



Софийски университет „Св.
Климент Охридски“ гр. София

Факултет по математика и информатика

Документация

към проект по „Бази от знания“ на тема:

„Онтология за книги – извличане на информация“ – група А

Изготвено от: Кристияна Николова,

ИИ, 9MI3400442

Идея на проекта:

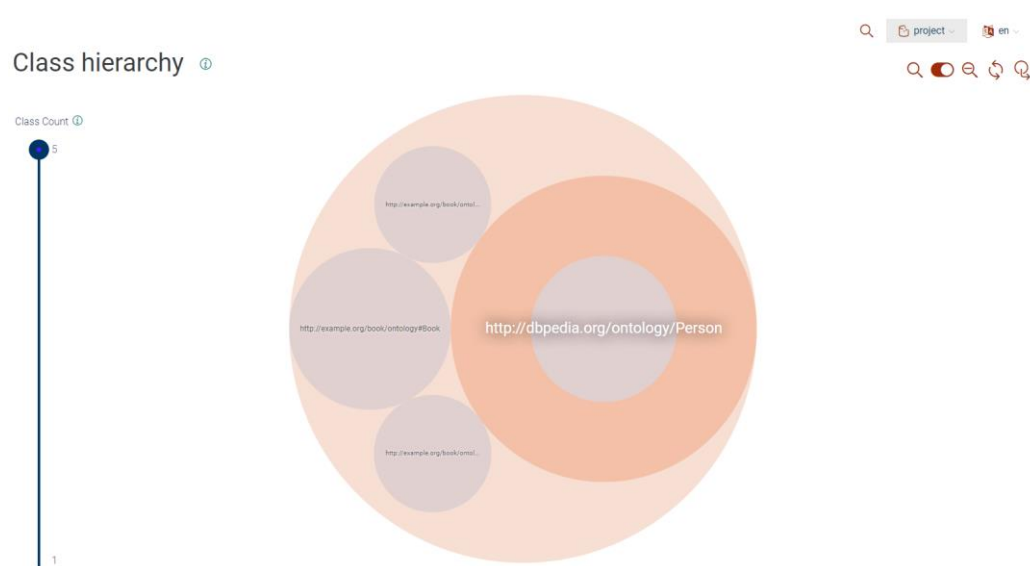
Идеята зад използването на онтология за книги е да създадем структурирано представяне на информацията, свързана с книгите и литературата като цяло. Целта е да предоставим по-лесен начин за търсене, организация и анализ на информацията за книгите, както и да подпомогнем автоматизирането на някои процеси в областта на литературата и издателствата. Може да е от полза на читатели, книгоиздатели, критици, библиотекари и други, които работят с книги и литературни данни.

Ето някои от предимствата и целите на използването на тази онтология:

1. **Извличане на информация за различни книги** – потребителите могат бързо да намерят книги по зададени характеристики:
 - a. Конкретен автор
 - b. Жанр
 - c. Конкретно издателство
 - d. Година на издаване
 - e. Брой страници
2. **Класифициране на книгите по жанрове**
3. **Извличане на книги спрямо конкретно издателство**
4. **Извличане на всички книги от конкретен жанр**
5. **Извличане на статистики/информация за автор/издателство**

Използването на онтология за книги може значително да подобри начина, по който работим с информацията за книгите, като предостави по-ефективни и интелигентни инструменти за търсене, анализ и управление на литературни данни.

Описание на данните:



Данните се състоят от 4 класа – Author, който е подклас на Person, Book, Genre, Publisher.

Всички property-та на класа Book, които можем да използваме, за да извеждаме информация, можем да извлечем по следния начин:

query

1

PREFIX

rdfs:

<http://www.w3.org/2000/01/rdf-schema#>

2

PREFIX

bko:

<http://example.org/book/ontology#>

3

PREFIX

dbo:

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

4

SELECT

DISTINCT

?property

WHERE

{

5

VALUES

?domainClass

{

6

bko:Book

dbo:Person

7

}

8

?property

rdfs:domain

?domainClass

.

