Ontologies and Semantic Web

Semantic Web: querying data – SPARQL

N. Aussenac-Gilles

IRIT- CNRS

aussenac@irit.fr

MELODI group

http://www.irit.fr/

Tributes to other lectures + links

- J. Euzenat : semantic web

http://www.inrialpes.fr/exmo/teaching/swc/

- C. Comparot : Interrogation du web des données – SPARQL (mai 2011)
- http://www.linkeddatatools.com/semantic-modeling
- https://www.w3.org/TR/sparql11-overview/
  https://www.w3.org/TR/sparql11-query/

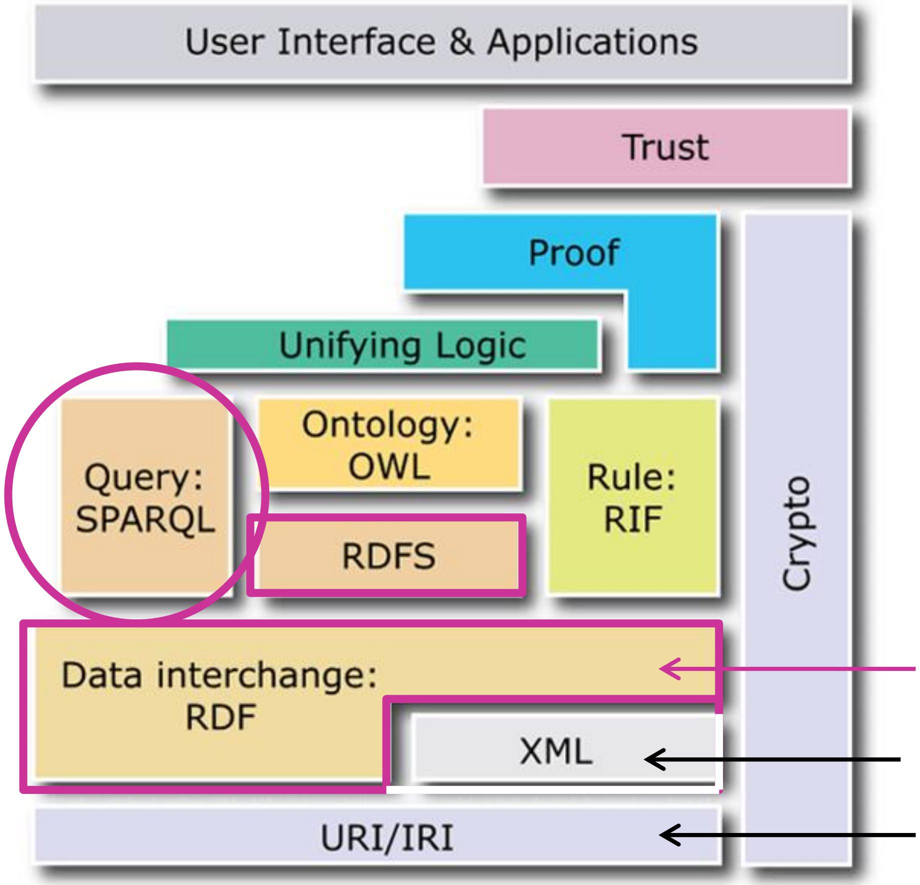
Possibilités de stages

- En entreprise (TAL) : SYNAPSE_développement
  1. Chatbot auto-apprenant multilingue : https://www.synapse-developpement.fr/stage-ia-chatbot-auto-apprenant-h-f/
  2. Génération de questions à partir de documents (type FAQ) : https://www.synapse-developpement.fr/stage-ia-generation-de-questions-a-de-documents-h-f/

- Au sein de l'équipe MELODI
  - Métadonnées sémantiques pour la science ouverte :
    - Aide à l'écriture de schéma en SHACL ; application aux données MétéoFrance
    - Génération d'interface pour décrire des jeux de données à partir de formes SHACL
    - Developpement de services sémantiques au sein de l'entrepôt de science ouverte DataVerse
    - D'une ontologie de l'informatique et de services à une ontologie de métadonnées pour décrire des workflows dans la plateforme CALISTO
  - Extraction de relations n-aires pour détecter des événements par apprentissage non supervisé à partir de textes
  - Représentation sémantique de règles juridiques et aide à la décision ; Apprentissage à partir de décisions à partir de décisions passées
  - Reconnaissance de locuteurs au sein de dialogue par apprentissage
  - Détection de discours haineux, de discours sexistes

Poursuite en thèse possible

The Semantic Web layer cake (2010)



Query language

Semantic

Abstract syntax

Concrete syntax

Naming

SPARQL : Simple Protocol And RDF Query Language

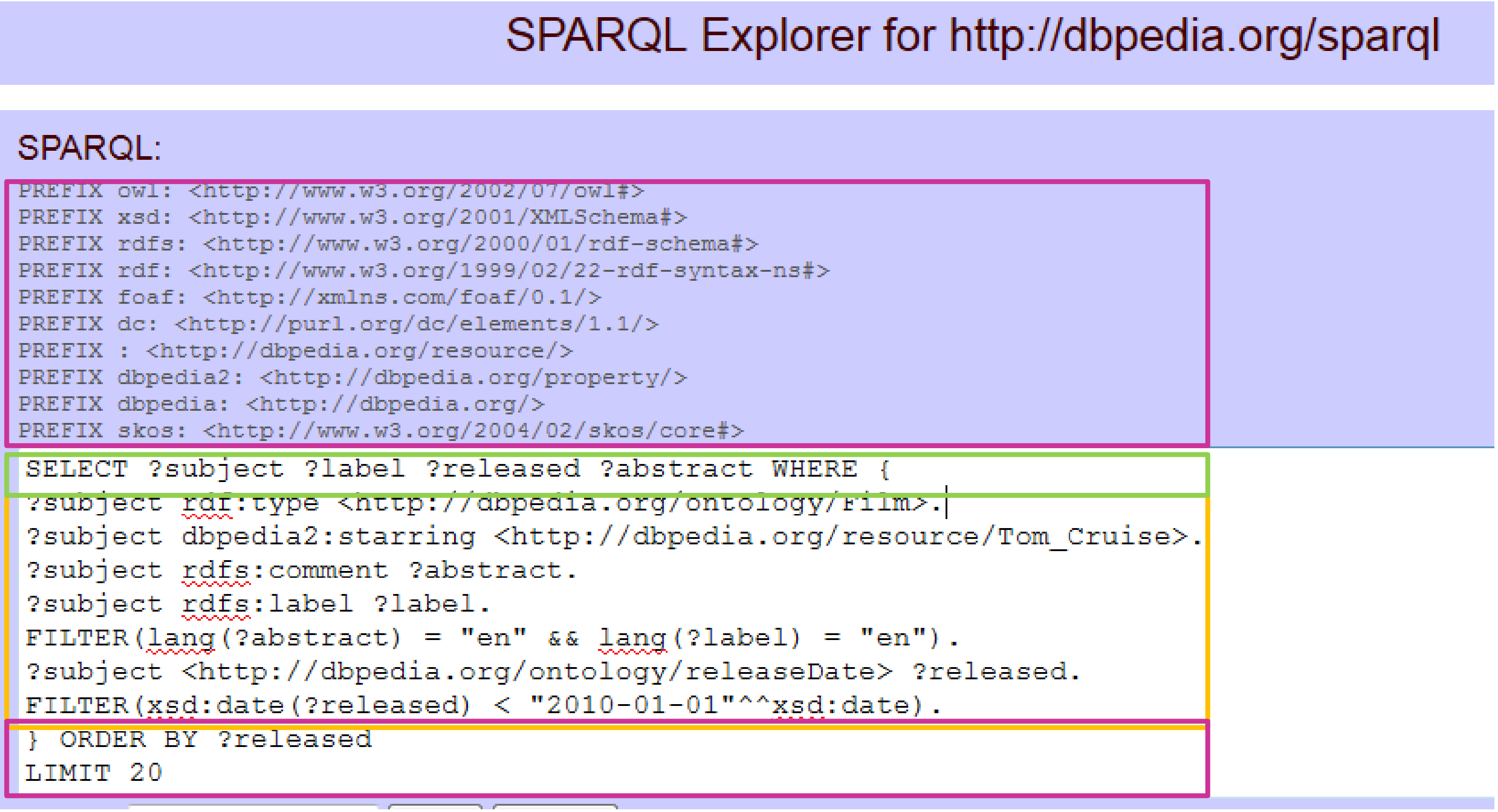
- SPARQL 1.0
  - Language to query RDF data stores
  - Results are in XML format
  - Protocol to exchange SPARQL queries between client applications and a SPARQL query processor
- SPARQL 1.1
  - Updates : SPARUL or SPARQL/Update
  - Compatibility with SPARQL1.0
  - Explicit management of RDF graphs
  - More functions
- Running SPARQL queries: 2 ways
  - On-line command on a client site
  - Interaction protocol with an RDF data server (SPARQL endpoint) thanks to the SPARQL/HTTP protocol

SPARQL endPoints : examples

- Online access
  https://wiki.dbpedia.org/OnlineAccess
- Faceted search http://dbpedia.org/fct/
- Endpoint http://dbpedia.org/sparql/

DBPedia Endpoint: query

http://dbpedia.org/sparql



SPARQL Explorer for http://dbpedia.org/sparql

SPARQL:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT ?subject ?label ?released ?abstract WHERE {
?subject rdf:type <http://dbpedia.org/ontology/Film>.
?subject dbpedia2:starring <http://dbpedia.org/resource/Tom_Cruise>.
?subject rdfs:comment ?abstract.
?subject rdfs:label ?label.
FILTER(lang(?abstract) = "en" && lang(?label) = "en").
?subject <http://dbpedia.org/ontology/releaseDate> ?released.
FILTER(xsd:date(?released) < "2010-01-01"^^xsd:date).
} ORDER BY ?released
LIMIT 20
```

Warning: DbPedia changed > query in 2021

