountains in Rome in 1571 ms with their names in Italian, geographical coordinates, and images (where available). Below are some examples from the query results:

Table

42 resultados en 1571 ms

Código

Descargar

Enlace

fountain	fountainLabel	coordinates	image
wd:Q267971	Fontana del Tritone	Point(12.48833333 41.90361111)	commons:Fontana del Tritone,Rome.jpg
wd:Q655470	fontana del Quattro Fiumi	Point(12.473075 41.898953)	commons:PiazzaNavonaPanoramicaVerticale.jpg
wd:Q936296	Fontana delle Tartarughe	Point(12.4775 41.89380555)	commons:The Turtle Fountain.jpg
wd:Q1114231	fontana della Barcaccia	Point(12.482212 41.905806)	commons:Barcaccia e scalinata.jpg
wd:Q1217302	Quattro Fontane	Point(12.490736 41.901962)	commons:Piazza Quattro Fontane 270deg Piano.jpg
wd:Q3747369	Fontana del porto di Ripetta	Point(12.475139 41.904811)	commons:Campo Marzio - Ripetta fontana 1200060.JPG
wd:Q3747400	Fontana delle Cariatidi	Point(12.46446991 41.90937042)	commons:Fuente de las Cariatides, Roma, Italia, 2022-09-16, DD 01.jpg
wd:Q3747420	Fontana di Papa Giulio III	Point(12.473308587 41.918315769)	commons:Fontana di palazzo Borromeo (Rome).jpg
wd:Q3747421	Fontana di Ponte Sisto	Point(12.4697976 41.8916861)	commons:Roma fontana di ponte Sisto.jpg

Image grid

42 resultados en 1571 ms

Código

Descargar

Enlace

commons:Fontana ...

Point(12.48833333 41.90361111)

Q Fontana del Tritone

commons:Piazz...

Point(12.473075 41.898953)

Q Fontana del Quattro Fiumi

commons:The Turtle Fountain.jpg

Point(12.4775 41.89380555)

Q Fontana delle Tartarughe

commons:Barcaccia e scalinata.jpg

Point(12.482212 41.905806)

Q fontana della Barcaccia

commons:Piazza Quattro Fontane 270deg Piano.jpg

Point(12.490736 41.901962)

Q Quattro Fontane

commons:Campo Marzio - Ripetta fontana 1200060.JPG

Point(12.475139 41.904811)

Q Fontana del porto di Ripetta

commons:Fuente de las Cariatides, Roma, Italia, 2022-09-16, DD 01.jpg

Point(12.46446991 41.90937042)

Q Fontana delle Cariatidi

commons:Fontana di palazzo Borromeo ...

Point(12.473308587 41.918315769)

Q Fontana di Papa Giulio III

commons:Roma fontana di ponte S...

Point(12.4697976 41.8916861)

Q Fontana di Ponte Sisto

commons:Roma fontana di ponte S...

Point(12.4697976 41.8916861)

Q Fontana di Ponte Sisto

The geographical coordinates obtained from the query were plotted on a map of Rome using the Map View.

Map

42 resultados en 1571 ms

Código

Descargar

Enlace

Additional Experiments

To enhance the analysis, additional experiments were conducted:

- Extracting more details about each fountain, such as the year of construction and the architect.

Wikidata Query Service

Ejemplos
Ayuda
Más herramientas
Constructor de consultas
español

```

1 SELECT ?fountain ?fountainLabel ?coordinates ?image ?yearOfConstruction ?architectLabel WHERE {
2   ?fountain wdt:P91 wd:Q489453.           # Fountain
3   ?fountain wdt:P131 wd:Q220.             # Located in Rome
4   ?fountain wdt:P625 ?coordinates.        # Coordinates
5   OPTIONAL { ?fountain wdt:P18 ?image. }   # Image
6   OPTIONAL { ?fountain wdt:P571 ?yearOfConstruction. } # Year of construction
7   OPTIONAL { ?fountain wdt:P84 ?architect. } # Architect
8   SERVICE wikibase:label {
9     bd:serviceParam wikibase:language "it".
10    ?fountain rdfs:label ?fountainLabel.
11    ?architect rdfs:label ?architectLabel.
12  }
13 }
14