9

}

keyboard shortcuts

След пускане на заявката получаваме следните 8 property-та:

Table

Raw response

Pivot Table

Google Chart

Download as

Filter query results

Compact view

Hide row numbers

Showing results from 0 to 8 of 8. Query took 0.1s, moments ago.

	property
1	bko.followedBy
2	bko.releaseDate
3	bko.hasAuthor
4	bko.hasGenre
5	bko.publishedBy
6	bko.numberOfPages
7	bko.following
8	bko.nationality

Можем да извлечем и съответните домейни:

query

1

PREFIX

rdfs:

<http://www.w3.org/2000/01/rdf-schema#>

2

PREFIX

bko:

<http://example.org/book/ontology#>

3

PREFIX

dbo:

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

4

SELECT

DISTINCT

?property

?domainClass

WHERE

{

5

VALUES

?domainClass

{

6

bko:Book

dbo:Person

7

}

8

?property

rdfs:domain

?domainClass

.

9

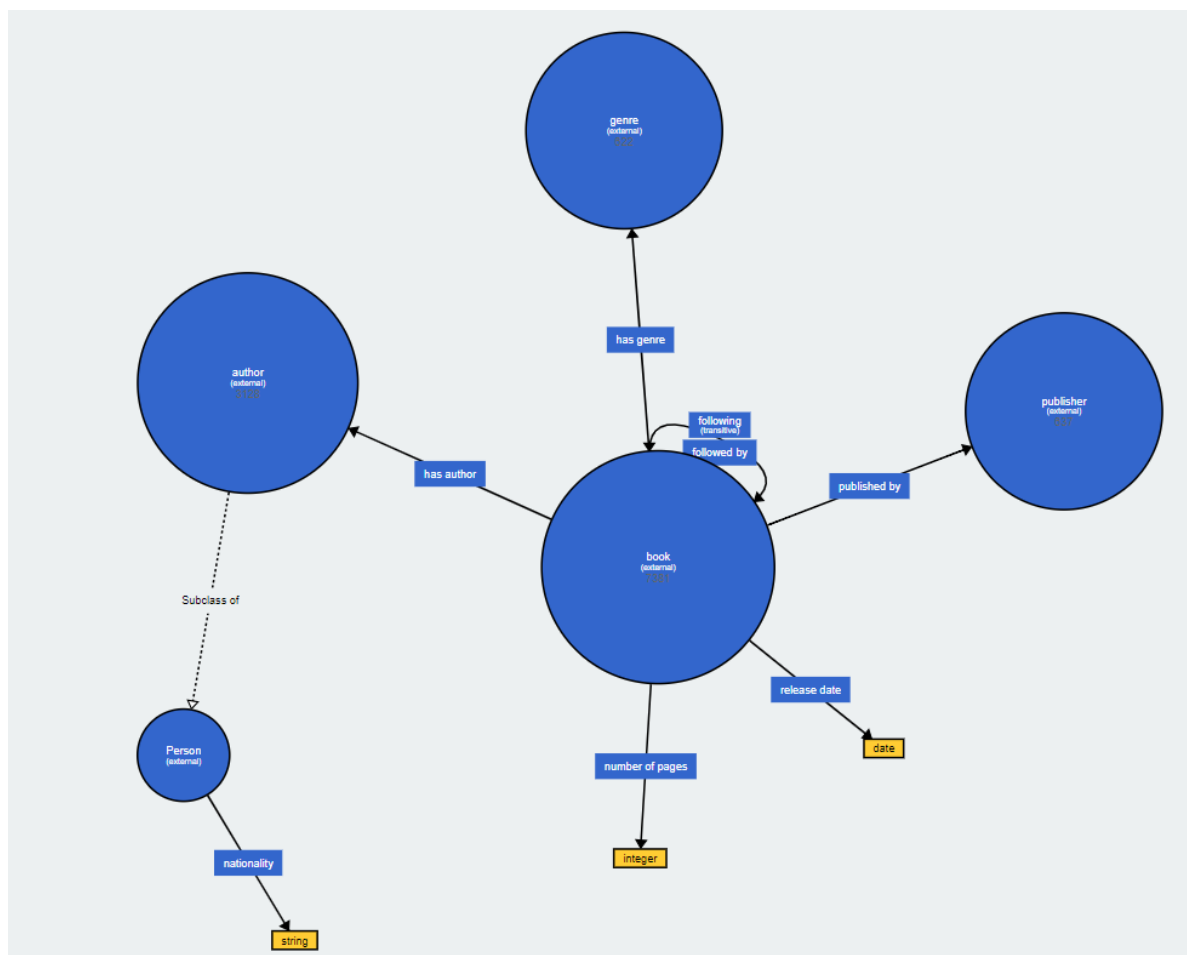
}

Press Alt+Enter to autocomplete

keyboard shortcuts

	property	domainClass
1	bko.followedBy	http://example.org/book/ontology#Book
2	bko.releaseDate	http://example.org/book/ontology#Book
3	bko.hasAuthor	http://example.org/book/ontology#Book
4	bko.hasGenre	http://example.org/book/ontology#Book
5	bko.publishedBy	http://example.org/book/ontology#Book
6	bko.numberOfPages	http://example.org/book/ontology#Book
7	bko.following	http://example.org/book/ontology#Book
8	bko.nationality	dbo:Person

Визуализации на графа:



Прости заявки върху индивидите в базата

- 1. Можем да изведем всички автори, които са част от нашата онтология:

Заявка:

```
query X ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT DISTINCT ?authorName WHERE {
4     ?author a bko:Author ;
5             dbp:name ?authorName .
6 }
```

Резултат: Заявката извежда общо 3125 уникални резултата.

Chart Config Showing results from 0 to 1000 of 3125. Query took 0.1s, minutes ago.

authorName
Émile Durkheim
Arthur Rimbaud
Nicolas Sarkozy