```
SELECT DISTINCT ?film ?label ?titre ?abstract ?date
WHERE
{  ?film dbo:starring <http://dbpedia.org/resource/Tom_Cruise> .
  <http://dbpedia.org/resource/Tom_Cruise> rdfs:label ?label .
   ?film a dbo:Film .
   ?film rdfs:comment ?abstract .
   ?film rdfs:label ?titre .
   optional { ?film dbo:releaseDate ?date . }
   filter (lang(?titre) = 'en' and lang(?abstract) = 'en'  and lang(?label) = 'en' ).
}
```

No more dbo:releaseDate

SPARQL: structure of a query

- Namespace declaration

  PREFIX pref: <http://www.exemple.com/ressources#> **...**

- Expected result (variables)

  SELECT …?subject ?label

- Pattern query definition with searched criteria: graph pattern and fiilters

  WHERE {

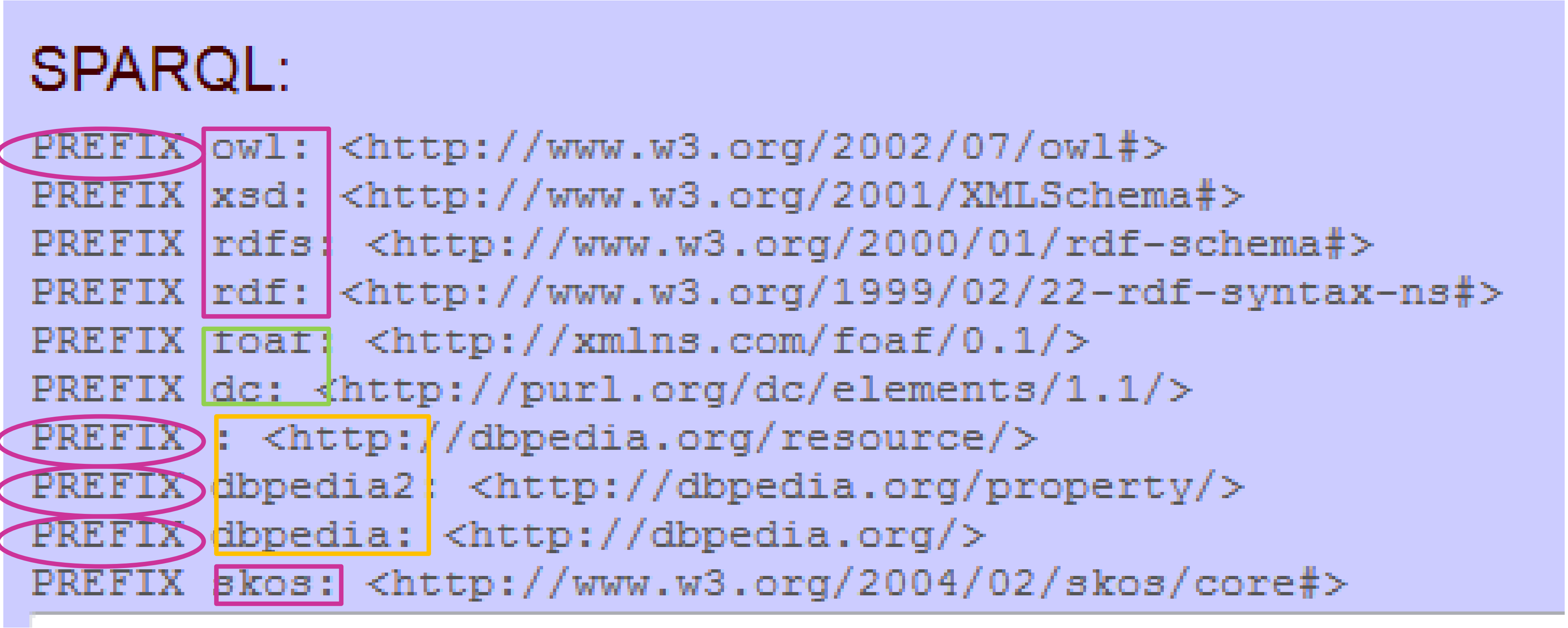
  …

  }

- Browsing / filtering results

  ORDER BY ... LIMIT ..

SPARQL: query prefixes

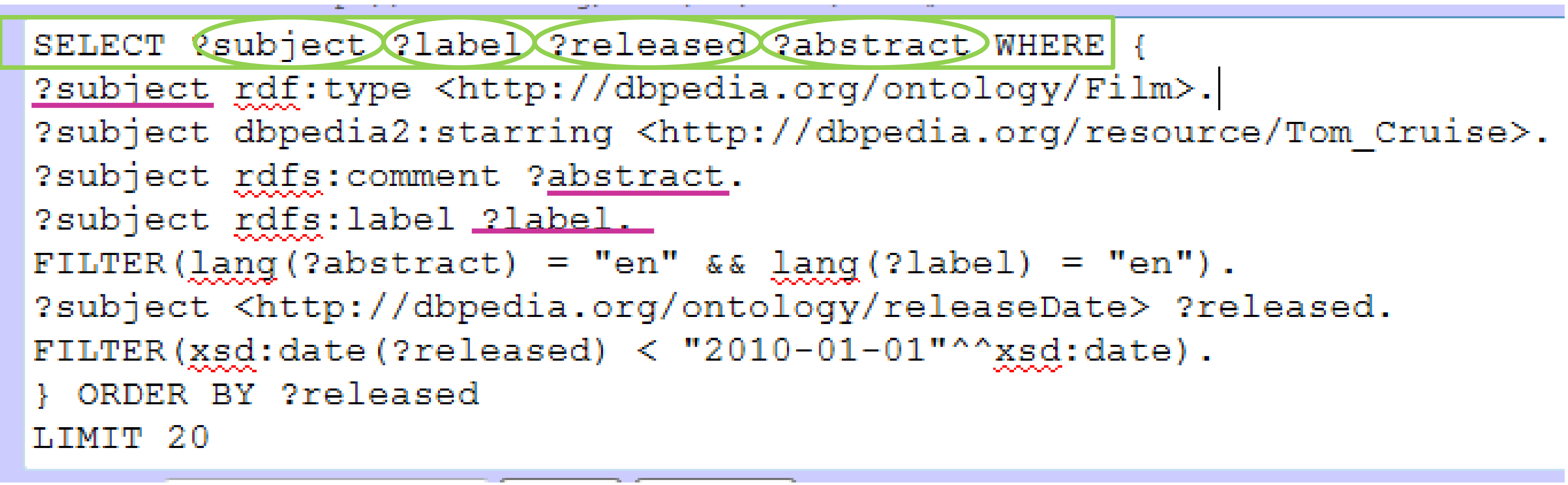


SPARQL:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

- PREFIX
  - Namespace declarations : W3C standards, data stores and standard “ontologies”

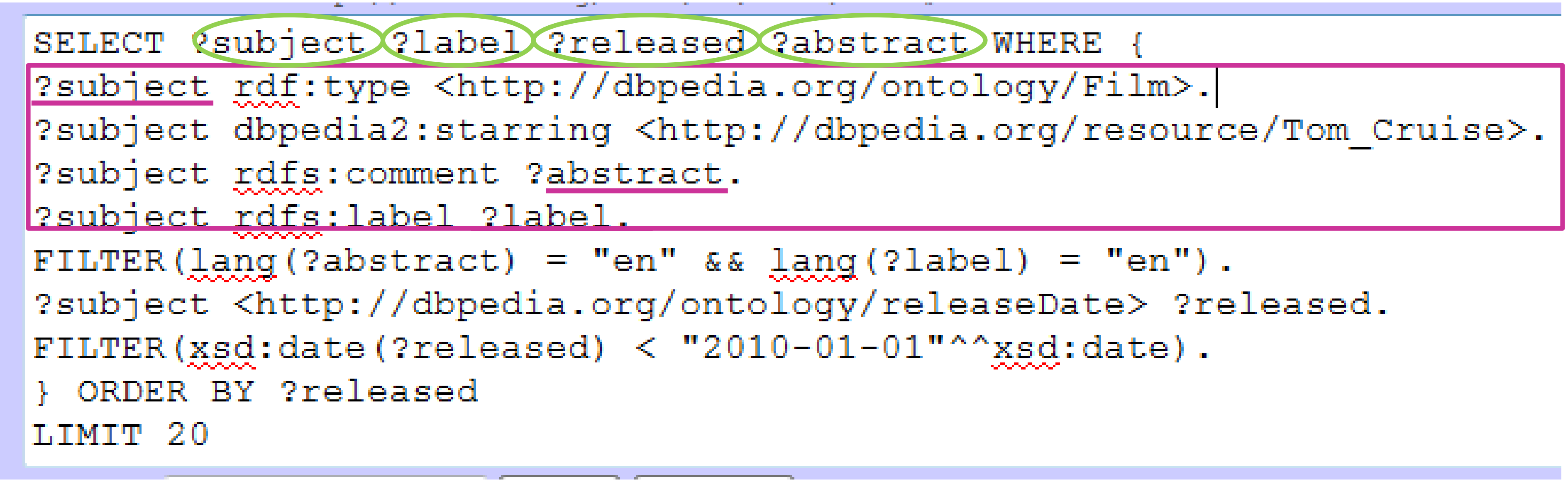
SPARQL: structure of a query



```
SELECT ?subject ?label ?released ?abstract WHERE {
?subject rdf:type <http://dbpedia.org/ontology/Film>.
?subject dbpedia2:starring <http://dbpedia.org/resource/Tom_Cruise>.
?subject rdfs:comment ?abstract.
?subject rdfs:label ?label.
FILTER(lang(?abstract) = "en" && lang(?label) = "en").
?subject <http://dbpedia.org/ontology/releaseDate> ?released.
FILTER(xsd:date(?released) < "2010-01-01"^^xsd:date).
} ORDER BY ?released
LIMIT 20
```

. SELECT : list of variables to be displayed as results

Structure of a query



