

# Ontologies and the Semantic Web

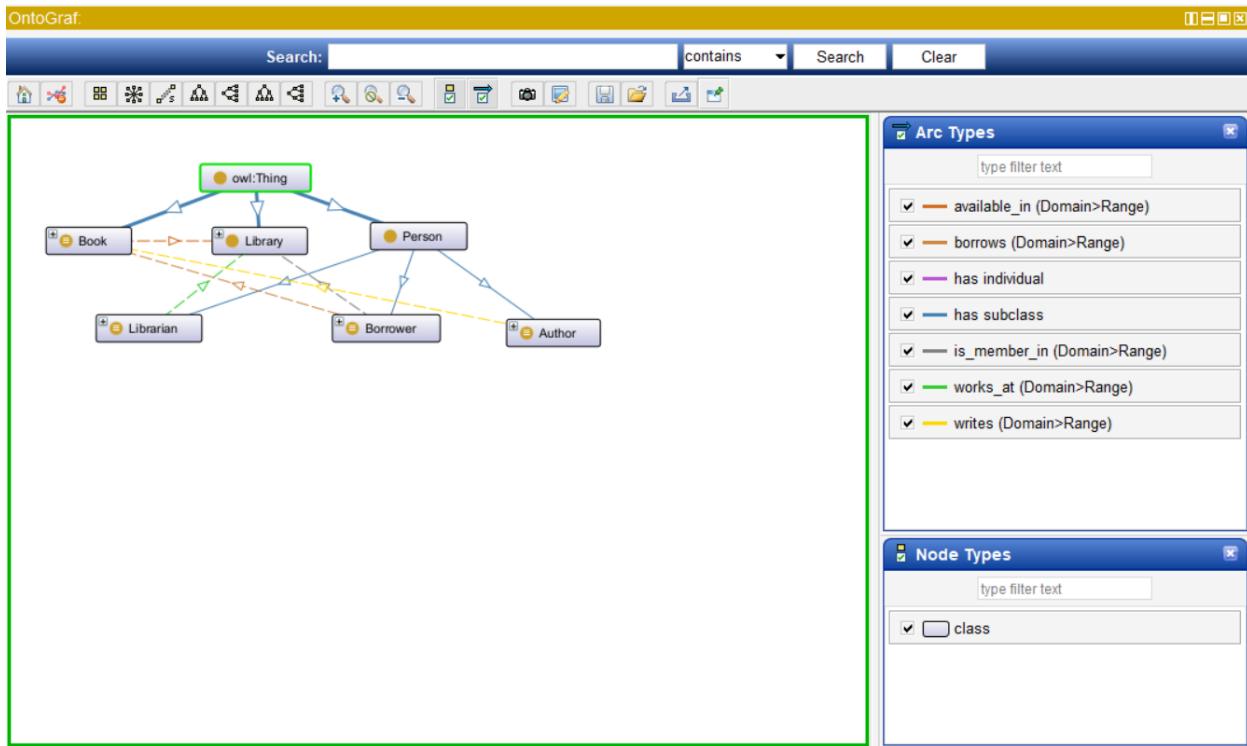
## Course Project

Public Library Ontology

Maryam Ahmed Galal Nouh	18P2824
Eman Khaled Ahmed	18P9713
Farah Amr Abdelfatah	18P3784
Zyad Ahmed Aly Hedar Yakan	18P9538

# I. Ontology

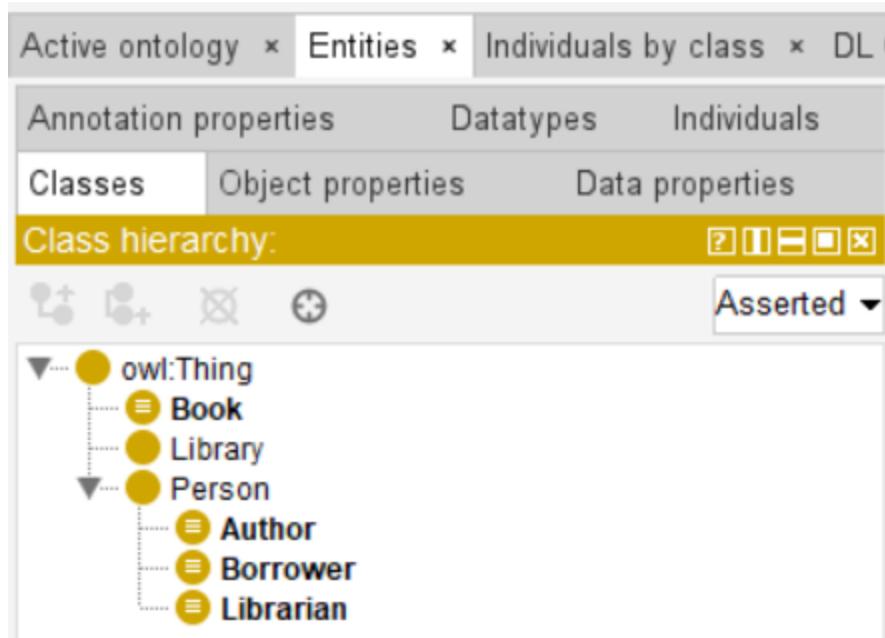
A public library is a library that is accessible by the general public and is usually funded from public sources. It has books, with their corresponding authors, members, and a librarian.



