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 based on the birth date.

```sql
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?name ?birth ?death
WHERE {
  ?person rdfs:label ?name FILTER (lang(?name) = "en").
  OPTIONAL { ?person dbo:deathDate ?death . }
  FILTER ( (?birth > "1900-01-01"^^xsd:date) &&
    (?birth < "1950-01-01"^^xsd:date)) .
}
ORDER BY (?birth)
```
Query 2: Display names and dates of birth of all actors born in London after year 1930. who are still alive

```sparql
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?name ?birth
WHERE {
  ?person foaf:name ?name .
  ?person dbo:birthDate ?birth
  FILTER (?birth > "1930-01-01"^^xsd:date) .
  FILTER NOT EXISTS { ?person dbo:deathDate ?death. }
}
```
More comprehensive solution (takes into account cases where the occupation is given as a plain string):

```sparql
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?name ?birth
WHERE {
    {?person dbo:occupation dbr:Actor .}
    UNION
    {?person dbp:occupation ?occupation
        FILTER (contains(str(?occupation), "Actor") ||
            contains(str(?occupation), "Actress")) }
    ?person foaf:name ?name .
    FILTER NOT EXISTS { ?person dbo:deathDate ?death. }
}
```
Query 3: Display names of all capital cities that are in the CET time zone. Sort the results based on the size of the population of the capitals from the result set.
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?name
WHERE {
  ?country rdf:type dbo:Country;
  FILTER NOT EXISTS {{ ?country dbo:dissolutionDate ?endDate } UNION
                   { ?country dbo:dissolutionYear ?endDate }}
  ?city rdfs:label ?name FILTER ( lang(?name) = "en" )
  { ?city dbo:timeZone dbr:Central_European_Time .
     UNION
     { ?city dbp:timezone ?timezone FILTER (str(?timezone) = "CET") } }
}
ORDER BY DESC (?population)

This filter is to eliminate the former capitals, that is, capitals of the countries that ceased to exist.
QUERY EXAMPLES OVER LINKED MOVIE DATABASE

SPARQL endpoint:
http://data.linkedmdb.org/sparql

Dataset description:
http://linkedmdb.org/
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 a movie directed by Martin Scorsese and also in a 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 titles of movie topics (subjects) with at least 50 films being on that topic

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 ?title (COUNT(?movie) AS ?count)
WHERE {
  { ?movie skos:subject ?subject . }
  UNION
  { ?movie mov:film_subject ?subject . }
}
GROUP BY ?title
HAVING (COUNT(?movie) >= 50)
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 (regex(?subjectLbl, "mafia", "i")) .
  ?director mov:director_name ?directorName .
}
ORDER BY (?directorName)
QUERY EXAMPLES OVER EduBase Dataset
(PART OF DATA.GOV.UK)

SPARQL Endpoint:
http://gov.tso.co.uk/education/sparql

SPARQL examples and query editor:
http://openuplabs.tso.co.uk/sparql/gov-education
Query 1: Display names of all schools that belong to the “City of London” district; 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 Web sites of all schools that belong to the “City of London” district and offer nursery service; sort the result by the schools names

Hints:
- execute DESCRIBE query to examine how schools are described
- note that information about Web address is not available for all the schools
PREFIX sch-ont: <http://education.data.gov.uk/def/school/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name ?website
WHERE {
    ?school a sch-ont:School;
        sch-ont:establishmentName ?name;
        sch-ont:districtAdministrative ?district;
        sch-ont:nurseryProvision "true"^^xsd:boolean .
    ?district rdfs:label "City of London" .
    OPTIONAL {?school sch-ont:websiteAddress ?website. }
} ORDER BY ?name
Query 3: Display the name and the opening date of the oldest school in England that was for boys and girls (mixed gender)

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 names of all schools in England that were opened after 2005, and belong to the category (type of establishment) of “Community School”.
  • Use the EduBase SPARQL Endpoint
Write a SPARQL query that returns names of the universities that were established in English speaking countries before the beginning of the 20th century. For each university, return also the name of the country it is located in, and if available, the number of students. Sort the results based on the date of the universities’ founding dates.

• Use DBpedia SPARQL Endpoint
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?uniName ?cName ?students
WHERE {
    ?uni rdf:type dbo:University;
        foaf:name ?uniName FILTER (lang(?uniName)="en").
    ?uni dbp:established ?creationDate;
        dbo:country ?country .
    ?country dbo:language dbr:English_language;
        foaf:name ?cName FILTER (lang(?cName)="en").
    OPTIONAL { ?uni dbo:numberOfUndergraduateStudents ?students . }
    FILTER ( ?creationDate < "1900-01-01"^^xsd:date ) .
}
ORDER BY (?creationDate)