```
SELECT ?subject ?label ?released ?abstract WHERE {
?subject rdf:type <http://dbpedia.org/ontology/Film>.
?subject dbpedia2:starring <http://dbpedia.org/resource/Tom_Cruise>.
?subject rdfs:comment ?abstract.
?subject rdfs:label ?label.
FILTER(lang(?abstract) = "en" && lang(?label) = "en").
?subject <http://dbpedia.org/ontology/releaseDate> ?released.
FILTER(xsd:date(?released) < "2010-01-01"^^xsd:date).
} ORDER BY ?released
LIMIT 20
```

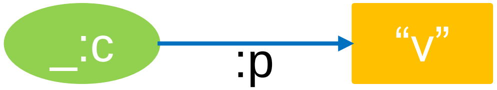
. WHERE

sequence of query patterns = triples with at least one blank node

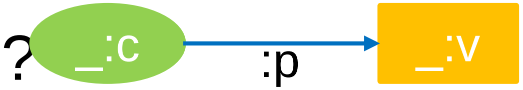
+ FILTERs

SPARQL: structure of a query

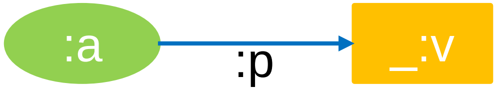
- Graph patterns = triples with empty nodes
  - Which class has value”v” for property :p ?


_:c
:p
“v”

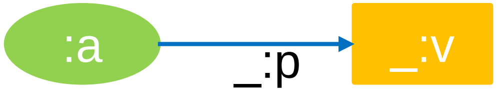
  - Which triples are connected by property :p ?


_:c
:p
_:v

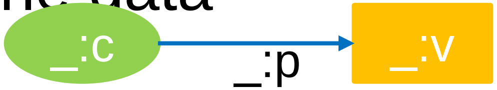
  - Which are the values of property :p for class :a ?


:a
:p
_:v

  - Which are the properties of :a ?


:a
_:p
_:v

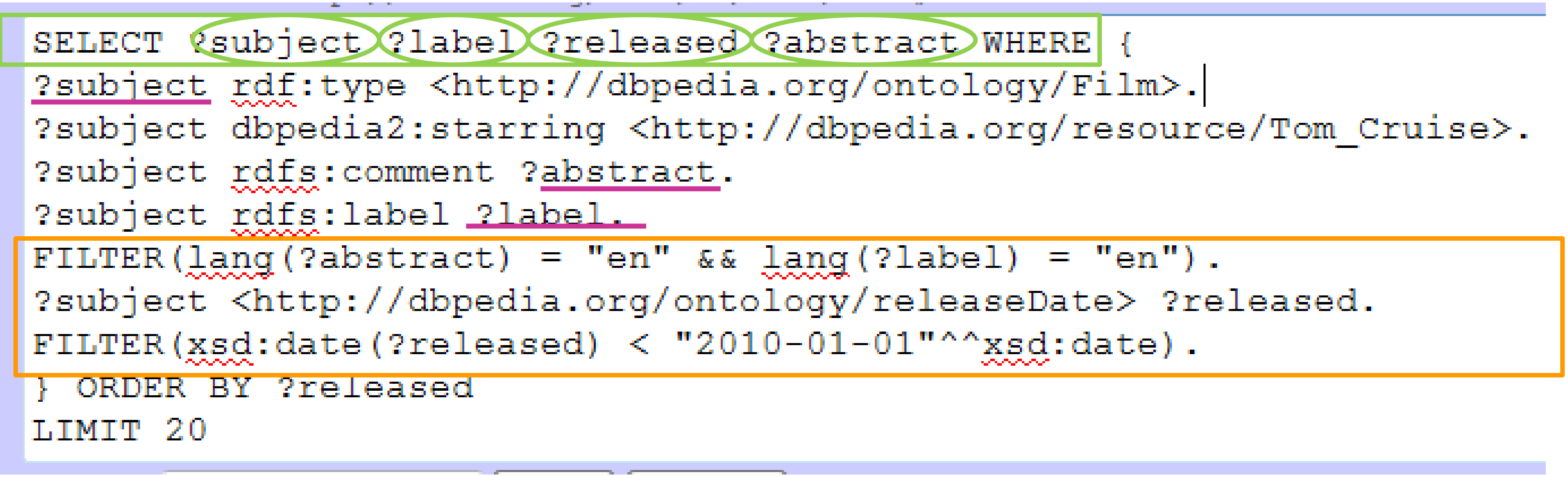
  - Which are all the classes and properties in the data store ?


_:c
_:p
_:v

  - …

DBPedia Endpoint: query

http://dbpedia.org/sparql



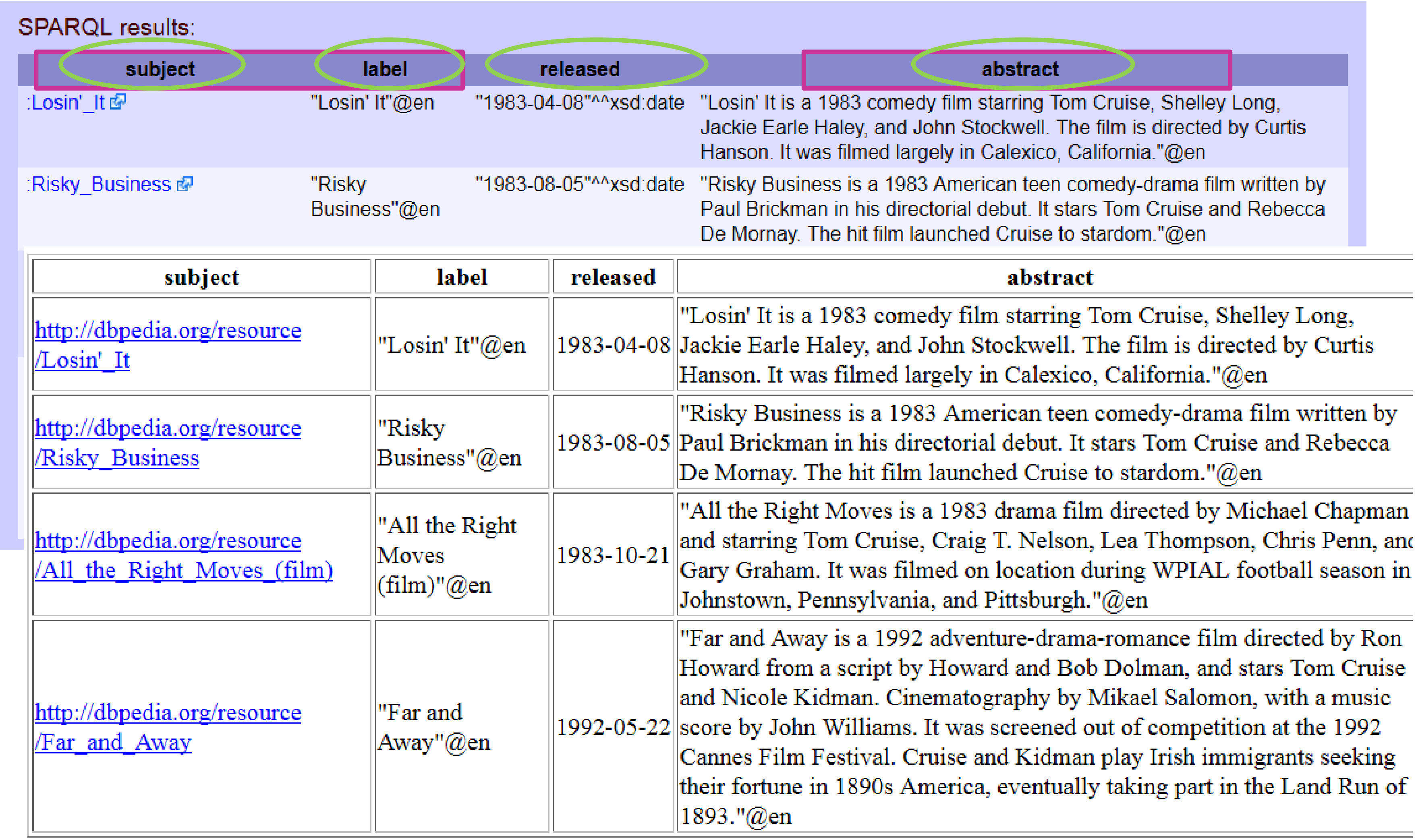
```
SELECT ?subject ?label ?released ?abstract WHERE {
?subject rdf:type <http://dbpedia.org/ontology/Film>.
?subject dbpedia2:starring <http://dbpedia.org/resource/Tom_Cruise>.
?subject rdfs:comment ?abstract.
?subject rdfs:label ?label.
FILTER(lang(?abstract) = "en" && lang(?label) = "en").
?subject <http://dbpedia.org/ontology/releaseDate> ?released.
FILTER(xsd:date(?released) < "2010-01-01"^^xsd:date).
} ORDER BY ?released
LIMIT 20
```

FILTER: uses regular expressions

DBPedia endpoint: result presentations

- XML
  - Predefined vocabulary to build table results
  - Can be converted into HTML thanks to an XSLT translation
- JSON (JavaScript Object Notation)
  - Ligh interchange format for data in web applications (Javascript)
- RDF
  - Results can be serialized as RDF triples
  - Storage in any RDF format: RDF/XML, N-Triples, Turtle

SPARQL query: results presentation



SPARQL results:

subject	label	released	abstract
:Losin'_It	"Losin' It"@en	"1983-04-08"^^xsd:date	"Losin' It is a 1983 comedy film starring Tom Cruise, Shelley Long, Jackie Earle Haley, and John Stockwell. The film is directed by Curtis Hanson. It was filmed largely in Calexico, California."@en
:Risky_Business	"Risky Business"@en	"1983-08-05"^^xsd:date	"Risky Business is a 1983 American teen comedy-drama film written by Paul Brickman in his directorial debut. It stars Tom Cruise and Rebecca De Mornay. The hit film launched Cruise to stardom."@en

subject	label	released	abstract
http://dbpedia.org/resource/Losin'_It	"Losin' It"@en	1983-04-08	"Losin' It is a 1983 comedy film starring Tom Cruise, Shelley Long, Jackie Earle Haley, and John Stockwell. The film is directed by Curtis Hanson. It was filmed largely in Calexico, California."@en
http://dbpedia.org/resource/Risky_Business	"Risky Business"@en	1983-08-05	"Risky Business is a 1983 American teen comedy-drama film written by Paul Brickman in his directorial debut. It stars Tom Cruise and Rebecca De Mornay. The hit film launched Cruise to stardom."@en
http://dbpedia.org/resource/All_the_Right_Moves_(film)	"All the Right Moves (film)"@en	1983-10-21	"All the Right Moves is a 1983 drama film directed by Michael Chapman and starring Tom Cruise, Craig T. Nelson, Lea Thompson, Chris Penn, anc Gary Graham. It was filmed on location during WPIAL football season in Johnstown, Pennsylvania, and Pittsburgh."@en
http://dbpedia.org/resource/Far_and_Away	"Far and Away"@en	1992-05-22	"Far and Away is a 1992 adventure-drama-romance film directed by Ron Howard from a script by Howard and Bob Dolman, and stars Tom Cruise and Nicole Kidman. Cinematography by Mikael Salomon, with a music score by John Williams. It was screened out of competition at the 1992 Cannes Film Festival. Cruise and Kidman play Irish immigrants seeking their fortune in 1890s America, eventually taking part in the Land Run of 1893."@en

How to write a SPARQL query?

SPARQL: structure of a query (reminder)

- Namespace declaration

  PREFIX pref: <http://www.exemple.com/ressources#> **...**

- Expected result

  SELECT ...

- Pattern query definition with searched criteria: graph pattern

  WHERE {

  …

  }

- Browsing / filtering results

  FILTER … (inside WHERE)

  ORDER BY ... (after WHERE)

Search for instances

- Let's consider swl an ontology with 3 classes :
  swl:Man, swl:Woman, swl:HumanBeing

- Resource with an unknown URI
  swl:x rdf:type swl:Man
  swl:y rdf:type swl:Woman

- RDFs data
  swl:Man   rdfs:subClassOf swl:HumanBeing
  swl:Woman   rdfs:subClassOf swl:HumanBeing

- SPARQL query
  SELECT _:h WHERE { _:h rdf:type swl:HumanBeing }

_:h
swl:x
swl:y

Search for instances

- Resource with an unknown URI

