Revision Exercises 2019

1. Consider the following XML document:

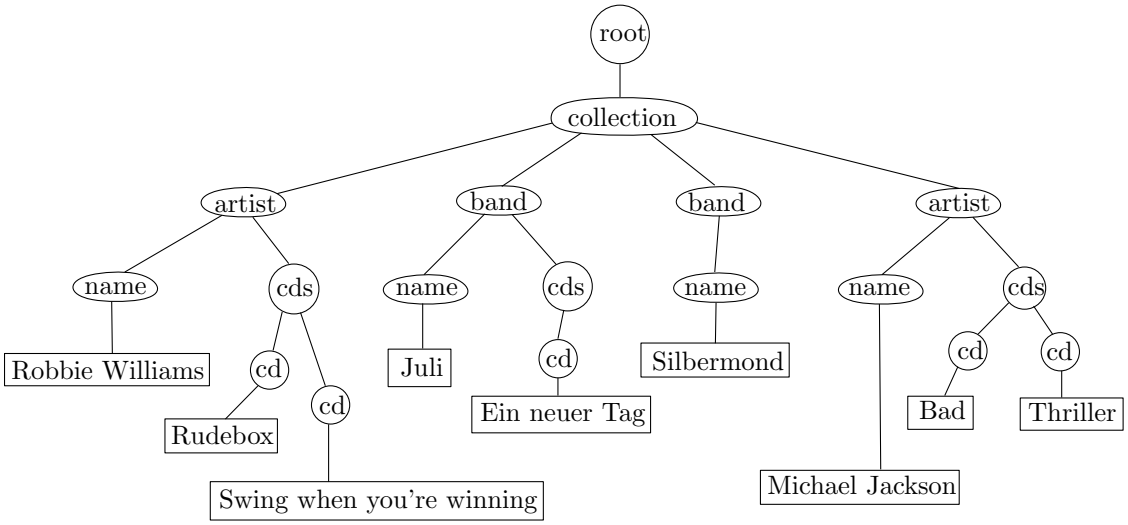
```
<collection>
  <artist>
    <name>Robbie Williams</name>
    <cds>
      <cd>Rudebox</cd>
      <cd>Swing when you're winning</cd>
    </cds>
  </artist>
  <band>
    <name>Juli</name>
    <cds>
      <cd>Ein neuer Tag</cd>
    </cds>
  </band>
  <band>
    <name>Silbermond</name>
  </band>
  <artist>
    <name>Michael Jackson</name>
    <cds>
      <cd>Bad</cd>
      <cd>Thriller</cd>
    </cds>
  </artist>
</collection>
```

(a) give a tree representation of this XML document;

(b) give XPath queries that can find the following in the document:

- the names of all artists;
- the name of the artist who recorded the 'Rudebox' CD.
- the names of all bands that do not have any CDs.

Answer: (a) Tree representation of the XML document:



(b)

- the names of all artists: `//artist/name` or `//artist/name/text()`
- the name of the artist who recorded the 'Rudebox' CD: `//*[cd='Rudebox']/parent::artist/name` or `//artist[cds/cd='Rudebox']/name` possibly followed by `/text()`
- the names of all bands that do not have any CDs: `//band[not(cds)]/name` possibly followed by `/text()`

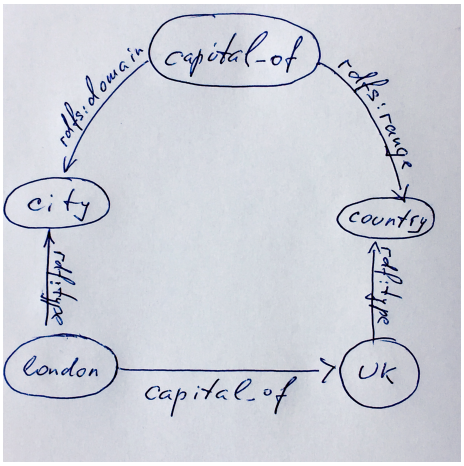
2. Consider the following RDF document:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:geo="http://geography.org/#"
         xml:base="http://geography.org/">

