Knowledge Graph LAB Assignment Implement SDM UPC

Linhan Wang [*], Min Zhang[*]

{linhan.wang,min.zhang}@estudiantat.upc.edu

In this project we will be using GraphDB to create and manage knowledge graphs, and query and process them. We will use innovative methods to solve these problems. The problem list is divided into four parts: TBOX deninition, ABOX deninition, Create the final ontology, Querying the ontology.

1 B.1 Ontology creation

We used the same datasets(Scorpus) in the first lab and built our TBOX using grafo. And the visual representation of the TBOX can be seen in Figure 1, which was drawn by grafo.[1]

In the graph we defined superclasses and subclasses, which can demonstrate the ontology clearly. And in order to define the hierarchy, we made an assumption that reviewers are authors, so they can share some common properties with authors. So we defined superclasses as follows: Person is the superclass of Author, while Author is the superclass of Reviewer. Proceeding is the superclass of Proceeding_conference and Proceeding_workshop.

In this TBOX definition, there are three kinds of owl: owl:Class, owl:DatatypeProperty, owl:ObjectProperty. For each type, details are shown in Table 1.

owl:ObjectProperty	cites collection_of_conference collection_of_workshop has_keyword paper_belong_to_journal published_in_proceeding reviewed_by written_by
owl:DatatypeProperty	author_id author_name doi journal_name keyword title ......
owl:Class	Author Conference Journal Keywords Paper ......

Table 1: TBOX Definition

2 B.2 ABOX deninition(programmatically)

We built the ABOX using Python's library RDFLib, and the ABOX was build in 2 steps: (1) Converting each row in csv file into RDF triples(Subject, predicate and object), (2) Serialize and output the ABOX into file.

The triples for each property and class are shown in Table 2.

When each csv is transformed into RDF triples, ABOX is constructed. We demonstrated this process by the example of Authors data conversion.

[*] Universitat Politècnica de Catalunya (UPC), Erasmus Mundus Big Data Management and Analytics (BDMA)