Voltaire
Andreas Capellanus
Chrétien de Troyes
Jean Bodel

Нека да видим и от каква националност са тези автори:

Заявка:

```
query X ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT DISTINCT ?authorName ?nationality WHERE {
4     ?author a bko:Author ;
5             dbp:name ?authorName ;
6             bko:nationality ?nationality .
7 }
```

Резултат:

Vera Sebert	Germany
Steve Pilot	Germany
Holger Twele	Germany
Georgi Gospodinov	Bulgaria
Ivan Vazov	Bulgaria
Lyubomir Miletich	Bulgaria
Petar Beron	Bulgaria
Stanislav Stratiev	Bulgaria
Hristo Gorov	Bulgaria
Dimitar Talev	Bulgaria
Nikolay Yordanov	Bulgaria

Сега, нека изведем всички книги, чиито автори са българи:

Заявка:

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?bookName ?authorName WHERE {
6     ?book a bko:Book .
7     ?book bko:hasAuthor ?author ;
8           dbp:title ?bookName .
9     ?author a bko:Author ;
10            dbp:name ?authorName ;
11            bko:nationality "Bulgaria" .
12 }
13 ORDER BY DESC(?authorName)
```

Резултат: резултатът е сортиран в намаляващ ред – общо 12 резултата

bookName	authorName
Mammoth	Stanislav Stratiev
Primer with Various Instructions	Petar Beron
Notes from Hell	Nikolay Yordanov
The Destruction of the Thracian Bulgarians in 1913	Lyubomir Miletich
Powers of Horror	Julia Kristeva
Under the Yoke	Ivan Vazov
A Newspaperman?	Ivan Vazov
The Outcasts	Ivan Vazov

2. Извличане на всички книги на даден автор:

За целта избираме да извлечем информация за всички произведения на Иван Вазов.

Заявка:

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?bookName WHERE {
6     ?book a bko:Book .
7     ?book bko:hasAuthor ?author ;
8           dbp:title ?bookName .
9     ?author a bko:Author ;
10            dbp:name "Ivan Vazov" .
11 }
12 }
```

Резултат:

Chart Config Showing results from 0 to 3 of 3. Query took 0.1s, minutes ago.

