

SPARQL exercises

1. Find the class name for films
(use **owl:Class** and **regex**(*var*, *pattern*, "i") for case-insensitive pattern matching)
2. Find the object property (**owl:ObjectProperty**) for film directors
3. Find your favourite film
4. Find all the films made by your favourite director
5. Find the datatype property (**owl:DatatypeProperty**) for release dates
6. Find all the films released in January 2000.

Answers to SPARQL exercises

1. Find the class name for films

(use **owl:Class** and **regex**(*var*, *pattern*, "i") for case-insensitive pattern matching)

```
SELECT ?c
WHERE
{
  ?c rdf:type owl:Class .
  FILTER regex(?c, "film", "i")
}
```

<http://dbpedia.org/ontology/Film>

2. Find the object property (**owl:ObjectProperty**) for film directors

```
SELECT ?p
WHERE
{
  ?p rdf:type owl:ObjectProperty .
  FILTER regex(?p, "direct", "i")
}
```

<http://dbpedia.org/ontology/director>

Answers to SPARQL exercises (2)

3. Find your favourite film

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
SELECT ?f
WHERE
{
  ?f rdf:type dbpedia:Film .
  FILTER regex(?f, "blow", "i")
}
```

4. Find all the films made by your favourite director

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
SELECT ?f ?d
WHERE
{
  ?f rdf:type dbpedia:Film .
  ?f dbpedia:director ?d .
  FILTER regex(?d, "antonioni", "i")
}
```

NB. One can use `<http://dbpedia.org/ontology/Film>` instead of `dbpedia:Film`

Answers to SPARQL exercises (3)

5. Find the datatype property (`owl:DatatypeProperty`) for release dates

```
SELECT ?p
WHERE
{
  ?p rdf:type owl:DatatypeProperty .
  FILTER regex(?p, "release", "i")
}
```

<http://dbpedia.org/ontology/releaseDate>

6. Find all the films released in January 2000.

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
SELECT ?f ?d
WHERE
{
  ?f rdf:type dbpedia:Film .
  ?f dbpedia:releaseDate ?d .
  FILTER ((?d >= "2000-01-01"^^xsd:date)
    && (?d < "2000-02-01"^^xsd:date))
}
```