

Домашно др. 2  
**SPARQL**  
Константин Богданоски, 161048

**A) Домашна задача**

**1. Што е SPARQL?**

**SPARQL** е RDF query јазик, поточно семантичен query јазик наменет за датабази кои работат и можат да вратат податоци напишани во RDF (Resource Definition Format) форма.

---

**2. Кои типови на SPARQL прашања постојат?**

Постојат **SELECT**, **CONSTRUCT**, **ASK**, **DESCRIBE** и **UPDATE** queries

---

**3. Во каква логичка релација се наоѓаат меѓусебно triple pattern-ите кои се наведени во стандарден WHERE дел од SPARQL прашање?**

Query-то ќе ги врати сите тројки кои ги исполнуваат сите tripple pattern услови во **WHERE** делот. Таму дефинираме според што да селектира **SPARQL**.

---

**4. Како можеме да означиме во едно SPARQL прашање дека бараниот triple pattern не е задолжителен при пребарувањето?**

Се додава **OPTIONAL** во **WHERE** делот од query-то.

---

**5. Дали еден SPARQL endpoint може да се користи и од страна на апликации? Објаснете.**

SPARQL endpoints не се наменети да бидат читани од страна на човекот, нивната цел е да се добијат податоците од веб-от и да не се форматираат тие за да може човекот да ги разбере. Поради тоа добро е излезот од SPARQL да биде влез на соодветна апликација, која успешно ќе ги процесира, парсира и прикаже на начин кој што е разбирлив за човекот.

Пример:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:name ?name .
}
```

ни ги дава сите личности што имаат релација име, но добиваме:

[www.google.com/Konstantin-Bogdanoski](http://www.google.com/Konstantin-Bogdanoski) foaf:name Konstantin Bogdanoski

не е баш разбирливо, затоа преку соодветна апликација можеме да направиме парсирање и да добиеме [www.google.com/Konstantin-Bogdanoski](http://www.google.com/Konstantin-Bogdanoski) has name Konstantin Bogdanoski

## Б) Вовед

DBpedia (<http://www.dbpedia.org>) е проект кој ги извлекува структурираните информации од Wikipedia, семантички ги аотира (со RDF) и ги објавува како јавно достапни информации. DBpedia дозволува овие податоци да се поврзуваат со други RDF податоци низ WWW, да се користат за сопствени бази на знаење, а дозволува и поставување на SPARQL прашања врз податоците со кои располага.

За да го овозможи последново, DBpedia нуди пристапна точка за пишување SPARQL прашања преку веб, преку нивниот SPARQL Endpoint, поставен на: <http://dbpedia.org/sparql>.

Напомена: Доколку некогаш имате проблеми со SPARQL Endpoint-от, можете да го користите SPARQL Explorer-от, достапен на: <http://dbpedia.org/snorql>. Доколку DBpedia е недостапна во моментот, нејзините податоци можете да ги пребарувате и преку <http://lod.openlinksw.com/sparql>.

## В) Практична задача

### I. Работа со податоци за личност, режисер

1. DBpedia своите ресурси ги означува со URI-а кои се изведени од оригиналните адреси на англиските страни на Wikipedia. Тие се добиваат на следниот начин:

- Се одбира Wikipedia страна (на пр: [http://en.wikipedia.org/wiki/Woody\\_Allen](http://en.wikipedia.org/wiki/Woody_Allen))
- Префиксот <http://en.wikipedia.org/wiki/> се заменува со префиксот <http://dbpedia.org/resource/>
- На тој начин, се добива DBpedia URI-то за ресурсот: [http://dbpedia.org/resource/Woody\\_Allen](http://dbpedia.org/resource/Woody_Allen)
- Доколку го отворите овој ресурс преку интернет прелистувач, ќе ви се отвори страната [http://dbpedia.org/page/Woody\\_Allen](http://dbpedia.org/page/Woody_Allen), на која се излистани сите RDF тројки кои DBpedia ги извлекла од Wikipedia за дадениот ресурс.
- Првата половина RDF тројки се оние во кои ресурсот се среќава како Subject. Втората половина RDF тројки, во кои релациите се означени како 'is relation of', се оние во кои ресурсот се среќава како Object.

2. Разгледајте ги тројките (релациите и вредностите) кои ги има на DBpedia за ресурсот [http://dbpedia.org/resource/Woody\\_Allen](http://dbpedia.org/resource/Woody_Allen). Секоја од релациите е означена линк кој носи кон уникатно URI кое ја идентификува релацијата и на кое е поставена нејзината дефиниција.

Преку SPARQL Endpoint-от (поставен на адресата <http://dbpedia.org/sparql>), напишете SPARQL прашање со кое ќе го добиете вистинското име на Вуди Ален (dbo:birthName).

