Table of Contents

[1. Introduction 2](#_Toc34619602)

[2. Description 3](#_Toc34619603)

[2.1 University Knowledge Graph 3](#_Toc34619604)

[2.1.1 University 3](#_Toc34619605)

[2.1.2 Course 3](#_Toc34619606)

[2.1.3 Topic 3](#_Toc34619607)

[2.1.4 Student 3](#_Toc34619608)

[2.1.5 Course Grade 3](#_Toc34619609)

[2.2 Automated Knowledge Base Construction 3](#_Toc34619610)

[2.3 Knowledge Queries 4](#_Toc34619611)

[3. Architecture 4](#_Toc34619612)

[3.1 Vocabularies Used 4](#_Toc34619613)

[3.1.1 Reused Vocabularies 4](#_Toc34619614)

[3.1.2 Our Schema 5](#_Toc34619615)

# Introduction

Knowledge graphs are information represented in the form of graphs where nodes are entities and the relationships between them are the edges. The fact that they help form better connections between data helps machines to process and connect loads of information for better results. Knowledge graphs are the foundation for artificial intelligence and are the main tool that is required to build intelligent agents such as chatbots that help answer questions based on the user’s input. Each fact in a knowledge graph is represented as a triple (subject, predicate, object) which are interconnected with each other to form the whole graph.

In this project, we create a knowledge graph that stores the information about Universities and its academic details. Here, we concentrate on information about Concordia University alone. The goal of this project is to build a knowledge graph in such a way that it can answer questions about the University through SPARQL queries. This project is the first step of building an intelligent agent that is capable of replying to questions related to the university which would be further done in our next project. Knowledge graphs are created from the data extracted from the web. Here, we take information about the courses, topics related to it and further details regarding it from the Concordia University websites and its open databases.

We have constructed the University knowledge graph using RDF and RDFS standard in turtle format. The graph is built to store information about universities along with its DBpedia entry URI. But since we restrict our project to just one university, the graph stores information about all the courses offered by Concordia University and the topics that are covered in each course. It also stores student information such as name, email etc. For testing purposes, we have used dummy information for ten students and their course history.

# Description

## University Knowledge Graph

The University graph is stored in the form of RDF schema and contains mainly five components, namely, ‘University’, ‘Course’, “Topic’ and ‘Student’ and ‘Course Grade’. These are stored in the form of the classes. Each of these components has attributes that store more information about them.

### University

This contains the name and the DBpedia URI for the given university. Since we use only Concordia University in our project, this has only once instance.

### Course

This contains the name of the course, course identifier and the description for the course.

### Topic

This contains the name of the topic and the DBpedia URI entry for the topic.

### Student

This contains the name of the student; first name and last name, ID number and email address of the student.

### Course Grade

This contains a course class instance along with the grade that the student scored for that course.

The RDF Schema for the University graph is stored in Turtle format, in the files universityKG.ttl and DataGraph.ttl. We first created the universityKG.ttl as the base graph which stores the definitions for the classes described above

## Automated Knowledge Base Construction

In this step, we have first generated .csv files that stores information about the instances.

*Courses.csv* stores the data related to courses and its properties.

*Grades.csv* stores the courses and the grade scores by each student.

*Student.csv* stores information about students such name, email, id etc.

*Topics.csv* stores information about Topics.

*Universities.csv* stores information about universities and its DBpedia entries.

These data was scraped from Concordia University Graduate/Under Graduate websites. We have scraped the data from over 99 URLs which gave information about the courses that are offered by the University, along with its course code and course description. We have found 2742 courses offered by Concordia from all the Departments. We have used Beautiful Soup to extract specific data from the Webpage, specifically under the “courses” section, such as course name, course code and course description. We then used python spotlight library which links to the DBpedia spotlight to find the Topics covered under a topic. This finds the DBpedia URI for the topics which are found inside the course description.

## Knowledge Queries

We use SPARQL queries to ask and get information from the RDF Knowledge graph. We have tested our knowledge graph for various queries. The queries are run on python using rdflib but we have also tested our queries on Apache Jena Fuseki.

# Architecture

In this section, we describe the architecture of our RDF schema in details.

## Vocabularies Used

### Reused Vocabularies

#### RDFS Schema

*rdfs:*[*http://www.w3.org/2000/01/rdf-schema#*](http://www.w3.org/2000/01/rdf-schema#)*:*

rdfs:label This is used to provide a human readable name for all the entities in our KG.

rdfs:comment A description of the subject resource.

rdfs:domain Domain of the property defined

rdfs:range Range of the property defined

rdfs:Class A class

#### RDF Schema

*rdf:* [*http://www.w3.org/1999/02/22-rdf-syntax-ns#*](http://www.w3.org/1999/02/22-rdf-syntax-ns#)

rdf:Property

#### XML Schema

*xsd:* [*http://www.w3.org/2001/XMLSchema#*](http://www.w3.org/2001/XMLSchema#)

*xsd:string*

#### FOAF Vocabulary

*foaf:* [*http://xmlns.com/foaf/0.1/*](http://xmlns.com/foaf/0.1/)

*foaf:name* This is used for giving name for entities such as course, topic and university.

