COMP 6741 – WINTER 2024

Moodle Team Name: FS_G_01

Team Members:

Student Name	Student Id	Specialization
Venkatasubrahmanyamanisaisri Bandaru	40264171	Masters in Applied Computer Science
Anmol Arora	40172251	Masters in Applied Computer Science

Project Repository:

https://github.com/Sriluharshini/RoboProf

We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality.

Signature:

Anmol Arora - 40172251 - 22-03-2024	Venkatasubrahmanyamanisaisri Bandaru - 40264171 - 22-03-2024

I. Vocabulary

Classes:

- The definition of the classes has been written in the vocabulary for University, Course, Lectures, Topic, Student, Slides, Reading, Worksheets, and Other Material. Every class is a distinct entity.
- Justification: Having classes helps in organizing the data entities so that they can have distinctions between them.

Object Properties:

- The object properties such as hasCourse, hasStudent, hasTopic, hasLectures, etc have been defined in order to construct the relations between the instances of different classes. For example, hasTopic creates a relation between a lecture and the topics.
- Inverse properties such as enrolledAt and enrolledIn establish bidirectional relationships between Students and Universities/Courses.
- Justification: Object properties make it easier to navigate and query the knowledge base by defining the relationships between various items.

Datatype Properties:

- Datatype properties define attributes or characteristics of entities. For example, courseName, courseDescription, studentName, etc., capture textual or URI values associated with entities.
- In the vocabulary, datatype properties were used to define the attributes of an entity(class). For instance, course units, courseDescription, studentName, etc., capture textual or URI values associated with entities.
- Justification: In-depth descriptions and information about entities can be stored and retrieved from the knowledge base using datatype properties and this is a sophisticated and a structured approach.

Subclassing:

- Relation between the classes such as one class is the subclass of another has also been defined in order to model the entities in a specialized manner. For instance, focu:Student is defined to be a subclass of foaf:Person.
- Justification: Subclassing facilitates in describing hierarchical relationships and improves entity modeling by focusing on specific characteristics and behaviors.

Labels and Comments:

- Multilingual labels and comments improve the schema's readability and usefulness and increase its accessibility for a wider range of users.
- Justification: Comments and labels act as the schema's documentation, including background information, justifications wherever needed.

II. Knowledge Base Construction

Dataset

The knowledge base was populated with a dataset that includes information from multiple sources that are necessary to build a thorough grasp of data related to Concordia University's courses. The course material was taken from the open data portal at Concordia University. Course names, subjects, numbers, credits, descriptions, URLs, and outlines have been included in this dataset.
The CSV files `lectures.csv`, `slides.csv`, `students.csv`, `topics.csv`, and `university.csv` consist of the data for Lectures, Slides, Students, Topics, and University. Each of these files includes details about the university as well as information about lecture topics, student profiles, lecture slides, and lecture information.

Process and Developed Tools:

- **Import Libraries:**
  The Python script imports necessary libraries such as ‘rdflib’, ‘pandas’, etc for handling CSV files and graph manipulation. These libraries are essential for efficiently processing and transforming the dataset into RDF triples.

- **Specify File Path and Namespaces:**
  The script specifies the file paths for each CSV file containing the dataset. Additionally, it defines RDF namespaces to represent the schema and data within the RDF graph. This step is crucial for organizing and structuring the RDF triples generated from the dataset.

- **Create RDF Graph:**
  Using the `Graph` class from the `rdflib` library, the script creates an RDF graph. This graph serves as the primary container for storing RDF triples that represent the information extracted from the dataset.

- **Read CSV Data:**
  The script reads the data from each CSV file, extracting information about lectures, slides, students, topics, and the university from their respective files.

- **Iterate Over Rows and Add Triples to Graph:**
  RDF triples are generated for each attribute of the entities represented in the dataset. For example, for each course, lecture, student, etc RDF triples are created to capture their properties and relationships.

- **Serialize RDF Graph:**
  Finally, the RDF graph is serialized to a Turtle file using the `serialize` method provided by the `rdflib` library. This Turtle file represents the populated knowledge base containing information derived from the dataset.

Instructions for the construction of the knowledge base and execution:

- Ensure Python is installed on your system.
- Install the required libraries (`pandas` and `rdflib`) using pip if not already installed.
- Open each python script to be executed.
- Specify the destination path and filename correctly in the python script..
- Run the code to populate data into the destination file..
- Merge all the TTL files to populate the main Knowledge_Base graph.
- Run the Apache-fuseki server and upload the Knowledge_Base.ttl file.
- Click on the query tab above and run the queries given in the queries folders.
- Check and compare the results with the knowledge graph and refer vocabulary graph for detailed information on the relationships.