```
SELECT ?birthName
```

```
WHERE {
```

```
<http://dbpedia.org/resource/Woody\_Allen> dbo:birthName ?birthName
}
```

Напомена: На линкот Namespace Prefixes (поставен на SPARQL Endpoint страната) можете да ги погледнете префиксите кои може да ги користите кај ресурсите од различни извори, без да ги наведувате експлицитно во самото SPARQL прашање. На пример, ресурсот [http://dbpedia.org/resource/Woody\\_Allen](http://dbpedia.org/resource/Woody_Allen) може да се наведе како **dbpedia:Woody\_Allen**, а ресурсот <http://dbpedia.org/ontology/birthName> како **dbo:birthName** во SPARQL

прашањата, без да ги вклучите овие два префикси експлицитно. Доколку префиксот кој сакате да го користите не е прикажан во Namespace Prefixes, дефинирајте го експлицитно.

3. Напишете SPARQL прашање со кое ќе ги добиете сите филмови на Вуди Ален (филмови кои ги режирал, филмови кои ги напишал и филмови во кои глумел).

```
SELECT *  
WHERE {  
  {?directorOf dbo:director <http://dbpedia.org/resource/Woody_Allen> .}  
UNION  
  {?creatorOf dbo:creator <http://dbpedia.org/resource/Woody_Allen> .}  
UNION  
  {?starringIn dbo:starring <http://dbpedia.org/resource/Woody_Allen> .}  
}
```

---

4. Проширете го претходното SPARQL прашање за да ги добиете сите филмови на Вуди Ален, но подредени од најновите кон најстарите (според датумот на објавување).

```
ORDER BY ASC(dbo:date)
```

---

5. Проширете го претходното SPARQL прашање за да ги добиете сите филмови на Вуди Ален, со буџет поголем од \$500.000 долари.

```
SELECT *  
WHERE {  
  {?directorOf dbo:director <http://dbpedia.org/resource/Woody_Allen> .  
    ?directorOf dbo:budget ?directorbudget .  
  FILTER (?directorbudget>500000)  
}  
UNION  
  {?creatorOf dbo:creator <http://dbpedia.org/resource/Woody_Allen> .  
    ?creatorOf dbo:budget ?creatorbudget .  
  FILTER (?creatorbudget>500000)  
}  
UNION  
  {?starringIn dbo:starring <http://dbpedia.org/resource/Woody_Allen> .  
    ?starringIn dbo:budget ?starringbudget .  
  FILTER (?starringbudget>500000)  
}  
}  
ORDER BY ASC(dbo:date)
```

---

6. Напишете SPARQL прашање со кое ќе ги добиете имињата (label, на англиски) и сликите (thumbnail) на сите актери кои играле (starring) во филм на Вуди Ален. Ограничете го излезот на само 10 актери. Отстранете ги дупликатите од резултатите.

```
SELECT DISTINCT ?directorOf ?actor
WHERE {
  ?directorOf dbo:director <http://dbpedia.org/resource/Woody_Allen>.
  {
    SELECT DISTINCT ?actor
    WHERE {
      ?directorOf dbo:starring ?actor.
    }
    LIMIT 10
  }
}
ORDER BY ?directorOf
```

---

## II. Работа со податоци за книга

7. Отворете го ресурсот [http://dbpedia.org/resource/The\\_Little\\_Prince](http://dbpedia.org/resource/The_Little_Prince). Погледнете ги својствата (тројките) кои ги има оваа книга. Напишете SPARQL прашање со кое ќе ги добиете авторот и јазикот на книгата.

