Vocabulary

The combination of existing vocabularies with custom vocabularies allows to effectively represent the data. It enables to comprehensively model and query the information related to universities, courses, lectures, topics, students, and their relationship in the educational domain.

Reuse of existing vocabularies

Using existing libraries ensures compatibility and adheres to the best practices in RDF data modeling, which facilitates integration and shows consistency.

- **RDF(S):** RDF and RDFS are namespaces that define classes and properties with their relationships. Additionally, **RDFS:labels** was used to name things such as Course, Topic, University, Lecture, etc. in the knowledge base. Moreover, we use **RDFS:seeAlso** to create a link between a resource and a webpage.
- **XDS:** The XSD namespace defines data types such as strings, integers, and decimals.
- **FOAF:** FOAF is an existing vocabulary that describes social networks of humans. providing properties to describe a person and their relationship. Therefore, FOAF:familyName, FOAF:givenName and FOAF:mbox was used to present Student's properties such as name, email, etc. This allows the knowledge base to connect with other knowledge bases.

Vocabulary extension

Developing a custom vocabulary customized to the educational domain allows to precisely model the relationships that are specific to universities, courses, lectures, topics, and students. This will enhance the semantics of the knowledge base.

- **RBPS:** Developed a namespace to define the specific concepts related to the educational institution, courses, lectures, topics, and students.(http://roboprof.com/schema#)
- **RBPD:** Developed a namespace to bind data.(http://roboprof.com/data#)
- **DBP:** A resource media.(http://dbpedia.org/resource/)
-

Vocabulary Content

The vocabulary has 7 Classes:

- University
- Course
- Lecture
- LectureContent: This class has 4 subclasses

o Slides
o Worksheets o Readings
o OtherMaterial

- Topic
- Student
- Attempt

For each property, comments and labels are provided to enhance understandability.

Properties

For each property, comments and labels are provided to enhance understandability.

The expected values for each Property are also defined. Domain will define the subject, and Range will define the Object.

University: No property

Course

Property	Description
CourseSubject	• Domain: Course • Range: String
CourseNumber	• Domain: Course • Range: Integer
CourseOutline	• Domain: Course • Range: Ressource
CourseDescription	• Domain: Course • Range: String
CourseCredits	• Domain: Course • Range: Decimal
CourseOfUniversity	• Domain: Course • Range: University
CoveredTopic	• Domain: Course • Range: Topic

Lecture

Property	Description
LectureNumber	• Domain: Lecture • Range: Integer
TopicLecture	• Domain: Lecture • Range: Topic
hasLectureContent	• Domain: Lecture • Range: LectureContent
LectureOfCourse	• Domain: Lecture • Range: Course

LectureContent: No properties **Topic**

Property	Description
TopicProvenance	• Domain: Topic • Range: LectureContent

Student

Property	Description
IDNumber	• Domain: Student • Range: Integer
hasAttempt	• Domain: Student • Range: Attempt
Competency	• Domain: Student • Range: Topic
hasEnrolled	• Domain: Student • Range: University

Attempt

Property	Description
AttemptCourse	• Domain: Attempt • Range: Course

AttemptGrade	• Domain: Attempt • Range: Decimal

Knowledge Base Construction

A. Dataset

1. To populate the knowledge base for the intelligent agent, Roboprof, two datasets were used from https://opendata.concordia.ca/datasets/: **CATALOG.csv** and **CU_SR_OPEN_DATA_CATALOG.csv**
    - **CATALOG.csv** stores Keys, Faculties, Departments, Programs, Levels, Degrees, Course Codes (Course Subjects), Course Numbers, Titles, Descriptions, Metadata, Types, and Course Websites. Therefore, we use the file to extract information related to a course, such as Course Subject, Course Number, Course Description, and Course Website. Using Titles as Course Names was avoided because some Titles contain characters that may not display correctly due to potential encoding or formatting issues.
    - **CU_SR_OPEN_DATA_CATALOG.csv** stores CourseIDs, Course Subjects, Catalogs, Long Titles, Class Units, Component Codes, Component Descriptions, Prerequisite Descriptions, Careers, and Equivalent Courses. Therefore, the file to extract Course Subject, Course Number, and Course Credits was used. Additionally, Long Titles to represent Course Names in our Knowledge Base was used.
    -
2. To generate a dataset named **Students_Grades.csv**.

    - **Students_Grades.csv** stores example data of Students including their student_id, first_name, last_name and email with their Grades for an enrolled course_subject and course_number. Therefore, a Student was used to generate a Grade for a course_subject and course_number (namely COMP 445 or COMP 474). This file also illustrates that a student might attempt a course multiple times to achieve a different grade.