# II. RDF

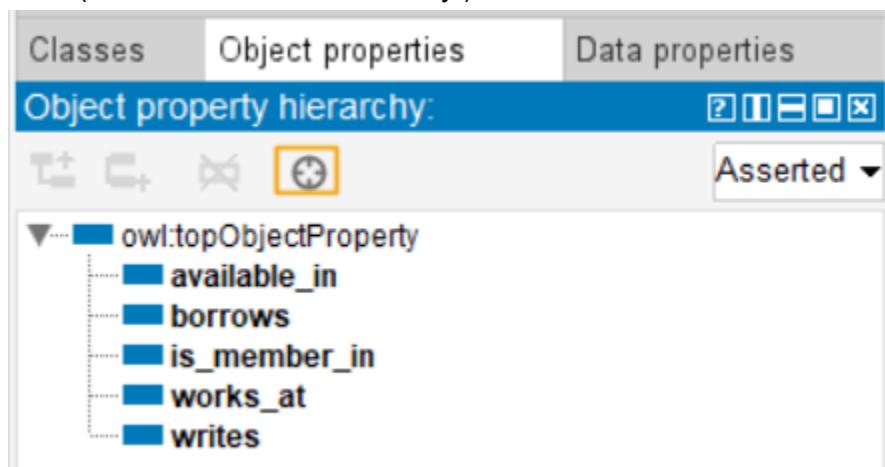
List of entities:

- Library
- Book
- Person
  - Author
  - Librarian
  - Library member / book borrower



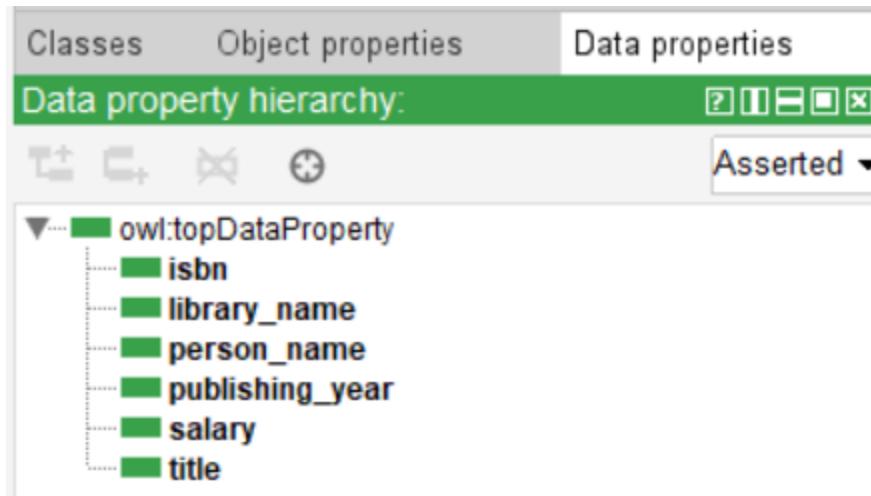
List of object properties:

- Member in ( library member is member in a library )
- Writes ( author writes book )
- Works at ( librarian works at a library )
- Borrows ( library member / book borrower borrows book )
- Available in ( book is available in a library )



List of data properties:

- Library name
- Book title
- Person name
- Book isbn
- Librarian salary
- Publishing year



Constraints:

- Member can borrow up to 4 books
- Borrower is member in at least one library

**Description: Borrower**

Equivalent To [+](#)

- borrows **max 4 Book**
- is\_member\_in **min 1 Library**

- Author must write at least one book

**Description: Author**

Equivalent To [+](#)

- writes **min 1 Book**

- Librarian works at exactly one library

**Description: Librarian**

Equivalent To [+](#)

- works\_at **exactly 1 Library**

- Book is available in at least one library

**Description: Book**

Equivalent To [+](#)

- available\_in **min 1 Library**

### III. Instances

Books:

Description: Book	
Instances	<a href="#">+</a>
◆	<a href="#">Animal_Farm</a>
◆	<a href="#">Crime_and_Punishment</a>
◆	<a href="#">Franny_and_Zooey</a>
◆	<a href="#">Go_Set_a_Watchman</a>
◆	<a href="#">Harry_Potter_and_the_Philosophers_Stone</a>
◆	<a href="#">Moby_Dick</a>
◆	<a href="#">Nineteen_Eighty-Four</a>
◆	<a href="#">Pride_and_Prejudice</a>
◆	<a href="#">Sense_and_Sensibility</a>
◆	<a href="#">Tender_Is_the_Night</a>
◆	<a href="#">The_Adventures_of_Sherlock_Holmes</a>
◆	<a href="#">TheBeautiful_and_Damned</a>
◆	<a href="#">The_Catcher_in_the_Rye</a>
◆	<a href="#">The_Da_Vinci_Code</a>
◆	<a href="#">The_Girl_Who_Played_with_Fire</a>
◆	<a href="#">The_Girl_with_the_Dragon_Tattoo</a>
◆	<a href="#">The_Great_Gatsby</a>
◆	<a href="#">The_Hobbit</a>
◆	<a href="#">The_Hunger_Games</a>
◆	<a href="#">The_Lord_of_the_Rings</a>
◆	<a href="#">The_Lost_Symbol</a>
◆	<a href="#">The_War_of_the_Worlds</a>
◆	<a href="#">To_Kill_a_Mockingbird</a>

Authors:

Description: Author

SubClass Of (Anonymous Ancestor)

