1. Unit: SPARQL

Exercise 1.1 (SPARQL-Queries) Give SPARQL queries against `mondial.n3` that yield answers to the following questions:

- Name and population (ordered) of all countries that have more than 10.000.000 inhabitants.
- Name of all countries that have at least one city with more than 1.000.000 inhabitants.
- Name of all countries that have no city with more than 1.000.000 inhabitants.
- Name of all european countries that have no membership in the European Union.
- Abbreviations of all organizations whose headquarter is located in the capital of a member country (together with the names of the country and the city).

Exercise 1.2 (SPARQL Optional)

Give a SPARQL query against `mondial.n3` that yield answers to the following question:

- For each country, give the name, and the population.
  If more than 1/4 of the population are living in its capital, give also the name and the population of the capital.
- Give the same query in SQL (against relational Mondial) and in XML/XQuery (against mondial.xml).

Exercise 1.3 (SPARQL Formal Semantics) Consider the SPARQL Formal Semantics.

a) Define a "null-tolerant join" for the relational algebra that acts like the $\bowtie$ of the SPARQL algebra.

b) Which SQL construct is similar to the "\" operator in the SPARQL algebra?

c) In the SPARQL algebra, OPT is expressed via left outer join, which is defined via "\" (while a corresponding MINUS does not exist in the SPARQL syntax).
Such a MINUS (cf. part (b) of this exercise) provides a more intuitive idea of negation than "! bound(x)". Give a general pattern how to express (P_1 MINUS P_2) in SPARQL 1.0 syntax.

d) Recall the definition of $\sqsupset\!\bowtie$ in the relational algebra (DB lecture) and define SPARQL's $\sqsupset\!\bowtie$ in a similar way.

Exercise 1.4 (Outer Join) Recall that SPARQL's OPTIONAL corresponds to a left outer join.

a) Give a general pattern how to express a *full* outer join (i.e., "outer" to both sides) in the SPARQL algebra (consider as input two mappings R and S and give an expression for $R \sqsupset\!\bowtie\!\sqsubset S$) and in SPARQL.

b) Give all cities (name as ?XN) that are the capital of a country (:capital) or that are located at a river (:locatedAt) or both (return the names ?CN of the country and/or the river (?RN)).

Exercise 1.5 (SPARQL Formal Semantics: OPTIONAL) Consider the SPARQL Formal Semantics.

Prove or show a counterexample:

The statement (from W3C SPARQL Working Draft 20061004)

> If OPT(A, B) is an optional graph pattern, where A and B are graph patterns, then S is a solution of OPT(A,B) if
>
> - S is a pattern solution of A and of B, or
> - S is a solution to A, but not to A and B.

describes the same semantics as above.

Exercise 1.6 (SPARQL: Filter-Safe Expressions) Consider the following definition:

Definition 1 ([PAG06, AG 08]) A SPARQL expression is *filter-safe*, if for every subexpression of the form $(P\ \ \text{FILTER}\ \ R)$, $\text{var}(R) \subseteq \text{var}(P)$.

Filters of the forbidden form are rather commonly used, e.g.,

```
{ ?P1 a :Person;  :age ?A1. ?P2 a :Person
  OPTIONAL { ?P2 :age ?A2 . FILTER ( ?A2 > ?A1 ) }}
```

a) Sketch an algorithm that rewrites non-filter safe queries into safe ones. First, try it on your own, then maybe look in [AG08].

b) Give a SPARQL query for "Give the names of all countries, such that there is some city in that country where more than 1/4 of the population are living in", and make it filter-safe.
Give also its SPARQL algebra expression.

c) Give a SPARQL query for "Give the names of all countries, such that there is *no* city in that country where more than 1/4 of the population are living in", and make it filter-safe.
Give also its SPARQL algebra expression.

d) Is there a similar thing in SQL, in the relational algebra, and in the relational calculus?