

### Problem 3.

Design a family ontology in a manner that it contains only brotherhood, sisterhood, fatherhood and motherhood relationships between its members; then define grandfather, grandmother, aunts, uncles and cousin classes and all possible relationships like 'grandfather is.'. Finally by using the available Reasoner in Protege find all grandfather, grandmother, grandchild, aunts, nephew, niece, uncles, relationships among members in your ontology.

## Problem 1.

Define a RDFS file for a hospital that includes three rooms, six doctors, eight patients, and four employees. Also each room has atleast one patient and utmost three patients. Then transform your file with IsaVis tool to graphical model.

---

## Problem 2.

For this part, open swpub.rdf schema file in Protege then write and execute the following queries with SPARQL language. Attach queries and the output of them in your report. <sup>1</sup>

1. Retrieve the titles of all papers that are "articles in a periodical."
2. For all papers written between 2001 and 2005, inclusive, retrieve the title, human-friendly name (i.e., rdfs:label) of the type (e.g., article, book chapter, etc.) and year of the paper.
3. Find all papers that appear in a venue with "ISWC" in the name (using partial string matching), and return a list of title, venue, year and topic label (i.e., not the URI of the topic), sorted by year in descending order, followed by topic in ascending order (for those papers in the same year).
4. For each author, retrieve their name and a count of the number of papers they have authored. Sort the result by author name. You should assume that two authors are the same if and only if their full names match exactly. For sorting purposes, just use the name as it appears, do not worry about first name or last name distinctions. Hint: You need to use specific features of SPARQL 1.1 to do this.

1. Define the following types in an XSD (XML Schema Document) file using the XML Schema primitive types:

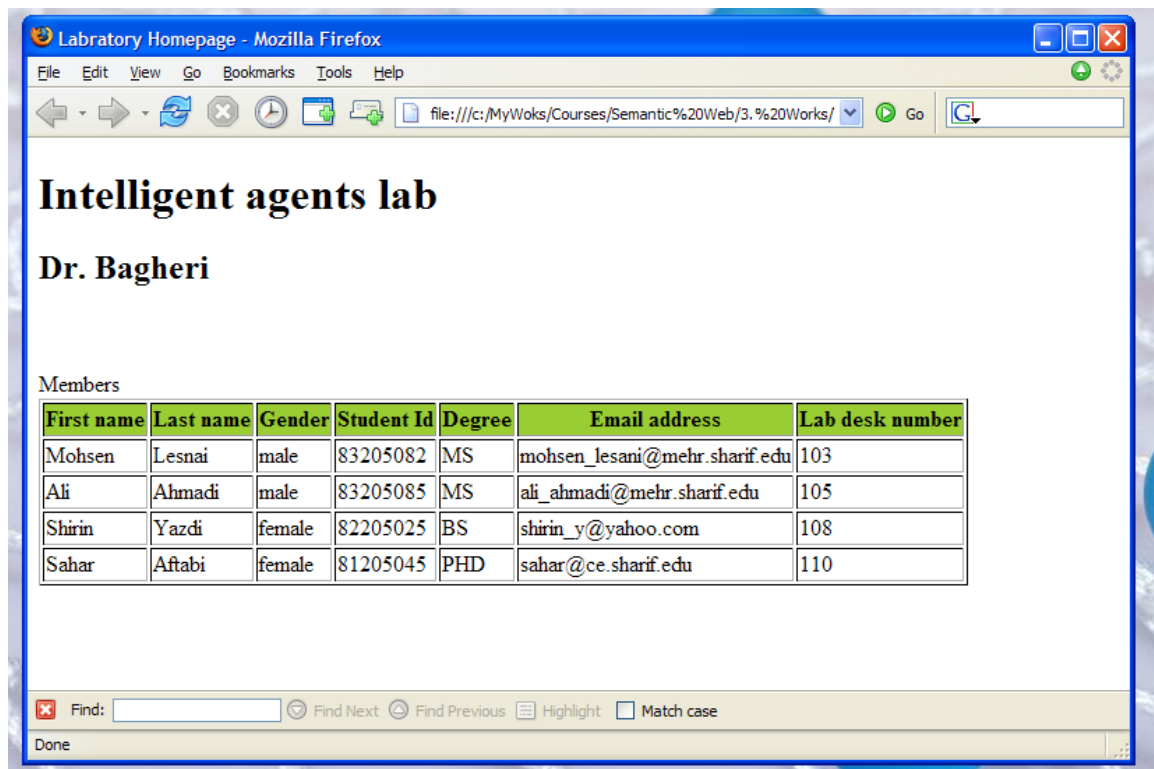
String, Int(as positive integer numbers), Date, Id(as a string of length three of digits), Gender(“female” or “male”), Degree(“BS”, “MS” or “PHD”), StudentId(as string of length eight of digits), EmailAddress(as a string of alphabets and dot, a @ and another string of alphabets and dot).