*foaf:givenName* This

*foaf:familyName*

*foaf:mbox*

*foaf:Person*

#### DBpedia Property

*dbp:* [*http://dbpedia.org/property/*](http://dbpedia.org/property/)

*dbp:score* Connects the Grade for a student as a string literal

*dbp:id* Connects the Identity number for a student

#### DCMI Metadata Terms

*dc:* [*http://purl.org/dc/elements/1.1/*](http://purl.org/dc/elements/1.1/)

*dc:source*

*dc:subject*

*dc:identifier*

*dc:description*

### Our Schema

#### ISP

*isp:* [*http://intelligentsystemproj1.io/schema#*](http://intelligentsystemproj1.io/schema#)

*isp:University* Class for a University

*isp:Course* Class for a Course

*isp:Topic* Class for a Topic

*isp:Student* Class for a student

*isp:CourseGrade* Class for courses with it’s grade for a student

*isp:studiesAt* Property that links the university that the student studies at

*isp:tookCourse* Property that links the Courses completed by a student

*isp:hasPart* Property that links the Topics covered for a Course

*isp:coversCourse* Property that links the Courses covered in a University

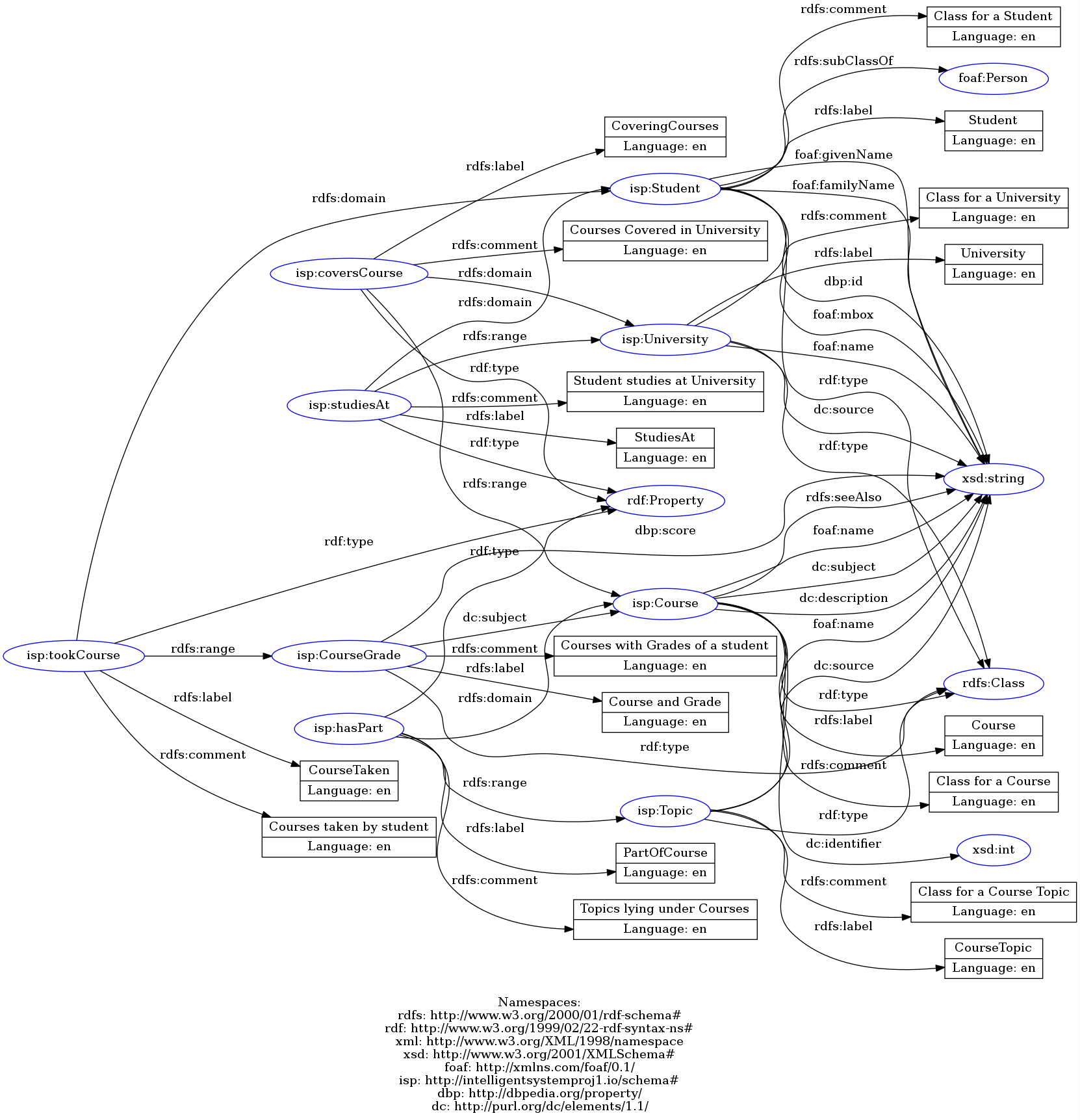


Figure 1: Shows the RDF graph visualisation for the base knowledge graph.

## SPARQL Queries