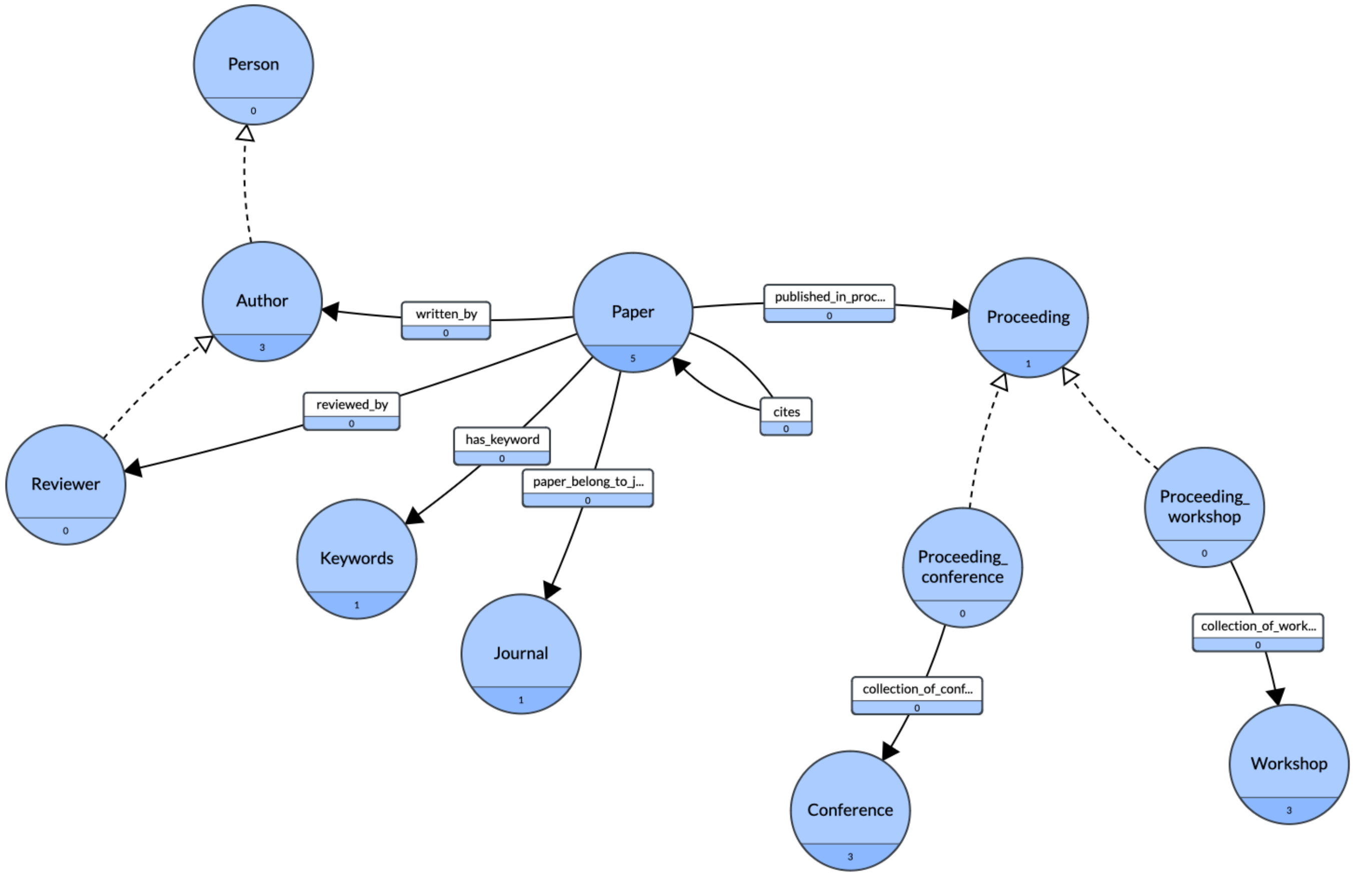


Figure 1: Comprehensive view of the Knowledge Graphs.

semantic	subject	predicate	object
cites	Paper	cites	Paper
has_keyword	Paper	has_keyword	Keyword
reviewed_by	Paper	reviewed_by	Reviewer
written_by	Paper	written_by	Author
......	......	......	......
author_id	Author	author_id_type	author_id
doi	Paper	doi_type	doi
keyword	Keyword	keyword_type	keyword
title	Paper	title_type	title
......	......	......	......
Author	Author	RDF.type	author_type
Journal	Journal	RDF.type	journal_type
Paper	Paper	RDF.type	paper_type
......	......	......	......

Table 2: RDF triples

2.1 CSV to RDF Conversion Code for Authors

We used rdflib[2] to implement this conversion. The following Python code snippet reads data from a CSV file (`authors.csv`) and converts it into RDF triples, which are then added to an RDF graph. Each author in the CSV file is represented as a resource with properties such as type, name, ID, and affiliation.

```python
# authors.csv
csv_data = []
with open("data/authors.csv", newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        csv_data.append(row)

for row in csv_data:
    # author_type
    author_uri = URIRef(EX[row["author_id"]])
    author_type = URIRef("http://www.gra.fo/schema/untitled-ekg#Author")
    abox_graph.add((author_uri, RDF.type, author_type))

    # author_name
    author_name = URIRef(urllib.parse.quote(row["author_name"]))
    author_name_type = URIRef("http://www.gra.fo/schema/untitled-ekg#author_name")
    abox_graph.add((author_uri, author_name_type, author_name))

    # author_id
    author_id = URIRef(row["author_id"])
    author_id_type = URIRef("http://www.gra.fo/schema/untitled-ekg#author_id")
    abox_graph.add((author_uri, author_id_type, author_id))

    # author_affiliation
    author_aff = URIRef(urllib.parse.quote(row["author_affiliation"]))
    author_aff_type = URIRef("http://www.gra.fo/schema/untitled-ekg#author_affiliation")
    abox_graph.add((author_uri, author_aff_type, author_aff))
```

Listing 1: CSV to RDF Conversion Code for Authors

The code performs the following steps:

1. **Reading the CSV File**: Opens the `authors.csv` file and reads its contents using a dictionary reader.
2. **Processing Each Row**: For each row, it creates RDF triples for the author's type, name, ID, and affiliation.
3. **Defining the Author Resource**: Converts the author ID into a URI and adds a triple to specify that it is of type `Author`.
4. **Adding the Author Name**: Encodes the author name as a URI and links it to the author resource.
5. **Adding the Author ID**: Directly links the author ID to the author resource.
6. **Adding the Author Affiliation**: Encodes the author affiliation as a URI and links it to the author resource.
7. **Adding RDF Triples to the Graph**: Add each triple to the RDF graph.

2.2 ABOX Step 2 - Output ABOX

After converting all the csv files into rdf, using serialize function to output ABOX in turtle format.

3 B.3 Create the final ontology

After generating the TBOX and ABOX, we created the GraphDB repository and imported both files into it. When importing .owl file, TBOX is imported; when importing .ttl file, ABOX is imported.

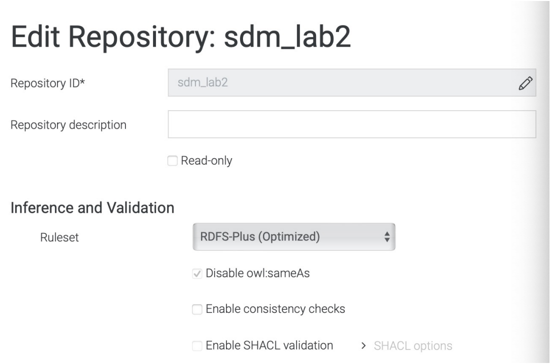
Label	**Count**
Number of Classes	20
Number of Main Properties	23
Number of Instances of Main Classes	2719
Number of Triples	31011

Table 3: Summary of Knowledge Graph Statistics

Class	**Instances**	**Triples**
Paper	483	19896
Author	1942	10819
Conference	178	178
Journal	116	118

Table 4: Statistics for individual main Classes

We configured the ruleset of our repository to RDFS (Optimized), as shown in Figure 2. According to GraphDB, this ruleset supports the standard model-theoretic semantics of RDFS.



Edit Repository: sdm_lab2

Repository ID* sdm_lab2

Repository description

□ Read-only

Inference and Validation

Ruleset RDFS-Plus (Optimized)

☑ Disable owl:sameAs

□ Enable consistency checks

□ Enable SHACL validation

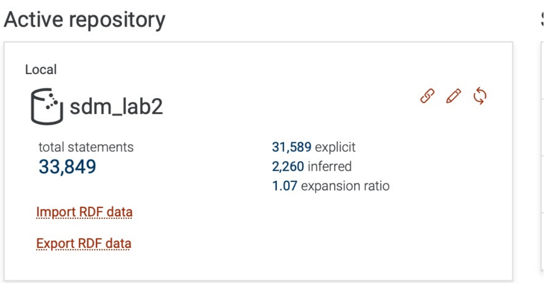
Figure 2: Repository-Setting

From Table3 and Table4, we can see the statistics information after of GraphDB.

From Figure 3 and Figure 4, we can see the overview of repository and graph in GraphDB. And from Figure 5, we can see the hierarchy of the GraphDB.

4 B.4 Querying

In this part, we listed all the 6 queries, as well as the screenshot of query results. Moreover, we also downloaded all the query results as CSV files in the query_results folder attached to our submission.