```

Table

42 resultados en 642 ms

Código Descargar Enlace

fountain	fountainLabel	coordinates	image	yearOfConstruction	architectLabel
Qd:Q267971	Fontana del Tritone	Point(12.488333333 41.903611111)	commons:Fontana del Tritone,Rome.jpg		
Qd:Q655470	fontana del Quattro Fiumi	Point(12.473075 41.898953)	commons:PiazzaNavonaPanoramicaVerticale.jpg	1 de enero de 1640	Gian Lorenzo Bernini
Qd:Q1114231	fontana della Barcaccia	Point(12.482212 41.905806)	commons:Barcaccia e scalinata.jpg	1 de enero de 1629	Pietro Bernini
Qd:Q1217302	Quattro Fontane	Point(12.490736 41.901962)	commons:Piazza Quattro Fontane 270deg Pano.jpg		
Qd:Q936296	Fontana delle Tartarughe	Point(12.4775 41.893805555)	commons:The Turtle Fountain.jpg		Giacomo Della Porta

- **OPTIONAL { ?fountain wdt:P571 ?yearOfConstruction. }**: This part of the query retrieves the year of construction of the fountain if the information is available.
- **wdt:P571**: Property for "inception" or year of construction.
- **OPTIONAL { ?fountain wdt:P84 ?architect. }**: This part of the query retrieves the architect of the fountain if the information is available.
- **wdt:P84**: Property for "architect".
- **?architectLabel**: The label (name) of the architect in Italian.

Conclusion

The SPARQL query successfully retrieved detailed information about fountains in Rome, including their names in Italian, geographical coordinates, and images. The results were visualized on a map to provide a clear representation of their locations. Additional experiments provided further insights about who designed them and when.

This report demonstrates the power of SPARQL in extracting and analyzing structured data from the Wikidata knowledge graph, showcasing its potential for various applications in data analysis and research.

Task 2: Australian Archbishops

Step-by-Step Query Construction

To construct the SPARQL query for listing the names of Australian archbishops documented in DBpedia, sorted by their birth date, the following steps were followed:

1. **Select Variables:** The query needs to select the archbishop's name and birth date.
 - ?archbishop: Represents the archbishop entity.
 - ?name: Represents the name of the archbishop.
 - ?birthDate: Represents the birth date of the archbishop.
2. **Define the Entity Type and Nationality:** The query needs to filter entities that are archbishops and are Australian.
 - Using dbo:Archbishop: The DBpedia ontology class for archbishops.
 - Using dbo:birthPlace: The property to filter Australian nationality.
 - Using dbr:Australia: The DBpedia entity for Australia.
3. **Sort the Results:** The query results should be sorted by birth date from the most modern to the oldest.
4. **Construct the Query:**

```
SELECT ?archbishop ?name ?birthDate WHERE {  
  ?archbishop a dbo:Archbishop .  
  ?archbishop dbo:birthPlace dbr:Australia .  
  ?archbishop foaf:name ?name .  
  ?archbishop dbo:birthDate ?birthDate .  
} ORDER BY DESC(?birthDate)
```

Parts of the query:

1. **SELECT ?archbishop ?name ?birthDate:** Specifies the variables to be returned in the results.
2. **WHERE { ... }:** Defines the pattern to match in the graph.
 - **?archbishop a dbo:Archbishop:** Matches entities that are instances of the class "Archbishop".
 - **?archbishop dbo:birthPlace dbr:Australia:** Matches entities whose birthplace is Australia.
 - **?archbishop foaf:name ?name:** Matches the name of the archbishop.
 - **?archbishop dbo:birthDate ?birthDate:** Matches the birth date of the archbishop.
3. **ORDER BY DESC(?birthDate):** Sorts the results by birth date in descending order, from the most modern to the oldest.

Running the Query

The query was executed using the DBpedia SPARQL endpoint at <https://dbpedia.org/sparql>. The endpoint provides various output formats including Table and CSV, which are useful for analyzing the data.

Results

The query returned a list of Australian archbishops with their names in English and birth dates, sorted from the most modern to the oldest. Below are some examples from the query results in HTML, Turtle and CSV format. This provides a lot of flexibility for its integration in different tools.

