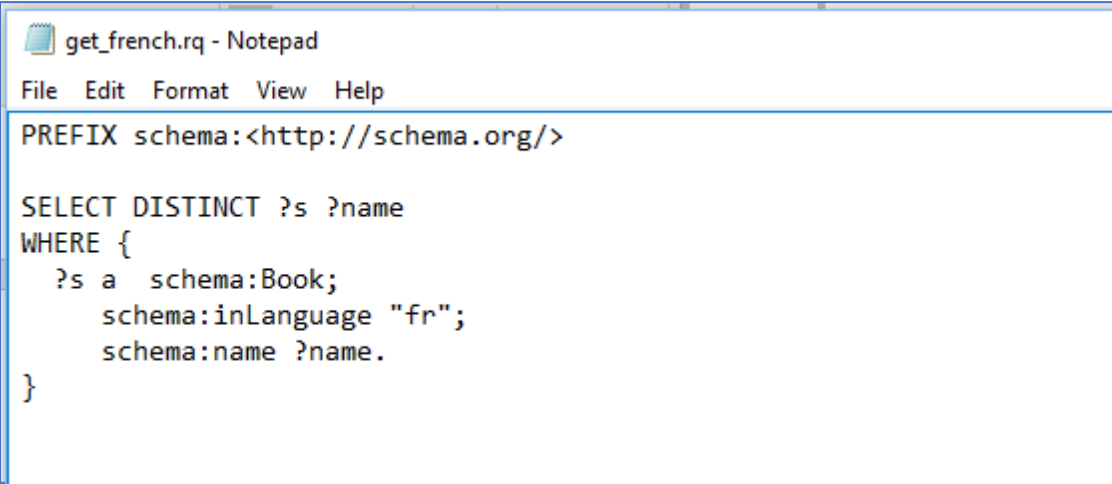


SIMPLE QUERY #2 – OPTIONAL AND TURNING AN OBJECT INTO SUBJECT

Now that you know all the classes and properties used in the dataset (**SEE** Exploratory Queries section), as well as which properties are used to describe languages and all the possible language codes (**SEE** Simple Query No. 1 section), let's write a more specific query. Let's limit the type of Creative Works you are looking for to *Books*, and the language they are written in to *French*.

All you need to do is string together a few triple statements. The following query (**Figure 12**) first uses a **placeholder variable** (“*?s*”) and asks for all resources of the **class** *Book*. Then, it asks for those resources in the French language, using the **literal value** “*fr*” as the object of the second statement. The third statement asks for the names of these resources using a “*?name*” variable because, as the results will demonstrate, the placeholder variable used in the first statement is only going to give us *URIs*, which *may not be very meaningful to humans*. Note that these three triple statements are tied together in the **WHERE** clause, which means that *only* resources that are Books, have been described as written in the French language, and who have a name attached to them will be returned in the results table. This is because none of the three triple statements are **OPTIONAL**.

A screenshot of a Notepad window titled "get_french.rq - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text content is a SPARQL query:

```
PREFIX schema:<http://schema.org/>

SELECT DISTINCT ?s ?name
WHERE {
  ?s a schema:Book;
     schema:inLanguage "fr";
     schema:name ?name.
}
```

Figure 12: SPARQL query to retrieve all *books* written in *French*

	A	B	C	D	E	F	G	H	I	J	K	L
1	s	name										
2	http://www.worldcat.org/oclc/36259367	Administration et bibliothèques										
3	http://www.worldcat.org/oclc/635595613	Administration et bibliothèques										
4	http://www.worldcat.org/oclc/752366283	Mener l'enquête : guide des études de publics en bibliothèque										
5	http://www.worldcat.org/oclc/778919701	Mener l'enquête : guide des études de publics en bibliothèque										
6	http://www.worldcat.org/oclc/469939413	Livres, lecteurs et bibliothèques de l'Italie médiévale (IXe-XVe siècles) : sources, textes et usages										
7	http://www.worldcat.org/oclc/757912796	Libri, lettori e biblioteche dell'Italia medievale (secoli IX-XV) : fonti, testi, utilizzazione del libro : atti della Ta										
8	http://www.worldcat.org/oclc/459516529	Règles pour la rédaction d'un catalogue collectif de périodiques										
9	http://www.worldcat.org/oclc/889625908	Thésaurus de l'Éducation Unesco : BIE : liste par facettes de termes destinés à l'indexation et à la rech										
10	http://www.worldcat.org/oclc/929717898	Les collections des arts du spectacle et leur traitement										
11	http://www.worldcat.org/oclc/49085268	Guide de la cartothèque										
12	http://www.worldcat.org/oclc/163649462	Catalogue des fonds coloniaux de la Bibliothèque Administrative 2 Les colonies dans les expositions / par Bâ										
13	http://www.worldcat.org/oclc/797005198	Bibliothèque de feu J. de Chantepie du Désert										
14	http://www.worldcat.org/oclc/427063079	Rapport sur la documentation en Suède										
15	http://www.worldcat.org/oclc/918222891	La Russie au XVIIIe : sources et histoire.										

Figure 13: SPARQL result set with URIs in the “s” column and human readable results in the “name” column

You find that there are quite a few books in the dataset in French. What if you want to narrow them down further? As always, you must know what properties are used to describe them first. You could go back to the list of all the properties from the entire dataset generated earlier, but you would be guessing which ones were used to describe resources of the **class Book** (plus, the **subgraph of French books** might use even fewer properties).