Instances +

- ◆ Arthur\_Conan\_Doyle
- ◆ Dan\_Brown
- ◆ F.Scott\_Fitzgerald
- ◆ Fyodor\_Dostoevsky
- ◆ George\_Orwell
- ◆ H.G.\_Wells
- ◆ Harper\_Lee
- ◆ Herman\_Melville
- ◆ J.D.\_Salinger
- ◆ J.K.\_Rowling
- ◆ J.R.R.\_Tolkien
- ◆ Jane\_Austen
- ◆ Stieg\_Larsson
- ◆ Suzanne\_Collins

Libraries:

Instances +

- ◆ Library\_at\_The\_Dock
- ◆ The\_Library\_Club
- ◆ The\_London\_Library

Borrowers:

Description: Borrower	
Instances	
Adam_Wilson	
Ava_Davis	
Benjamin_Patel	
Caleb_Brown	
Charlotte_Singh	
Christopher_Nguyen	
Daniel_Lee	
David_Smith	
Dylan_Rodriguez	
Emily_Mitchell	
Ethan_Park	
Gabriel_Smith	
Grace_Wong	
Isaac_Johnson	
Isabella_Baker	
Julia_Martinez	
Leah_Anderson	
Lily_Chen	
Lucas_Wilson	
Madison_Harrison	
Mia_Lee	
Natalie_Martinez	
Nathan_Scott	
Olivia_Taylor	
Samuel_Kim	
Sophia_Kim	
Sophia_Nguyen	
Timothy_Nelson	
Victoria_Patel	
William_Hernandez	

Librarians:

Instances	
Jasmine_Chen	
Matthew_Rodriguez	
Olivia_Kim	

# IV. SparQL Queries

## 1. Protégé

- 1.1. Search for all the authors and the titles of their books ordered by authors in ascending order.

The screenshot shows the Snap SPARQL Query interface. The query is:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uni: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
SELECT ?author_name ?book_title
WHERE {
    ?author a uni:Author.
    ?author uni:person_name ?author_name.
    ?author uni:writes ?book.
    ?book uni:title ?book_title.
}
ORDER BY ASC (?author_name)
```

The results table has two columns: ?author\_name and ?book\_title. The data is as follows:

?author_name	?book_title
Arthur Conan Doyle	The Adventures of Sherlock Holmes
Dan Brown	The Da Vinci Code
Dan Brown	The Lost Symbol
F.Scott Fitzgerald	Tender Is the Night
F.Scott Fitzgerald	The Great Gatsby
F.Scott Fitzgerald	The Beautiful and Damned
Fyodor Dostoevsky	Crime and Punishment
George Orwell	1984
George Orwell	Animal Farm
H.G. Wells	The War of the Worlds
Harper Lee	To Kill a Mockingbird
Harper Lee	Go Set a Watchman
Herman Melville	Moby-Dick
J.D. Salinger	The Catcher in the Rye
J.D. Salinger	Franny and Zooey

23 results

Synchronising

Activate Windows  
Go to Settings to activate Windows.

- 1.2. List the books available in a certain library.

The screenshot shows the Snap SPARQL Query interface. The query is:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uni: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
SELECT DISTINCT ?book_title
WHERE {
    ?book a uni:Book.
    ?library a uni:Library.
    ?book uni:available_in ?library.
    ?library uni:library_name ?library_name.
    ?book uni:title ?book_title.
}
FILTER regex (?library_name, "Library at the Dock", "i")
```

The results table has one column: ?book\_title. The data is as follows:

?book_title
1984
Crime and Punishment
The Girl with the Dragon Tattoo
The Hunger Games
Tender Is the Night
The War of the Worlds
The Hobbit
Harry Potter and the Philosopher's Stone
The Girl Who Played with Fire
The Catcher in the Rye
Go Set a Watchman

11 results

Synchronising

- 1.3. Find all books with a certain title keyword (here, "Girl").

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT DISTINCT ?book_title
WHERE {
?book a uri:Book.
?book uri:title ?book_title.
?library a uri:Library.
?book uri:available_in ?library
FILTER REGEX (?book_title, "Girl" , "i").
}

```

?book_title
The Girl with the Dragon Tattoo
The Girl Who Played with Fire

2 results

Synchronising

#### 1.4. Find all books whose ISBN end with certain numbers

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT ?book_title ?book_isbn
WHERE {
?book a uri:Book.
?book uri:title ?book_title.
?book uri:isbn ?book_isbn.
?library a uri:Library.
?book uri:available_in ?library
FILTER REGEX (?book_isbn, "4$" , "i").
}
ORDER BY ASC (?book_isbn)

```

?book_title	?book_isbn
Moby-Dick	978-0143105954
Tender Is the Night	978-0684801544

2 results

Synchronising

#### 1.5. Find all books borrowed by a certain borrower.

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT ?book_title
WHERE {
?borrower a uri:Borrower.
?borrower uri:borrowed ?book.
?book a uri:Book.
?book rdfs:title ?book_title.
?borrower rdfs:person_name ?borrower_name.
FILTER ( ?borrower_name = "Adam Wilson" )
}
ORDER BY ASC (?book_title)

```

?book_title
Animal Farm^^xsd:string
Pride and Prejudice^^xsd:string
Sense and Sensibility^^xsd:string

3 results

Synchronising

Activate Windows  
Go to Settings to activate Windows.

## 1.6. Find all borrowers who have borrowed a certain book.

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT ?borrower_name ?book_title
WHERE {
?borrower a uri:Borrower.
?borrower uri:borrowed ?book.
?book a uri:Book.
?book rdfs:title ?book_title.
?borrower rdfs:person_name ?borrower_name.
FILTER ( ?book_title = "1984" )
}


