Semantic Web Engineering

Exercise 3 - SPARQL
Due: 12:00 am, November 4, 2010 reif@ifi.uzh.ch
Required Reading: RDF Primer http://www.w3.org/TR/rdf-primer/

Use the RDF Primer and the lecture slides to answer the following questions. Be prepared to present your solutions in the lecture on November 5, 2010.

Following tools might be helpful for the exercise:
- ARQ http://seal.ifi.uzh.ch/semweb
- Twinkle http://www.ldodds.com/projects/twinkle/

1. SPARQL Queries

Use the follofing RDF graph to define your queries. You find the graph online under `cd.ttl`.

```
@prefix store:    <http://example.com/store#> .
@prefix mytunes:  <http://mytunes.org/music#> .
@prefix artist:   <http://mytunes.org/artist#> .

store:cd1    mytunes:artist  artist:morissette ;
             mytunes:album   "alanis unplugged" ;
             mytunes:year    1999;
             store:price     20 ;
             store:quantity  3 .

store:cd2    mytunes:artist  artist:morissette ;
             mytunes:album   "Jagged Little Pill" ;
             store:price     15 ;
             store:quantity  1 .

store:cd3    mytunes:artist  artist:pinkfloyd ;
             mytunes:album   "The Wall" ;
             mytunes:year    1994;
             store:price     25 ;
             store:quantity  5 .

store:cd4    mytunes:artist  artist:nirvana ;
             mytunes:album   "Nevermind" ;
             store:price     10 ;
             store:quantity  10 .

store:cd5    mytunes:artist  artist:nirvana ;
             mytunes:album   "Nirvana unplugged" ;
             mytunes:year    1993;
             store:quantity  4 .

store:cd6    mytunes:artist  artist:pinkfloyd ;
             mytunes:album   "Dark Side of the Moon" ;
             store:price     18 ;
             store:quantity  4 .


artist:morissette mytunes:name  "Alanis Morissette" .

artist:pinkfloyd  mytunes:name  "Pink Floyd" .
```

```
artist:nirvana    mytunes:name  "Nirvana" .
```

Query 1

Your query should return the *title of the album* and the *price* of all CDs of the artist with the name `"Alanis Morissette"` and is available at least 2 times (`store:quantity`).

Query 2

Your query should return the *title of the album*, the *price*, and the *year* of **all** CDs in the RDF graph (i.e. there should be a line in the search result for every CD even if this line cannot be filled completely).

Query 3

Your query should return a table with the *Name of the artist* and the sum of all CDs of this artist that are on stock in the store.

Query 4

Your query should return an RDF graph which contains for **every** CD in the original graph triples that follow the the structure below:

```
store:cd1 msstore:album   "alanis unplugged" ;
          msstore:artist  "Alanis Morissette" ;
          msstore:year    1999 .
```

The additional prefix is defined as:

```
PREFIX  msstore:   <http://msstore.com/money#>
```

If the year value is missing in the original graph, the corresponding triple will also be missing in the graph of the search result.

2. Root of a Class Hierarchy

Write a SPARQL query that returns the root of the class hierarchy. Use a) only features of SPARQL 1.0 and b) features of SPARQL 1.1. Useful hints for the SPARQL 1.0 query you might find in the Operator section of the W3C SPARQL Query recommendation. (http://www.w3.org/TR/rdf-sparql-query/#OperatorMapping)
You can use the following graph as an example class hierarchy. You find the graph online under `vehicle.owl`.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/Ontology1228297369.owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1228297369.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Van">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="MotorVehicle"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="NonMotorVehicle">
    <rdfs:subClassOf>
```

```
      <owl:Class rdf:ID="Vehicle"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Bicycle">
    <rdfs:subClassOf rdf:resource="#NonMotorVehicle"/>
  </owl:Class>
  <owl:Class rdf:ID="Truck">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#MotorVehicle"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#MotorVehicle">
    <rdfs:subClassOf rdf:resource="#Vehicle"/>
  </owl:Class>
  <owl:Class rdf:ID="Limousine">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </owl:Class>
</rdf:RDF>
```