SPARQL | HTML5 table

archbishop	name	birthDate
http://dbpedia.org/resource/Peter_Comensoli	"Peter Andrew Comensoli"@en	1964-03-25
http://dbpedia.org/resource/Philip_Freier	"Philip Freier"@en	1955-02-09
http://dbpedia.org/resource/Peter_Jensen_(bishop)	"Peter Jensen"@en	1943-07-11
http://dbpedia.org/resource/Denis_Hart	"Denis James Hart"@en	1941-05-16
http://dbpedia.org/resource/John_Bathersby	"John Bathersby"@en	1936-07-26
http://dbpedia.org/resource/John_Vockler	"John Vockler"@en	1924-07-22

```
@prefix res: <http://www.w3.org/2005/sparql-results#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
_:a a res:ResultSet .
_:i res:resultVariable "archbishop" , "name" , "birthDate" .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
_:i res:solution [
  res:binding [ res:variable "archbishop" ; res:value dbr:Peter_Comensoli ] ;
  res:binding [ res:variable "name" ; res:value "Peter Andrew Comensoli"@en ] ;
  res:binding [ res:variable "birthDate" ; res:value "1964-03-25"^^xsd:date ] ] .
_:i res:solution [
  res:binding [ res:variable "archbishop" ; res:value dbr:Philip_Freier ] ;
  res:binding [ res:variable "name" ; res:value "Philip Freier"@en ] ;
  res:binding [ res:variable "birthDate" ; res:value "1955-02-09"^^xsd:date ] ] .
_:i res:solution [
  res:binding [ res:variable "archbishop" ; res:value <http://dbpedia.org/resource/Peter_Jensen_(bishop)> ] ;
  res:binding [ res:variable "name" ; res:value "Peter Jensen"@en ] ;
  res:binding [ res:variable "birthDate" ; res:value "1943-07-11"^^xsd:date ] ] .
_:i res:solution [
  res:binding [ res:variable "archbishop" ; res:value dbr:Denis_Hart ] ;
  res:binding [ res:variable "name" ; res:value "Denis James Hart"@en ] ;
  res:binding [ res:variable "birthDate" ; res:value "1941-05-16"^^xsd:date ] ] .
_:i res:solution [
  res:binding [ res:variable "archbishop" ; res:value dbr:John_Bathersby ] ;
  res:binding [ res:variable "name" ; res:value "John Bathersby"@en ] ;
  res:binding [ res:variable "birthDate" ; res:value "1936-07-26"^^xsd:date ] ] .
_:i res:solution [
  res:binding [ res:variable "archbishop" ; res:value dbr:John_Vockler ] ;
  res:binding [ res:variable "name" ; res:value "John Vockler"@en ] ;
  res:binding [ res:variable "birthDate" ; res:value "1924-07-22"^^xsd:date ] ] .
```

	A	B	C
1	archbishop	name	birthDate
2	http://dbpedia.org/resource/Peter_Comensoli	Peter Andrew Comensoli	25/03/1964
3	http://dbpedia.org/resource/Philip_Freier	Philip Freier	09/02/1955
4	http://dbpedia.org/resource/Peter_Jensen_(bishop)	Peter Jensen	11/07/1943
5	http://dbpedia.org/resource/Denis_Hart	Denis James Hart	16/05/1941
6	http://dbpedia.org/resource/John_Bathersby	John Bathersby	26/07/1936
7	http://dbpedia.org/resource/John_Vockler	John Vockler	22/07/1924