```

?borrower_name	?book_title
Emily Mitchell	1984

## 1.7. How many authors are there and how many books ?

```

Snap SPARQL Query:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT ( COUNT(?book) AS ?book_count )
( COUNT(DISTINCT (?author)) AS ?author_count )
WHERE {
    ?author uri:writes ?book
}

```

Execute	
?book_count	?author_count
23	23

## 1.8. Which authors wrote how many books are available?

```

Snap SPARQL Query:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT ?author_name (COUNT (?book) AS ?book_count)
WHERE {
    ?book a uri:Book.
    ?author a uri:Author.
    ?library a uri:Library.
    ?book uri:available_in ?library.
    ?author uri:writes ?book.
    ?author uri:person_name ?author_name.
}
GROUP BY ?author_name
ORDER BY DESC ( ?book_count ) |
```

Execute	
?author_name	?book_count
J.R.R.Tolkien	4
Harper Lee	4
Stieg Larsson	4
F.Scott Fitzgerald	3
Jane Austen	3
J.K. Rowling	3
H.G. Wells	2
Fyodor Dostoevsky	2
J.D. Salinger	2
George Orwell	2
Dan Brown	2
Arthur Conan Doyle	1
Suzanne Collins	1
Herman Melville	1
14 results	

1.9. Search for all books that are available in either  
“Library\_at\_The\_Dock” or “The\_London\_Library” or both.

Snap SPARQL Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT DISTINCT ?book_title
WHERE{
{
    ?book a uri:Book.
    ?library a uri:Library.
    ?book uri:available_in ?library.
    ?library uri:library_name ?library_name.
    ?book uri:title ?book_title.
    FILTER (?library_name = "Library at the Dock" )
}
UNION
{
    ?book a uri:Book.
    ?library a uri:Library.
    ?book uri:available_in ?library.
    ?library uri:library_name ?library_name.
    ?book uri:title ?book_title.
    FILTER (?library_name = "The London Library")
}
}
```

?book_title
1984
Crime and Punishment
The Girl with the Dragon Tattoo
The Hunger Games
Tender Is the Night
The War of the Worlds
The Hobbit
Harry Potter and the Philosopher's Stone
The Girl Who Played with Fire
The Catcher in the Rye
Go Set a Watchman
The Beautiful and Damned
Moby-Dick
The Lord of the Rings
Pride and Prejudice
To Kill a Mockingbird
Sense and Sensibility
Animal Farm
The Lost Symbol

19 results

Synchronising

## 1.10. List all the members in a certain library

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT DISTINCT ?member_name
WHERE {
    ?borrower a uri:Borrower.
    ?borrower uri:borrows ?book.
    ?borrower uri:person_name ?member_name.
    ?borrower uri:is_member_in ?library.
    ?library uri:library_name ?library_name
    FILTER (?library_name = "The London Library")
}
ORDER BY ASC (?member_name)

```

**Execute**

?member_name
Adam Wilson
David Smith
Dylan Rodriguez
Grace Wong
Isaac Johnson
Julia Martinez
Madison Harrison
Nathan Scott
Sophia Kim
Vicotria Patel
10 results

Synchronising

## 1.11. List the members that borrowed a certain book “pride and prejudice”, who are not a member in a certain library “The\_London\_Library”

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
SELECT ?member_name ?library_name
WHERE {
    ?borrower a uri:Borrower.
    ?borrower uri:borrows ?book.
    ?borrower uri:person_name ?member_name.
    ?borrower uri:is_member_in ?library.
    ?library uri:library_name ?library_name.
    ?book uri:title ?book_title.
    FILTER (?book_title = "Pride and Prejudice")
    MINUS {?library uri:library_name "The London Library".}
}

```

**Execute**

?member_name	?library_name
Charlotte Singh	The Library Club
Natalie Martinez	The Library Club

## 1.12. List all the libraries in the system

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
SELECT ?library_name
WHERE {
    ?library a uri:Library.
    ?library uri:library_name ?library_name
}
ORDER BY ASC (?library_name)

