Universitat Politècnica de Catalunya

Facultad de Informática de Barcelona

Lab Assignment 2: Knowledge Graph

Semantic Data Management

Spring 2024

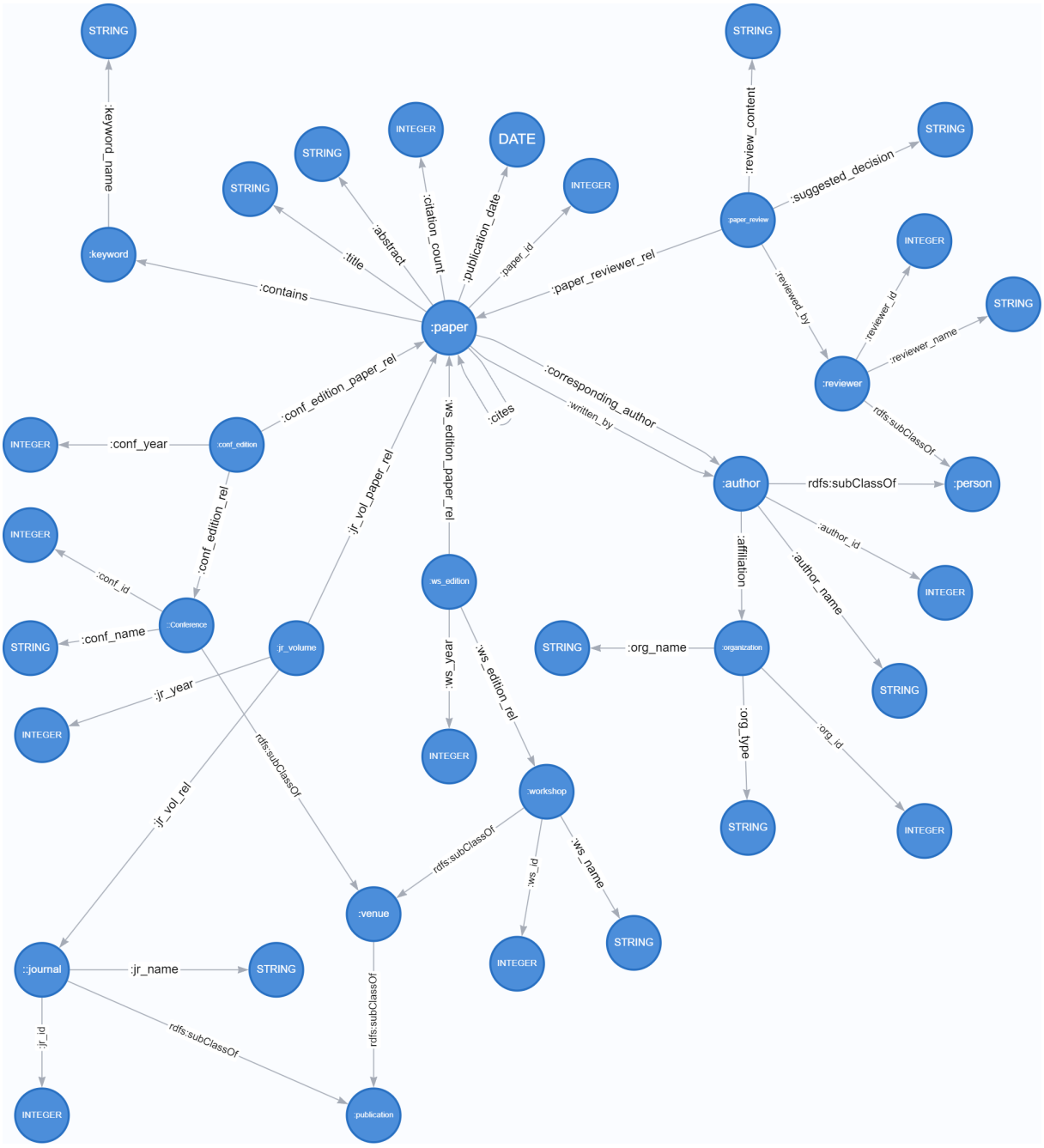
Group:
BDMA 12-A

Authors:
Sony Shrestha
Aayush Paudel

Professor:
Oscar Romero

B.1. TBOX Definition

The following diagram depicts the visual representation of the knowledge graph designed.