bookName
Under the Yoke
A Newspaperman?
The Outcasts

3. Търсене на автора на конкретна книга:

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?authorName WHERE {
6   ?book a bko:Book .
7   ?book bko:hasAuthor ?author ;
8         dbp:title "Under the Yoke" .
9   ?author a bko:Author ;
10          dbp:name ?authorName .
11
12 }
```

Резултат: Ivan Vazov

4. Извличане на информация за всички жанрове, част от нашата онтология:

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?genreName WHERE {
6   ?book a bko:Book .
7   ?book bko:hasGenre ?genre .
8   ?genre dbp:name ?genreName .
9 }
```

Резултат: 614 различни жанра

Chart Config	Showing results from 0 to 614 of 614. Query took 0.1s, moments ago.
genreName	
sociology	
essay	
poetry	
nonfiction	
philosophical fable	
philosophical fiction	
novella	
Bildungsroman	

5. Извличане на жанра на дадено произведение:

Ще извлечем информация за жанра на всички произведения в онтологията, които са написани от български автори.

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?authorName ?genreName WHERE {
6   ?book a bko:Book .
7   ?book bko:hasAuthor ?author ;
8         dbp:title ?bookName ;
9         bko:hasGenre ?genre .
10  ?genre dbp:name ?genreName .
11  ?author a bko:Author ;
12         dbp:name ?authorName ;
13         bko:nationality "Bulgaria" .
14 }
```

Резултат: 13 резултата

Chart Config Showing results from 0 to 13 of 13. Query took 0.1s, moments ago.

bookName	authorName	genreName
Powers of Horror	Julia Kristeva	essay
Auto-da-Fé	Elias Canetti	novel
The physics of sorrow	Georgi Gospodinov	novel
Under the Yoke	Ivan Vazov	historical fiction
A Newspaperman?	Ivan Vazov	comedy

6. Извличане на всички произведения от даден жанр:
Например, нека извлечем информация за всички произведения, които са поеми.

Заявка:

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?genreName WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8           bko:hasGenre ?genre .
9     ?genre dbp:name ?genreName .
10    FILTER(?genreName = "poem" )
11 }
```

Резултат: 22 поеми

Chart Config Showing results from 0 to 22 of 22. Query took 0.1s, moments ago.

bookName	genreName
The Maid of Orleans	poem
Don Juan in Hell	poem
On Wings of Song	poem
A Funeral Rhapsody in Memory of General Bem	poem
La prose du Transsibérien et de la Petite Jehanne de France	poem

7. Извличане на информация за всички издателства, част от нашата онтология:

Заявка:

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?publisherName WHERE {
6     ?book a bko:Book .
7     ?book bko:publishedBy ?publisher .
8     ?publisher dbp:name ?publisherName .
9 }
```


Резултат: 637 уникални издателства

Chart Config	Showing results from 0 to 637 of 637. Query took 0.2s, moments ago.
publisherName	
Plon	
Q55698991	
Pierre-Jules Hetzel	
Éditions Gallimard	
fiction literature	
University of Wales Press	
Alfred A. Knopf	

8. Извличане на издателството на дадено произведение:

Заявка:

```
query X ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?publisherName WHERE {
6   ?book a bko:Book .
7   ?book bko:publishedBy ?publisher ;
8         dbp:title "Under the Yoke" .
9   ?publisher dbp:name ?publisherName .
10 }
```

Решение: 1 резултат – Todor Chipev

9. Извличане на всички произведения, издадени от конкретно издателство:

Ще извлечем всички произведения, които са издадени под името на “Viking Press”.

Заявка:

```
query X ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName WHERE {
6   ?book a bko:Book .
7   ?book dbp:title ?bookName ;
8         bko:publishedBy ?publisher .
9   ?publisher dbp:name "Viking Press" .
10 }
```

Резултат: 34 резултата

Chart Config	Showing results from 0 to 34 of 34. Query took 0.1s, moments ago.
bookName	
The Thibaults	
Monsieur	
The Eye in the Door	
The Finishing School	
The Birthday Present	

10. Извличане на годината на издаване на дадено произведение:

Заявка:

```
query × ⊕
1 ▾ PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 ▾ SELECT ?bookName ?releaseDate WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:releaseDate ?releaseDate .
9 }
```

Резултат:

bookName	releaseDate
The Rules of Sociological Method	Jan 1, 1895
Le Bateau ivre	Jan 1, 1871
Poésies	Jan 1, 1895
Voyelles	Oct 5, 1883

11. Извличане на информация за броя на страниците на дадено произведение:

Заявка:

```
query × ⊕
1 ▾ PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 ▾ SELECT ?bookName ?pages WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:numberOfPages ?pages .
9 }
```

Резултат:

bookName	pages
Irène	62
L'Ecosaise	204
Master of the World	317
Paris in the Twentieth Century	216
No Exit	65
A Happy Death	230

Операции с множества

1. Операция UNION:

Ще изведем всички произведения, които са или трилъри, или са написани от J. K. Rowling.

Заявка:

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?authorName ?genreName WHERE {
6   {
7     ?book a bko:Book .
8     ?book dbp:title ?bookName ;
9           bko:hasGenre ?genre ;
10          bko:hasAuthor ?author .
11     ?genre dbp:name ?genreName .
12     ?author dbp:name ?authorName .
13     FILTER(?genreName = "thriller")
14   }
15   UNION
16   {
17     ?book a bko:Book .
18     ?book dbp:title ?bookName ;
19           bko:hasGenre ?genre ;
20           bko:hasAuthor ?author .
21     ?genre dbp:name ?genreName .
22     ?author dbp:name ?authorName .
23     FILTER(?authorName = "J. K. Rowling")
24   }
25 }
```

Резултат: 217 резултата общо – извеждаме името на произведението, автора и жанра

Interview mit Ed	Ivar Leon Menger	thriller
Das böse Zimmer - Teil 1	Hendrik Buchna	thriller
Das System	Karl Olsberg	thriller
my bad heart	Wulf Dorn	thriller
The Casual Vacancy	J. K. Rowling	political fiction
The Casual Vacancy	J. K. Rowling	tragicomedy
The Casual Vacancy	J. K. Rowling	black comedy
Harry Potter and the Philosopher's Stone	J. K. Rowling	adventure fiction

В този случаи се обединяват всички произведения, които са трилъри, с всички произведения, които са написани от J. K. Rowling, независимо дали те са трилъри или не.

2. Операция MINUS:

Ще изведем всички произведения, които написани от J. K. Rowling, но ще премахнем тези, които са crime fiction.

Първо да видим колко са всички книги, написани от J. K. Rowling.

Заявка:

```
query × ⊕
1 * PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 * SELECT ?bookName ?authorName ?genreName WHERE {
6   ?book a bko:Book .
7   ?book dbp:title ?bookName ;
8         bko:hasGenre ?genre ;
9         bko:hasAuthor ?author .
10  ?genre dbp:name ?genreName .
11  ?author dbp:name ?authorName .
12  FILTER(?authorName = "J. K. Rowling")
13 }
```

Резултат: 33 резултата общо

Chart Config			Showing results from 0 to 33 of 33. Query took 0.1s, moments ago.			
bookName	authorName	genreName				
The Casual Vacancy	J. K. Rowling	political fiction				
The Casual Vacancy	J. K. Rowling	tragicomedy				
The Casual Vacancy	J. K. Rowling	black comedy				
Harry Potter and the Philosopher's Stone	J. K. Rowling	adventure fiction				
Harry Potter and the Philosopher's Stone	J. K. Rowling	fantasy				
Harry Potter and the Philosopher's Stone	J. K. Rowling	young adult fiction				
Harry Potter and the Chamber of Secrets	J. K. Rowling	adventure fiction				
Harry Potter and the Chamber of Secrets	J. K. Rowling	fantasy				
Harry Potter and the Chamber of Secrets	J. K. Rowling	young adult fiction				
Harry Potter and the Goblet of Fire	J. K. Rowling	adventure fiction				

Сега да видим колко са резултатите, след като премахнем всички произведения на J. K. Rowling, които са „crime fiction”.

Заявка:

```
query × ⊕
1 * PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 * SELECT ?bookName ?authorName ?genreName WHERE {
6   {
7     ?book a bko:Book .
8     ?book dbp:title ?bookName ;
9           bko:hasGenre ?genre ;
10          bko:hasAuthor ?author .
11    ?genre dbp:name ?genreName .
12    ?author dbp:name ?authorName .
13    FILTER(?authorName = "J. K. Rowling")
14  }
15  MINUS
16  {
17    ?book a bko:Book .
18    ?book dbp:title ?bookName ;
19          bko:hasGenre ?genre ;
20          bko:hasAuthor ?author .
21    ?genre dbp:name ?genreName .
22    ?author dbp:name ?authorName .
23    FILTER(?genreName = "crime fiction")
24  }
25 }
```

Резултат: 27 резултата общо

Chart Config

Showing results from 0 to 27 of 27. Query took 0.1s, minutes ago.

bookName	authorName	genreName
The Casual Vacancy	J. K. Rowling	political fiction
The Casual Vacancy	J. K. Rowling	tragicomedy
The Casual Vacancy	J. K. Rowling	black comedy
Harry Potter and the Philosopher's Stone	J. K. Rowling	adventure fiction
Harry Potter and the Philosopher's Stone	J. K. Rowling	fantasy
Harry Potter and the Philosopher's Stone	J. K. Rowling	young adult fiction
Harry Potter and the Chamber of Secrets	J. K. Rowling	adventure fiction

Можем да заключим, че J. K. Rowling има 5 написани произведения, които са от жанр „crime fiction”.

Отговаряне на по-сложни въпроси чрез FILTER

Чрез операцията FILTER можем да филтрираме произведения по зададени критерии – да включваме/изключваме определени жанрове, автори, да задаваме диапазон на годините на издаване и диапазон на брой страници.

1. Ще изведем всички произведения, които притежават следните характеристики:
- a. Жанр – fantasy, young adult fiction, comedy

b. Автор – всички автори, освен тези от френски произход

c. Година на издаване – между 2000г. и 2024г.

d. Брой страници – между 360 и 520 страници

e. Издателство – без ограничения

Чрез задаване на толкова конкретни критерии, можем значително по-лесно да намираме произведенията, които ни интересуват.

Заявка:

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?authorName ?nationality ?genreName ?year ?pages WHERE {
6     BIND(YEAR(?releaseDate) as ?year)
7     ?book a bko:Book .
8     ?book dbp:title ?bookName ;
9         bko:hasGenre ?genre ;
10        bko:hasAuthor ?author ;
11        bko:releaseDate ?releaseDate ;
12        bko:numberOfPages ?pages .
13    ?genre dbp:name ?genreName .
14    ?author dbp:name ?authorName ;
15        bko:nationality ?nationality .
16    FILTER(?genreName in ("fantasy", "young adult fiction", "comedy"))
17    FILTER(?nationality not in ("France"))
18    FILTER(?year ≥ 2000 && ?year ≤ 2024)
19    FILTER(?pages ≥ 360 && ?pages ≤ 520)
20 }
```

Резултат: получаваме само 4 произведения, които отговарят на зададените критерии

bookName	authorName	nationality	genreName	year	pages
The Legend of Sigurd and Gudrún	Christopher Tolkien	United Kingdom	fantasy	2,009	384
The Legend of Sigurd and Gudrún	J. R. R. Tolkien	United Kingdom	fantasy	2,009	384
The Buried Giant	Kazuo Ishiguro	United Kingdom	fantasy	2,015	415
Something Rotten	Jasper Fforde	United Kingdom	fantasy	2,004	393

Създаване на нови променливи чрез BIND

Можем да използваме функцията BIND, за да създаваме нови променливи в рамките на заявката чрез присвояване на стойности и резултати от изрази. Това е полезно за изпълнение на различни операции като изчисления, преобразувания или добавяне на допълнителни данни към резултатите. В нашия случай, например, чрез BIND можем да създадем допълнителна променлива, която указва дали произведението е сравнително ново (издадено след 2022г.), дали произведението е кратко, средно дълго или дълго и т.н.

Ще преработим малко горната заявка.

Заявка:

query × ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?authorName ?nationality ?genreName ?year ?basedOnYear ?pages ?basedOnPages WHERE {
6   VALUES ?genres {"fantasy" "young adult fiction" "comedy"}
7   ?book a bko:Book .
8   ?book dbp:title ?bookName ;
9         bko:hasGenre ?genre ;
10        bko:hasAuthor ?author ;
11        bko:releaseDate ?releaseDate ;
12        bko:numberOfPages ?pages .
13   ?genre dbp:name ?genreName .
14   ?author dbp:name ?authorName ;
15          bko:nationality ?nationality .
16   FILTER(?genreName in (?genres))
17   FILTER(?nationality not in ("France"))
18   FILTER(?year ≥ 2000 && ?year ≤ 2024)
19   FILTER(?pages ≥ 140 && ?pages ≤ 520)
20   BIND(YEAR(?releaseDate) as ?year)
21   BIND(IF(?year ≥ 2015, "New Book", "Old Book") as ?basedOnYear)
22   BIND(IF(?pages ≤ 150, "Short Length", IF(?pages ≤ 450, "Medium Length", "Long Length"))) as ?basedOnPages)
23 }
```