III. Graph Queries:

#1. List all courses offered by [Concordia_University]

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?course ?CourseName ?CourseCredits ?CourseSubject ?CourseCatalog ?Description ?Career
WHERE {
  focudata:Concordia_University focu:hasCourse ?course .
        ?course
                focudata:CourseCredits ?CourseCredits ;
                focudata:CourseName ?CourseName ;
                focudata:CourseSubject ?CourseSubject ;
                focudata:career ?Career ;
                focudata:catalog ?CourseCatalog ;
                focudata:Descr ?Description .
}
```

Output:

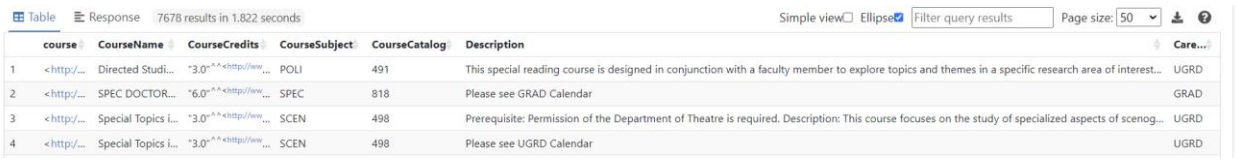


Table | Response | 7678 results in 1.822 seconds — Simple view ☐ Ellipse ☑ — Filter query results — Page size: 50

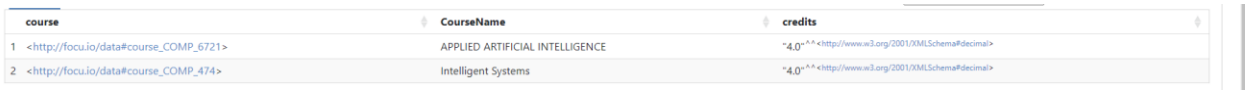
	course	CourseName	CourseCredits	CourseSubject	CourseCatalog	Description	Care...
1	<http:/...	Directed Studi...	"3.0"^^<http://ww...	POLI	491	This special reading course is designed in conjunction with a faculty member to explore topics and themes in a specific research area of interest...	UGRD
2	<http:/...	SPEC DOCTOR...	"6.0"^^<http://ww...	SPEC	818	Please see GRAD Calendar	GRAD
3	<http:/...	Special Topics i...	"3.0"^^<http://ww...	SCEN	498	Prerequisite: Permission of the Department of Theatre is required. Description: This course focuses on the study of specialized aspects of scenog...	UGRD
4	<http:/...	Special Topics i...	"3.0"^^<http://ww...	SCEN	498	Please see UGRD Calendar	UGRD

#2 In which courses is [topic = Machine_Learning] discussed?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?course ?CourseName ?credits
WHERE {
  ?course focu:hasTopic focudata:Machine_Learning;
                        focudata:CourseName ?CourseName ;
          focudata:CourseCredits ?credits .
}
```

Output:



	course	CourseName	credits
1	<http://focu.io/data#course_COMP_6721>	APPLIED ARTIFICIAL INTELLIGENCE	"4.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
2	<http://focu.io/data#course_COMP_474>	Intelligent Systems	"4.0"^^<http://www.w3.org/2001/XMLSchema#decimal>

 # 3. Which [topics] are covered in [course_COMP_6721] during [lecture_3]?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>


SELECT ?topicName ?lectures
WHERE {
    focudata:course_COMP_6721 a focu:Course ;
          focu:hasLectures ?lectures .
  ?lectures focudata:lectureNumber 3 .
   ?lectures focu:hasTopics ?topicName .
}
```

Output:

	topicName	lectures
1	<http://focu.io/data#Machine_Learning>	<http://focu.io/schema#lecture_3>

 # 4. List all [courses] offered by [Concordia_University] within the [subject = COMP] (e.g., "COMP", "SOEN").

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>


SELECT ?COURSES ?subjectName
WHERE {
  focudata:Concordia_University focu:hasCourse ?COURSES .
  ?COURSES focudata:CourseSubject ?subjectName .
}
```

Output:

	COURSES	subjectName
1	<http://focu.io/data#course_POLI_491>	POLI
2	<http://focu.io/data#course_SPEC_818>	SPEC
3	<http://focu.io/data#course_SCEN_498>	SCEN
4	<http://focu.io/data#course_APLI_635>	APLI
5	<http://focu.io/data#course_EDUC_494>	EDUC

#5. What [materials] (slides, readings) are recommended for [topic = Knowledge_Representation] in [course_COMP_474]?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>