B. Process and developed tools

- To generate Knowledge_Base.nt and Knowledge_Base.ttl, run the command:

```
python create_knowledge_base.py
```

- The script **create_knowledge_base.py** is used to generate the knowledge base.
- Firstly, a University instance named Concordia University manually was created, and then added to the knowledge base

- In the script, **pandas** was used to load the course information from CATALOG.csv CU_SR_OPEN_DATA_CATALOG.csv and save the information in a dictionary. This process is in the method **load_course_info().**
- Additionally, topics such as Intelligent systems, Knowledge graph, etc., were created then added to our knowledge base**.** This process is executed in the **add_topics_to_graph()** method.
- Moreover, course materials for COMP 474 and COMP 445 was collected, organizing them into folders such as slides, reading materials, and worksheets. A course outline for each course was also included. Furthermore, instances of the course materials was created and added to the knowledge base. The process occurs in the **add_materials_to_course()** method.
- The **load_students()** method loads student information from the Students_Grades.csv file. In the method, each student is iterated, adding their details (student_id, first_name, last_name and email) and enrolled courses along with grades to the Students_Grades.csv file.
- The **add_student_to_graph()** method creates a Student Node in the knowledge graph using their IDnumber, the **FOAF** namespace and the vocabularies. The Node includes:
    - Basic Information, including first_name, last_name, email using **RDF.type**, **FOAF.givenName**, **FOAF.familyName** and **FOAF.mbox**
    - Student enrollment in a University using **rbps.hasEnrolled**
    - Attempt instances was also created, which represent the number of times a student has attempted to complete a course. This allows the knowledge base to represent the idea that a student can retake a course to achieve a different grade. After creating the Student instances, they were added to the knowledge base.
- Whenever an instance is created, relations between that instance and other relevant instances in the knowledge base is established (i.e Student and Attempt, Student and Course, Course and Lecture, etc.)
- To create and add instances and relations to the knowledge base, the RDFLib Python package was used.
- After completing the knowledge base, it is written to files (i.e Knowledge_Base.nt, Knowledge_Base.ttl) in both N-Triples and Turtle format

Graph Queries

Question 1

SPARQL Endpoint: /Roboprof/query

Content Type (SELECT): JSON

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rbps: <http://roboprof.com/schema#>

SELECT ?course ?courseLabel
WHERE {
    ?course rdfs:label ?courseLabel.
    ?course rbps:CourseOfUniversity ?university.
    ?university rdfs:label "Concordia University"@en
}
```

Table | Response | 1464 results in 0.064 seconds

	course	courseLabel
1	<http://roboprof.com/data#CHST608>	"FIELD OBSERVATIONS"@en
2	<http://roboprof.com/data#MECH452>	"Heat Transfer II"@en
3	<http://roboprof.com/data#PHOT399>	"SPECIAL TOPICS IN PHOTOGRAPHY"@en
4	<http://roboprof.com/data#COMP5361>	"DISCRETE STR.+FORMAL LANG."@en

This query returns all courses offered by Concordia University in the knowledge graph.

Question 2

SPARQL Query

To try out some SPARQL queries against the selected dataset, enter your query here.

Example Queries: Selection of triples | Selection of classes

Prefixes: rdf | rdfs | owl | xsd

SPARQL Endpoint: /Roboprof/query

Content Type (SELECT): JSON

Content Typ: Turtle

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rbps: <http://roboprof.com/schema#>

SELECT ?course ?courseLabel
WHERE {
    ?course rdf:type rbps:Course ;
            rdfs:label ?courseLabel ;
            rbps:CoveredTopic ?topic.

    ?topic rdf:type rbps:Topic ;
           rdfs:label "Intelligent system"@en.
}
```

Press CTRL + <spacebar> to autocomplete

Table | Response | 1 result in 0.011 seconds

Simple view | Ellip

	course	courseLabel
1	<http://roboprof.com/data#COMP474>	"Intelligent Systems"@en

This query returns all the courses in which the topic: “Intelligent system” is discussed.