Active repository

Local

sdm_lab2

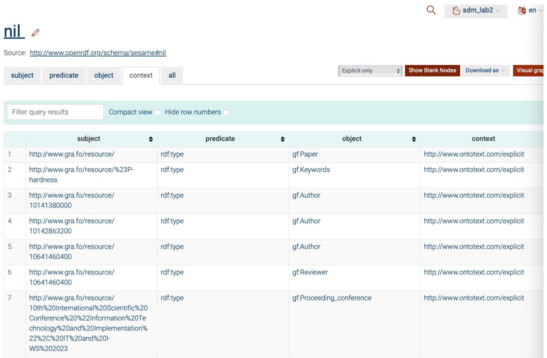
total statements
33,849

31,589 explicit
2,260 inferred
1.07 expansion ratio

Import RDF data

Export RDF data

Figure 3: Repository Overview



nil

Source: http://www.openrdf.org/schema/sesame#nil

	subject	predicate	object	context
1	http://www.gra.fo/resource/	rdf:type	gf:Paper	http://www.ontotext.com/explicit
2	http://www.gra.fo/resource/%23P-hardness	rdf:type	gf:Keywords	http://www.ontotext.com/explicit
3	http://www.gra.fo/resource/10141380000	rdf:type	gf:Author	http://www.ontotext.com/explicit
4	http://www.gra.fo/resource/10142863200	rdf:type	gf:Author	http://www.ontotext.com/explicit
5	http://www.gra.fo/resource/10641460400	rdf:type	gf:Author	http://www.ontotext.com/explicit
6	http://www.gra.fo/resource/10641460400	rdf:type	gf:Reviewer	http://www.ontotext.com/explicit
7	http://www.gra.fo/resource/10th%20International%20Scientific%20Conference%20%22Information%20Technology%20and%20Implementation%22%2C%20IT%20and%20I-WS%202023	rdf:type	gf:Proceeding_conference	http://www.ontotext.com/explicit

Figure 4: Graph Overview

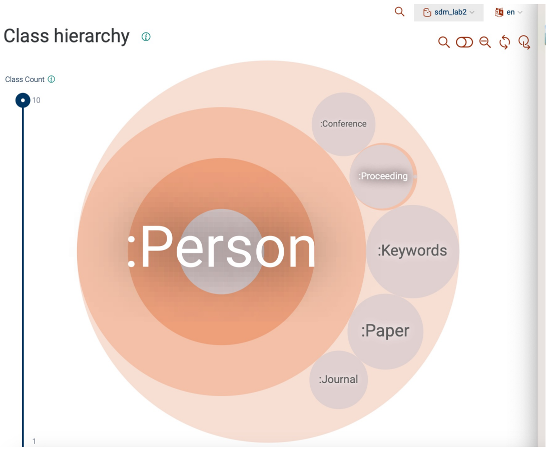


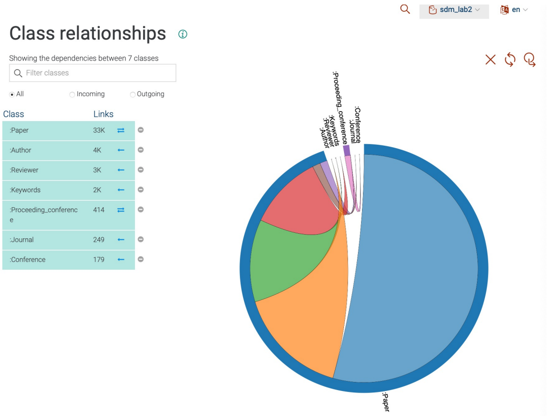
Figure 5: Class hierarchy

4.1 1. Find all Authors.

The following Cypher query finds the top 3 most cited papers of each conference:

```
PREFIX gf: <http://www.gra.fo/schema/untitled-ekg#>
PREFIX ex: <http://www.gra.fo/resource/>

SELECT DISTINCT ?author ?authorName
WHERE {
  ?author a gf:Author .
  ?author gf:author_name ?authorName .
}
```



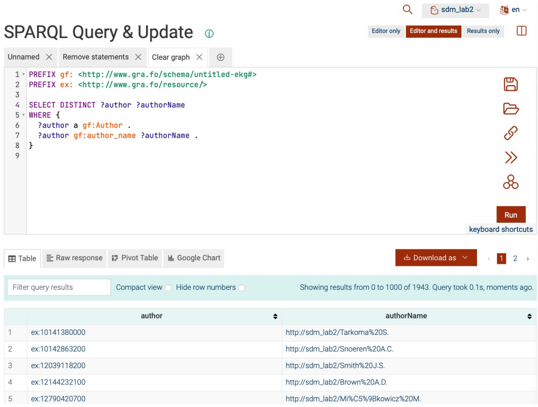
Class relationships

Showing the dependencies between 7 classes

Class	Links	
:Paper	33K	⇄
:Author	4K	←
:Reviewer	3K	←
:Keywords	2K	←
:Proceeding_conference	414	⇄
:Journal	249	←
:Conference	179	←

Figure 6: Class relationship

Listing 2: SPARQL Query to Select Distinct Authors and Their Names



SPARQL Query & Update

```
PREFIX gf: <http://www.gra.fo/schema/untitled-ekg#>
PREFIX ex: <http://www.gra.fo/resource/>

SELECT DISTINCT ?author ?authorName
WHERE {
  ?author a gf:Author .
  ?author gf:author_name ?authorName .
}
```

Showing results from 0 to 1000 of 1943. Query took 0.1s, moments ago.

	author	authorName
1	ex:10141380000	http://sdm_lab2/Tarkoma%20S.
2	ex:10142863200	http://sdm_lab2/Snoeren%20A.C.
3	ex:12039118200	http://sdm_lab2/Smith%20J.S.
4	ex:12144232100	http://sdm_lab2/Brown%20A.D.
5	ex:12790420700	http://sdm_lab2/Mi%C5%9Bkowicz%20M.