```

**Execute**

?library_name
Library at the Dock
The Library Club
The London Library

### 1.13. List the authors with their books ordered by the authors' Last names.

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
SELECT DISTINCT ?author_name ?book_title ?author_last_name
WHERE {
    ?author a uri:Author.
    ?book a uri:Book.
    ?author uri:writes ?book.
    ?book urititle ?book_title.
    ?author uriperson_name ?author_name
    BIND(strafter(str(?author_name), " ") AS ?author_last_name)
    FILTER(BOUND(?author_last_name))
}
ORDER BY ASC (?author_last_name)

```

**Execute**

?author_name	?book_title	?author_last_name
Jane Austen	Pride and Prejudice	Austen
Jane Austen	Sense and Sensibility	Austen
Dan Brown	The Da Vinci Code	Brown
Dan Brown	The Lost Symbol	Brown
Suzanne Collins	The Hunger Games	Collins
Arthur Conan Doyle	The Adventures of Sherlock Holmes	Conan Doyle
Fyodor Dostoevsky	Crime and Punishment	Dostoevsky
F Scott Fitzgerald	The Great Gatsby	Fitzgerald
F Scott Fitzgerald	The Beautiful and Damned	Fitzgerald
F Scott Fitzgerald	Tender Is the Night	Fitzgerald
Stieg Larsson	The Girl with the Dragon Tattoo	Larsson
Stieg Larsson	The Girl Who Played with Fire	Larsson
Harper Lee	To Kill a Mockingbird	Lee

23 results

Synchronising

?author_name	?book_title	?author_last_name
Stieg Larsson	The Girl with the Dragon Tattoo	Larsson
Stieg Larsson	The Girl Who Played with Fire	Larsson
Harper Lee	To Kill a Mockingbird	Lee
Harper Lee	Go Set a Watchman	Lee
Herman Melville	Moby-Dick	Melville
George Orwell	1984	Orwell
George Orwell	Animal Farm	Orwell
J.K. Rowling	Harry Potter and the Philosopher's Stone	Rowling
J.D. Salinger	The Catcher in the Rye	Salinger
J.D. Salinger	Franny and Zooey	Salinger
J.R.R. Tolkien^^xsd:string	The Lord of the Rings	Tolkien
J.R.R. Tolkien^^xsd:string	The Hobbit	Tolkien
H.G. Wells	The War of the Worlds	Wells
23 results		
<input type="checkbox"/> Synchronising		

## 1.14. Retrieve all librarians, which library they work at, and their salaries.

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>

SELECT ?librarian_name ?salary ?library
WHERE
{
    ?librarian a uri:Librarian.
    ?library a uri:Library.
    ?librarian rdfs:person_name ?librarian_name.
    ?librarian uri:works_at ?library.
    ?librarian uri:salary ?salary.
}
ORDER BY ASC (?salary)

```

Execute

?librarian_name	?salary	?library
Matthew Rodriguez^^xsd:string	45000.0	<https://dbpedia.org/page/Library_at_The_Dock>
Olivia Kim^^xsd:string	47500.0	<https://dbpedia.org/page/The_London_Library>
Jasmine Chen^^xsd:string	50000.0	<https://dbpedia.org/page/The_Library_Club>

3 results

Synchronising

Activate Windows  
Go to Settings to activate Windows.

## 2. Jena

### 2.1. Search for all the authors whose names start with “J” and if available, their books have “and” in the books’ titles or start with “the”.

SPARQL Endpoint /library.owl/query Content Type (SELECT) JSON Content Type (GRAPH) Turtle

```
1- PREFIX owl: <http://www.w3.org/2002/07/owl#>
2- PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3- PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4- PREFIX url: <http://www.semanticweb.org/mouh/ontologies/2023/3/untitled-ontology-3#>
5- SELECT DISTINCT ?author_name ?book_title
6- WHERE {
7-   ?author a url:Author .
8-   ?author a url:Book .
9-   ?author url:writes ?book .
10-  ?book url:person_name ?author_name
11-  FILTER REGEX (?author_name, "J", "i")
12-  OPTIONAL {
13-    ?author url:writes ?book .
14-    ?book url:title ?book_title }
```

Table Response 7 results in 0.062 seconds

author_name	book_title
1 J.D. Salinger	The Catcher in the Rye
2 J.D. Salinger	Franny and Zooey
3 J.K. Rowling	Harry Potter and the Philosopher's Stone
4 J.R.R. Tolkien	The Hobbit
5 J.R.R. Tolkien	The Lord of the Rings
6 Jane Austen	Pride and Prejudice
7 Jane Austen	Sense and Sensibility

Showing 1 to 7 of 7 entries

SPARQL Endpoint /library.owl/query Content Type (SELECT) JSON Content Type (GRAPH) Turtle

```
11- FILTER REGEX (?author_name, "J", "i")
12- OPTIONAL {
13-   ?author url:writes ?book .
14-   ?book url:title ?book_title
15-   FILTER REGEX (?book_title, "and", "i")
16- }
17- OPTIONAL {
18-   ?author url:writes ?book .
19-   ?book url:title ?book_title
20-   FILTER REGEX (?book_title, "the", "i")
21- }
22- }
23- 
```

Table Response 7 results in 0.062 seconds

author_name	book_title
1 J.D. Salinger	The Catcher in the Rye
2 J.D. Salinger	Franny and Zooey
3 J.K. Rowling	Harry Potter and the Philosopher's Stone
4 J.R.R. Tolkien	The Hobbit
5 J.R.R. Tolkien	The Lord of the Rings
6 Jane Austen	Pride and Prejudice
7 Jane Austen	Sense and Sensibility

Showing 1 to 7 of 7 entries

## 2.2. Find authors who wrote more than 2 books that are available.

SPARQL Endpoint Content Type (SELECT)

/library\_ont/query JSON

```

1 PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/united-ontology-3#>
2 SELECT ?author_name (COUNT(?book) AS ?num_books)
3 WHERE {
4   ?author a uri:Author.
5   ?book a uri:Book.
6   ?library a uri:Library.
7   ?author uri:writes ?book.
8   ?book uri:title ?book_title.
9   ?book uri:available_in ?library.
10  ?author uri:person_name ?author_name.
11 }
12 GROUP BY ?author_name
13 HAVING (?num_books >= 2)

```

Table Response 11 results in 0.386 seconds

author_name	num_books
Dan Brown	2
J.R.R. Tolkien	4
Jane Austen	3
F.Scott Fitzgerald	3
Fyodor Dostoevsky	2
George Orwell	2
Stieg Larsson	4
H.G. Wells	2
Harper Lee	4
J.K. Rowling	3
J.D. Salinger	2