TBOX was developed using the **Python library RDFLib**. The corresponding python script is provided in Github repo here and output turtle file generated for TBOX here.
The diagram above provides a visual representation of the TBOX structure. In addition to the predicates visualized in the diagram, several other predicates have been defined in relation to the predicate subPropertyOf. These additional predicates are provided below.

```
:corresponding_author rdfs:subPropertyOf :written_by

:conf_year rdfs:subPropertyOf :pub_year
:jr_year rdfs:subPropertyOf :pub_year
```

```
:ws_year rdfs:subPropertyOf :pub_year

:conf_edition rdfs:subPropertyOf :pub_edition
:ws_edition rdfs:subPropertyOf :pub_edition

:conf_edition_paper_rel rdfs:subPropertyOf :pub_edition
:ws_edition_paper_rel rdfs:subPropertyOf :pub_edition

:jr_edition_paper_rel rdfs:subPropertyOf :prop_publication
:pub_edition rdfs:subPropertyOf :prop_publication
```

The python script creates TBOX for publications. Namespaces such as RDF, RDFS, XSD, and a custom namespace for publications (pub) are defined for use in the graph. Classes related to academic publications, such as Paper, Author, Reviewer, Person, Conference, Journal, Workshop, Keyword are created. Properties that link these classes are defined, for example, paper_id, title, abstract, citation_count, publication_date. Hierarchical Structure between classes are represented by making use of subclass. For instance, Reviewer and Author are subclass of Person, and Conference and Workshop are subclass of Venue, and Venue and Journal are subclass of Publication. Domain and range of properties are defined which represents either relationship between two resources or between a resource and a literal. Finally, the graph is serialized in Turtle format, which is a compact, human-readable serialization format for RDF data.

B.2. ABOX Definition

For the purpose of creating ABOX, we have used **RDFLib library** provided by Python. Using this library, publication data present in CSV file is mapped into RDF output dataset. The corresponding python script is provided in Github repo here and output turtle file generated for ABOX here.

Sample code for generating ABOX for Author is provided below.

```
def authors():
    author_df = pd.read_csv('./data/authors.csv')
    for index, row in author_df.iterrows():
        subject = URIRef(pub + 'author/'+str(row['authorId']))
        author_name_literal = Literal(row['authorName'], datatype = XSD.string)
        author_aff_org_literal = URIRef(pub + 'organization/'+str(row['
    affiliatedOrg']))
        g.add((subject, pub.author_name, author_name_literal))
        g.add((subject, pub.affiliation, author_aff_org_literal))
    return g
```

B.3. Create the final Ontology

We are considering RDFS (Optimized) inference regime entailment. This regime includes rule for:

- rdfs:subClassOf for subclass relationships.
- rdfs:subPropertyOf for subproperty relationships.
- rdfs:domain which sets property domain constraints.
- rdfs:range which sets property range constraints.

However, this regime does not automatically infer rdf:type properties in GraphDB. So, this needs to be explicitly provided by user. In our TBOX, we have added following statement for every properties.

```
:predicate rdf:type rdf:Property
```

Due to the inference rule that RDFS makes use of, it was not needed to define classes separately once we defined domain and range for properties. Given following code for creation of TBOX

```
pub:jr_vol_rel rdfs:domain pub:jr_volume
pub:jr_vol_rel rdfs:range pub:journal
```

It is not needed to explicitly specify

```
pub:jr_volume rdf:type rdfs:Class
pub:journal rdf:type rdfs:Class
```

The statistics about knowledge graph providing number of classes, properties, instances are provided below.

S.N.	Description	Count
1	Number of classes	16
2	Number of properties	42
3	Number of instances for main classes	25,295
3	Number of triples using main properties	381,869

The SPARQL query used to generated above statistics are provided in Github repo here.

B.4. Querying the ontology

This section contains SPARQL queries used to answer questions provided in Section B.4 with two additional queries showing Property Paths and Aggregations in Knowledge Graph respectively. The SPARQL query used to generated above statistics are prodived in Github repo here.

