COMP 872 Social Semantic Web Fall 2014 Assignment 4—Solutions

Due Friday, October 3

1 (5 pts.). Write a Jena application that creates a vCard that describes you and write out the model in N3 format. For creating vCard representations in general, see **IntroRDF_and_JenaEsterline.ppt**, slides 2-7. For writing an RDF model to file, see slides 28-30 of the same.

Solution

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.vocabulary.VCARD;

public class prob1 {
   public static void main(String [] args) throws IOException{
      FileWriter outFile = new FileWriter("william.n3");
      PrintWriter out = new PrintWriter(outFile);
      String personURI = "http://somewhere/LiamNick";
      String givenName = "William";
      String familyName = "Nick";
      String fullName = "Liam" + " " + familyName;
      // create an empty Model
      Model model = ModelFactory.createDefaultModel();
      // create the resource and add the properties cascading style
      Resource johnSmith
         = model.createResource(personURI)
            .addProperty(VCARD.FN, fullName)
            .addProperty(VCARD.N,
               model.createResource()
                  .addProperty(VCARD.Given, givenName)
                  .addProperty(VCARD.Family, familyName));
      model.write(out, "N3");
   }
}
```

2 (7 pts.). The following is a partial listing of file **cd.n3**, which you download from the location in Blackboard where you downloaded this assignment. (Additional CDs will be added shortly.)

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix cd:      <http://www.recshop.fake/cd#> .
@prefix cdterms: <http://www.recshop.fake/terms#> .
@prefix vcard:   <http://www.w3.org/2001/vcard-rdf/3.0#> .
cd:7 cdterms:artist _:bobdylan.
cd:7 cdterms:title "Empire Burlesque";
   cdterms:country "US";
   cdterms:company "Columbia";
   cdterms:price "10.90";
   cdterms:year "1985".
cd:8 cdterms:artist _:bonnietyler.
cd:8 cdterms:title "Hide your Heart";
   cdterms:country "UK";
   cdterms:company "CBS Records";
   cdterms:price "9.90";
   cdterms:year "1988".
_:bobdylan vcard:FN "Bob Dylan";
   vcard:N
    [ vcard:Family "Zimmerman";
      vcard:Given "Robert"].
_:bonnietyler vcard:FN "Bonnie Tyler";
   vcard:N
    [ vcard:Family "Hopkins";
      vcard:Given "Gaynor"].
```

Write a SPARQL query on an ARQ file (to be executed from the command line) that uses **cd.n3 and** queries for all CDs whose artist is Bob Dylan.

Solution

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX cdterms: <http://www.recshop.fake/terms#>
PREFIX cd: <http://www.recshop.fake/cd#>

SELECT ?title
WHERE { ?x cdterms:artist ?y .
        ?x cdterms:title ?title .
        ?y vcard:FN "Bob Dylan" .}
```

3 (8 pts.). The following is a listing of file **first_dataN.n3**, which you download from Blackboard.

```
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vCard:   <http://www.w3.org/2001/vcard-rdf/3.0#> .
<http://somewhere/RebeccaSmith/>
      vCard:FN "Becky Smith" ;
      vCard:N [ vCard:Family "Smith" ;
                vCard:Given "Rebecca"
              ] .
<http://somewhere/MattJones/>
      vCard:FN "Matt Jones" ;
      vCard:N [ vCard:Family "Jones" ;
                vCard:Given "Matthew"
<http://somewhere/SarahJones/>
      vCard:FN "Sarah Jones" ;
      vCard:N [ vCard:Family "Jones" ;
                vCard:Given "Sarah"
              ] .
<http://somewhere/JohnSmith/>
      vCard:FN "John Smith" ;
      vCard:N [ vCard:Family "Smith" ;
                vCard:Given "John"
              ] .
<http://somewhere/JohnSmith/>
      vCard:FN "Johnny Jones" ;
      vCard:N [ vCard:Family "Jones" ;
                vCard:Given "John"
              ] .
```

Write a Java/Jena/SPARQL program that inputs **first_dataN.n3** and outputs

- the full name of all those with family name “Jones” and
- the family name of all those with given name “John”

Solution

```
import java.io.IOException;
import java.io.InputStream;
import java.lang.*;
import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.QuerySolution;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.util.FileManager;
```

```
public class JenaSemanticWeb {
    public static void main(String[] args) {
      Model model = ModelFactory.createDefaultModel();
      // use the FileManager to find the input file
      InputStream in = FileManager.get().open("vc-db-1.rdf");
      // read the RDF/XML file
      model.read(in, "N3");
      //Now add query
     StringBuffer queryStr = new StringBuffer();
     queryStr.append(
         "SELECT DISTINCT ?fname "
       + "WHERE { ?x <http://www.w3.org/2001/vcard-rdf/3.0#Family> \"Jones\". \n"
       + " ?uri <http://www.w3.org/2001/vcard-rdf/3.0#N> ?x;"
       +" <http://www.w3.org/2001/vcard-rdf/3.0#FN> ?fname.}");
      Query query = QueryFactory.create(queryStr.toString());
      QueryExecution qexec = QueryExecutionFactory.create(query,model);
      System.out.println("First Query Results");
      try {
        ResultSet response = qexec.execSelect();
        while( response.hasNext()){
          QuerySolution soln = response.nextSolution();
          RDFNode x = soln.get("?fname");
          if( x != null ){
            System.out.println( x.toString() );
          }
          else
            System.out.println("Nothing found!");
        }
      } finally { qexec.close();}
      String queryString = "SELECT ?family "
         +"WHERE { ?x <http://www.w3.org/2001/vcard-rdf/3.0#Given> \"John\". "
         + "?x <http://www.w3.org/2001/vcard-rdf/3.0#Family> ?family.}";
      System.out.println("\nSecond Query Results");
      Query query2 = QueryFactory.create(queryString);
      QueryExecution qexec2 = QueryExecutionFactory.create(query2,model);
      try {
        ResultSet response = qexec2.execSelect();
        while( response.hasNext()){
          QuerySolution soln = response.nextSolution();
          RDFNode x = soln.get("?family");
          if( x != null ){
            System.out.println( x.toString() );
          }
          else
            System.out.println("Nothing found!");
        }
      } finally { qexec.close();}
    }
}
```

For submission, zip the N3 and Java files into a single file.