Define the types in the following scenario using the just defined types:

A Laboratory has an Id, a name, an advisor (name) and some Members. A Member is a student that has a first name, a last name, a gender, a student Id, a degree, an email address, a desk Id and can work on any number of projects. A Project has a title, a start date and a finish date.

2. Define an XML file presenting information about your lab according to the defined schema.

3. Define three XSL (EXtensible Stylesheet Language) files for transforming the XML file to XHTML files of laboratory homepage and two student homepages as follows:



Labratory Homepage - Mozilla Firefox

file:///c:/MyWoks/Courses/Semantic%20Web/3.%20Works/

# Intelligent agents lab

Dr. Bagheri

Members

First name	Last name	Gender	Student Id	Degree	Email address	Lab desk number
Mohsen	Lesnai	male	83205082	MS	mohsen_lesani@mehr.sharif.edu	103
Ali	Ahmadi	male	83205085	MS	ali_ahmadi@mehr.sharif.edu	105
Shirin	Yazdi	female	82205025	BS	shirin_y@yahoo.com	108
Sahar	Aftabi	female	81205045	PHD	sahar@ce.sharif.edu	110

Find:  Find Next Find Previous Highlight Match case

Done

Homepage - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

file:///c:/MyWoks/Courses/Semantic%20Web/3.%20Works/

## Sahar Aftabi

Student Id: 81205045  
Gender: female  
Degree : PHD  
Email Address: sahar@ce.sharif.edu  
Lab Desk No.: 110

Projects:

Title	Start Date	Finish Date
Semantic Agents on the web	2004-9-19	

Find:  Find Next Find Previous Highlight Match case

Done

Homepage - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

file:///c:/MyWoks/Courses/Semantic%20Web/3.%20Works/

## Ali Ahmadi

Student Id: 83205085  
Gender: male  
Degree : MS  
Email Address: ali\_ahmadi@mehr.sharif.edu  
Lab Desk No.: 105

Projects:

Title	Start Date	Finish Date
Multiagent Systems	2004-5-11	2005-1-5
Artificial Life	2005-5-11	

Find:  Find Next Find Previous Highlight Match case

Done



2.الف) داده های RDF زیر را که دارای ساختار سه تایی (N3) هستند را در قالب فایل data.rdf با ساختار RDF/XML ذخیره و تحویل دهید:

```
<http://example.org/#SW><http://example.org/#OfferedBy><http://example.org/#Dr.Abolhassani>
<http://example.org/#DM><http://example.org/#OfferedBy><http://example.org/#Dr.Abolhassani>
<http://example.org/#sayyadi><http://example.org/#TAOf><http://example.org/#SW>
<http://example.org/#bagheri><http://example.org/#TAOf><http://example.org/#SW>
<http://example.org/#sayyadi><http://example.org/#enrolledIn><http://example.org/#DM>
<http://example.org/#bagheri><http://example.org/#enrolledIn><http://example.org/#Database2>
<http://example.org/#std1><http://example.org/#enrolledIn><http://example.org/#SW>
<http://example.org/#std2><http://example.org/#enrolledIn><http://example.org/#DM>
<http://example.org/#sayyadi><http://example.org/#fullName>"Hassan Sayyadi"
<http://example.org/#bagheri><http://example.org/#fullName>"Babak Bagheri"
<http://example.org/#std1><http://example.org/#fullName>"Student1"
<http://example.org/#std2><http://example.org/#fullName>"Student2"
<http://example.org/#SW><http://example.org/#unit> 3
<http://example.org/#DM><http://example.org/#unit> 3
<http://example.org/#SW><http://example.org/#name> "Semantic Web"
<http://example.org/#DM><http://example.org/#name> "Data Mining"
```

ب) query های لازم برای بازیابی اطلاعات زیر را در قالب RDQL بنویسید:

- پیدا کردن نام کامل (fullName) تمام دانشجویان دکتر ابولحسنی (Dr.Abolhassani).
- پیدا کردن نام و تعداد واحد (unit) دروسی که کمک مربیان دکتر ابولحسنی در آنها ثبت نام شده اند (enrolledIn).
- ج) از یکی از ابزار های موجود مانند Jena استفاده کرده و نتیجه اجرای query های قسمت (ب) را گرفته و در فایل result.txt ذخیره و تحویل دهید.

توضیحات: