NSWI144 – Linked Data – Lecture 6 – 22th November 2011

SPARQL

Martin Svoboda

Faculty of Mathematics and Physics

Charles University in Prague

SPARQL Basics

- Query components
  - `PREFIX …`
    `SELECT | DESCRIBE | ASK | CONSTRUCT …`
    `FROM …`
    `WHERE { … }`
    `ORDER BY … LIMIT … OFFSET …`
- Graph patterns
  - Basic, group, optional, alternative
- Filters

Sample Data

- Schema
  - `@prefix i: <http://is.cuni.cz/is#> .`
  - Classes
    - Student, Teacher, Name, Address, Course, Registration, Event, Term, Room, Building, Department

Sample Data

- Student dataset
  - `@prefix s: <http://is.cuni.cz/students/> .`
  - `s:student14 i:idNumber "14" ;`
    `i:name [i:first "Martin" ;`
    `i:last "Svoboda" ] ;`
    `i:email "svoboda@ksi.mff.cuni.cz" .`
  - `s:student14 i:enroll [ i:course t:course17 ;`
    `i:term t:term1112W ;`
    `i:result "2" ] .`

Sample Data

- Teaching dataset
  - `@prefix t: <http://is.cuni.cz/teaching/> .`
  - `t:teacher25 i:name [i:first "Irena" ;`
    `i:last "Mlynkova" ] ;`
    `i:affiliation f:department8;`
  - `t:teacher25 i:teach [`
    `i:course t:course17 ; i:term t:term1112W ;`
    `i:day "MON" ; i:time "9:00" ;`
    `i:place f:roomS5 ; i:length "90" ; ] .`
  - `t:term1112W i:year "2011/12" ; i:period "W" .`
  - `t:course17 i:title "XML Technologies" ;`
    `i:code "NPRG036" ;`
    `i:prerequisity t:course22 .`

Sample Data

- Faculty dataset
  - `@prefix f: <http://is.cuni.cz/faculty/> .`
  - `f:department8 i:name "Dep. of Soft. Eng." ;`
    `i:code "DSE" .`
  - `f:roomS5 i:number "S5" ;`
    `i:building f:buildingS ;`
    `i:capacity "116" .`
  - `f:buildingS i:address [`
    `i:street "Malostranske n. 25" ;`
    `i:city "Prague" ] .`

Assignment 1

- Select all courses with prefix *NSWI*.
  - Use *regex* function.
  - Return course titles and codes.

Assignment 1

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?code ?title
FROM <http://is.cuni.cz/teaching/>
WHERE {
  ?course i:code ?code; i:title ?title.
  FILTER ( regex(?code, "^NSWI") )
}
```

Assignment 2

- Select all courses that are taught on Mondays or Fridays.

Assignment 2

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?code ?title
FROM <http://is.cuni.cz/teaching/>
WHERE {
  ?course i:code ?code; i:title ?title.
  ?event i:course ?course; i:day ?day.
  FILTER ( (?day = "MON") || (?day = "FRI") )
}
```

Assignment 3

- Select all courses that are not taught on Mondays and nor on Fridays.

Assignment 3

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?code ?title
FROM <http://is.cuni.cz/teaching/>
WHERE {
  ?course i:code ?code; i:title ?title.
  OPTIONAL {
    ?event i:course ?course; i:day ?day.
    FILTER ( (?day = "MON") || (?day = "FRI") )
  }
  FILTER ( ! bound(?event) )
}
```

Negation

- EXISTS
  - `Pattern1 FILTER [NOT] EXISTS { pattern2 }`
  - Tests for the presence of a pattern
  - Does not generate any additional bindings
- MINUS
  - `Pattern1 MINUS { pattern2 }`
  - Evaluates both operands and preserves only solutions of the left-hand pattern that are not compatible with solutions of the right-hand one

Assignment 4

- Find other equivalent query expressions for the problem from Assignment 3.
  - Use NOT EXISTS and MINUS.

Assignment 4

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?code ?title
FROM <http://is.cuni.cz/teaching/>
WHERE {
  ?course i:code ?code; i:title ?title.
  FILTER NOT EXISTS {
    ?event i:course ?course; i:day ?day.
    FILTER ( (?day = "MON") || (?day = "FRI") )
  }
}
```