Figure 7: Result of Finding all Authors

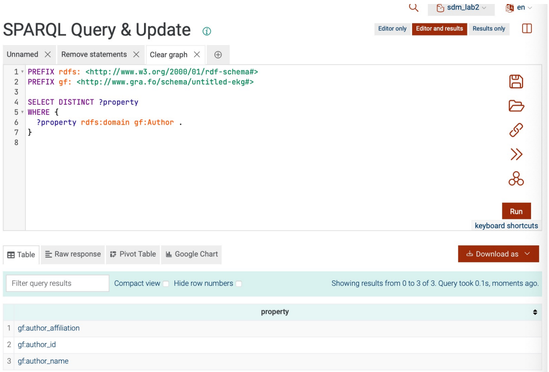
4.2 2. Find all properties whose domain is Author.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX gf: <http://www.gra.fo/schema/untitled-ekg#>

SELECT DISTINCT ?property
WHERE {
  ?property rdfs:domain gf:Author .
}
```

Listing 3: SPARQL Query to Select Distinct Properties of Authors

4.3 3. Find all properties whose domain is either Conference or Journal.



```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX gf: <http://www.gra.fo/schema/untitled-ekg#>

SELECT DISTINCT ?property
WHERE {
  ?property rdfs:domain gf:Author .
}

```

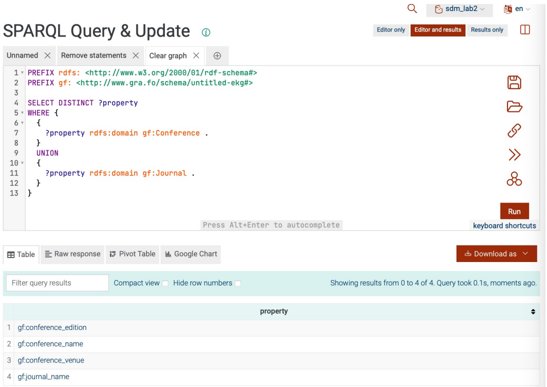
Showing results from 0 to 3 of 3. Query took 0.1s, moments ago.

	property
1	gf:author_affiliation
2	gf:author_id
3	gf:author_name

Figure 8: Result of Finding all properties whose domain is Author

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX gf: <http://www.gra.fo/schema/untitled-ekg#>

SELECT DISTINCT ?property
WHERE {
  {
    ?property rdfs:domain gf:Conference .
  }
  UNION
  {
    ?property rdfs:domain gf:Journal .
  }
}
```



```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX gf: <http://www.gra.fo/schema/untitled-ekg#>

SELECT DISTINCT ?property
WHERE {
  {
    ?property rdfs:domain gf:Conference .
  }
  UNION
  {
    ?property rdfs:domain gf:Journal .
  }
}
```

Showing results from 0 to 4 of 4. Query took 0.1s, moments ago.

	property
1	gf:conference_edition
2	gf:conference_name
3	gf:conference_venue
4	gf:journal_name

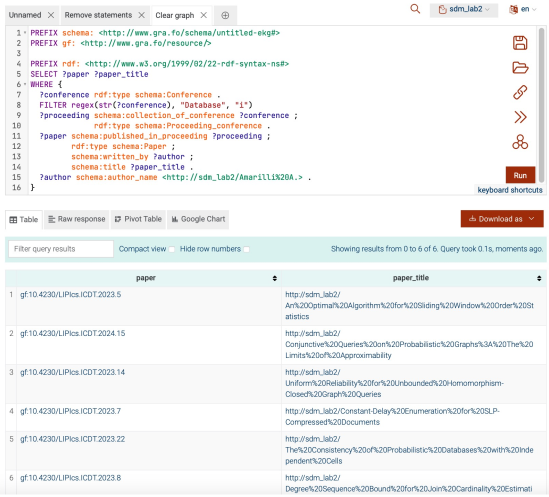
Figure 9: Result of Finding all properties whose domain is either Conference or Journal.