```
SELECT ?authorName ?language
WHERE {
  <http://dbpedia.org/resource/The_Little_Prince> dbo:author ?authorName;
  dbp:language ?language.
}
```

---

8. Напишете SPARQL прашање со кое ќе го добиете апстрактот на овој ресурс. Апстрактот нека биде на англиски јазик.

```
SELECT ?abstract
WHERE {
  <http://dbpedia.org/resource/The_Little_Prince> dbo:abstract ?abstract.
  FILTER (langMatches(lang(?abstract), "en"))
}
```

---

9. Преку истиот ресурс (книгата „Малиот Принц“), освен името на авторот извлечете ги информациите и за датумот и местото на раѓање на авторот, како и датумот и местото на умирање.

```
SELECT *
WHERE {
  <http://dbpedia.org/resource/The_Little_Prince> dbo:author ?author.
  ?author dbo:birthDate ?authorBirthDate;
  dbo:birthPlace ?authorBirthPlace;
  dbo:deathDate ?authorDeathDate;
  dbo:deathPlace ?authorDeathPlace.
}
```

10. Преку истиот ресурс (книгата „Малиот Принц“), извлекете ги имињата и на останатите книги кои ги напишал нејзиниот автор, нивните години на издавање и сортирајте ги според годината.

```
SELECT DISTINCT ?book ?author ?releaseDate
WHERE {
  <http://dbpedia.org/resource/The_Little_Prince> dbo:author ?author.
  ?book dbo:author ?author.
  ?book dbp:releaseDate ?releaseDate.
}
ORDER BY ASC(?releaseDate)
```

11. Напишете SPARQL прашање за да добиете одговор во која година прв пат е издадена оваа книга, од која издавачка куќа и под кој оригинален наслов.

```
SELECT *
WHERE {
  <http://dbpedia.org/resource/The_Little_Prince> dbp:releaseDate ?releaseDate;
                                                    dbo:publisher ?publisher;
                                                    prov:wasDerivedFrom ?firstName.
}
ORDER BY ASC(?releaseDate)
```

---

### III. Работа со податоци за ресурс по слободен избор

12. Отворете Wikipedia article по ваш избор. Отворете го соодветниот DBpedia ресурс и разгледајте ги тројките кои постојат за тој ресурс. Напишете барем три SPARQL прашања по ваш избор, во кој ќе ги пробате останатите опции од аудиториски вежби.

Пример идеи:

- Музички бенд / артист:
  - прикажете ги имињата и датумите на објава на албумите;
  - прикажете ги имињата и местата на раѓање на членовите на бендот;
- Личност:
  - прикажете краток опис и биографија, на англиски;
  - прикажете ги датумите и местата на раѓање / смрт;
  - прикажете ги линковите до неговиот / нејзиниот веб сајт, социјален профил, итн.;
- Локација:
  - прикажете краток опис и линк кон слика од локацијата;
  - прикажете ги longitude и latitude координатите, како и надморската височина;
- итн.

#### **IV. Работа со податоци за потребите на конкретна апликација**

13. Размислете за некоја мобилна или веб апликација која би сакале да ја развивате. Согласно темата, пробајте да напишете едно или повеќе SPARQL прашања со кои од DBpedia ќе ги извлечете сите потребни податоци за нејзино реализирање.

Пример идеи:

- најдете ги сите филмови објавени во тековната година и прикажете ги нивните имиња, краток опис, жанр, како и датумот на објава; сортирајте ги по датум на објава;
- најдете ги музичките албуми кои се објавени во тековната година и прикажете го артистот, името на албумот, жанрот, датумот на објава и линкот кон слика од албумот; сортирајте ги по датум на објава;
- итн.