Question 3

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rbps: <http://roboprof.com/schema#>
PREFIX rbpd: <http://roboprof.com/data#>

SELECT ?topic ?topicLabel
WHERE {
    ?topic rdfs:label ?topicLabel .
    ?lecture rdf:type rbps:Lecture ;
             rbps:LectureOfCourse ?course ;
             rbps:LectureNumber '3'^^xsd:integer;
             rbps:TopicLecture ?topic .
  ?course rdf:type rbps:Course ;
          rdfs:label "Intelligent Systems"@en
}
```

Table | Response | 2 results in 0.011 seconds | Simple view ☐ Ellipse ☑ | Filter query res

	topic	topicLabel
1	<http://roboprof.com/data#Linked_data>	"Linked data"@en
2	<http://roboprof.com/data#SPARQL>	"SPARQL"@en

This query returns which topic is covered in lecture 1 of the course “Data Communication and Computer Network”

Question 4

SPARQL Endpoint: /Roboprof/query

Content Type (SELECT): JSON

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rbps: <http://roboprof.com/schema#>
PREFIX rbpd: <http://roboprof.com/data#>

SELECT ?course ?courseLabel
WHERE {
    ?course rdf:type rbps:Course ;
            rdfs:label ?courseLabel ;
            rbps:CourseOfUniversity ?university ;
            rbps:CourseSubject "COMP"@en .
  ?university rdf:type rbps:University ;
              rdfs:label "Concordia University"@en.
}
```

Table | Response | 43 results in 0.011 seconds

	course	courseLabel
1	<http://roboprof.com/data#COMP5361>	"DISCRETE STR.+FORMAL LANG."@en
2	<http://roboprof.com/data#COMP428>	"Parallel Programming"@en
3	<http://roboprof.com/data#COMP5541>	"TOOLS+TECHNIQUES / SW ENGR."@en
4	<http://roboprof.com/data#COMP492>	"Computer Science Project II"@en

This query returns a list of all the courses offered by Concordia University within the subject name “COMP”

Question 5

```
SELECT ?material ?material_label ?material_type_label
WHERE {
    ?material rdf:type ?material_type ;
              rdfs:label ?material_label.

    ?material_type rdfs:subClassOf rbps:LectureContent ;
                   rdfs:label ?material_type_label.

    ?lecture rbps:haslectureContent ?material;
             rbps:TopicLecture rbpd:Machine_learning;
             rbps:LectureOfCourse ?course.

    ?course rdf:type rbps:Course ;
            rbps:CourseSubject "COMP"@en ;
            rbps:CourseNumber "474"@en.
}
```

Table | Response | 3 results in 0.008 seconds | Simple view ☐ Ellipse ☑ | Filter query results | Page size: 5

	material	material_label	material_type_label
1	<http://concordiauniversity.on.worldcat.org/oclc/314121652>	"Introduction to Information Retrieval"@en	"Readings"@en
2	<http://concordiauniversity.on.worldcat.org/oclc/314121652>	"Collective intelligence in action"@en	"Readings"@en
3	<file:///home/roboprof/COMP474/lectures/slides05.pdf>	"slides05.pdf"@en	"Slides"@en

This query returns what materials are recommended for the topic “Machine_learning” from the course “COMP” “474”

Question 6

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rbps: <http://roboprof.com/schema#>
PREFIX rbpd: <http://roboprof.com/data#>

SELECT ?credits
WHERE {
    ?course rdf:type rbps:Course ;
            rbps:CourseSubject "ACCO"@en ;
            rbps:CourseNumber "355"@en ;
            rbps:CourseCredits ?credits .
}
```

Table | Response | 1 result in 0.013 seconds | Simple view ☐ Ellipse ☑ | Filter query results | Page size: 50

	credits
1	"3.00"^^<http://www.w3.org/2001/XMLSchema#decimal>

This query returns the number of credits in the course “ACCO” “355”

Question 7

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rbps: <http://roboprof.com/schema#>
Prefix rbpd: <http://roboprof.com/data#>

SELECT ?additionalResource
WHERE {
    ?course rdf:type rbps:Course ;
            rbps:CourseSubject "BLDG"@en ;
            rbps:CourseNumber "390"@en ;
            rdfs:seeAlso ?additionalResource.
}
```

Table | Response | 1 result in 0.007 seconds

Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	additionalResource
1	<https://www.concordia.ca/ginacody/computer-science-software-eng/research/groups.html>

This query returns the additional resources that are available for the course “BLDG” “390”

Question 8

```
    ?material rdf:type ?material_type ;
              rdfs:label ?material_label.

    ?material_type rdfs:subClassOf rbps:LectureContent ;
                   rdfs:label ?material_type_label.

    ?lecture rbps:hasLectureContent ?material ;
             rbps:LectureNumber "2"^^xsd:integer ;
             rbps:LectureOfCourse ?course.

    ?course rdf:type rbps:Course ;
            rbps:CourseSubject "COMP"@en ;
            rbps:CourseNumber "474"@en.
}
```