    <rdf:Description rdf:about="#UK">
        <rdf:type rdf:resource="#country"/>
    </rdf:Description>

    <rdf:Description rdf:about="#capital_of">
        <rdf:type rdf:resource="&rdf;Property"/>
        <rdf:domain rdf:resource="#city"/>
        <rdf:range rdf:resource="#country"/>
    </rdf:Description>

    <rdfs:Class rdf:about="#country"/>

    <geo:city rdf:about="#london">
        <geo:capital_of rdf:resource="#UK"/>
    </geo:city>

    <rdfs:Class rdf:about="#city"/>
</rdf:RDF>
```

(a) Describe in English the content of this document.
(b) Give a graph representation of the document.
(c) Represent this graph in the Turtle syntax.

Answer. (a) UK is a country (which is a class). London is a capital of UK, where 'capital of' is a property with domain 'city' and range 'country.' Thus, London is a city.

(b)



(c)

```
ex:UK rdf:type ex:country .
ex:capital_of rdf:type rdf:Property .
ex:capital_of rdfs:domain ex:city .
ex:capital_of rdfs:range ex:country .
ex:country rdf:type rdfs:Class .
ex:london rdf:type ex:city .
ex:london ex:capital_of ex:UK .
ex:city rdf:type rdfs:Class .
```

3. Represent the following information by means of RDF/S triples, using the Turtle syntax:

- Document 1 has been created by Paul on 29 November 2013.
- Document 2 and document 3 have been created by the same (unknown) author.
- Document 3 says that document 1 has been published by W3C.
- Somebody knows somebody else who is a lecturer and whose phone number is 654-321.

Represent the following query in SPARQL:

- Find the authors of documents together with the date of creation.
- Find the phone number of the author and the date of creation of any document having an author and, if available, the date of creation.

Answer. (a) Turtle representation:

```
myns:doc1 dc:creator myns:paul .
myns:doc1 dc:date "29-11-2013"^^xsd:date .
myns:paul dc:phone "123-456" .
myns:doc2 dc:creator _:x .
myns:doc3 dc:creator _:x .
myns:doc3 myns:claims _:y .
_:y rdf:subject myns:doc1 .
_:y rdf:predicate dc:Publisher .
_:y rdf:object "W3C" .
_:y rdf:type rdf:statement .
[] foaf:knows [ rdf:type foaf:lecturer ;
                dc:phone "654-321" ] .
```

(b) SPARQL queries:

```
SELECT ?a ?d
WHERE { ?x dc:creator ?a .
        ?x dc:date ?d .  }

SELECT ?p ?d
WHERE { ?x dc:creator ?a .
WHERE { ?a dc:phone ?p .
        OPTIONAL { ?x dc:date ?d .  }}
```

4. Consider the database comprising 5 tables:

1. The table *uni1.student* contains the local ID, first and last names of the students.

s_id	**first_name**	**last_name**
1	Mary	Smith
2	John	Doe

The column *s_id* is a primary key.

2. The table *uni1.academic* contains the local ID, first and last names of the academic staff, but also information about their position.

a_id	**first_name**	**last_name**	**position**
1	Anna	Chambers	1
2	Edward	May	9
3	Rachel	Ward	8

The column *position* is populated with magic numbers:

$1 \rightarrow$ Full Professor

$2 \rightarrow$ Associate Professor

$3 \rightarrow$ Assistant Professor

$8 \rightarrow$ External Teacher

9 → PostDoc

The column *a_id* is a primary key.

3. The table *uni1.course* contains the local ID and the course title.

c_id	**title**
1234	Linear Algebra

The column *c_id* is a primary key.

4. The table *uni1.teaching* contains the relation between courses and teachers.

c_id	**a_id**
1234	1
1234	2

There is no primary key, but two foreign keys to the tables *uni1.course* and *uni1.academic*.

5. The table *uni1.course-registration* contains the relation between courses and students.

c_id	**s_id**
1234	1
1234	2

There is no primary key, but two foreign keys to the tables *uni1.course* and *uni1.student*.

Create mappings that populate concepts **Course**, **Student** (together with data properties **firstName** and **lastName)**, **FullProfessor** and object property **isTaughtBy.**

Answer:

Course

Target:

```
ex:uni1/course/{c_id} a :Course .
```

Source:

```
SELECT c_id FROM uni1.course
```

Student

Target:

```
ex:uni1/student/{s_id} a :Student ;
    foaf:firstName {first_name}^^xsd:string ;
    foaf:lastName {last_name}^^xsd:string .
```

Source:

```
SELECT * FROM uni1.student
```

FullProfessor

Target:

```
ex:uni1/academic/{a_id} a :FullProfessor .
```

Source:

```
SELECT a_id FROM uni1.academic WHERE position = 1
```

isTaughtBy

Target:

```
ex:uni1/academic/{a_id} :teaches ex:uni1/course/{c_id} .
```

Source:

```
SELECT * FROM uni1.teaching
```

5. Represent the following story by means of RDF/S triples, using the Turtle syntax:

> 'The elementary school of Freiburg has three employees: the two teachers Mr. Maier and Mrs. Schmidt, and the schoolmaster Mrs. Koster. In addition to their administrative duties, Mrs. Koster also does some teaching. In particular, Mr. Maier is assigned to the first-graders, while Mrs. Schmidt and Mrs. Koster together teach the second-, third-, and fourth-graders. Mr. Maier has specialised in sports and therefore is assigned to physical education for all four grades of school. Each grade has a class representative and at least one pupil. Actually, Marie is a fourth-grader. Her favourite subjects in school are physical education, painting, and mathematics.'

Use URIs, Blank Nodes, Literals, and RDF containers in your RDF graph. Whenever it makes sense, also use the rdfs vocabulary, in particular rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, and rdfs:range.

Answer: (prefixes and namespaces are omitted)
classes and properties:

ElementarySchool rdf:type rdfs:Class .
Employer rdf:type rdfs:Class .
ElementarySchool rdfs:subClassOf Employer .
Teacher rdf:type rdfs:Class .
Schoolmaster rdf:type rdfs:Class .
Pupil rdf:type rdfs:Class .
Schoolmaster rdfs:subClassOf Teacher .
Teacher rdfs:subClassOf Person .
Pupil rdfs:subClassOf Person .
SchoolSubject rdf:type rdfs:Class .
Grade rdf:type rdfs:Class .
name rdf:type rdf:Property .
name rdfs:domain Person .
name rdfs:range xsd:string .
gender rdf:type rdf:Property .
name rdfs:domain Person .
gender rdfs:range xsd:string .
works rdf:type rdf:Property .
works rdfs:domain Person .
works rdfs:range Employer .

grader rdf:type rdf:Property .
grader rdfs:domain Grade .
grader rdfs:range xsd:integer .
teaches rdf:type rdf:Property .
teaches rdfs:domain Teacher .
teaches rdfs:range Grade .
teachesPhysEdu rdf:type rdf:Property .
teachesPhysEdu rdfs:subPropertyOf teaches .
teachesPhysEdu rdfs:domain Teacher .
teachesPhysicalEdu rdfs:range Grade .
visits rdf:type rdf:Property .
visits rdfs:domain Pupil .
visits rdfs:range Grade .
represents rdf:type rdf:Property .
represents rdfs:domain Pupil .
represents rdfs:range Grade .
represents rdfs:subPropertyOf visits .
favoriteSubjects rdf:type rdf:Property .
favoriteSubjects rdfs:domain Pupil .
favoriteSubjects rdfs:range rdf:Bag .

Facts:

ESFr rdf:type ElementarySchool .
ESFr rdfs:label "Elementary School Freiburg" .
T1 rdf:type Teacher .
T1 name "Maier" .
T1 gender "male" .
T1 works ESFr .
T2 rdf:type Teacher .
T2 name "Schmidt" .
T2 gender "female" .
T2 works ESFr .
T3 rdf:type Schoolmaster .
T3 name "Koster" .
T3 gender "female" .
T3 works ESFr .
G1 rdf:type Grade .
G1 grader "1" .
G1 rdfs:label "First-grader" .
...
T1 teaches G1 .
...
T1 teachesPhysEdu G1 .
...
_:P1a rdf:type Pupil .
_:P1a represents G1 .
_:P1b rdf:type Pupil .
_:P1b visits G1 .

...
Marie represents G4 .
Marie favoriteSubjects _:SL .
_:SL rdf:type rdf:Bag .
_:SL rdf:_1 PhysEdu .
_:SL rdf:_2 Painting .
_:SL rdf:_3 Math .
PhysEdu rdf:type SchoolSubject .
Painting rdf:type SchoolSubject .
Math rdf:type SchoolSubject .

6. Represent, if possible, the following information in RDFS using the Turtle syntax: John believes that Mary wants to marry him.

This is a statement about statements, which requires reification:
'John believes X', where X is 'Mary wants Y', where Y is 'Mary marries John'.