```
swl:x  rdf:type  swl:Man
swl:x  foaf:firstName ”Thomas”@fr

swl:y  rdf:type  swl:Woman
swl:y  foaf:firstName  ”Elise”@fr
```

- RDFs data

```
swl:Man          rdfs:subClassOf  swl:HumanBeing
swl:Woman        rdfs:subClassOf  swl:HumanBeing
```

- SPARQL query

```
SELECT _:h _:name WHERE {
    _:h            rdf:type               swl:HumanBeing
    _:h            foaf:firstName          _:name }
```

_:h	_:name
swl:x	”Thomas”@fr
swl:y	”Elise”@fr

Searching for classes

- RDFS data

```
swl:Man  rdfs:subClassOf  swl:HumanBeing
swl:Woman  rdfs:subClassOf  swl:HumanBeing
swl:HumanBeing rdfs:subClassOf foaf:person
```

- SPARQL

```
SELECT _:c WHERE { _:c rdfs:subClassOf foaf:person }
```

_:c
swl:HumanBeing
swl:Man
swl:Woman

More complex queries: SELECT

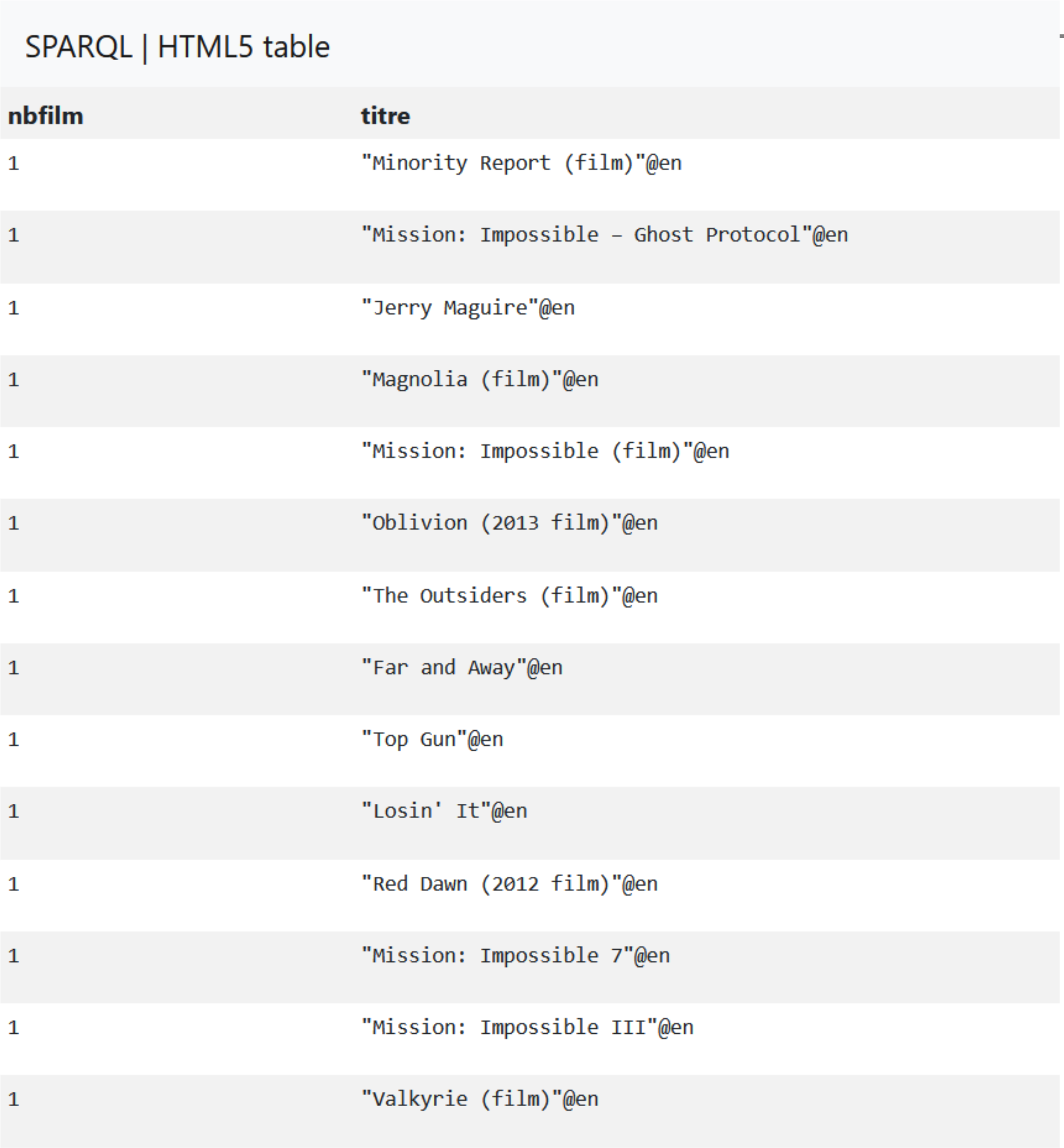
- SELECT DISTINCT variable
  - Suppresses duplicate instances of ”variable” that are part of result triples

- SELECT (COUNT (DISTINCT ?var) AS ?nbvar)
  - Counts the number of (dictinct) entities represented as ?var and satisfying the request ; this value is ?nbvar

- Other operators : MIN(), MAX(), SUM(), AVG()

- Using COUNT(), MIN(), MAX(), SUM(), AVG() with GROUP BY can produce summary values for groups of triples

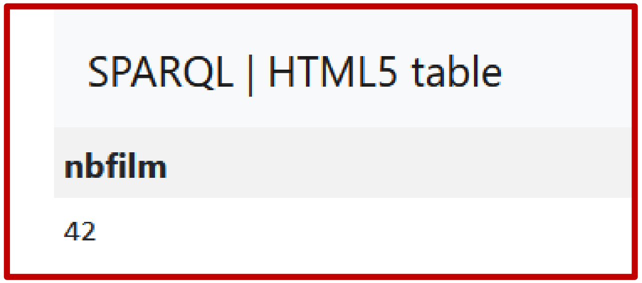
```
SELECT (COUNT (distinct ?film) AS ?nbfilm) ?titre
WHERE
{  ?film dbo:starring <http://dbpedia.org/resource/Tom_Cruise> .
  <http://dbpedia.org/resource/Tom_Cruise> rdfs:label ?label .
  ?film a dbo:Film .
  ?film rdfs:comment ?abstract .
  ?film rdfs:label ?titre .
filter (lang(?titre) = 'en' and lang(?abstract) = 'en'  and lang(?label) =
'en' ).
}
```



SPARQL | HTML5 table

nbfilm	titre
1	"Minority Report (film)"@en
1	"Mission: Impossible – Ghost Protocol"@en
1	"Jerry Maguire"@en
1	"Magnolia (film)"@en
1	"Mission: Impossible (film)"@en
1	"Oblivion (2013 film)"@en
1	"The Outsiders (film)"@en
1	"Far and Away"@en
1	"Top Gun"@en
1	"Losin' It"@en
1	"Red Dawn (2012 film)"@en
1	"Mission: Impossible 7"@en
1	"Mission: Impossible III"@en
1	"Valkyrie (film)"@en

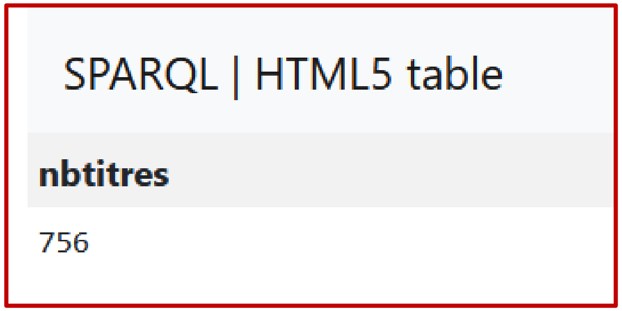
```
SELECT (COUNT (distinct ?film) AS ?nbfilm)
WHERE
{  ?film dbo:starring <http://dbpedia.org/resource/Tom_Cruise> .
  <http://dbpedia.org/resource/Tom_Cruise> rdfs:label ?label .
  ?film a dbo:Film .
?film rdfs:comment ?abstract .
?film rdfs:label ?titre .
optional { ?film dbo:releaseDate ?date . }
filter (lang(?titre) = 'en' and lang(?abstract) = 'en'  and
lang(?label) = 'en' ).
}
```



SPARQL | HTML5 table

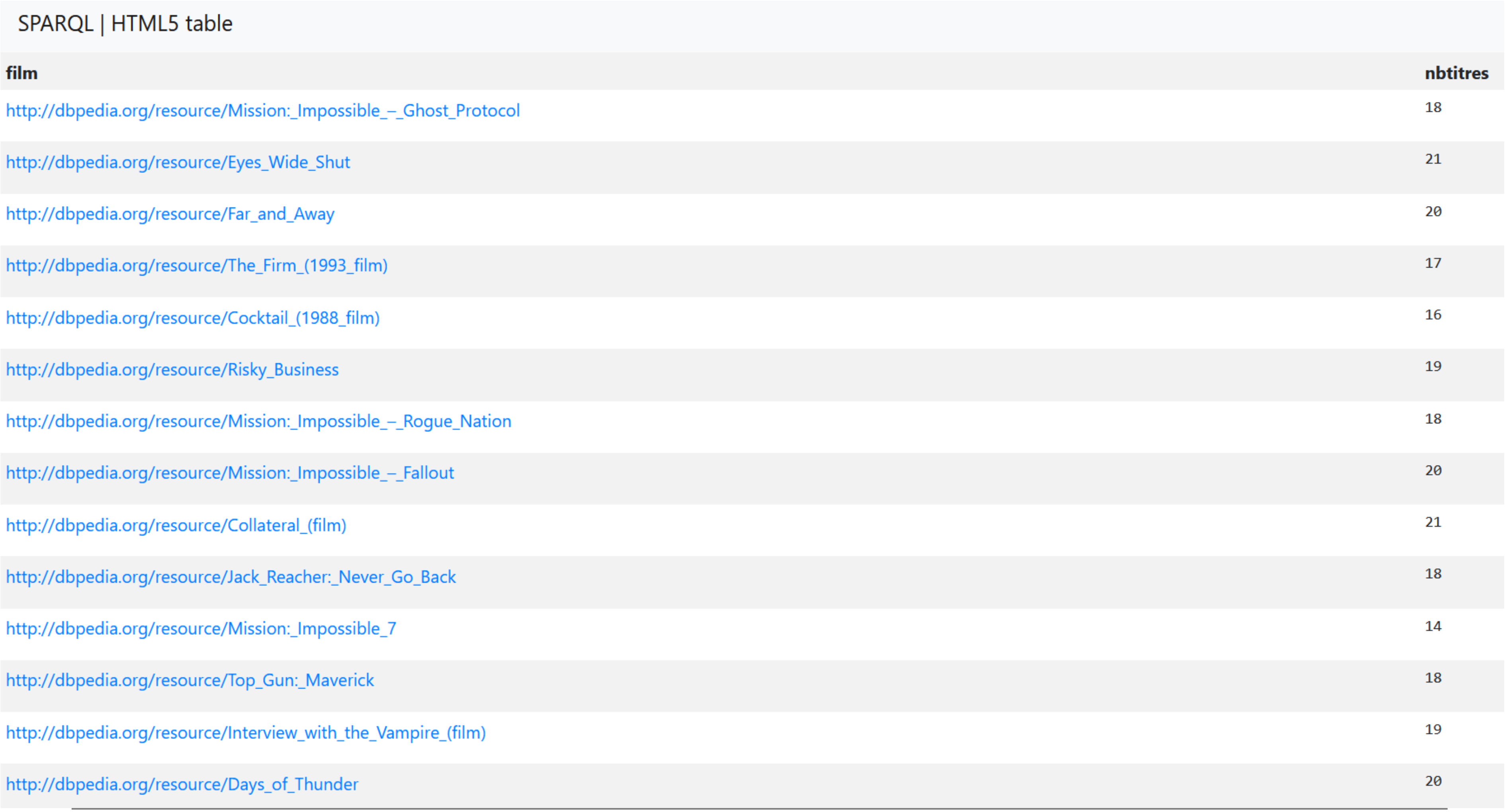
nbfilm
42

