

ПРИМЕРИ SPARQL УПИТА

Примери упита над
DBpedia SPARQL Endpoint-ом
(<http://dbpedia.org/sparql>)

Пример 1: Приказати имена и датуме рођења и смрти свих људи рођених у Лондону у периоду 1900-1950 године; сортирати резултате

```
PREFIX db-ont: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?person ?name ?birth ?death
WHERE {
  ?person db-ont:birthPlace <http://dbpedia.org/resource/London> .
  ?person db-ont:birthDate ?birth .
  ?person foaf:name ?name .
  OPTIONAL { ?person db-ont:deathDate ?death . }
  FILTER ((?birth > "1900-01-01"^^xsd:date) &&
    (?birth < "1950-01-01"^^xsd:date)) .
}
ORDER BY (?name)
```

Пример 2: Приказати имена и датуме рођења свих глумаца рођених у Лондону након 1930 године који су још увек живи

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX db-ont: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?person ?name ?birth
WHERE {
    ?person db-ont:birthPlace <http://dbpedia.org/resource/London> .
    ?person db-ont:occupation <http://dbpedia.org/resource/Actor> .
    ?person foaf:name ?name .
    ?person db-ont:birthDate ?birth
    FILTER (?birth > "1930-01-01"^^xsd:date) .
    FILTER NOT EXISTS { ?person db-ont:deathDate ?death. }
}
```

Обухватније решење (узима у обзир и случајеве кад је занимање особе представљено као обичан стринг):



```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX db-ont: <http://dbpedia.org/ontology/>
PREFIX db-prop: <http://dbpedia.org/property/>
SELECT DISTINCT ?person ?name ?birth
WHERE {
  ?person db-ont:birthPlace <http://dbpedia.org/resource/London> .
  { ?person db-ont:occupation <http://dbpedia.org/resource/Actor> . }
  UNION
  {?person db-prop:occupation ?occupation
    FILTER ((str(?occupation) = "Actor") || (str(?occupation) = "Actress")) }
  ?person foaf:name ?name .
  ?person db-ont:birthDate ?birth
  FILTER (?birth > "1930-01-01"^^xsd:date) .
  FILTER NOT EXISTS { ?person db-ont:deathDate ?death. }
}
```

Пример 3: Приказати имена свих престоница које припадају средње европској часовној зони. Приказати и изворна имена оних градова за које је тај податак расположив.

```
PREFIX db-ont: <http://dbpedia.org/ontology/>
```

```
PREFIX db-prop: <http://dbpedia.org/property/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT DISTINCT ?name ?native
```

```
WHERE {
```

```
  ?country rdf:type <http://dbpedia.org/ontology/Country> ; db-ont:capital ?city .
```

```
  { ?city foaf:name ?name .} UNION { ?city rdfs:label ?name . }
```

```
    FILTER ( lang(?name) = "en" )
```

```
  {?city db-ont:timeZone <http://dbpedia.org/resource/Central_European_Time> . }
```

```
    UNION
```

```
  {?city db-prop:timezone ?timezone FILTER (str(?timezone) = "CET") }
```

```
  OPTIONAL { ?city db-prop:nativeName ?native } .
```

```
}
```

Примери упита над

[EduBase dataset](#) (део data.gov.uk)

SPARQL Endpoint-ом (

<http://openuplabs.tso.co.uk/demos/sparqleditor>)

Пример 1: Приказати имена свих школа које припадају административном округу “City of London”; сортирати резултате према називу школа

```
PREFIX sch-ont: <http://education.data.gov.uk/def/school/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?name
WHERE {
    ?school a sch-ont:School;
            sch-ont:establishmentName ?name;
            sch-ont:districtAdministrative ?district .
    ?district rdfs:label "City of London" .
} ORDER BY ?name
```


Пример 2:

Приказати имена и адресе свих школа које припадају административном округу “City of London”;

сортирати резултате према називу школа;

напомена: подаци о адреси нису расположиви за све школе

```
PREFIX sch-ont: <http://education.data.gov.uk/def/school/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?name ?address1 ?address2 ?postcode ?town
WHERE {
    ?school a sch-ont:School;
        sch-ont:establishmentName ?name;
        sch-ont:districtAdministrative ?district .
    ?district rdfs:label "City of London" .
OPTIONAL {
    ?school sch-ont:address ?address .
    ?address sch-ont:address1 ?address1 ;
        sch-ont:address2 ?address2 ;
        sch-ont:postcode ?postcode ;
        sch-ont:town ?town . }
} ORDER BY ?name
```

Пример 3: Приказати име и датум отварања најстарије школе у Енглеској