Table | Response | 3 results in 0.007 seconds

Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	material	material_label	material_type_la...
1	<http://concordiauniversity.on.worldcat.org/oc...	"A Developer's Guide to the Semantic...	"Readings"@en
2	<file:///home/roboprof/COMP474/worksheets...	"worksheet02.pdf"@en	"Worksheets"@en
3	<file:///home/roboprof/COMP474/lectures/sli...	"slides02.pdf"@en	"Slides"@en

This query returns the detailed lecture content from the lecture “2” of the course “COMP” “474”

Question 9

```
PREFIX rbpd: <http://roboprof.com/data#>

SELECT ?material ?material_label
WHERE {
    ?material rdf:type rbps:Readings ;
              rdfs:label ?material_label.

    ?lecture rbps:hasLectureContent ?material ;
             rbps:TopicLecture rbpd:Intelligent_system ;
             rbps:LectureOfCourse ?course.

    ?course rdf:type rbps:Course ;
            rdfs:label "Intelligent Systems"@en.
}
```

Table | Response | 3 results in 0.008 seconds

Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	material	material_label
1	<http://en.wikipedia.org/wiki/ELIZA>	"ELIZA"@en
2	<http://en.wikipedia.org/wiki/Watson_(computer)>	"Watson"@en
3	<http://plato.stanford.edu/entries/artificial-intelligence/>	"Artificial Intelligence"@en

This query returns the recommended reading material for studying “Intelligent_systems” from the course “Intelligent Systems”

Question 10

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rbps: <http://roboprof.com/schema#>

SELECT ?competency ?competencyLabel
WHERE {
    ?course rdf:type rbps:Course ;
            rbps:CourseSubject "COMP"@en;
            rbps:CourseNumber "474"@en;
            rbps:CoveredTopic ?competency.

    ?competency rdfs:label ?competencyLabel.
}
```

Table | Response | 7 results in 0.011 seconds

Simple view ☐ Ellipse ☑ Filter query results

	competency	competencyLabel
1	<http://roboprof.com/data#Recommender_system>	"Recommender system"@en
2	<http://roboprof.com/data#Intelligent_system>	"Intelligent system"@en
3	<http://roboprof.com/data#Linked_data>	"Linked data"@en
4	<http://roboprof.com/data#Knowledge_graph>	"Knowledge graph"@en
5	<http://roboprof.com/data#Machine_learning>	"Machine learning"@en
6	<http://roboprof.com/data#Personalization>	"Personalization"@en
7	<http://roboprof.com/data#SPARQL>	"SPARQL"@en

This query returns all the competencies (topics) that a student gains after completing the course “COMP” “474”

Question 11

```
WHERE {
  ?student rdf:type rbps:Student ;
           rbps:IDNumber "40074296"^^xsd:integer ;
           rbps:hasAttempt ?attemptedCourse.

  ?attemptedCourse rdf:type rbps:Attempt;
                   rbps:AttemptCourse ?course ;
                   rbps:AttemptGrade ?grade.

  ?course rdf:type rbps:Course ;
          rbps:CourseSubject "COMP"@en ;
          rbps:CourseNumber "474"@en.

}
```

Table | Response | 2 results in 0.012 seconds

Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	grade
1	"45.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
2	"79.0"^^<http://www.w3.org/2001/XMLSchema#decimal>

This query returns the grades that student with ID “40074296” achieved in the course “COMP” “474”

Question 12

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rbps: <http://roboprof.com/schema#>

SELECT ?student ?id ?firstName ?lastName
WHERE {
    ?course rdf:type rbps:Course ;
            rbps:CourseSubject "COMP"@en ;
            rbps:CourseNumber "445"@en .

    ?student rdf:type rbps:Student ;
             rbps:IDNumber ?id;
```

Table | Response | 6 results in 0.009 seconds

Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	student	id	firstName	lastName
1	<http://roboprof.com/data#40182...	"40182191"^^<http://www.w3.org/2001/XMLSchema...	"Trenton"@en	"Doyle"@en
2	<http://roboprof.com/data#40057...	"40057323"^^<http://www.w3.org/2001/XMLSchema...	"Deandre"@en	"Beasley"...
3	<http://roboprof.com/data#40012...	"40012687"^^<http://www.w3.org/2001/XMLSchema...	"Elaina"@en	"Burns"@en
4	<http://roboprof.com/data#40137...	"40137428"^^<http://www.w3.org/2001/XMLSchema...	"August"@en	"Thomas"...
5	<http://roboprof.com/data#40057...	"40057320"^^<http://www.w3.org/2001/XMLSchema...	"Sutton"@en	"Randall"...
6	<http://roboprof.com/data#40289...	"40289039"^^<http://www.w3.org/2001/XMLSchema...	"Elizabeth"...	"Landry"@en