```
SELECT (COUNT (distinct ?titre) AS ?titres)
WHERE
{  ?film dbo:starring <http://dbpedia.org/resource/Tom_Cruise> .
  <http://dbpedia.org/resource/Tom_Cruise> rdfs:label ?label .
  ?film a dbo:Film .
?film rdfs:comment ?abstract .
?film rdfs:label ?titre .
}
```


SPARQL | HTML5 table

nbtitres
756

```
SELECT ?film (COUNT (distinct ?titre) AS ?nbtitres)
WHERE
{  ?film dbo:starring
<http://dbpedia.org/resource/Tom_Cruise> .
  <http://dbpedia.org/resource/Tom_Cruise> rdfs:label
?label .
?film a dbo:Film .
?film rdfs:comment ?abstract .
?film rdfs:label ?titre .
}
GROUP BY (?film)
```



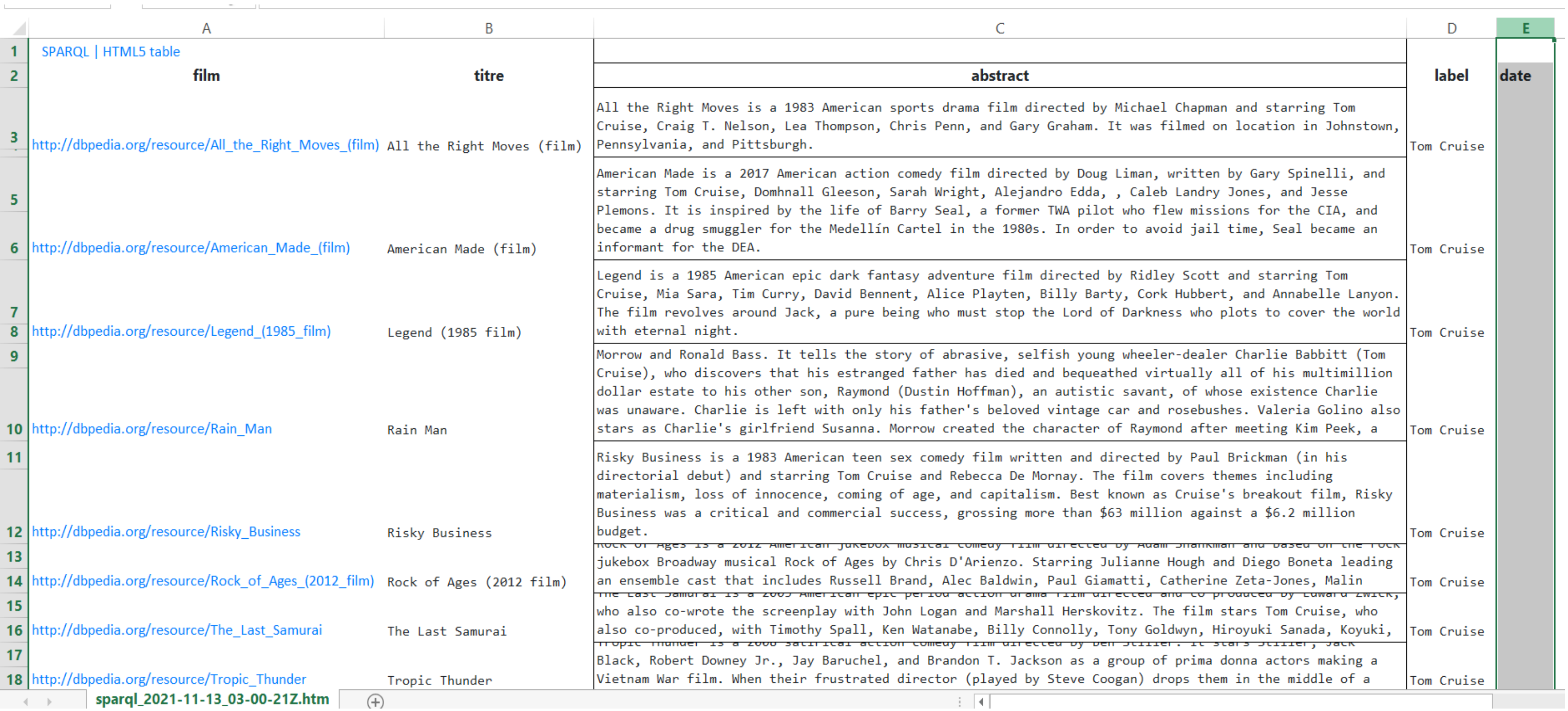
SPARQL | HTML5 table

film	nbtitres
http://dbpedia.org/resource/Mission:_Impossible_–_Ghost_Protocol	18
http://dbpedia.org/resource/Eyes_Wide_Shut	21
http://dbpedia.org/resource/Far_and_Away	20
http://dbpedia.org/resource/The_Firm_(1993_film)	17
http://dbpedia.org/resource/Cocktail_(1988_film)	16
http://dbpedia.org/resource/Risky_Business	19
http://dbpedia.org/resource/Mission:_Impossible_–_Rogue_Nation	18
http://dbpedia.org/resource/Mission:_Impossible_–_Fallout	20
http://dbpedia.org/resource/Collateral_(film)	21
http://dbpedia.org/resource/Jack_Reacher:_Never_Go_Back	18
http://dbpedia.org/resource/Mission:_Impossible_7	14
http://dbpedia.org/resource/Top_Gun:_Maverick	18
http://dbpedia.org/resource/Interview_with_the_Vampire_(film)	19
http://dbpedia.org/resource/Days_of_Thunder	20

OPTIONAL

If a variable is empty, the whole line is not displayed.
> Use OPTIONAL to allow to display a line of variables including empty values

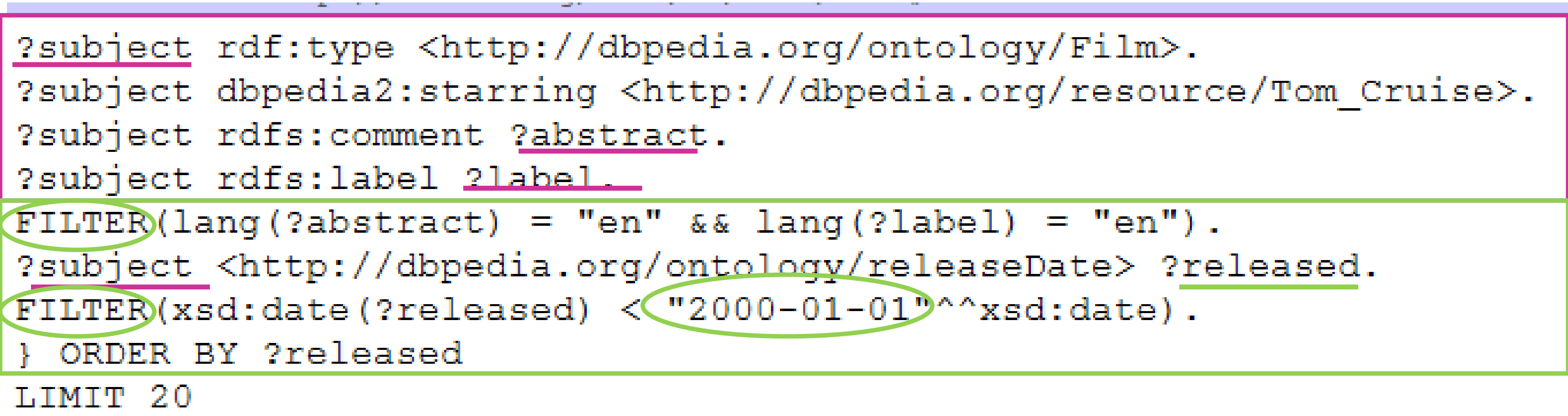
```
SELECT ?film ?titre ?abstract ?label ?date
WHERE
{  ?film dbo:starring <http://dbpedia.org/resource/Tom_Cruise> .
  <http://dbpedia.org/resource/Tom_Cruise> rdfs:label ?label .
  ?film a dbo:Film .
?film rdfs:comment ?abstract .
?film rdfs:label ?titre .
OPTIONAL { ?film dbo:releaseDate ?date . }
filter (lang(?titre) = 'en' and lang(?abstract) = 'en'  and lang(?label) =
'en' ).
}
```



	A	B	C	D	E
1	SPARQL | HTML5 table				
2	**film**	**titre**	**abstract**	**label**	**date**
3	http://dbpedia.org/resource/All_the_Right_Moves_(film)	All the Right Moves (film)	All the Right Moves is a 1983 American sports drama film directed by Michael Chapman and starring Tom Cruise, Craig T. Nelson, Lea Thompson, Chris Penn, and Gary Graham. It was filmed on location in Johnstown, Pennsylvania, and Pittsburgh.	Tom Cruise	
5 6	http://dbpedia.org/resource/American_Made_(film)	American Made (film)	American Made is a 2017 American action comedy film directed by Doug Liman, written by Gary Spinelli, and starring Tom Cruise, Domhnall Gleeson, Sarah Wright, Alejandro Edda, , Caleb Landry Jones, and Jesse Plemons. It is inspired by the life of Barry Seal, a former TWA pilot who flew missions for the CIA, and became a drug smuggler for the Medellín Cartel in the 1980s. In order to avoid jail time, Seal became an informant for the DEA.	Tom Cruise	
7 8	http://dbpedia.org/resource/Legend_(1985_film)	Legend (1985 film)	Legend is a 1985 American epic dark fantasy adventure film directed by Ridley Scott and starring Tom Cruise, Mia Sara, Tim Curry, David Bennent, Alice Playten, Billy Barty, Cork Hubbert, and Annabelle Lanyon. The film revolves around Jack, a pure being who must stop the Lord of Darkness who plots to cover the world with eternal night.	Tom Cruise	
9 10	http://dbpedia.org/resource/Rain_Man	Rain Man	Morrow and Ronald Bass. It tells the story of abrasive, selfish young wheeler-dealer Charlie Babbitt (Tom Cruise), who discovers that his estranged father has died and bequeathed virtually all of his multimillion dollar estate to his other son, Raymond (Dustin Hoffman), an autistic savant, of whose existence Charlie was unaware. Charlie is left with only his father's beloved vintage car and rosebushes. Valeria Golino also stars as Charlie's girlfriend Susanna. Morrow created the character of Raymond after meeting Kim Peek, a	Tom Cruise	
11 12	http://dbpedia.org/resource/Risky_Business	Risky Business	Risky Business is a 1983 American teen sex comedy film written and directed by Paul Brickman (in his directorial debut) and starring Tom Cruise and Rebecca De Mornay. The film covers themes including materialism, loss of innocence, coming of age, and capitalism. Best known as Cruise's breakout film, Risky Business was a critical and commercial success, grossing more than $63 million against a $6.2 million budget.	Tom Cruise	