Showing 1 to 11 of 11 entries

## 2.3. Find borrowers who borrowed more than 1 book.

SPARQL Endpoint Content Type (SELECT)

/library\_ont/query Content Type (GRAPH)

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX uril: <http://www.semanticweb.org/mnouh/ontologies/2023/3/united-ontology-3#>
5 SELECT ?borrower_name (COUNT(?book) AS ?num_books)
6 WHERE {
7   ?borrower a uril:borrower.
8   ?book a uril:book.
9   ?library a uril:library.
10  ?borrower uril:borrows ?book.
11  ?book uril:title ?book_title.
12  ?book uril:available_in ?library.
13  ?borrower uril:person_name ?borrower_name
14 }
15 GROUP BY ?borrower_name
16 HAVING (?num_books >= 2)

```

Table Response 23 results in 0.139 seconds

borrower_name	num_books
Nathan Scott	3
Natalie Martinez	2
Mia Lee	2
Samuel Kim	2
Isabella Baker	2
Lily Chen	2
Celine Brown	3
Leah Anderson	2

Table Response 23 results in 0.139 seconds

borrower_name	num_books
Nathan Scott	3
Natalie Martinez	2
Mia Lee	2
Samuel Kim	2
Isabella Baker	2
Lily Chen	2
Caleb Brown	3
Leah Anderson	2
Olivia Taylor	2
David Smith	3
Adam Wilson	4
Julia Martinez	3
Ethan Park	2
Timothy Nelson	2
Christopher Nguyen	4
Ava Davis	3
Daniel Lee	2
Madison Harrison	2
Charlotte Singh	2
Sophia Kim	2
Lucas Wilson	4
Benjamin Patel	4
Emily Mitchell	3

Showing 1 to 23 of 23 entries < 1 >

## 2.4. Retrieve the titles of all books that are available in libraries other than "The\_Library\_Club" and if available starts with "the".

SPARQL Endpoint /library.owl/query Content Type (SELECT) Content Type (GRAPH)

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdt: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX wri: <http://www.semanticweb.org/mehdi/ontologies/2023/3/united-ontology#>
5 SELECT book_title ?library_name
6 WHERE {
7   ?book a wri:Book,
8   ?book a wri:Library,
9   ?book wri:availableIn_1 ?library,
10  ?book wri:bookTitle ?book_title,
11  ?library wri:libraryName ?library_name
12  FILTER (?library_name != "The Library Club")
13  OPTIONAL {
14    FILTER REGEX (?book_title , "the", "i")
15  }
16 }

```

Table Response 23 results in 0.142 seconds

book_title	library_name
Crime and Punishment	Library at the Dock
Crime and Punishment	The London Library
The Lost Symbol	The London Library
Tender Is the Night	Library at the Dock
The Beautiful and Damned	The London Library
1984	Library at the Dock
Animal Farm	The London Library
The War of the Worlds	Library at the Dock
To Kill a Mockingbird	The London Library
Go Set a Watchman	Library at the Dock
Go Set a Watchman	The London Library
Harry Potter and the Philosopher's Stone	Library at the Dock
Harry Potter and the Philosopher's Stone	The London Library
Moby-Dick	Library at the Dock
The Catcher in the Rye	The London Library
The Hobbit	Library at the Dock
The Lord of the Rings	Library at the Dock
Pride and Prejudice	The London Library
Sense and Sensibility	The London Library

book_title	library_name
Crime and Punishment	Library at the Dock
Crime and Punishment	The London Library
The Lost Symbol	The London Library
Tender Is the Night	Library at the Dock
The Beautiful and Damned	The London Library
1984	Library at the Dock
Animal Farm	The London Library
The War of the Worlds	Library at the Dock
To Kill a Mockingbird	The London Library
Go Set a Watchman	Library at the Dock
Go Set a Watchman	The London Library
Harry Potter and the Philosopher's Stone	Library at the Dock
Harry Potter and the Philosopher's Stone	The London Library
Moby-Dick	The London Library
The Catcher in the Rye	Library at the Dock
The Hobbit	The London Library
The Lord of the Rings	Library at the Dock
Pride and Prejudice	The London Library
Sense and Sensibility	Library at the Dock
The Girl with the Dragon Tattoo	The London Library
The Girl Who Played with Fire	Library at the Dock
The Girl Who Played with Fire	The London Library
The Hunger Games	Library at the Dock

Showing 1 to 23 of 23 entries

## 2.5. ASK Query, is there an author with a book?

SPARQL Endpoint
Content Type  
(SELECT)
Content Type  
(GRAPH)

JSON
Turtle

```

1 v PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 v PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 v PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 v PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
5 v ASK
6 v WHERE {
7 v   ?author a uri:Author.
8 v   ?book a uri:Book.
9 v   ?author uri:person_name ?author_name.
10 v  ?author uri:writes ?book.
11 v }
12 v
13 v

```

Press CTRL - <spacebar> to autocomplete

Table
Response
Response in 0.08 seconds

✓ True

## 2.6. Ask Query, is there a borrower with a book?

SPARQL Endpoint      Content Type      Content Type  
/library\_ont/query      (SELECT)      (GRAPH)  
JSON      Turtle

```
1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
5 ASK
6 WHERE {
7   ?borrower a uri:Borrower.
8   ?book a uri:Book.
9   ?borrower uri:borrows ?book.
10  ?book uri:available_in ?library.
11  ?book uri:title ?book_title.
12  ?library uri:library_name ?library_name
13 }
```