Query 1: Find all Authors

```
PREFIX pub: <http://www.example.edu/publication/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT distinct ?author_name
WHERE {
    ?sub rdf:type pub:author.
    ?sub pub:author_name ?author_name
}
```

The result of query above is provided below.

author_name
Shiv Verma
Ashish R. Mittal
Ginger Tsueng
Aleksandar Makelov
A. Jović
D. Shuman

Query 2: Find all properties whose domain is Author

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX pub: <http://www.example.edu/publication/>
SELECT distinct ?prop
WHERE {
    ?prop rdfs:domain pub:author
}
```

The result of query above is provided below.

	prop
1	pub:affiliation
2	pub:author_id
3	pub:author_name

Query 3: Find all properties whose domain is either Conference or Journal

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX pub: <http://www.example.edu/publication/>
SELECT distinct ?prop
WHERE {
    {
        ?prop rdfs:domain pub:conference
    }
    union
    {
        ?prop rdfs:domain pub:journal
    }
}
```

The result of query above is provided below.

	prop
1	pub:conf_id
2	pub:conf_name
3	pub:journal_id
4	pub:journal_name

Query 4: Find all the papers written by a given author that where published in database conferences

Here, we have assumed that database conferences are the ones whose conference name contains substring database in it.

Also, here we are trying to find all papers written by author **A. Bonifati** that were published in database conference.

```
PREFIX pub: <http://www.example.edu/publication/>
SELECT ?paper_title
WHERE {
  ?paper pub:title ?paper_title .
    ?paper pub:written_by/pub:author_name ?author_name .
    ?conference pub:conf_name ?conf_name .
    ?conference ^pub:conf_edition_rel/pub:conf_edition_paper_rel ?paper .
    FILTER(CONTAINS(STR(LCASE(?conf_name)), "database"))
    FILTER(?author_name='A. Bonifati')
}
```

The result of query above is provided below.

	paper_title
1	"Hierarchical Clustering for Property Graph Schema Discovery"
2	"Big Data Technology Hierarchical Clustering for Property Graph Schema Discovery"
3	"Schema Inference for Property Graphs"
4	"Big Data Technology Schema Inference for Property Graphs"

Query 5: Find all conferences with number of papers published in it in descending order of number of publications

```
PREFIX pub: <http://www.example.edu/publication/>
SELECT ?conf_name (count(distinct(?paper)) AS ?publication_count)
WHERE {
    ?conference pub:conf_name ?conf_name .
    ?conference ^pub:conf_edition_rel/pub:conf_edition_paper_rel ?paper
}
GROUP BY ?conf_name
ORDER BY DESC(?publication_count)
```

The result of query above is provided below.

	conf_name	publication_count
1	"Knowledge Discovery and Data Mining"	"34"^^xsd:integer
2	"Neural Information Processing Systems"	"30"^^xsd:integer
3	"ACM SIGMOD Conference"	"26"^^xsd:integer
4	"International Conference on Learning Representations"	"22"^^xsd:integer
5	"Very Large Data Bases Conference"	"22"^^xsd:integer
6	"International Conference on Machine Learning"	"22"^^xsd:integer

Query 6: List name of all reviewers who reviewed more than 7 papers related to data

Here, we have assumed that paper related to data are one whose keyword contains data in it.

```
PREFIX pub: <http://www.example.edu/publication/>
SELECT ?reviewer (COUNT(?paper) AS ?cnt_paper)
WHERE {
    ?paper ^pub:paper_reviewer_rel/pub:reviewed_by/pub:reviewer_name ?reviewer .
    ?paper pub:contains/pub:keyword_name ?keyword .
    FILTER(CONTAINS(LCASE(?keyword), "data"))
}
GROUP BY ?reviewer
HAVING (?cnt_paper>=7)
```

The result of query above is provided below.

	reviewer	cnt_paper
1	"Neil I. Dewar"	"8"^^xsd:integer
2	"H. Greenspan"	"7"^^xsd:integer
3	"F. V. van Eeuwijk"	"10"^^xsd:integer
4	"R. Stoleru"	"7"^^xsd:integer
5	"Zhonglin Cao"	"7"^^xsd:integer
6	"R. Rastogi"	"13"^^xsd:integer