13 14	http://dbpedia.org/resource/Rock_of_Ages_(2012_film)	Rock of Ages (2012 film)	Rock of Ages is a 2012 American jukebox musical comedy film directed by Adam Shankman and based on the rock jukebox Broadway musical Rock of Ages by Chris D'Arienzo. Starring Julianne Hough and Diego Boneta leading an ensemble cast that includes Russell Brand, Alec Baldwin, Paul Giamatti, Catherine Zeta-Jones, Malin	Tom Cruise	
15 16	http://dbpedia.org/resource/The_Last_Samurai	The Last Samurai	The Last Samurai is a 2003 American epic period action drama film directed and co-produced by Edward Zwick, who also co-wrote the screenplay with John Logan and Marshall Herskovitz. The film stars Tom Cruise, who also co-produced, with Timothy Spall, Ken Watanabe, Billy Connolly, Tony Goldwyn, Hiroyuki Sanada, Koyuki,	Tom Cruise	
17 18	http://dbpedia.org/resource/Tropic_Thunder	Tropic Thunder	Tropic Thunder is a 2008 satirical action comedy film directed by Ben Stiller. It stars Stiller, Jack Black, Robert Downey Jr., Jay Baruchel, and Brandon T. Jackson as a group of prima donna actors making a Vietnam War film. When their frustrated director (played by Steve Coogan) drops them in the middle of a	Tom Cruise	

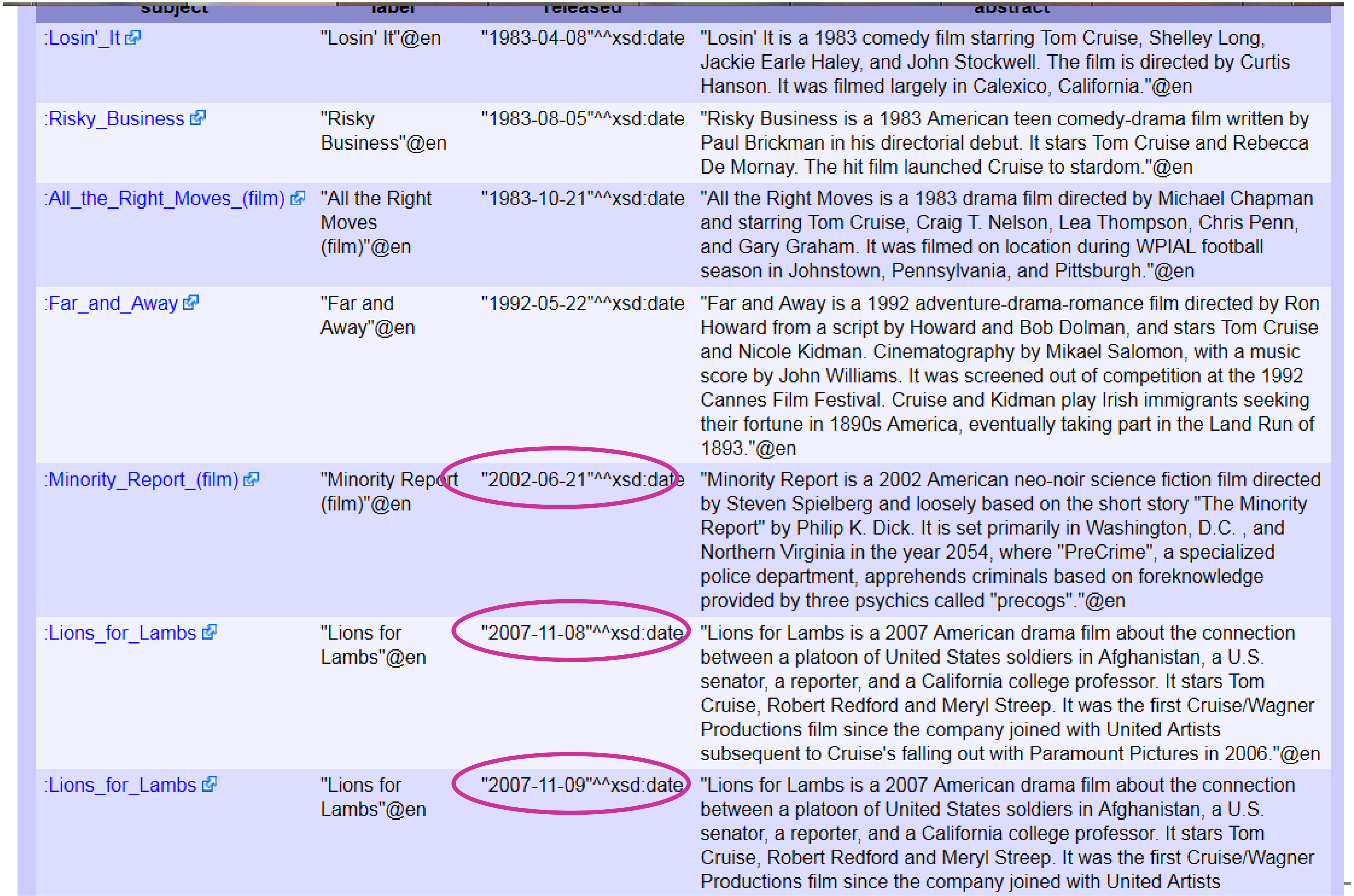
More complex queries: FILTER

- Result filtering using regular expressions involving one or several variables
- Can be inserted between query patterns (triples) atfer the first use of the variables they use



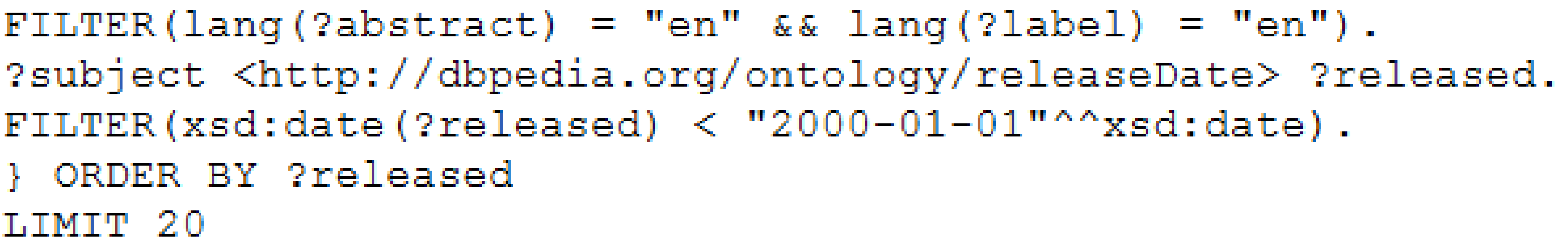
```
?subject rdf:type <http://dbpedia.org/ontology/Film>.
?subject dbpedia2:starring <http://dbpedia.org/resource/Tom_Cruise>.
?subject rdfs:comment ?abstract.
?subject rdfs:label ?label.
FILTER(lang(?abstract) = "en" && lang(?label) = "en").
?subject <http://dbpedia.org/ontology/releaseDate> ?released.
FILTER(xsd:date(?released) < "2000-01-01"^^xsd:date).
} ORDER BY ?released
LIMIT 20
```

DBPedia Endpoint: more results



subject	label	released	abstract
:Losin'_It	"Losin' It"@en	"1983-04-08"^^xsd:date	"Losin' It is a 1983 comedy film starring Tom Cruise, Shelley Long, Jackie Earle Haley, and John Stockwell. The film is directed by Curtis Hanson. It was filmed largely in Calexico, California."@en
:Risky_Business	"Risky Business"@en	"1983-08-05"^^xsd:date	"Risky Business is a 1983 American teen comedy-drama film written by Paul Brickman in his directorial debut. It stars Tom Cruise and Rebecca De Mornay. The hit film launched Cruise to stardom."@en
:All_the_Right_Moves_(film)	"All the Right Moves (film)"@en	"1983-10-21"^^xsd:date	"All the Right Moves is a 1983 drama film directed by Michael Chapman and starring Tom Cruise, Craig T. Nelson, Lea Thompson, Chris Penn, and Gary Graham. It was filmed on location during WPIAL football season in Johnstown, Pennsylvania, and Pittsburgh."@en
:Far_and_Away	"Far and Away"@en	"1992-05-22"^^xsd:date	"Far and Away is a 1992 adventure-drama-romance film directed by Ron Howard from a script by Howard and Bob Dolman, and stars Tom Cruise and Nicole Kidman. Cinematography by Mikael Salomon, with a music score by John Williams. It was screened out of competition at the 1992 Cannes Film Festival. Cruise and Kidman play Irish immigrants seeking their fortune in 1890s America, eventually taking part in the Land Run of 1893."@en
:Minority_Report_(film)	"Minority Report (film)"@en	"2002-06-21"^^xsd:date	"Minority Report is a 2002 American neo-noir science fiction film directed by Steven Spielberg and loosely based on the short story "The Minority Report" by Philip K. Dick. It is set primarily in Washington, D.C. , and Northern Virginia in the year 2054, where "PreCrime", a specialized police department, apprehends criminals based on foreknowledge provided by three psychics called "precogs"."@en
:Lions_for_Lambs	"Lions for Lambs"@en	"2007-11-08"^^xsd:date	"Lions for Lambs is a 2007 American drama film about the connection between a platoon of United States soldiers in Afghanistan, a U.S. senator, a reporter, and a California college professor. It stars Tom Cruise, Robert Redford and Meryl Streep. It was the first Cruise/Wagner Productions film since the company joined with United Artists subsequent to Cruise's falling out with Paramount Pictures in 2006."@en
:Lions_for_Lambs	"Lions for Lambs"@en	"2007-11-09"^^xsd:date	"Lions for Lambs is a 2007 American drama film about the connection between a platoon of United States soldiers in Afghanistan, a U.S. senator, a reporter, and a California college professor. It stars Tom Cruise, Robert Redford and Meryl Streep. It was the first Cruise/Wagner Productions film since the company joined with United Artists

SPARQL: using FILTER



