

SPARQL QUERY LANGUAGE

Examples of SPARQL queries

QUERY EXAMPLES OVER DBPEDIA

SPARQL endpoint: <http://dbpedia.org/sparql>

SPARQL editors:

- <http://yasgui.laurensrietveld.nl/>
- <http://dbpedia.org/snorql/>

Query 1: Display names and dates of birth and death of all people born in London between 1900-1950; sort the results

```
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)
```

Query 2: Display names and dates of birth of all actors born in London after year 1930. but which are still alive

```
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. }
}
```

More comprehensive solution (takes into account cases where occupation is given as a plain string):

```
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. }
```

```
}
```

Query 3: Display names of all capitals that are in the CET time zone. Display also native names of those cities if available.

```
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 db-ont: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 } .
```

```
}
```

QUERY EXAMPLES OVER LINKED MOVIE DATABASE

SPARQL endpoint:

<http://data.linkedmdb.org/sparql>

SPARQL editor:

- <http://yasgui.laurensrietveld.nl/>
- <http://www.snee.com/sparql/spreadsheetSPARQL.html>

Query 1: Display names of all actors that played in movies with a given actor, e.g. with Kevin Bacon

```
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).
}
```


Query 2: Display names of all actors that played in any movie directed by Martin Scorsese-a and any movie directed by Steven Spielberg

```
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.
}
```

Query 3: Display identifier and title of all movie topics defined in Linked Movie Database

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX mov: <http://data.linkedmdb.org/resource/movie/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?subject ?title
WHERE {
  { ?movie skos:subject ?subject . }
    UNION
  { ?movie mov:film_subject ?subject . }
  ?subject rdfs:label ?title.
}
ORDER BY ?title
```

Query 4: Display names of all directors that filmed movies about the topic 'mafia'; sort results by the director names

```
PREFIX mov: <http://data.linkedmdb.org/resource/movie/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?directorName
WHERE {
  {?movie skos:subject ?subject .} UNION {?movie mov:film_subject ?subject .}
  ?subject rdfs:label ?subjectLbl
      FILTER (strStarts(lcase(?subjectLbl), "mafia") ).
  ?movie mov:director ?director.
  ?director mov:director_name ?directorName .
}
ORDER BY (?directorName)
```

U slučaju da SPARQL engine ne podržava SPARQL 1.1, ovu liniju zameniti sa:
FILTER (regex(?subjectLbl, "^mafia", "i")) .

QUERY EXAMPLES OVER
EDUBASE DATASET (PART OF
DATA.GOV.UK)

SPARQL Editor:

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

Query 1: Display names of all schools that belong to the district of “City of London”; sort the results by the school names

```
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
```

Query 2:

Display names and addresses of all schools that belong to the district of “City of London”; sort the result by the schools names; REMARK: information about an address is not available for all schools

```
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
```

Query 3: Display a name and a date of opening of an oldest school in England

```
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
```


Tasks

- Write a SPARQL query that lists titles of all movies in which two of your favorite actors played (choose names of those actors)
 - Use the Linked Movie Database SPARQL Endpoint
- Write a SPARQL query that lists titles of all schools in England that were opened in 2008.
 - Use the EduBase SPARQL Endpoint

Tasks

- Write a SPARQL query that lists names, and numbers of students if available, of all universities that were established in 19th century in all English speaking countries
 - Use 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)
```