Assignment 4

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?code ?title
FROM <http://is.cuni.cz/teaching/>
WHERE {
  ?course i:code ?code; i:title ?title.
  MINUS {
    ?event i:course ?course; i:day ?day.
    FILTER ( (?day = "MON") || (?day = "FRI") )
  }
}
```

Assignment 5

- Select all teachers that have conflicts in their teaching schedule.
  - A teacher has a conflict in his/her schedule, if we can find at least two schedule events with at least partially overlapping times.
  - Moreover, we need at least 60 minutes for transfers between different buildings and 10 minutes for breaks within the same building.

Assignment 5

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?first ?last
FROM <http://is.cuni.cz/teaching/>
FROM <http://is.cuni.cz/faculty/>

WHERE {
  ?teacher i:name [i:first ?first; i:last ?last];
           i:teach ?e1; i:teach ?e2.
```

▼

Assignment 5

▲

```
  ?e1 i:term ?s1; i:day ?d1; i:time ?t1;
      i:place [i:building ?b1]; i:length ?l1.
  ?e2 i:term ?s2; i:day ?d2; i:time ?t2;
      i:place [i:building ?b2].
  FILTER (
    (?e1 != ?e2) &&
    (?s1 = ?s2) && (?d1 = ?d2) && (?t1 <= ?t2) &&
    ( ((?b1 != ?b2) && (?t2-(?t1+?l1)<60)) ||
      ((?b1 = ?b2) && (?t2-(?t1+?l1)<10)) )
  )
}
```

Property Paths

- Motivation
  - Ability to match arbitrary length paths
    - Path of length 1 is an ordinary triple
    - Path of length 0 connects a graph node to itself
    - Matching cycles in the graph is possible
- Limitations
  - Variables can not be used inside paths
  - Cycles are considered at most once

Property Paths

- Allowed syntax
  - Sequences
    - `path1/path2`
  - Alternatives
    - `path1|path2`
  - Groups
    - `(path)`
  - Negation
    - `!item`
    - `!(item1|item2|…)`

Property Paths

- Allowed syntax
  - Occurrences
    - `path*`
    - `path+`
    - `path?`
    - `path{n,m}`
    - `path{n}`
    - `path{n,}`
    - `path{,m}`
  - Inverse paths (from object to subject)
    - `^path`

Assignment 6

- Find all courses that are pre-requisites or co-requisites for a course with code *NSWI144*.
  - Search these dependencies recursively.
  - Return course codes together with titles.
  - Order the list using course codes.

Assignment 6

```
PREFIX i: <http://is.cuni.cz/is#>

SELECT DISTINCT ?code ?title
FROM <http://is.cuni.cz/teaching/>

WHERE {
  ?s i:code "NSWI144".
  ?s (i:prerequisity|i:requisity)+ ?course.
  ?course i:code ?code; i:title ?title.
}

ORDER BY ASC(?code)
```

Aggregates

- Available functions
  - `COUNT`
  - `SUM, AVG, MIN, MAX`
- Query syntax
  - ```
    SELECT …
    FROM …
    WHERE …
    GROUP BY …
    HAVING …
    ```
    ```

Assignment 7

- Return average examination results for all students over their courses enrolled in a winter semester 2011/12.
  - Ignore enrollments with undefined results.
  - Order the list using computed averages.

Assignment 7

```
PREFIX i: <http://is.cuni.cz/is#>
PREFIX t: <http://is.cuni.cz/teaching/>

SELECT ?last ?first (AVG(?result) AS ?average)
FROM <http://is.cuni.cz/students/>
WHERE {
  ?stud i:name [ i:first ?first ; i:last ?last ] ;
        i:enroll [ i:term t:term1112W ;
                   i:result ?result ] .
}
GROUP BY ?first ?last
ORDER BY ?average, ?last, ?first
```