This query returns which student, with their ID, that has completed the course “COMP” “445”

Question 13

```
16
17      ?student rdf:type rbps:Student ;
18               rbps:IDNumber '40182191'^^xsd:integer;
19               foaf:givenName 'Trenton'@en ;
20               foaf:familyName 'Doyle'@en ;
21               rbps:hasAttempt ?attempt .
22
23      ?attempt rdf:type rbps:Attempt;
24               rbps:AttemptCourse ?course;
25               rbps:AttemptGrade ?grade.
26  }
27
```

Table | Response | 3 results in 0.024 seconds | Simple view ☐ Ellipse ☑ | Filter query results | Page size: 50

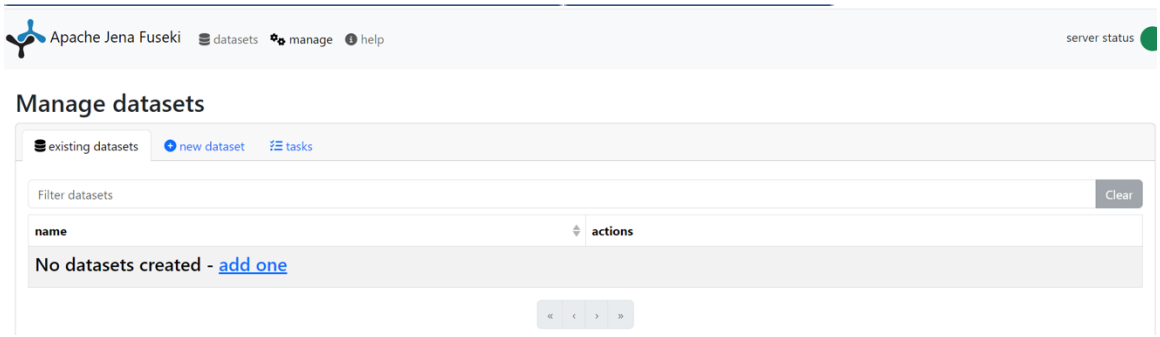
	universityLabel	courseLabel	courseSubject	courseNumber	grade
1	"Concordia University"@en	"Data Communication and Computer Networks"@en	"COMP"@en	"445"@en	"75.2"^^<http://www.w3.org/2001/XMLSchema#decimal>
2	"Concordia University"@en	"Intelligent Systems"@en	"COMP"@en	"474"@en	"40.6"^^<http://www.w3.org/2001/XMLSchema#decimal>
3	"Concordia University"@en	"Intelligent Systems"@en	"COMP"@en	"474"@en	"80.8"^^<http://www.w3.org/2001/XMLSchema#decimal>

Showing 1 to 3 of 3 entries < 1 >

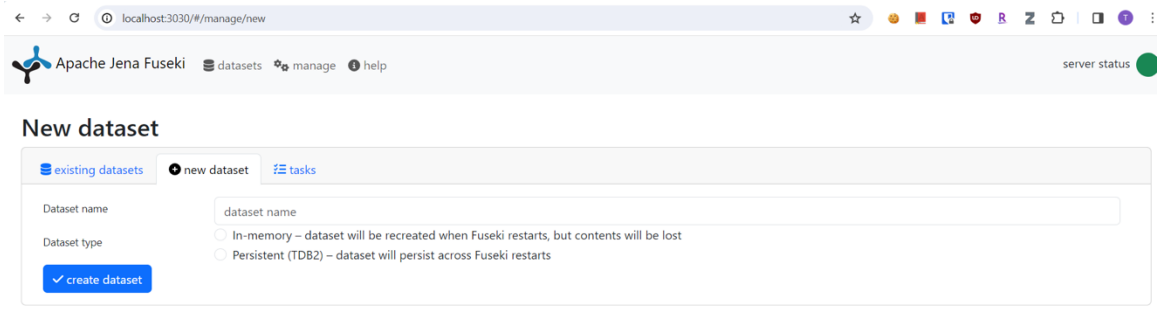
This query returns a transcript for student “40182191” that shows all the courses he has taken with their grades.