```
:john :believes _:X .
_:X rdf:type rdf:Statement .
_:X rdf:subject :mary .
_:X rdf:predicate :wants .
_:X rdf:object _:Y .

_:Y rdf:type rdf:Statement .
_:Y rdf:subject :mary .
_:Y rdf:predicate :marries .
_:Y rdf:object :john .
```

7. Create a DL knowledge base that models the following statements:

(1) Every pizza is a meal.

(2) Pizzas always have two toppings.

(3) Every Margherita pizza has a tomato topping.

(4) Everything with a topping is a pizza.

(5) No Margherita pizza has a meat topping.

(6) Pizzas, meat and toppings are different things.

(7) Property 'has topping is inverse-functional and has domain 'pizza' and range 'topping.'

Which of them can be satisfactory represented in RDF(S)?

Answer

(1) Every pizza is a meal.

$\mathsf{Pizza} \sqsubseteq \mathsf{Meal}$

(2) Pizzas always have two toppings.

$\mathsf{Pizza} \sqsubseteq \geq 2\mathsf{hasTopping}.\top$

(3) Every Margherita pizza has a tomato topping.

$\mathsf{Margherita} \sqsubseteq \exists \mathsf{hasTopping}.\mathsf{Tomato}$

(4) Everything with a topping is a pizza.

$\exists \mathsf{hasTopping}.\top \sqsubseteq \mathsf{Pizza}$

(5) No Margherita pizza has a meat topping.

$\mathsf{Margherita} \sqsubseteq \neg\exists \mathsf{hasTopping}.\mathsf{Meat}$

(6) Pizzas, meat and toppings are different things.

$\mathsf{Pizza} \sqsubseteq \neg\mathsf{Meat}, \quad \mathsf{Pizza} \sqsubseteq \neg\mathsf{Topping}, \quad \mathsf{Topping} \sqsubseteq \neg\mathsf{Meat},$

(7) Property 'has topping is inverse-functional and has domain 'pizza' and range 'topping.'

$\top \sqsubseteq \leq 1\mathsf{hasTopping}^{-}, \quad \exists \mathsf{hasTopping}.\top \sqsubseteq \mathsf{Pizza}, \quad \top \sqsubseteq \forall \mathsf{hasTopping}.\mathsf{Topping}$

Which of them can be satisfactory represented in RDF(S)? Only (1), (4) because it can be expressed as 'the domain of hasTopping is Pizza' and the last two statements of (7).