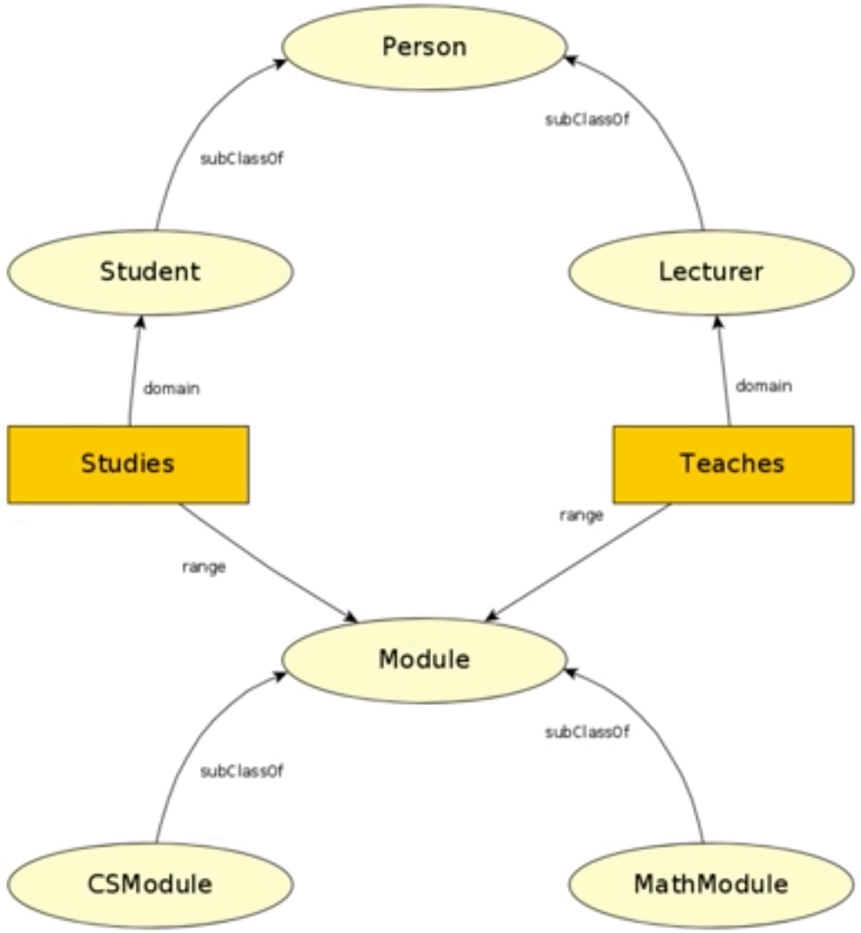
University Ontology

Reference:
https://www.youtube.com/watch?v=R9ERlUgvgwM&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1

Graph Diagram of University Ontology

https://www.youtube.com/watch?v=R9ERlUgvgwM&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=1



Class Hierarchy

https://www.youtube.com/watch?v=g5IGHpCSIow&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=2

- Create two base-classes and two sub-classes under each
  - Module
    - CSModule
    - MathModule
  - Person
    - Student
    - Lecturer
- Make each subclass as ‘Disjoint’ with the other subclass
  - To avoid overlapping between classes or to avoid multiple inheritance
  - Subclass, instance, or an individual can only be instance of one class

Object Properties

https://www.youtube.com/watch?v=wCsP36wFxdA&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=3

- Describes relationship between two instances (individuals) of classes (objects/instances)
- Create two object properties
  - Studies
  - Teaches
- Each property has a domain and range
  - Teaches
    - Domain: Lecturer
    - Range: Module
  - Studies
    - Domain: Student
    - Range: Module

Data Properties

https://www.youtube.com/watch?v=BaepddOkv4g&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=4

- Describes relationship between instances (individuals) and data values
  - Give CSModule a name, number etc.
  - Give Lecturer a staff id, first name, last name, phone num, email etc.
  - Give Student a student id, first name, last name, email etc.
- Each data property has a domain and range

• first_name
  • Domain: Person
  • Range: String
• last_name
  • Domain: Person
  • Range: String
	• staff_id
	  • Domain: Lecturer
	  • Range: Integer
	• student_id
	  • Domain: Student
	  • Range: Integer

Run Reasoner

- To check if everything make sense

Creating Individuals

https://www.youtube.com/watch?v=2UDX2Ho8ZEg&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=5