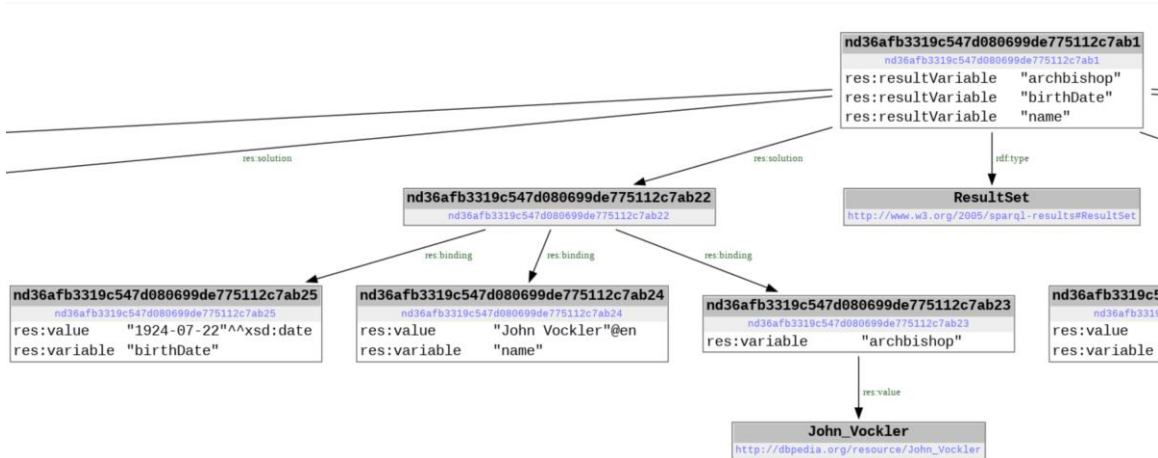
Using RDFLib, we can even represent the Knowledge Graph


```

[4] 1 g = Graph()
2 # create graph using turtle
3 turtleData = ""
4 # SPARQL query
5 # SPARQL query
6 # SPARQL query
7 # SPARQL query
8 # SPARQL query
9 # SPARQL query
10 # SPARQL query
11 # SPARQL query
12 # SPARQL query
13 # SPARQL query
14 # SPARQL query
15 # SPARQL query
16 # SPARQL query
17 # SPARQL query
18 # SPARQL query
19 # SPARQL query
20 # SPARQL query
21 # SPARQL query
22 # SPARQL query
23 # SPARQL query
24 # SPARQL query
25 # SPARQL query
26 # SPARQL query
27 # SPARQL query
28 # SPARQL query
29 # SPARQL query
30 # SPARQL query
31 # SPARQL query
32 # SPARQL query
33 # SPARQL query
34 # SPARQL query
35 # SPARQL query
36 # SPARQL query
37 # SPARQL query
38 # SPARQL query
39 # SPARQL query
40 # SPARQL query
41 # SPARQL query
42 # SPARQL query
43 # SPARQL query
44 # SPARQL query
45 # SPARQL query
46 # SPARQL query
47 # SPARQL query
48 # SPARQL query
49 # SPARQL query
50 # SPARQL query
51 # SPARQL query
52 # SPARQL query
53 # SPARQL query
54 # SPARQL query
55 # SPARQL query
56 # SPARQL query
57 # SPARQL query
58 # SPARQL query
59 # SPARQL query
60 # SPARQL query
61 # SPARQL query
62 # SPARQL query
63 # SPARQL query
64 # SPARQL query
65 # SPARQL query
66 # SPARQL query
67 # SPARQL query
68 # SPARQL query
69 # SPARQL query
70 # SPARQL query
71 # SPARQL query
72 # SPARQL query
73 # SPARQL query
74 # SPARQL query
75 # SPARQL query
76 # SPARQL query
77 # SPARQL query
78 # SPARQL query
79 # SPARQL query
80 # SPARQL query
81 # SPARQL query
82 # SPARQL query
83 # SPARQL query
84 # SPARQL query
85 # SPARQL query
86 # SPARQL query
87 # SPARQL query
88 # SPARQL query
89 # SPARQL query
90 # SPARQL query
91 # SPARQL query
92 # SPARQL query
93 # SPARQL query
94 # SPARQL query
95 # SPARQL query
96 # SPARQL query
97 # SPARQL query
98 # SPARQL query
99 # SPARQL query
100 # SPARQL query

```

Zooming in, we can see the results of the query, for example, for John Vockler,



Conclusion for Task 2

The SPARQL query successfully retrieved detailed information about Australian archbishops, including their names in English and birth dates, sorted from the most modern to the oldest. This task demonstrates the capability of SPARQL and the DBpedia knowledge graph in extracting and organizing specific data, highlighting its utility for historical and biographical research.