Author(s): Rafael Barros Date: November 5th 2024

SPARQL - Introduction: Origins, Challenges and Applications

Covering points of this presentation:

- Origin: Contex and limitations of Traditional Relational Databases;
- Early Challenges with RDF Data;
- Sparql: Overview, Purpose and Basic Syntax;
- Its Application in LOD;

- Examples (Isaac Newton data from Wikipedia);
- Real World Applications;
- Limitations and Future Analysis for Improvements in Knowledge Graphs;
- References;

Contextual Analysis: SPARQL - deep diving into the query of semantic web

- **Data Access** and **Query Language** in order to handle large, complex and interconnected datasets;
- **Semantic Web Principle**: data should be able to get accessed across platforms and universally understood;
- **Data Silos and its Interoperability**: database tables with rows and columns makes difficult to understand and link the relationships between different datasets;

- **Relational Databases**: rely on restructure while adding new data type when the pre-structured or predefined schema is modified;
- **SQL Queries Limitations**: as the data are more interconnected, the traditional SQL queries struggle with optimized ways of querying and retrieving complex interlinked data and its relationships;
- **Limitation of Data Interpretation**: previous data systems lacked the ability to reason and interpret data meaningfully;
- **Sharing Data on Internet**: old databases weren't made for sharing data across internet;

Early Challenges in RDF Data

- **Missing standardized way to query RDF data**:
- ***Eminent problem***: Researches and developers faced limitations in querying RDF data across different sources, as each system often required unique approaches for data retrieval.
- **Isolated data:**

- ***Eminent problem***: data fragmented in differents systems and format, turning difficult to integrate and link. **Example**: a dataset about Cologne's landscape may be stored separately from the city's population, which makes hard to connect them.
- **SPARQL made LOD gain richer insights:**

- ***Problem:*** how to combine geographic data, population statistics, and historical records? SPARQL's ability to pull data from different RDF datasets turned data more accessible.

What is SPARQL? Overview and Purpose:

SPARQL (SPARQL Protocol and RDF Query Language) is a powerful language created by the **World Wide Web Consortium (W3C)** to query and work with **RDF (Resource Description Framework)** data format. As the standard query language for RDF, SPARQL allows users to easily retrieve and manipulate data stored in the form of triples—connections between entities, like “Cologne is in Germany.” SPARQL is especially important for **Linked Open Data** and **knowledge graphs** in the **Semantic Web**, enabling users to uncover relationships and insights across vast and interconnected datasets.

SPARQL Types:

1. **ASK**: check if at least one match exists and return boolean (TRUE or FALSE);

- ***Use Case***: Useful for checks in data validation;

```
- ASK WHERE {
 dbr:Cologne dbo:country dbr:Germany .
}
```

2. **CONSTRUCT**: Create new RDF triples from matches;
3. **DESCRIBE**: Create RDF triples about a resource;
4. **SELECT(Most used)**: Return matches as a collection of solution bindings;

Application in LOD

- ***Linked Open Data*** is a network of interlinked datasets from different sources acessible through RDF and SPARQL;
- ***SPARQL*** viable users to query datasets such as DBpedia, Wikidata etc.. That's the limitation of formers Relational Datasets Retriever;

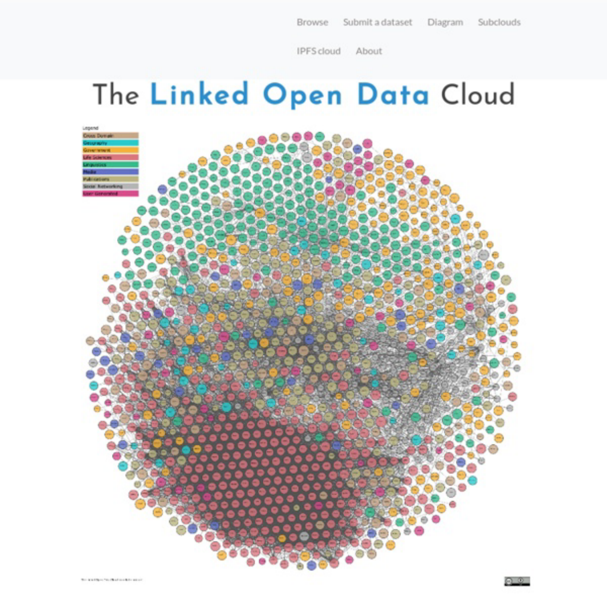


Figure 1: LOD Cloud

Syntax:

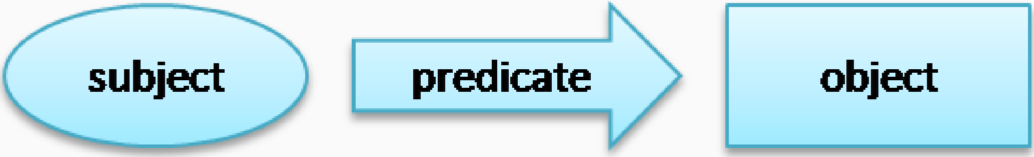
Similar to SQL, SPARQL queries involve selecting specific data from structured triples (subject, predicate, object).

Example of RDF Triples

Subject	Predicate	Object
`<John>`	`<likes>`	`<Pizza>`
`<City:Cologne>`	`<isInCountry>`	`<Country:Germany>`

- another example:

```
SELECT ?person
 WHERE {
   ?person rdf:type :Person .
   ?person :interest :SustainableEnergies .
   }
```



Code - Triple pattern query:

Basic SPARQL Query Structure

A basic SPARQL query contains the following components:

1. **SELECT** - Specifies which variables to return.
2. **WHERE** - Defines patterns to match against the RDF triples in the dataset.
3. **FILTER** - (Optional) Adds conditions to limit the results based on certain criteria.
4. **LIMIT** - That defines how many researched data should be retrieved.

SPARQL (Wikidata) Example:

Example SPARQL Query

The following highlighted SPARQL query retrieves all people who studied at Trinity College:

?person wdt:P69 wd:Q332342 .

wdt	wd
`<property>`	`<entity>`
`<educated at>`	`Trinity College`

```
SELECT ?person ?personLabel ?birth_placeLabel ?coodinates

WHERE{

  ?person wdt:P69 wd:Q332342 . # all those who studied at Trinity College

  ?person wdt:P106 wd:Q170790 . # ordered by occupation listed
  ?person wdt:P21 wd:Q6581097 . # male who studied at Trinity College
  ?person wdt:P19 ?birth_place . # place of birth , make sure the name giv
  ?birth_place wdt:P625 ?coodinates . # with Label after the name ensure t

   SERVICE wikibase:label {
       bd:serviceParam wikibase:language "en" . }
     #service provided by examples on wikidata query service

}
```

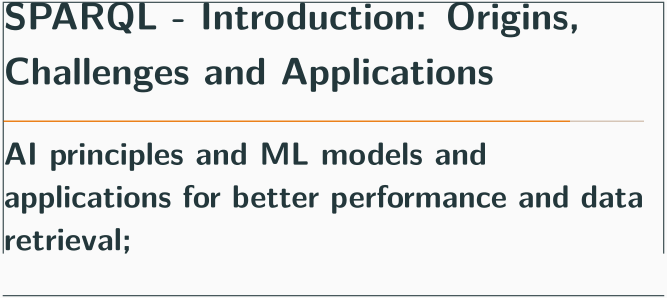
Real World Applications

- **Semantic Search**: search engines based on context search;
- **Knowledge Graphs**: query abilitiy on entities and its relationships in knowledge bases;
- **Data Integration**: Binding data from different sources; **Examples:** Google Knowledge Graph and Wikipedia (embedded to Wikidata);

Limitations of SPARQL:

- **Performance**: while querying large datasets, the general performance can get dropped or slowed;
- **Complexity**: complex queries require advanced knowledge in order to get the best use of it;
- **Interoperability**: many datasets are not available in RDF.

Future Directions:

- Enhanced query optimization for faster performance;
- Better tools for creating queries and visualize them;
- 

  ## SPARQL - Introduction: Origins, Challenges and Applications

  AI principles and ML models and applications for better performance and data retrieval;

- “In recent years, the field of **neural machine translation (NMT) for SPARQL query generation** has witnessed significant growth. Incorporating the copy mechanism with traditional encoder-decoder architectures and using pre-trained

References

Source_1: Using SPARQL and SPIN for Data Quality Management on the Semantic Web | SpringerLink

Source_2: A review of the semantic web field | Communications of the ACM

Source_3: SPARQL - An Absolute Beginner's Guide - DEV Community

Source_4: Comunica – SPARQL query types

Source_5: A Comprehensive Evaluation of Neural SPARQL Query Generation From Natural Language Questions | IEEE Journals & Magazine | IEEE Xplore

Thank you for attending to my presentation