```
FILTER(lang(?abstract) = "en" && lang(?label) = "en").
?subject <http://dbpedia.org/ontology/releaseDate> ?released.
FILTER(xsd:date(?released) < "2000-01-01"^^xsd:date).
} ORDER BY ?released
LIMIT 20
```

- FILTER (constraints on variables)
  - lang(?abstract) = ‘en’
  - xsd:date(?released) < “2000-01-01”^^xsd:date
  - Attribut(?var) = ‘value’
  - Conjunction of constraints &&

SPARQL: FILTER

- Operators and expressions
  - Logical operators: **!** non**, &&** et **, ||** ou
  - Mathematical operators: **+, -, *, /**
  - Comparison operators: **=, !=, >, <, ...**
  - Test functions: isURI, isBlank, isLiteral, bound, str, lang, datatype, sameTerm, langMatches, regex
- Regular expression to filter litterals: **regex**
- **http://www.w3.org/TR/xpath-functions/#regex-syntax**
  - regex(?name,"s“) : string containing “s”
  - regex(?name,"**^s“)** : string starting by “s”
  - regex(?name,“s**$“)** : string ending with“s”

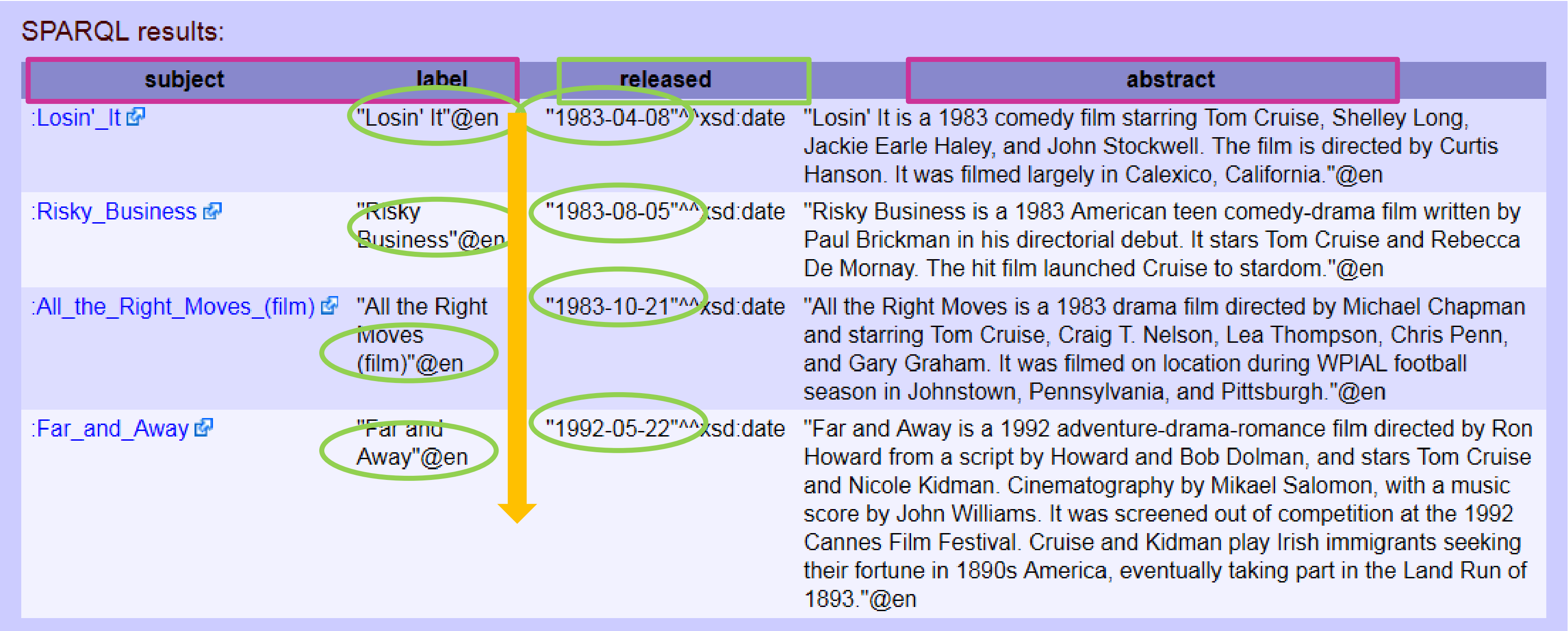
```
Example: Movies with « Man » in their label

