

Übungen zu Semantic Web

Aufgabenblatt 6

Beispiellösungen

Prof. Dr. Steffen Staab, Dipl.-Inform. Bernhard Schüler
Fachbereich Informatik, Universität Koblenz-Landau

Präsentation ab 15.01.2008

Auf der Internetseite zur Veranstaltung finden sie die Dateien Blatt2_Bsp_1.zip und Blatt2_Bsp_2.zip. Diese enthalten zwei unterschiedliche RDF-Schemata in den Dateien Rechnung.turtle und Speisekarte.turtle einerseits sowie rdfs.turtle andererseits. Außerdem enthalten sie zwei unterschiedliche relationale Schemata in den Dateien Relationales Modell Speisekarte_Rechnung 'Altes Brauhaus'.xls einerseits und relational.sql andererseits.

1 Anfragesprachen und Modellierung

Versuchen Sie aus Daten zu jedem der vier Schemata die folgenden Informationen mit jeweils einer Anfrage in SPARQL oder SQL zu gewinnen:

1. Nummer und Namen aller Getränke.

Angenommen vor dem Ausführen der SPARQL-Anfrage würden alle Tripel hinzugefügt, die mit Hilfe von syntaktischen Ableitungsregeln hergeleitet werden können: Gibt es andere SPARQL-Anfragen, die nun das gewünschte Ergebnis liefern?

Beispiel 1, SPARQL:

```
PREFIX men: <http://www.altesbrauhaus-koblenz.de/menue#>
SELECT ?nr ?name
WHERE {
  _:a men:hatArtikelnummer ?nr .
  _:a men:hatArtikelbezeichnung ?name .
  _:a men:hatMenge _:b .}
```

mit RDFS-Folgerung folgt der Typ von _:a aus der Property men:hatMenge:

```
PREFIX men: <http://www.altesbrauhaus-koblenz.de/menue#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
SELECT ?nr ?name
WHERE {
  _:a men:hatArtikelnummer ?nr .
```

```
_:a men:hatArtikelbezeichnung ?name .
_:a rdf:type men:getraenk .}
```

Beispiel 2, SPARQL:

```
PREFIX vsw: <http://isweb.uni-koblenz.de/Teaching/WS0708/SemWeb07#>
SELECT ?nr ?name
WHERE {
  ?a vsw:nmNr ?nr .
  ?a vsw:nmName ?name .
  ?a vsw:menge _:b .}
```

Oder (je nach Daten RDFS-Folgerung nötig):

```
PREFIX vsw: <http://isweb.uni-koblenz.de/Teaching/WS0708/SemWeb07#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?nr ?name
WHERE {
  ?a vsw:nmNr ?nr .
  ?a vsw:nmName ?name .
  ?a rdf:type vsw:Getraenk .}
```

Beispiel 1, SQL:

```
SELECT Artikel.Artikel_Bezeichnung, Artikel.Artikel
FROM Artikel, Kategorien
WHERE Artikel.Kategorie = Kategorien.Kategorie AND Kategorien.Artikeltyp="2";
```

Beispiel 2, SQL: Ob ein Nahrungsmittel ein Getränk ist, lässt sich bestenfalls an den Attributen menge und einheit ablesen.

```
SELECT nr, name
FROM nahrungsmittel as n, speisekarte as s
WHERE n.nr=s.nahrungsmittelnr AND n.menge IS NOT NULL;
```

2. Eine Liste aller Nahrungsmittel mit den jeweils enthaltenen Zusatzstoffen.

Beispiel 1, SPARQL:

```
PREFIX men: <http://www.altesbrauhaus-koblenz.de/menue#>
SELECT ?nm ?zus
WHERE {
  ?nm men:hatArtikelnummer _:a .
  OPTIONAL {?nm men:hatInhaltsstoff ?zus .} }
```

Beispiel 2, SPARQL:

```
PREFIX vsw: <http://isweb.uni-koblenz.de/Teaching/WS0708/SemWeb07#>
SELECT ?nm ?zus
WHERE {
  ?nm rdf:type vsw:Nahrungsmittel .
  OPTIONAL {?nm vsw:hatZusatzstoff ?zus .} }
```

Beispiel 1, SQL:

```
SELECT *
FROM Artikel LEFT OUTER JOIN Artikel_Inhaltstoffe;
ON Artikel.Artikel=Artikel_Inhaltstoffe.Artikel ;
```

Beispiel 2, SQL:

```
SELECT *
FROM   nahrungsmittel LEFT OUTER JOIN enthaeltzusatz
      ON nahrungsmittel.nr=enthaeltzusatz.nr ;
```

3. Alle Positionen von solchen Rechnungen, deren Summe weniger als 20 EUR beträgt.

Beispiel 1, SPARQL:

Zunächst einmal sind die Daten in Rechnung.turtle fehlerhaft:

```
rec:rechnung rec:hatBezeichnung "Rechnung";
              rec:hatRechnungsnr "2007090";
              ...
```

macht im Zusammenhang mit rec:rechnung rdf:type rdfs:Class keinen Sinn. Wir ersetzen oben rec:rechnung mit rec:rech1.

Dann bereitet die Verwendung von rdf:List Schwierigkeiten. Man muss eine Serie von in der Länge wachsenden Anfragen der folgenden Form stellen:

```
PREFIX men: <http://www.altesbrauhaus-koblenz.de/menue#>
SELECT ?pos1 ?pos2 ...
WHERE{ ?rech rec:hatPreis ?preis .
       FILTER (?preis<20)
       ?rech rec:hatPosition _:p1 .
       _:p1 rdf:first ?pos1 .
       _:p1 rdf:rest _:p2 .
       _:p2 rdf:first ?pos2 .
       _:p2 rdf:rest _p3 .
       ...
}
```

Beispiel 2, SPARQL:

```
PREFIX vsw: <http://isweb.uni-koblenz.de/Teaching/WS0708/SemWeb07#>
SELECT ?pos
WHERE{ ?pos vsw:gehörtZuRechnung ?rech .
       ?rech vsw:rechSumme ?sum .
       FILTER (?sum < 20) }
```

Beispiel 1, SQL: Angenommen, in Bestellung_Artikel wäre die Anzahl stets eins, so wäre dies eine Lösung:

```
SELECT b.Bestellung_Nr
FROM   Bestellung b, Bestellung_Artikel ba
WHERE  b.Bestellung_Nr=ba.Bestellung_Nr AND (
      SELECT SUM(a.Preis_Euro)
      FROM   Artikel a
      WHERE  a.Artikel=ba.Artikel
      ) < 20;
```

```
SELECT Artikel
FROM Bestellung_Artikel
WHERE Bestellung_Artikel.Bestellung_Nr IN
  (SELECT Bestellung_Artikel.Bestellung_Nr
   FROM Bestellung_Artikel, Artikel
   WHERE Bestellung_Artikel.Artikel = Artikel.Artikel
   GROUP BY Bestellung_Artikel.Bestellung_Nr
   HAVING SUM(Artikel.Preis_Euro*Bestellung_Artikel.Anzahl) < 20
  );
```

Beispiel 2, SQL:

```
SELECT *
FROM rechnung r, position p
WHERE p.rechnr=r.rechnr AND r.summe<20;
```

Was sind die Vor- und Nachteile der jeweiligen Modellierungen?