Universität Freiburg
Institut für Informatik
Prof. Dr. Peter Fischer
Lecuture Data Models and Query Languages
Summer Semester 2012

Due/Discussion **14.06.2012**
(June 7th is a holiday, therefore a “double” exercise)

Exercise Sheet 5+6

SPARQL and SPARQL semantics

Exercise 1 (SPARQL Understanding) 30 points

Consider the RDF database

D := { (_:P1,rdf:type,Person), (_:P1,name,”Pete"), (_:P1,age,”17"), (:P1,email,”pete@abc.com"),
(_:P2,rdf:type,Person), (_:P2,name,”John"), (_:P2,email,”john@abc.com"),
(_:P3,rdf:type,Person), (_:P3,name,”Sue"), (_:P3,age,”21"),
(_:P1,knows, _:P2), (_:P1,knows, _:P3), (_:P2,knows, _:P1), (_:P2,knows, _:P3) }.

Draw the RDF graph. Evaluate the following SPARQL queries and phrase their semantics in plain English as well as in the formal semantics described in the lecture

Assume that every of the following queries is preceded by the necessary namespace definitions and the Select * solution format.

a) { ?p rdf:type Person. ?p age ?age. FILTER (?age>20) }
b) { { ?p rdf:type Person. ?p name ?name. } OPTIONAL { ?p age ?age . } }
c) { { ?p rdf:type Person. ?p age ?age. } UNION { ?p rdf:type Person. ?p email ?email. } }
d) { { ?p rdf:type Person. OPTIONAL ?p email ?email. } FILTER (!bound(?email)) }

Verify your results using the Jena ARQ SPARQL engine. An installation instruction, the above RDF document D, and example query a) are provided at the exercise page of the lecture homepage.

Exercise 2 (SPARQL Queries) 20 points

Consider the RDF database D from the previous exercise. Specify the following requests as SPARQL queries and indicate the final results obtained when evaluating them on document D.

a) All pairs of distinct persons that have a common friend (i.e., it must hold that the intersection of persons they know is non-empty).
b) The names of all persons that know at least one person or are younger than 20 years. If present, the email address and, also if present, the age of this person should be included in the result.
c) Construct a new graph using the CONSTRUCT form that contains all persons (including their names) that know at least two persons.

Exercise 3 (SPARQL Semantics) 30 points

We write P1 ≡ P2 for two SPARQL queries P1, P2 if and only if P1 and P2 yield the same result on every possible RDF document.
Let A, B, and C be SPARQL graph patterns, i.e. they are composed of BGPs, Union, Join and Leftjoin.
For each of the following equivalences either prove that it holds or show (by counterexample) that the equivalence does not hold.
Assume that filter conditions are built from variables, IRIs, literals, the bound operator, equality and the logical connectives ;∧,∨;

a) Union(A;A) ≡ A
b) Leftjoin(A;A) ≡ A
c) Join(A;A) ≡ A
d) Union(A; Join(B;C)) ≡ Join(Union(A;B); Union(A;C))
e) Join(A; Union(B;C)) ≡ Union(Join(A;B); Join(A;C))
f) Leftjoin(A;B) ≡ Leftjoin(A; Join(A;B))
g) Leftjoin(Union(A;B);C) ≡ Union(Leftjoin(A;C); Leftjoin(B;C))
h) Let ?x be a variable that occurs in B. Filter(Leftjoin(A;B); bound(?x)) ≡ Join(A;B)
i) Let ?x be a variable that occurs in A and F and F be a filter condition.
Filter(Join(A;B); F) ≡ Join(Filter(A; F);B)
j) Let ?x be a variable that occurs in A and F and F be a filter condition.
Filter(Leftjoin(A;B); F) ≡ Leftjoin(Filter(A; F);B)

Exercise 4 (SPARQL/RDF Storage) 20 points

Consider the RDF database D from Exercise 1.

a) Provide the relational database instance that stores the RDF graph D according to the Vertical Partitioning scheme.
b) Translate the queries from Exercise 1 into SQL queries over the vertically partitioned scheme from part a).