

zad 1

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:lib="http://example.org/mylibrary#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdfs:Class rdf:ID="Multimedia" />
  <rdfs:Class rdf:ID="MusicCD">
<rdfs:subClassOf rdf:resource="#Multimedia"/>
<rdfs:label>CD z muzyka</rdfs:label>
<rdfs:comment>Klasa zawiera wszystkie CD.</rdfs:comment>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Book">
<rdfs:subClassOf rdf:resource="#Multimedia"/>
<rdfs:label>Books</rdfs:label>
<rdfs:comment>Klasa zawiera wszystkie Książki.</rdfs:comment>
  </rdfs:Class>

  <rdf:Description rdf:about="http://example.org/mylibrary#book-wz_projektowe">
    <dc:author>Alan Shalloway</dc:author>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Projektowanie zorientowane obiektowo: Wzorce
projektowe</dc:title>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2002-04-09</dc:date>
    <dc:publisher>Helion</dc:publisher>
    <dc:language>pl</dc:language>
<rdf:type rdf:resource="http://example.org/mylibrary#Book"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylibrary#book-asp">
    <dc:author>Bill Evjen</dc:author>
<dc:author>Scott Hanselman</dc:author>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ASP.NET z wykorzystaniem C# i VB</dc:title>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2011-06-18</dc:date>
    <dc:publisher>Helion</dc:publisher>
    <dc:language>pl</dc:language>
<rdf:type rdf:resource="http://example.org/mylibrary#Book"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylibrary#cd-LB1">
    <dc:artist>Limp Bizkit</dc:artist>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Significant Other</dc:title>
    <dc:publicationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1999-06-22</dc:publicationDate>
    <dc:language>en</dc:language>
<rdf:type rdf:resource="http://example.org/mylibrary#MusicCD"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylibrary#cd-LB2">
    <dc:artist>Limp Bizkit</dc:artist>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Three Dollar Bill, Y'All</dc:title>
    <dc:publicationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1997-07-01</dc:publicationDate>
    <dc:language>en</dc:language>
<rdf:type rdf:resource="http://example.org/mylibrary#MusicCD"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylibrary#cd-LP">
    <dc:artist>Linkin Park</dc:artist>
```

```
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Hybrid Theory</dc:title>
<dc:publicationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2000-10-24</dc:publicationDate>
<dc:language>en</dc:language>
<rdf:type rdf:resource="http://example.org/mylibrary#MusicCD"/>
</rdf:Description>

</rdf:RDF>
```

=====

zad 3 SPARQL

What semantic vocabularies are used in the queries? What are they for?

Najbardziej popularne to :FOAF, Dublin Core, Project Vocabulary. SPARQL to standardowy język zapytań, zaprojektowany tak żeby wspierać na wiele sposobów

interakcje z formatem RDF. Dzięki temu możemy nie tylko pisać zapytania o dane, ale także późniejszą transformację tych danych w alternatywny słownik i inną strukturę.

SELECT - pobiera zmienne, i umieszcza je w odpowiednich znacznikach, dzięki którym dane te zostają odpowiednio zaprezentowane.

CONSTRUCT - zwraca graf, na podstawie zapytania dostajemy trójki, dzięki którym możemy wygenerować graf.

zad 4

Plik znajduje się na:

<http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>

friends who have name and e-mail defined

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?email
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person ;
    ?x foaf:name ?name ;
    ?x foaf:mbox ?email .
}
```

friends who have name and e-mail defined and optional homepage

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?email ?homepage
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
```

```

WHERE {
  ?x rdf:type foaf:Person ;
  ?x foaf:name ?name ;
  ?x foaf:mbox ?email .
  OPTIONAL {?x foaf:homepage ?homepage}
}

```

friends who have name and e-mail defined and optional homepage, sorted by name descending

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?email ?homepage
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
  ?x rdf:type foaf:Person ;
  ?x foaf:name ?name ;
  ?x foaf:mbox ?email .
  OPTIONAL {?x foaf:homepage ?homepage}
}
ORDER BY DESC(?name)

```

zad 5

people whose name starts with 'K'

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
  ?x rdf:type foaf:Person ;
  ?x foaf:name ?name.
  FILTER regex(?name, "^K")
}

```

people who are older than 18 years old

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?age
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
  ?x rdf:type foaf:Person ;
  ?x foaf:name ?name ;
  ?x foaf:age ?age.
}

```

```
    FILTER (?age > 18)
}
```

people whose name starts with 'K' or are older than 18 years old, make search caseinsensitive

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?age
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person ;
    ?x foaf:name ?name ;
        ?x foaf:age ?age.
    FILTER (?age > 18) || regex(?name, „^K.*”, „i”)
}
```

people having e-mails on student.agh.edu.pl server

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?mail
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person ;
    ?x foaf:mbox ?mail.
    FILTER regex(?mail, 'student.agh.edu.pl$', 'i').
}
```

name of people, who have homepage or e-mail on student.agh.edu.pl server

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?mail
FROM <http://student.agh.edu.pl/~cezer/NKS/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person ;
    ?x foaf:mbox ?mail;
        ?x foaf:homepage ?homepage.
    FILTER regex(?homepage, '^http://student.agh.edu.pl') ||
    regex(?mail, 'student.agh.edu.pl$', 'i'))
}
```

zad 6

What are the main limitations of using (querying for information) the RDF datasets such as DBPedia or MusicBrainz?

Tak jak w przypadku zwykłych baz danych, im większa baza tym zapytanie trwa dłużej, a w naszym przypadku bazy danych są bardzo duże, i zwiększają się w zawrotnym tempie. Z tego powodu zapytania zajmują sporo czasu. Kolejnym problemem jest niespójność bazy danych i często zwrócone dane są nieczytelne.