Press CTRL - <spacebar> to autocomplete

Table Response Response in 0.022 seconds ✓ True

## 2.7. ASK Query, is there a book available in the library ?

SPARQL Endpoint      Content Type      Content Type  
/library\_ont/query      (SELECT)      (GRAPH)  
JSON      Turtle

```
1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
5 ASK
6 WHERE {
7   ?book a uri:Book.
8   ?library a uri:Library.
9   ?book uri:available_in ?library.
10  ?book uri:title ?book_title .
11  ?library uri:library_name ?library_name.
12  }
```

Table Response Response in 0.02 seconds ✓ True

## 2.8. ASK QUERY, is there an author who borrowed a book?

The screenshot shows a SPARQL endpoint interface with the following configuration:

- SPARQL Endpoint:** /library\_ont/query
- Content Type (SELECT):** JSON
- Content Type (GRAPH):** Turtle

```

PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

ASK WHERE {
    ?author a uri:Author .
    ?book a uri:Book .
    ?author uri:borrows ?book .
}
  
```

Below the query text, there are three tabs: **Table**, **Response**, and **Response in 0.044 seconds**. The **Response** tab displays the result:

**x False**

## 2.9. DESCRIBE QUERY, show all books available with their authors.

The screenshot shows a SPARQL endpoint interface with the following configuration:

- SPARQL Endpoint:** /library\_ont/query
- Content Type (SELECT):** JSON
- Content Type (GRAPH):** Turtle

```

PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#>
DESCRIBE Author Book
  
```

Below the query text, there are three tabs: **Table**, **Response**, and **215 results in 0.041 seconds**. The **Response** tab displays the results in a table format:

subject	predicate	object
http://dbpedia.org/page/Suzanne_Collins	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#person_name	Suzanne Collins
http://dbpedia.org/page/Suzanne_Collins	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#writes	http://dbpedia.org/page/The_Hunger_Games
http://dbpedia.org/page/Suzanne_Collins	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#NamedIndividual
http://dbpedia.org/page/Suzanne_Collins	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#Author
http://dbpedia.org/page/J.R.R._Tolkien	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#person_name	J.R.R. Tolkien
http://dbpedia.org/page/J.R.R._Tolkien	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#writes	http://dbpedia.org/page/The_Lord_of_the_Rings
http://dbpedia.org/page/J.R.R._Tolkien	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#writes	http://dbpedia.org/page/The_Hobbit
http://dbpedia.org/page/J.R.R._Tolkien	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#NamedIndividual
http://dbpedia.org/page/J.R.R._Tolkien	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#Author
http://dbpedia.org/page/The_Beautiful_and_Damned	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#type	The Beautiful and Damned
http://dbpedia.org/page/The_Beautiful_and_Damned	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#published_year	1922
http://dbpedia.org/page/The_Beautiful_and_Damned	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#isbn	978-0743273965
http://dbpedia.org/page/The_Beautiful_and_Damned	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#available_in	https://dbpedia.org/page/The_London_Library
http://dbpedia.org/page/The_Beautiful_and_Damned	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#NamedIndividual
http://dbpedia.org/page/The_Beautiful_and_Damned	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#Book
http://dbpedia.org/page/Stieg_Larsson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#person_name	Stieg Larsson
http://dbpedia.org/page/Stieg_Larsson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#writes	https://dbpedia.org/page/The_Girl_Who_Payed_with_Fire
http://dbpedia.org/page/Stieg_Larsson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#writes	https://dbpedia.org/page/The_Girl_With_the_Dragon_Tattoo
http://dbpedia.org/page/Fyodor_Dostoevsky	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Fyodor Dostoevsky
http://dbpedia.org/page/Fyodor_Dostoevsky	http://www.semanticweb.org/mnouh/ontologies/2023/3/untitled-ontology-3#person_name	http://dbpedia.org/page/Crime_and_Punishment