Triplestore and SPARQL Endpoint Setup SPARQL Endpoint Setup

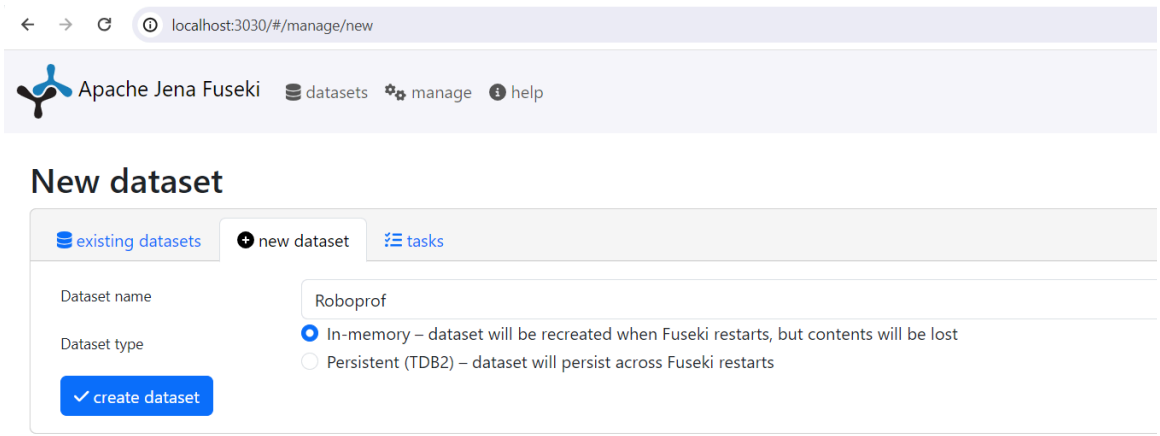
1. Download the fuseki binary distribution from https://jena.apache.org/download/index.cgi
2. Install and start Apache Fuseki server, which will listen for requests on port 3030
3. Access Fuseki's web interface by navigating to http://localhost:3030/ in your browser
4. Select the 'manage' tab