SELECT ?RecommendedSlides ?Readings
WHERE {
```

```
  focudata:course_COMP_474 focu:hasTopic focudata:Knowledge_Representation .
  focudata:Knowledge_Representation focu:hasSlides ?RecommendedSlides .
}
```

Output:

	RecommendedSlides	Readings
1	<http://focu.io/data#slides03>	

 # 6. How many credits is [course] [6721] worth?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>


SELECT ?courseName ?courseCredits
WHERE {
  focudata:course_COMP_6721 focudata:CourseCredits ?courseCredits ;
                    focudata:CourseName ?courseName .
}
```

Output:

	courseName	courseCredits
1	APPLIED ARTIFICIAL INTELLIGENCE	"4.0"^^<http://www.w3.org/2001/XMLSchema#decimal>

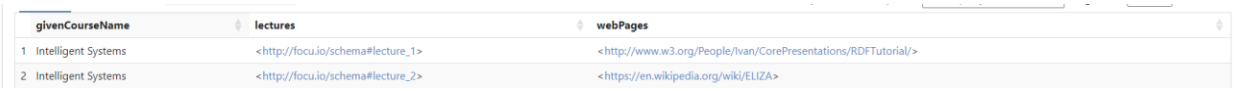
 # 7. For the course_COMP_474, what additional resources (links to web pages) are available?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?givenCourseName ?lectures ?webPages
WHERE {
  focudata:course_COMP_474 focu:hasLectures ?lectures ;
```

```
                focudata:CourseName ?givenCourseName .
  ?lectures focudata:lectureWebpage ?webPages .
}
```

Output:



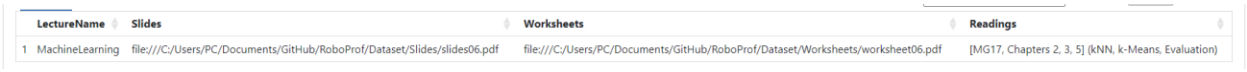
	givenCourseName	lectures	webPages
1	Intelligent Systems	<http://focu.io/schema#lecture_1>	<http://www.w3.org/People/Ivan/CorePresentations/RDFTutorial/>
2	Intelligent Systems	<http://focu.io/schema#lecture_2>	<https://en.wikipedia.org/wiki/ELIZA>

 # 8. Detail the content (slides, worksheets, readings) available for lecture_3 in course_COMP_474.

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?LectureName ?Slides ?Worksheets ?Readings
WHERE {
  focudata:course_COMP_474 focu:hasLectures focu:lecture_3 .
  focu:lecture_3
                    focudata:lectureName ?LectureName ;
                    focudata:Slides ?Slides ;
          focudata:Worksheets ?Worksheets ;
          focudata:Reading ?Readings .
}
```

Output:



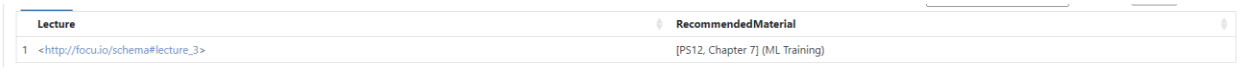
	LectureName	Slides	Worksheets	Readings
1	MachineLearning	file:///C:/Users/PC/Documents/GitHub/RoboProf/Dataset/Slides/slides06.pdf	file:///C:/Users/PC/Documents/GitHub/RoboProf/Dataset/Worksheets/worksheet06.pdf	[MG17, Chapters 2, 3, 5] (kNN, k-Means, Evaluation)

 # 9. What reading materials are recommended for studying [topic] in [course]?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>
```

```
SELECT ?Lecture ?RecommendedMaterial
WHERE {
  focudata:course_COMP_474 focu:hasLectures ?Lecture .
  ?Lecture focu:hasTopics focudata:Machine_Learning .
  ?Lecture focudata:OtherMaterial ?RecommendedMaterial .
}
```

Output:



	Lecture	RecommendedMaterial
1	<http://focu.io/schema#lecture_3>	[PS12, Chapter 7] (ML Training)

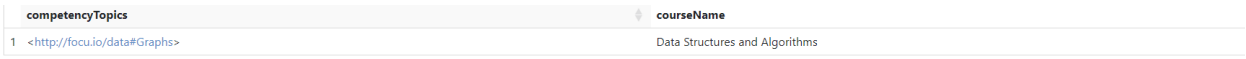
 # 10. COMPETENCY [TOPICS] THAT A STUDENT CAN GAIN AFTER COMPLETING course_COMP_352.

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?competencyTopics ?courseName
WHERE {
   focudata:course_COMP_352 focu:hasTopic ?competencyTopics ;
                focudata:CourseName ?courseName.

}
```

Output:



	competencyTopics	courseName
1	<http://focu.io/data#Graphs>	Data Structures and Algorithms

 # 11. What grades did [student] achieve in [course] [number]?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?courseNumber ?Grades
```

```
WHERE {
   focu:John_Doe focudata:completedCourses focudata:course_COMP_352 ;
                 focudata:courseGrade ?Grades.
 focu:John_Doe focudata:completedCourses ?courseNumber.