4.4 4. Find all the papers written by a given author - Amarilli 20A that where published in database conferences.

For this query, our query design is as follows: Firstly, we identify all papers in the database field by searching for all papers from conferences with the keyword ”database”. Secondly, regarding the specified author condition, we identify authors from the previous step who have published more than two papers, and then sort these authors by the number of papers published, identifying the author with the highest number of publications. The third step is to find all papers published by the specified author (i.e., the author with the most publications in the database field) at database conferences.

```
PREFIX schema: <http://www.gra.fo/schema/untitled-
    ekg#>
PREFIX gf: <http://www.gra.fo/resource/>

SELECT ?paper ?paper_title
WHERE {
  ?conference rdf:type schema:Conference .
  FILTER regex(str(?conference), "Database", "i")
  ?proceeding schema:collection_of_conference ?
    conference ;
              rdf:type schema:Proceeding_conference
    .
  ?paper schema:published_in_proceeding ?proceeding
    ;
         rdf:type schema:Paper ;
         schema:written_by ?author ;
         schema:title ?paper_title .
  ?author schema:author_name <https://gra.fo/
    Amarilli%20A.> .
}
```



```
PREFIX schema: <http://www.gra.fo/schema/untitled-ekg#>
PREFIX gf: <http://www.gra.fo/resource/>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?paper ?paper_title
WHERE {
  ?conference rdf:type schema:Conference .
  FILTER regex(str(?conference), "Database", "i")
  ?proceeding schema:collection_of_conference ?conference ;
              rdf:type schema:Proceeding_conference .
  ?paper schema:published_in_proceeding ?proceeding ;
         rdf:type schema:Paper ;
         schema:written_by ?author ;
         schema:title ?paper_title .
  ?author schema:author_name <http://sdm_lab2/Amarilli%20A.> .
}
```

Showing results from 0 to 6 of 6. Query took 0.1s, moments ago.

	paper	paper_title
1	gf:10.4230/LIPIcs.ICDT.2023.5	http://sdm_lab2/ An%20Optimal%20Algorithm%20for%20Sliding%20Window%20Order%20Statistics
2	gf:10.4230/LIPIcs.ICDT.2024.15	http://sdm_lab2/ Conjunctive%20Queries%20on%20Probabilistic%20Graphs%3A%20The%20Limits%20of%20Approximability
3	gf:10.4230/LIPIcs.ICDT.2023.14	http://sdm_lab2/ Uniform%20Reliability%20for%20Unbounded%20Homomorphism-Closed%20Graph%20Queries
4	gf:10.4230/LIPIcs.ICDT.2023.7	http://sdm_lab2/Constant-Delay%20Enumeration%20for%20SLP-Compressed%20Documents
5	gf:10.4230/LIPIcs.ICDT.2023.22	http://sdm_lab2/ The%20Consistency%20of%20Probabilistic%20Databases%20with%20Independent%20Cells
6	gf:10.4230/LIPIcs.ICDT.2023.8	http://sdm_lab2/ Degree%20Sequence%20Bound%20for%20Join%20Cardinality%20Estimati

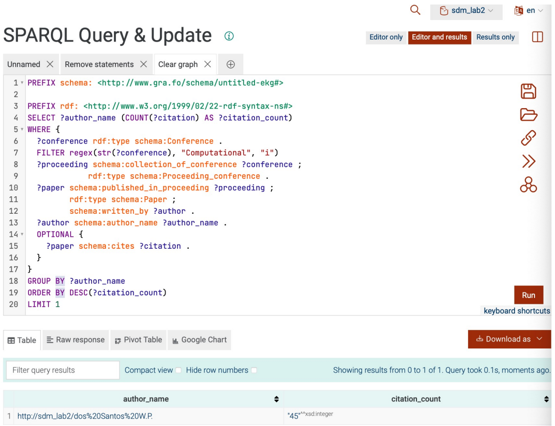
Figure 10: Result of Finding all the papers written by author Amarilli 20A. that where published in database conferences.

4.5 5. Find the author that having the most citations in Computational conferences.