SELECT ?film  ?label
WHERE {        ?film rdf:type <http://dbpedia.org/ontology/Film> .
               ?film rdfs:label ?label .
               FILTER regex(?label, "Man"). }
```

```
select distinct ?Concept ?Type
where
{[] a ?Concept.
?Concept rdf:type ?Type .
FILTER (?Type != owl:Class).
}
LIMIT 100
```

Organizing query results

- GROUP BY ?var
  - Results are displayed grouped by the selected variable
- ORDER BY
  - Result classification using one or several variables
- LIMIT n
  - Display the n first results
- OFFSET (when LIMIT and ORDER are used)
  - Page presentation of the results

Organizing query results: ORDER BY



SPARQL results:

subject	label	released	abstract
:Losin'_It	"Losin' It"@en	"1983-04-08"^^xsd:date	"Losin' It is a 1983 comedy film starring Tom Cruise, Shelley Long, Jackie Earle Haley, and John Stockwell. The film is directed by Curtis Hanson. It was filmed largely in Calexico, California."@en
:Risky_Business	"Risky Business"@en	"1983-08-05"^^xsd:date	"Risky Business is a 1983 American teen comedy-drama film written by Paul Brickman in his directorial debut. It stars Tom Cruise and Rebecca De Mornay. The hit film launched Cruise to stardom."@en
:All_the_Right_Moves_(film)	"All the Right Moves (film)"@en	"1983-10-21"^^xsd:date	"All the Right Moves is a 1983 drama film directed by Michael Chapman and starring Tom Cruise, Craig T. Nelson, Lea Thompson, Chris Penn, and Gary Graham. It was filmed on location during WPIAL football season in Johnstown, Pennsylvania, and Pittsburgh."@en
:Far_and_Away	"Far and Away"@en	"1992-05-22"^^xsd:date	"Far and Away is a 1992 adventure-drama-romance film directed by Ron Howard from a script by Howard and Bob Dolman, and stars Tom Cruise and Nicole Kidman. Cinematography by Mikael Salomon, with a music score by John Williams. It was screened out of competition at the 1992 Cannes Film Festival. Cruise and Kidman play Irish immigrants seeking their fortune in 1890s America, eventually taking part in the Land Run of 1893."@en

SPARQL: selecting a subset of data with named graphs: FROM NAMED, GRAPH

- FROM *RDF_source*
  - To query the default graph
- FROM NAMED *RDF_source*
  - To query graphs that are not in the source data set

```
# Default graph (located at http://example.org/foaf/aliceFoaf)
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
```

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

SELECT ?name FROM <http://example.org/foaf/aliceFoaf>
WHERE { ?x foaf:name ?name }
```

SPARQL: selecting a subset of data with named graphs: FROM NAMED, GRAPH

```
# Graph: http://example.org/bob
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Bob" .
_:a foaf:mbox <mailto:bob@oldcorp.example.org> .
_:a foaf:nick "Robert" .
```

```
...
FROM NAMED <http://example.org/alice>
FROM NAMED <http://example.org/bob>
...
```

SPARQL: selecting a subset of data with named graphs: FROM NAMED, GRAPH

- GRAPH *RDF source*

```
# Default graph (located at http://example.org/foaf/aliceFoaf)
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
_:a foaf:nick "Bobby" .
```

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

SELECT ?src ?bobNick
FROM NAMED <http://example.org/foaf/aliceFoaf>
FROM NAMED <http://example.org/foaf/bobFoaf>
WHERE { GRAPH ?src
        {        ?x foaf:nick ?bobNick } }
```

src	bobNick
<http://example.org/foaf/aliceFoaf>	« Bobby »
<http://example.org/foaf/bobFoaf>	« Robert »

SPARQL: result modification: OPTIONAL

- Goals
  - search for results where some information is missing
  - Search for data where the specified information is not known
- OPTIONAL (?x :p ?l)
  - If property :p is known, then it is displayed
  - Else, other results are displayed but not the property value

```
@prefix foaf:
<http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .
_:b foaf:name "Bob" .
_:c foaf:name "Claire" .
_:c foaf:nick "CT" .
```

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

SELECT ?nameX ?nameY ?nickY
WHERE
       { ?x foaf:knows ?y ;
            foaf:name ?nameX .
         ?y foaf:name ?nameY .
         OPTIONAL { ?y foaf:nick ?nickY }
       }
```

nameX	nameY	nickY
"Alice"	"Bob"	
"Alice"	"Claire"	"CT"

SPARQL: Alternatives in a Pattern: UNION

```
@prefix foaf:
<http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .
_:b foaf:name "Bob" .
_:c foaf:name "Claire" .
_:c foaf:nick "CT" .
```

```
SELECT ?name ?nickName
WHERE { { [] foaf:name ?name } UNION { [] foaf:nick ?nickName } }
```

name
"Alice"
"Bob"
"CT"
"Claire"

name	nickName
"Alice"	
"Bob"	
	"CT"
"Claire"	

Querying DBPedia : SPARQL endpoints

- Basic public SPARQL endpoint http://dbpedia.org/sparql
- OpenLink Interactive SPARQL query builder http://dbpedia.org/isparql

Example: Pairs of different movies with the same producer but exclude movies with the same label

```
SELECT distinct ?film ?film2 ?producer
WHERE { ?film       rdf:type <http://dbpedia.org/ontology/Film> ;
                    dbo:producer ?producer .
             OPTIONAL { ?film rdfs:label ?label1 }.
            ?film2     rdf:type <http://dbpedia.org/ontology/Film> ;
                    dbo:producer ?producer .
            OPTIONAL { ?film2  rdfs:label ?label2 } .
            FILTER (?label1 != ?label2 and ?film != ?film2 ) .
            }
```

SPARQL: result modification: BOUND

- bound (?x)
  - Returns true if ?x is bound to a value
- !bound (?film2)
  - returns true if ?film2 is NOT bound to a value

Example: list of movies with the same director if any

In <http://www.dbpedia.org/ontology/>

```
SELECT ?film ?film2 ?label2 ?director
WHERE { ?film       rdf:type dbo:Film .
         ?film       dbo:director ?director.
         ?film2      rdf:type dbo:Film .
         ?film2      dbo:director ?director .
         ?film2      rdfs:label ?label2 .
        FILTER ( lang(?label2) = ‘en’).
        FILTER (?film != ?film2 and BOUND (?director) ) .
}
ORDER BY ?director
LIMIT 100
```

SPARQL Query forms

SELECT

Returns all, or a subset of, the variables bound in a query pattern match.

ASK

Returns a boolean indicating whether a query pattern matches or not.

DESCRIBE

Returns an RDF graph that describes the resources found.

CONSTRUCT

Returns an RDF graph constructed by substituting variables
in a set of triple templates.

SPARQL: ASK query

ASK

- True if the query model has an answer in the data set
- False else

```
@prefix foaf:
<http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .
_:b foaf:name "Bob" .
_:c foaf:name "Claire" .
_:c foaf:nick "CT" .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
ASK { ?x foaf:name "Alice" }
```

```
true
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
ASK { ?x foaf:name "Arthur" }
```

```
false
```

```
<sparql xmlns="http://www.w3.org/2005/sparql-
results#">
   <head></head>
   <results> <boolean>true</boolean> </results>
</sparql>
```

Exemples ASK

https://codyburleson.com/blog/sparql-examples-ask

```
PREFIX prop: <http://dbpedia.org/property/>
ASK
 {
<http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
<http://dbpedia.org/resource/Nile> prop:length ?nile .
FILTER(?amazon > ?nile) .
}
```

```
PREFIX: <http://bedrock/>
ASK
WHERE { :fred :hasChild :?child .
        :?child :hasChild :?grandchild. }
```

WHERE is optional (in all SPARQL query)

SPARQL: DESCRIBE query

DESCRIBE

- returns a single result RDF graph
- Looks for all RDF information (triples) in the data source

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DESCRIBE ?x
WHERE { ?x foaf:name "Alice" }
```

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .
```

SPARQL: using CONSTRUCT queries

```
@prefix : <http://www.snee.com/ns/demo#> .

:jane :hasParent :gene .
:gene :hasParent :pat ;
      :gender :female .
:joan :hasParent :pat ;
      :gender :female .
:pat :gender :male .
:mike :hasParent :joan .
```

```
PREFIX : <http://www.snee.com/ns/demo#>

CONSTRUCT { ?s1 :hasSibling ?s2 . }
WHERE {?s1 :hasParent ?parent .
       ?s2 :hasParent ?parent . }
```

```
@prefix : <http://www.snee.com/ns/demo#> .

:gene :hasSibling :joan .
```

- Write a querry that generates :hasSibling relations

SPARQL: CONSTRUCT builds graphs

- Returns a single RDF graph
- Specified by a graph template (= a triple or a union of triples or a URI)
- More than a query language: you can create new data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@example.org> .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT { <http://example.org/person#Alice> vcard:FN ?name }
WHERE { ?x foaf:name ?name }
```

```
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .

<http://example.org/person#Alice> vcard:FN "Alice" .
```

Libraries to manage semantic web data

- **JENA:** A free and open source Java framework for building Semantic Web and Linked Data applications. http://jena.apache.org/
  - RDF core API
  - **ARQ -** A SPARQL Processor for Jena (query engine on RDF data)
  - **TDB** triple Store for RDF Data
  - **Fuseki** end point on the triple stores
  - **Ontology API and Inference API**

- **Python for semantic web**
  - Use libraries for URL management and JSON data management or CSV data reading using Pandas
  - http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/6148/pdf
  - query = """ SELECT …. WHERE { …} ""«
  - **Owlready2 module** : to manage and build ontologies
  - Rdflib
  - Segaran, T., Evans, C., and Taylor, J. (2009). Programming the Semantic Web. O'Reilly. ISBN 978-0-596-15381-6.
  - Jean-Baptiste LAMY (2019) Python et les ontologies. Editions ENI