}
```

Output:

	courseNumber	Grades
1	<http://focu.io/data#course_COMP_352>	A

#12. Which [students] have completed [course] [number]?

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT DISTINCT ?students ?courseNumber
WHERE {
   ?students focudata:completedCourses focudata:course_COMP_352 ;
          focudata:completedCourses ?courseNumber
}
```

Output:

	students	courseNumber
1	<http://focu.io/schema#John_Doe>	<http://focu.io/data#course_COMP_352>

#13. Print a transcript for a [student], listing all the course taken with their grades.

```
PREFIX fo: <http://www.w3.org/1999/XSL/Format#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX focu: <http://focu.io/schema#>
prefix focudata: <http://focu.io/data#>

SELECT ?studentID ?completedCourse ?grade
WHERE {
```

```
        focu:Alice_Johnson
                        focudata:studentID ?studentID ;
                        focudata:completedCourses ?completedCourse ;
                        focudata:courseGrade ?grade .
}
```

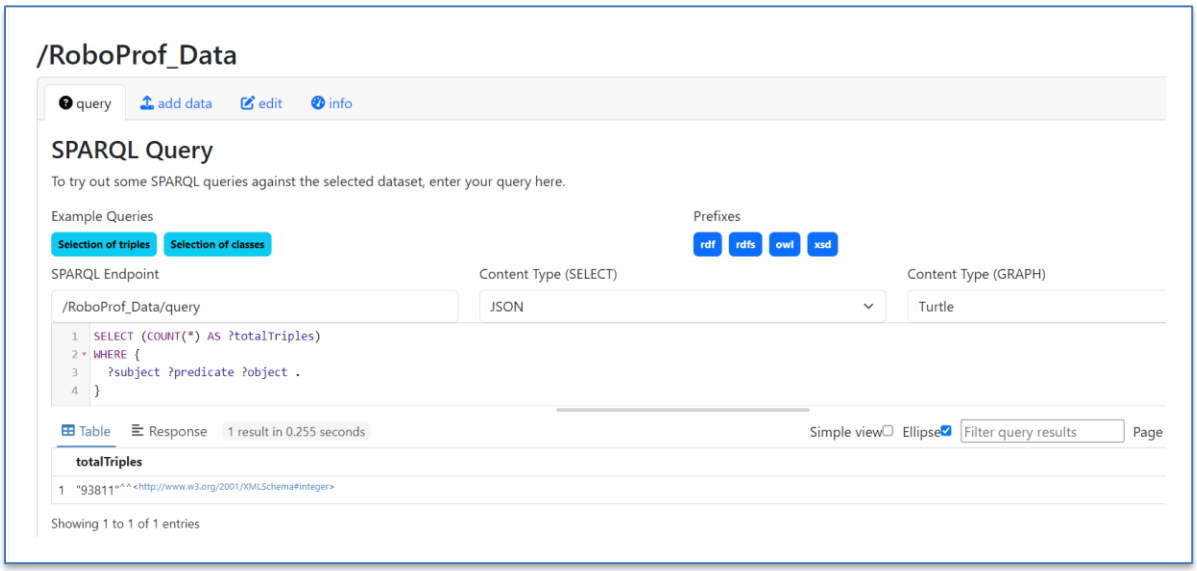
Output:

	studentID	completedCourse	grade
1	345678	<http://focu.io/data#course_COMP_201>	B

Knowledge Base Statistics:

- The total number of triples in the Knowledge Base is 93811

```
SELECT (COUNT(*) AS ?totalTriples)
WHERE {
  ?subject ?predicate ?object .
}
```



- The count of course URIs present is 7635.

```
SELECT (COUNT(DISTINCT ?course) AS ?totalCourses)
WHERE {
  ?course a <http://focu.io/schema#Course> .
}
```

IV. Triplestore and SPARQL Endpoint Setup:

Steps for setting up the endpoint:

1. We have installed Apache Jena Fuseki according to the documentation, unzipped the file and configured the path.
2. Start the server using the command - fuseki-server.
3. Prepare and populate the knowledge graph for ingestion into the triplestore.
4. Upload the graph populated in turtle format, enable the SPARQL endpoint and run basic queries to verify that the endpoint is functional.
5. Query the data and validate it with expected results.

Challenges:

1. Slow Query Execution: Working on large datasets leads to slow execution of queries. To solve that we have to optimize query performance by indexing frequently queried properties and employ query optimization techniques such as query rewriting and query plan analysis
2. Query Complexity: Complex query construction can be challenging for users. Solutions include providing query builders, templates, and documentation to aid users.