This query is designed to identify the author with the highest number of citations among those who have presented papers at conferences related to ”Computational.” The query first filters out conferences whose names contain ”Computational,” then finds the proceedings of these conferences and the papers published in them. From these papers, it extracts the authors and their names, counts the citations for each author, sorts the results in descending order of citation count, and finally returns the author with the most citations.

```
PREFIX schema: <http://www.gra.fo/schema/untitled-ekg#>

SELECT ?author_name (COUNT(?citation) AS ?citation_count)
WHERE {
  ?conference rdf:type schema:Conference .
  FILTER regex(str(?conference), "Computational", "i")
  ?proceeding schema:collection_of_conference ?conference ;
              rdf:type schema:Proceeding_conference .
  ?paper schema:published_in_proceeding ?proceeding ;
         rdf:type schema:Paper ;
         schema:written_by ?author .
  ?author schema:author_name ?author_name .
  OPTIONAL {
    ?paper schema:cites ?citation .
  }
}
GROUP BY ?author_name
ORDER BY DESC(?citation_count)
LIMIT 1
```



```
PREFIX schema: <http://www.gra.fo/schema/untitled-ekg#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?author_name (COUNT(?citation) AS ?citation_count)
WHERE {
  ?conference rdf:type schema:Conference .
  FILTER regex(str(?conference), "Computational", "i")
  ?proceeding schema:collection_of_conference ?conference ;
             rdf:type schema:Proceeding_conference .
  ?paper schema:published_in_proceeding ?proceeding ;
         rdf:type schema:Paper ;
         schema:written_by ?author .
  ?author schema:author_name ?author_name .
  OPTIONAL {
    ?paper schema:cites ?citation .
  }
}
GROUP BY ?author_name
ORDER BY DESC(?citation_count)
LIMIT 1
```

Showing results from 0 to 1 of 1. Query took 0.1s, moments ago.

	author_name	citation_count
1	http://sdm_lab2/dos%20Santos%20W.P.	"45"^^xsd:integer

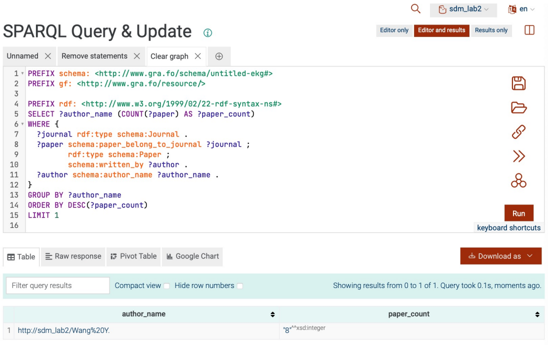
Figure 11: Result of Finding the author that having the most citations in Computational conferences.

4.6 6. Find the author that published most papers in journals.

This query is to identify the author who has published the most papers in journals. By matching journals and their related papers, the query retrieves each author's name and counts the number of papers they have published, ultimately returning the author with the highest paper count.

```
PREFIX schema: <http://www.gra.fo/schema/untitled-ekg#>
PREFIX gf: <http://www.gra.fo/resource/>

SELECT ?author_name (COUNT(?paper) AS ?paper_count)
WHERE {
  ?journal rdf:type schema:Journal .
  ?paper schema:paper_belong_to_journal ?journal ;
         rdf:type schema:Paper ;
         schema:written_by ?author .
  ?author schema:author_name ?author_name .
}
GROUP BY ?author_name
ORDER BY DESC(?paper_count)
LIMIT 1
```



```
PREFIX schema: <http://www.gra.fo/schema/untitled-ekg#>
PREFIX gf: <http://www.gra.fo/resource/>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?author_name (COUNT(?paper) AS ?paper_count)
WHERE {
  ?journal rdf:type schema:Journal .
  ?paper schema:paper_belong_to_journal ?journal ;
         rdf:type schema:Paper ;
         schema:written_by ?author .
  ?author schema:author_name ?author_name .
}
GROUP BY ?author_name
ORDER BY DESC(?paper_count)
LIMIT 1

```

Showing results from 0 to 1 of 1. Query took 0.1s, moments ago.

	author_name	paper_count
1	http://sdm_lab2/Wang%20Y.	"8"^^xsd:integer

Figure 12: Result of Finding the author that published most papers in journals.

References

[1] grafo. Design Knowledge Graphs, Simply, 2024.

[2] rdflib. rdflib 7.0.0 Documentation, 2024.