- Click on ‘individual’ tab and create a new individual (instance)

- CS101
  - Type: CSModule
- CS103
  - Type: CSModule
- M201
  - Type: MathModule
- M204
  - Type: MathModule
- Lecturer1
  - Type: Lecturer
- Lecturer2
  - Type: Lecturer
- Student1
  - Type: Student
- Student2
  - Type: Student

Data property assertions

- Lecturer1
  - first_name
    - Type: String
    - Value: Smith
  - last_name
    - Type: String
    - Value: Chess
  - staff_id:
    - Type: Integer
    - Value: 417686
- Student1
  - first_name
    - Type: String
    - Value: Joseph
  - last_name
    - Type: String
    - Value: Malcom
  - staff_id:
    - Type: Integer
    - Value: 200202

Object property assertions

- Lecturer1
  - teaches
    - Value: CS103
    - Value: M201
- Student1
  - Studies
    - Value: M204
    - Value: CS103

Upload University Ontology

https://www.youtube.com/watch?v=QY9M_j2Ta14&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=6

- User ‘scp’ to copy or upload the ontology file to a server
- One can actually commit to their local git repository

Running SPARQL Queries

https://www.youtube.com/watch?v=JZp70uFsZS0&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=7
https://www.youtube.com/watch?v=0zUos1zWB5k&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=8

- Download Apache Jena Fuseki distribution and run the server
  - Start ‘fuseki-server’ (./fuseki-server --update --mem /ds)
  - Fuseki server is now running on port 3030
- Open a browser and load Fuseki server page (http://localhost:3030)
  - Go to ’control panel’
  - Select the ‘/ds’ dataset
  - Upload the ontology file we created (upon success, it would show 68 triples)
  - Go back to the ‘control panel’ and select the ‘/ds’ dataset
  - Run the SPARQL queries
    - Select * {?x ?y ?z}
    - Prefix uni: <url_of_univ_ontology> select * { ?student uni:studies uni:M204}
    - Prefix uni: <url_of_univ_ontology> Prefix rdfs: <url_of_rdfs_ontology> select ?class where { ?class rdfs:subclassof uni:Person}

Create Sports Ontology

https://www.youtube.com/watch?v=Pn7oiDrtHmc&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=9

- Two main classes
  - FavoriteSport
    - IndoorSports
    - OutdoorSports
  - Person
    - Lecturer
    - Student
- Add object property
  - hasFavoriteSports
    - Domain: Person
    - Range: FavoriteSport
- Individuals
  - IndoorSport
    - TableTenis
  - OutdoorSport
    - Rugby
  - Person
    - Lecturer1
    - Student1
- Object Property Assertion
  - Lecturer1
    - FavoriteSport
      - TableTenis
  - Student1
    - FavoriteSport
      - Rugby

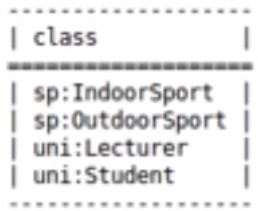
Upload Sport Ontology

- Upload the ‘sport’ ontology alongside the university ontology so that we can query both together

Running SPARQL Queries Together

https://www.youtube.com/watch?v=U_Sf-RJAXfs&list=PLea0WJq13cnAfCC0azrCyquCN_tPelJN1&index=10

- Download Apache Jena Fuseki distribution and run the server
  - Start ‘fuseki-server’ (./fuseki-server --update --mem /ds)
  - Fuseki server is now running on port 3030
- Open a browser and load Fuseki server page (http://localhost:3030)
  - Go to ’control panel’
  - Select the ‘/ds’ dataset
  - Upload the sport ontology file we created (upon success, it would show 34 triples)
  - Go back to the ‘control panel’ and select the ‘/ds’ dataset
  - Run the SPARQL queries
    - Select * {?x ?y ?z}
    - Prefix sp: <url_of_sport_ontology>
      Prefix uni: <url_of_univ_ontology>
      Prefix rdfs: <url_of_rdfs_ontology>
      select ?class where { {?class rdfs:subclassof sp:FevoriteSport} UNION {?class rdfs:subclassof uni:Person}}
  - Output is as shown in the figure on top right



class
sp:IndoorSport
sp:OutdoorSport
uni:Lecturer
uni:Student

Thank you!