## 2.10. DESCRIBE QUERY, show all the borrowers with their books.

SPARQL Endpoint: /library\_ont/query

Content Type (SELECT): JSON

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX murl: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#>
5 DESCRIBE ?borrower ?book
6 WHERE {
7   ?borrower a murl:Borrower .
8   ?book a murl:Book .
9   ?borrower murl:person_name ?borrower_name .
10  ?borrower murl:borrows ?book .
11  ?book murl:title ?book_title .
12 }
13

```

Content Type (GRAPH): Turtle

Table Response: 311 results in 0.03 seconds

subject	predicate	object
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Mia_Lee	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owlNamedIndividual
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Mia_Lee	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Borrower
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Mia_Lee	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Hobbit
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Mia_Lee	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#u1_member_in	http://dbpedia.org/page/Library_at_The_Dock
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Mia_Lee	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#person_name	Mia Lee
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Nathan_Scott	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owlNamedIndividual
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Nathan_Scott	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Borrower
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Nathan_Scott	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/Harry_Potter_and_the_Philosopher_Stone
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Nathan_Scott	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#u1_member_in	http://dbpedia.org/page/The_London_Library
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Nathan_Scott	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#person_name	Nathan Scott
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Samuel_Kim	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owlNamedIndividual
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Samuel_Kim	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Borrower
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Samuel_Kim	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Way_of_the_Worlds
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Samuel_Kim	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#u1_member_in	http://dbpedia.org/page/Library_at_The_Dock
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Samuel_Kim	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#person_name	Samuel Kim
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#XIII_A_Modigliani	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owlNamedIndividual
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#XIII_A_Modigliani	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Borrower
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#XIII_A_Modigliani	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Uncle_Cuba
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#XIII_A_Modigliani	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#u1_member_in	http://dbpedia.org/page/The_Uncle_Cuba
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#XIII_A_Modigliani	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#person_name	XIII_A_Modigliani
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Isabella_Baker	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owlNamedIndividual
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Isabella_Baker	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Borrower
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Isabella_Baker	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Way_of_the_Worlds
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Isabella_Baker	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#u1_member_in	http://dbpedia.org/page/Library_at_The_Dock
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Isabella_Baker	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#person_name	Isabella Baker
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Nineteen_Eighty_Four	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owlNamedIndividual

## 2.11. CONSTRUCT QUERY, create new RDF triples for authors and their books.

SPARQL Endpoint: /library\_ont/query

Content Type (SELECT): JSON

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX murl: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#>
5 CONSTRUCT {?author uri:writes ?book.}
6 WHERE {
7   ?author a owl:Author .
8   ?author rdfs:person_name ?author_name .
9   ?book uri:title ?book_title .
10 } LIMIT 10
11
12

```

Content Type (GRAPH): Turtle

Table Response: 10 results in 0.144 seconds

subject	predicate	object
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/Moby_Dick
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/The_Girl_with_the_Dragon_Tattoo
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/Harry_Potter_and_the_Philosophers_Stone
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/The_Great_Gatsby
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/Crime_and_Punishment
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/The_Da_Vinci_Code
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/Pride_and_Prejudice
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/The_Catcher_in_the_Rye
http://dbpedia.org/page/Arthur_Conan_Doyle	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#writes	http://dbpedia.org/page/Nineteen_Eighty_Four

Showing 1 to 10 of 10 entries

## 2.12. CONSTRUCT QUERY, create new RDF triples for borrowers and their borrowed books .

SPARQL Endpoint: /library\_cont/query

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX url: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#>
5 CONSTRUCT {borrower url:borrowes ?book}
6 WHERE {
7 ?book a url:Book,
8 ?borrower a url:Borrower,
9 ?borrower url:person_name ?borrower_name,
10 ?book url:title ?book_title.
11 } LIMIT 10
12
13

```

Table Response 10 results in 0.022 seconds

subject	predicate	object
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Ava_Davis	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Adam_Wilson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Charlotte_Singh	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Christopher_Nguyen	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Daniel_Lee	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Dylan_Rodriguez	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Emily_Mitchell	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Benjamin_Patel	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Caleb_Brown	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#David_Smith	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#borrows	http://dbpedia.org/page/The_Adventures_of_Sherlock_Holmes

Showing 1 to 10 of 10 entries

## 2.13. CONSTRUCT QUERY, create a new RDF triples with borrowers that are members in the Library.

SPARQL Endpoint: /library\_cont/query

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```

2 PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX url: <http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#>
5 CONSTRUCT {?borrower url:member_in ?library.}
6
7 WHERE {
8 ?borrower a url:Borrower,
9 ?library a url:Library,
10 ?borrower url:person_name ?borrower_name,
11 ?library url:library_name ?library_name,
12
13 } LIMIT 10
14

```

Table Response 10 results in 0.023 seconds

subject	predicate	object
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Adam_Wilson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/The_Library_Club
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Adam_Wilson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/The_London_Library
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Adam_Wilson	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/Library_at_The_Dock
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Ava_Davis	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/The_Library_Club
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Ava_Davis	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/The_London_Library
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Ava_Davis	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/Library_at_The_Dock
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Benjamin_Patel	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/The_Library_Club
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Benjamin_Patel	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/The_London_Library
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Benjamin_Patel	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/Library_at_The_Dock
http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#Caleb_Brown	http://www.semanticweb.org/mnouh/ontologies/2023/3/untilted-ontology-3#member_in	https://dbpedia.org/page/Library_at_The_Dock

Showing 1 to 10 of 10 entries

## V. Front End

### 1.0 Index page where the user writes their query.

A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/proj/`. Below the address bar, the text "Sparql Query" is displayed. A large text area labeled "Enter Sparql Query:" contains the following SPARQL query:

```
SELECT ?book_title ?book_isbn
WHERE {
  ?book a uri:Book.
  ?book uri:title ?book_title.
  ?book uri:isbn ?book_isbn.
  ?library a uri:library.
  ?book uri:available_in ?library.
  FILTER REGEX (?book_isbn, "4$" , "i").
}
ORDER BY ASC (?book_isbn)
```

At the bottom of the query area is a green "Execute" button.

### 2.0 Results page showing that the query was sent as a parameter for the end point

A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/proj/query?query=SELECT+?book_title+?book_isbn%0D%0A++WHERE++{+?book+a+uri:Book.+?book+uri:title+?book_title.+?book+uri:isbn+?book_isbn.+?library+a+uri:library.+?book+uri:available_in+?library.+FILTER+REGEX+(%?book_isbn,+4$+,+"i").+}+ORDER+BY+ASC+(%?book_isbn)+`. Below the address bar, the word "Results" is displayed. The main content area shows two book entries:

Hoby-Dick  
978-0143105954

Tender Is the Night  
978-0684801544