Another option you might consider would be picking a URI from the list of French books and writing a simple query to ask for all the properties used to describe it. However, that URI only represents *one resource*, and it's very unlikely that the same exact set of properties was consistently used to describe every single French book, so this query would likely omit some relevant results.

Instead, you should ask for *all the properties used to describe all the French books*. The **DISTINCT** keyword will, as usual, save you from duplicate results. Notice also that you can leave out the statement asking for the names of the books, because you aren't interested in that information this time. In fact, the *only* results you are asking to be returned from this query (**Figure 14**) are those from the “?p” (standing for “property” or “predicate”) variable.

```
get_french_props.rq - Notepad
File Edit Format View Help
PREFIX schema:<http://schema.org/>

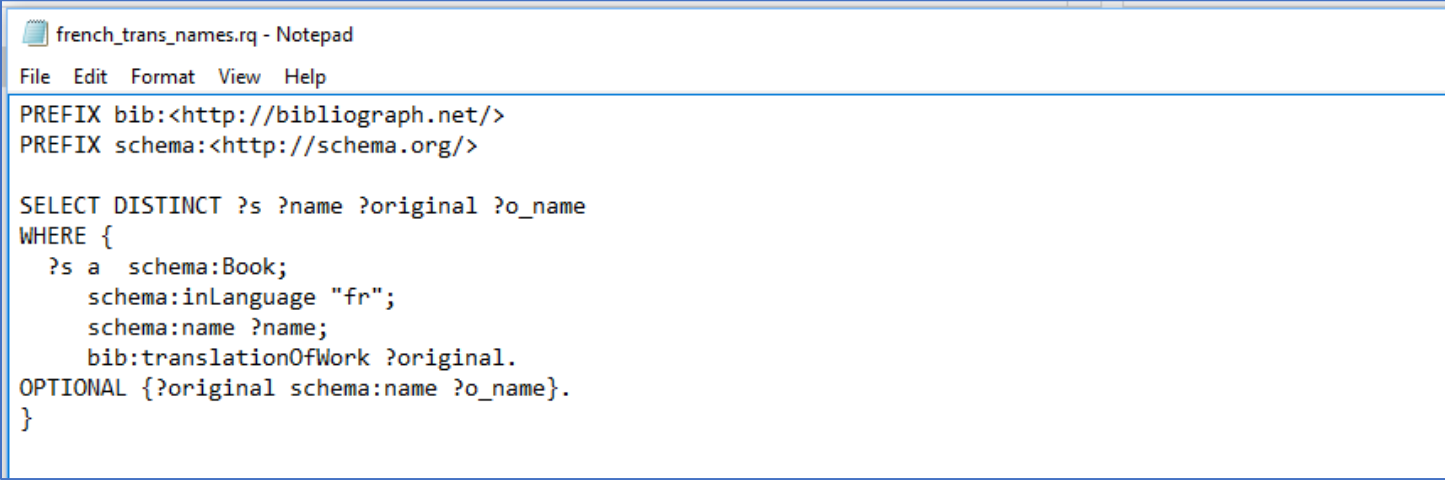
SELECT DISTINCT ?p
WHERE {
  ?s a schema:Book;
     schema:inLanguage "fr";
     ?p ?o.
}
```

Figure 14: SPARQL query to find *all properties* used to describe French books

```
p
-----
schema:name
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
schema:creator
schema:workExample
schema:bookFormat
schema:about
schema:exampleOfWork
schema:inLanguage
<http://purl.org/library/placeOfPublication>
schema:numberOfPages
schema:datePublished
<http://www.w3.org/2007/05/powder-s#describedby>
schema:isPartOf
schema:publication
```

Figure 15: Result set for triple statements sharing subject variable (truncated)

If you scroll through the result set (**Figure 15**), you find several dozen properties, each of which was used to describe at least one French book. It seems fair to assume that there will be quite a few French books which are translations of books in other languages. Let's find out how many with the following query:



```
french_trans_names.rq - Notepad
File Edit Format View Help
PREFIX bib:<http://bibliograph.net/>
PREFIX schema:<http://schema.org/>

SELECT DISTINCT ?s ?name ?original ?o_name
WHERE {
  ?s a schema:Book;
    schema:inLanguage "fr";
    schema:name ?name;
    bib:translationOfWork ?original.
  OPTIONAL {?original schema:name ?o_name}.
}
```

Figure 16: SPARQL query to retrieve *French books which are translations from another language*

Notice another new keyword has been added to the mix in this query (**Figure 16**). First, you used four triple statements that shared the same subject, with each statement specifying a condition that must be met for a resource to be included in the result set. For example, the fourth of these statements specified that each resource must have the property of being translated from an original work. Then, you added an **OPTIONAL** statement that said, if possible, also give us the original name of the work for which each resource is a translation.

This query *will still return resources even if the name of the original work is not included in the resource's description* – it will just leave a blank for that field for those entries in the results set. This is NOT the same as having a “NULL” value in a **relational database** one queries with **SQL**- a subtle distinction.

Also notice that, in order to get the names of the original works (if available), *you have taken the object of one triple statement (“?original”) and used it, in turn, as the subject of another triple statement*. In the **OPTIONAL** clause, the variable (“?original”) represents a new resource that you are interested in and you re-use the property (“*schema:name*”) to grab its name (represented as the object of this triple by the variable “?o_name”) exactly as you used this property to grab the name of the French version of the resource earlier in the query. This is what makes **triples, graphs, and the SPARQL query language** so powerful!