```
PREFIX sch-ont: <http://education.data.gov.uk/def/school/>
SELECT ?name ?date
WHERE {
    ?school a sch-ont:School;
           sch-ont:establishmentName ?name;
           sch-ont:openDate ?date.
} ORDER BY ASC(?date)
LIMIT 1
```

Примери упита над

[Linked Movie Database](#)

SPARQL Endpoint-ом (
<http://data.linkedmdb.org/sparql>)

коришћењем Веб алата:

<http://www.snee.com/sparql/spreadsheetSPARQL.html>

Пример 1: Приказати имена свих глумаца који су играли у филму са задатим глумцем, нпр. Kevin Bacon-ом

```
PREFIX mov: <http://data.linkedmdb.org/resource/movie/>
SELECT DISTINCT ?actorName
WHERE {
  ?kb mov:actor_name "Kevin Bacon".
  ?movie mov:actor ?kb;
        mov:actor ?actor.
  ?actor mov:actor_name ?actorName.
  FILTER (?kb != ?actor).
}
```

Пример 2: Приказати имена свих глумаца који су играли и у филму Martin Scorsese-а и у филму Steven Spielberg-а

```
PREFIX mov: <http://data.linkedmdb.org/resource/movie/>
```

```
SELECT DISTINCT ?actorName
```

```
WHERE {
```

```
  ?dir1  mov:director_name "Martin Scorsese".
```

```
  ?dir2  mov:director_name "Steven Spielberg".
```

```
  ?dir1movie mov:director ?dir1;
```

```
    mov:actor ?actor.
```

```
  ?dir2movie mov:director ?dir2 ;
```

```
    mov:actor ?actor .
```

```
  ?actor  mov:actor_name ?actorName.
```

```
}
```

Пример 3: Приказати идентификаторе и називе свих филмских тема дефинисаних у Linked Movie Database

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX mov: <http://data.linkedmdb.org/resource/movie/>
SELECT DISTINCT ?subject ?title
WHERE {
  { ?movie skos:subject ?subject . }
    UNION
  { ?movie mov:film_subject ?subject . }
  ?subject rdfs:label ?title.
}
ORDER BY ?title
```

Пример 4: Приказати имена свих режисера који су снимали филмове на тему мафије; сортирати резултате

```
PREFIX mov: <http://data.linkedmdb.org/resource/movie/>
```

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

```
SELECT DISTINCT ?directorName
```

```
WHERE {
```

```
  {?movie skos:subject <http://data.linkedmdb.org/resource/film_subject/352> .}
```

```
UNION
```

```
  {?movie mov:film_subject <http://data.linkedmdb.org/resource/film_subject/352> .}
```

```
  ?movie mov:director ?director.
```

```
  ?director mov:director_name ?directorName .
```

```
}
```

```
ORDER BY (?directorName)
```


- Написати SPARQL упит који приказује називе свих филмова у којима су играли двоје ваших омиљених глумаца (сами задајте имена тих глумаца)
 - Користити Linked Movie Database SPARQL Endpoint
- Написати SPARQL упит који приказује називе школа у Енглеској које су отворене током 2008. године
 - Користити EduBase SPARQL Endpoint

- Написати SPARQL упит који приказује називе, и уколико је расположив број студената, свих универзитета који су основани у 19. веку, у земљама које припадају енглеском говорном подручју
 - Користити Dbpedia SPARQL Endpoint

```
PREFIX db-ont: <http://dbpedia.org/ontology/>
PREFIX db-prop: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?students
WHERE {
  ?univ rdf:type <http://dbpedia.org/ontology/University> ;
    foaf:name ?name ;
    db-prop:established ?creationDate ;
    db-ont:country ?country .
  ?country db-ont:language
    <http://dbpedia.org/resource/English_language>
  OPTIONAL { ?univ db-ont:numberOfUndergraduateStudents ?students . }
  FILTER ((?creationDate > "1800-01-01"^^xsd:date) &&
    (?creationDate < "1900-01-01"^^xsd:date)) .
}
ORDER BY (?name)
```