SEMANTIC WEB - SPRAWOZDANIE 3

1. & 2.

RDF with Dublin Core & Schema:

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

 <rdfs:Class rdf:ID="Item" />
 <rdfs:Class rdf:ID="AudioCD">
        <rdfs:subClassOf rdf:resource="#Item"/>
        <rdfs:label>An audio CD</rdfs:label>
        <rdfs:comment>Class consisting of all audio CDs in library, only original work - longplays/studio albums.</
rdfs:comment>
 </rdfs:Class>
 <rdfs:Class rdf:ID="Book">
        <rdfs:subClassOf rdf:resource="#Item"/>
  <rdfs:label>A book</rdfs:label>
        <rdfs:comment>Pretty self explanatory, includes e-books for kindle.</rdfs:comment>
 </rdfs:Class>
 <rdfs:Class rdf:ID="Magazine">
        <rdfs:subClassOf rdf:resource="#Item"/>
  <rdfs:label>A magazine</rdfs:label>
        <rdfs:comment>Class consisting of magazines, periodicals and newspapers.</rdfs:comment>
 </rdfs:Class>

 <rdf:Description rdf:about="http://example.org/mylib#book-slowik_moskiewski">
        <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Grzegorz Gortat</dc:author>
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Slowik Moskiewski</dc:title>
  <dc:publisher>Wydawnictwo Dolnoslaskie</dc:publisher>
  <rdf:type rdf:resource="http://example.org/mylib#Book"/>
 </rdf:Description>

 <rdf:Description rdf:about="http://example.org/mylib#book-zeznanie">
        <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">John Grisham</dc:author>
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Zeznanie</dc:title>
  <dc:publisher>Amber</dc:publisher>
  <rdf:type rdf:resource="http://example.org/mylib#Book"/>
 </rdf:Description>

 <rdf:Description rdf:about="http://example.org/mylib#cd-fire_of_unknown_origin">
        <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Blue Oyster Cult</dc:author>
        <dc:title>Fire Of Unknown Origin</dc:title>
  <dc:date>1981</dc:date>
  <rdf:type rdf:resource="http://example.org/mylib#AudioCD"/>
 </rdf:Description>
```

```
  <rdf:Description rdf:about="http://example.org/mylib#book-ghost_story">
          <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Jim Butcher</dc:author>
          <dc:title>Ghost Story</dc:title>
   <dc:language>en</dc:language>
   <rdf:type rdf:resource="http://example.org/mylib#Book"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylib#book-krol_bolu">
          <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Jacek Dukaj</dc:author>
          <dc:title>Krol Bolu</title>
   <rdf:type rdf:resource="http://example.org/mylib#Book"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylib">
          <MyFavouriteBooks>
          <rdf:Bag>
          <rdf:li rdf:resource="http://example.org/mylib#book-ghost_story"/>
          <rdf:li rdf:resource="http://example.org/mylib#book-krol_bolu"/>
          <rdf:li rdf:resource="http://example.org/mylib#book-zeznanie"/>
          </rdf:Bag>
          </MyFavouriteBooks>
  </rdf:Description>

<rdf:Description rdf:about="http://example.org/mylib#book-interswiat">
   <authors rdf:parseType="Collection">
           <rdf:Description>
                   <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Neil Gaiman</rdf:first>
                   <rdf:rest>
                           <rdf:Description>
                                   <rdf:first rdf:datatype="http://www.w3.org/2001/
XMLSchema#string">Michael Reaves</rdf:first>
                                   <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdfsyntax-ns#nil" />
                           </rdf:Description>
                   </rdf:rest>
           </rdf:Description>
   </authors>
   <title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Interswiat</title>
   <rdf:type rdf:resource="http://example.org/mylib#Book"/>
</rdf:Description>

</rdf:RDF>
```

3.1.II

danja's miscellany - store, which contains a mix of experimental data about Keith Alexander.
Talisians Store - collection of different blog entries.
twitcrit reviews - collection of mini-reviews stored on Twitter's account called 'twitcrit'.

3.3.

The **SELECT** form of results returns variables and their bindings directly.

3.4.

The **CONSTRUCT** query form returns a single RDF graph specified by a graph template. The result is an RDF graph formed by taking each query solution in the solution sequence, substituting for the variables in the graph template, and combining the triples into a single RDF graph by set union.

4.5.

friends who have name and e-mail defined

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT $name $mbox $homepage FROM <http://student.agh.edu.pl/scanarch/test.rdf>
WHERE
 { $x a foaf:Person .
   $x foaf:name $name .
   $x foaf:mbox $mbox  }
```

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT $name $mbox $homepage FROM <http://student.agh.edu.pl/scanarch/test.rdf>
WHERE
 { $x a foaf:Person .
   $x foaf:name $name .
   $x foaf:mbox $mbox .
   optional {$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/>
SELECT $name $mbox $homepage FROM <http://student.agh.edu.pl/scanarch/test.rdf>
WHERE
 { $x a foaf:Person .
   $x foaf:name $name .
   $x foaf:mbox $mbox .
   optional {$x foaf:homepage $homepage}  }
ORDER BY DESC ($desc)
```

5.

people whose name starts with 'K':

```
SELECT DISTINCT $name
WHERE {
    $x rdf:type foaf:Person ;
    $x foaf:name $name .
    FILTER regex($name, „^K.*”)
}
```

people who are older than 18 years old:

```
SELECT DISTINCT $name, $age
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:
```
SELECT DISTINCT $name, $age
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:
```
SELECT DISTINCT $name, $mail
WHERE {
   $x rdf:type foaf:Person ;
   $x foaf:name $name ;
   $x foaf:mbox $mail
   FILTER regex($mail, „.*student.agh.edu.pl$”)
}
```
name of people, who have homepage or e-mail on student.agh.edu.pl server
```
SELECT DISTINCT $name, $mail, $page
WHERE {
   $x rdf:type foaf:Person ;
   $x foaf:name $name ;
                $x foaf:mbox $mail ;
   $x foaf:homepage $page ;
   FILTER regex($mail, „.*student.agh.edu.pl$”) || regex($page, „^http://
student.agh.edu.pl.*”)
}
```

6.

Aktualnie jednymi z głównych minusów wykorzystywania tego typu baz jest stosunkowo długi czas odpowiedzi oraz wymóg znajomości SPARQL do tworzenia zapytań. Przeciętny użytkownik internetu nie będzie mieć pojęcia o żadnym języku typu SQL, dlatego też użyteczność tego typu baz będzie dla niego znikoma. Dodatkowo niektóre dłuższe i bardziej skomplikowane zapytania potrzebują znacznej ilości czasu by się wykonać.
Dodatkowo biorąc pod uwagę że dbpedia nie zawiera jeszcze wszystkich danych z wikipedii to nigdy nie mamy pewności że otrzymaliśmy wszystkie wyniki dla naszego zapytania.