5. Select the “new dataset” tab

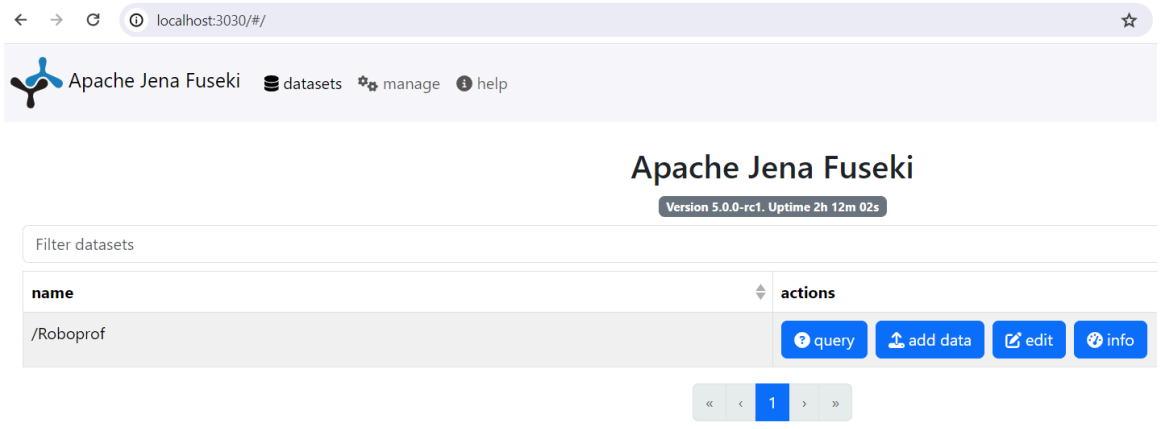


6. Create a dataset named **Roboprof**

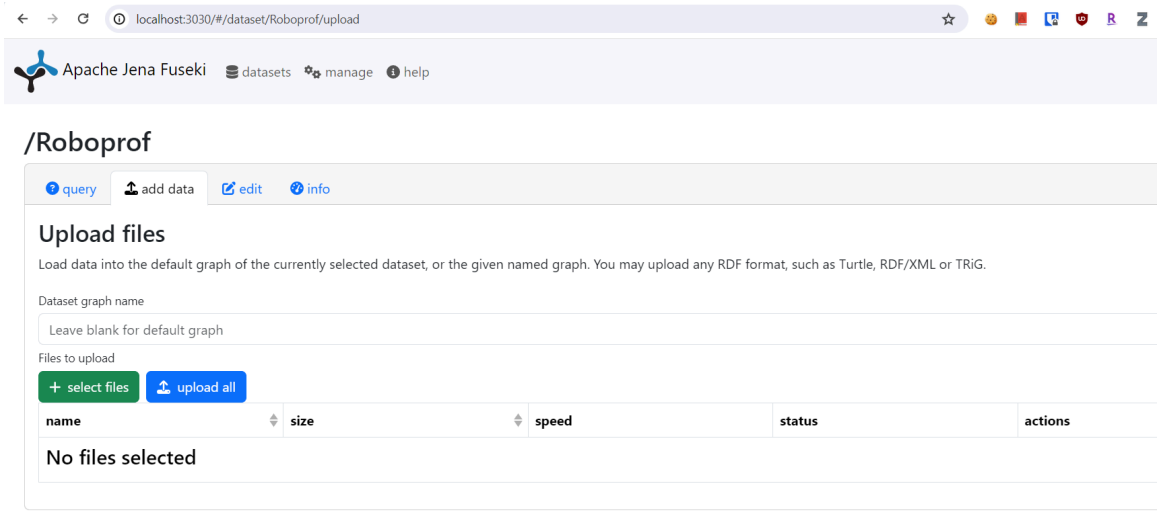


7. Under the “existing datasets” tab, select Roboprof dataset

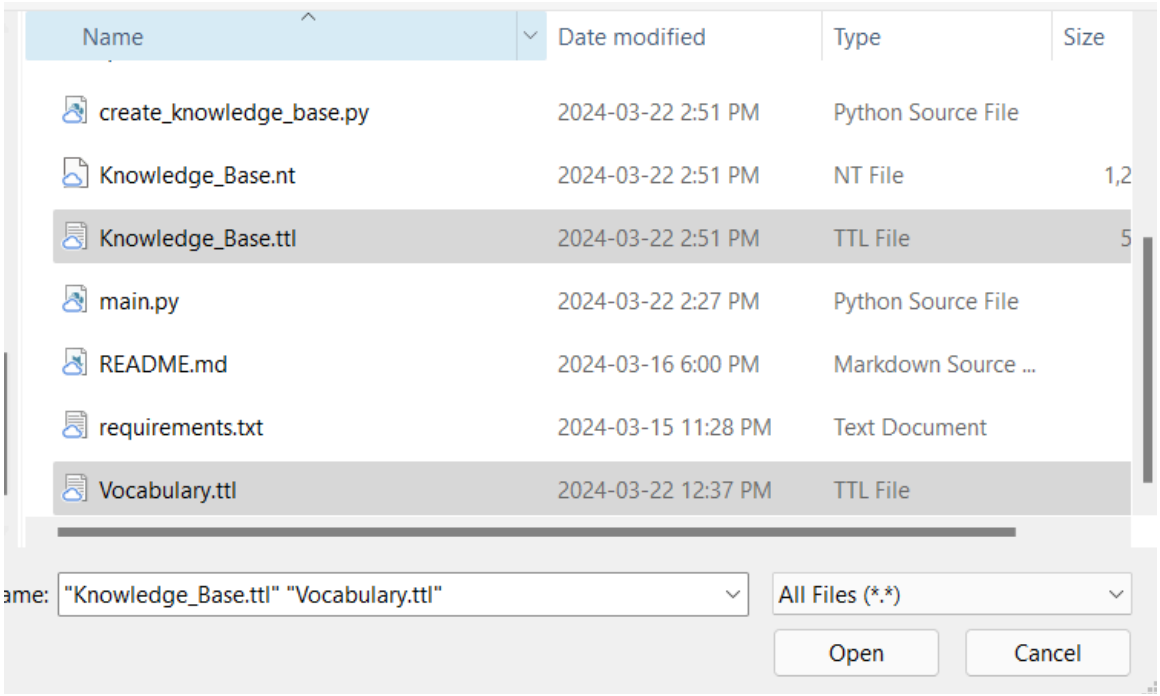
8. Select the “add data” button



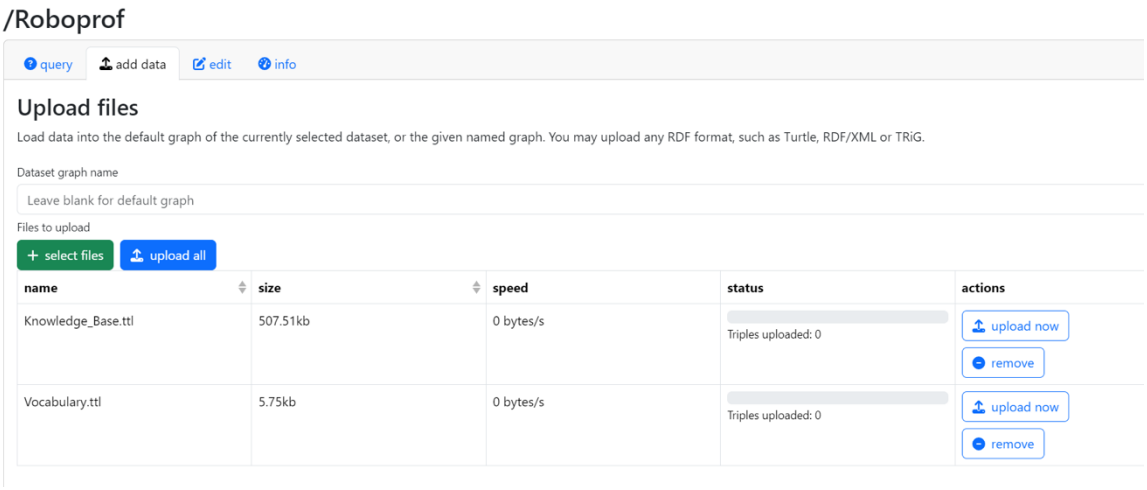
9. Select the “select files” button



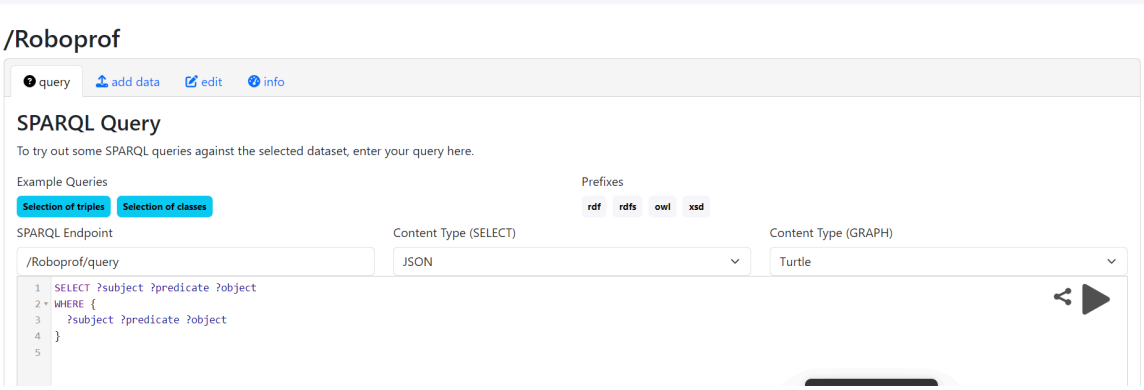
10. Select “Vocabulary.ttl” and “Knowledge_Base.ttl” and click “Open”



11. Select the “upload all” button



12. Now the SPARQL server is ready to query. 13. Go to the “query” tab to start querying.



Submit queries using Python

To submit queries using a Python application, I utilize SPARQLWrapper, a Python library that facilitates interaction between a Python application and the SPARQL server.

1. In **main.py**, create a method r**unQuery(query, format=CSV)**
2. In the method, establish a connection between the application and the server by

   writing:

   ```
   sparql = SPARQLWrapper("http://localhost:3030/Roboprof/query")
   ```

3. Set the result's format for a query
4. ```
   sparql.setReturnFormat(format)
   ```
5. Set a query we want to submit
6. ```
   sparql.setQuery(query)
   ```
7. Submit the query

```
try:
    ret = sparql.queryAndConvert()
    return ret
except Exception as e:
```

```
print(e)
```

6. Now, you can run a query from the application. For example:

```
7. 8. 9. 10. 11. 12. 13.

totalTriplesQuery = '''
    SELECT (COUNT(*) as ?NumberOfTriples)
    WHERE {
        ?subject ?predicate ?object .
    }

'''

ret = runQuery(totalTriplesQuery, JSON)
```

Furthermore, each query is stored in a .txt file. To run all queries, run:

```
 python main.py
```

This command will extract and submit the content of each query file to the SPARQL server and write its output into a CSV file.

Important Notes: Before running the command above, please ensure that you upload "Knowledge_Base.ttl" and "Vocabulary.ttl" to the SPARQL server