Чрез функцията VALUES задаваме, че променливата ?genres може да приема само стойностите в скобите. Така постигаме по-подреден вид на заявката.

Нека разгледаме ред 20 – чрез функцията YEAR извличаме само годината на издаване и я записваме в променливата ?year.

Нека разгледаме ред 21 – правим проверка дали произведението е ново – всичко, издадено след 2015 година, се счита за „ново“, а всичко преди това – „за старо“. Резултатът се записва съответно в променливата ?basedOnYear.

Нека разгледаме ред 22 – правим проверка дали произведението е кратко, средно дълго или дълго. Всичко под 150 страници се счита за „short length“, всичко между 150 и 450 – за „medium length“, а всичко над 450 страници – за „long length“. Резултатът се записва в променливата ?basedOnPages.

Резултат: 7 резултата

Chart Config		Showing results from 0 to 7 of 7. Query took 0.1s, moments ago.					
bookName	authorName	nationality	genreName	year	basedOnYear	pages	basedOnPages
The Legend of Sigurd and Gudrún	Christopher Tolkien	United Kingdom	fantasy	2,009	Old Book	384	Medium Length
The Legend of Sigurd and Gudrún	J. R. R. Tolkien	United Kingdom	fantasy	2,009	Old Book	384	Medium Length
The Buried Giant	Kazuo Ishiguro	United Kingdom	fantasy	2,015	New Book	415	Medium Length
Harry Potter and the Cursed Child	J. K. Rowling	United Kingdom	fantasy	2,016	New Book	328	Medium Length

Ще променим малко заявката, за да изведем книги с различна дължина.

Заявка: само книги под 150 страници

query	×	⊕
<pre>1 PREFIX bko: <http://example.org/book/ontology#> 2 PREFIX dbp: <http://dbpedia.org/property/> 3 PREFIX dbo: <http://dbpedia.org/ontology/> 4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> 5 SELECT ?bookName ?pages ?basedOnPages WHERE { 6 ?book a bko:Book . 7 ?book dbp:title ?bookName ; 8 bko:numberOfPages ?pages . 9 FILTER(?pages ≥ 0 && ?pages ≤ 150) 10 BIND(IF(?pages ≤ 150, "Short Length", IF(?pages ≤ 450, "Medium Length", "Long Length"))) as ?basedOnPages) 11 }</pre>		

Резултат: 49 резултата

Chart Config		Showing results from 0 to 49 of 49. Query took 0.1s, moments ago.	
bookName	pages	basedOnPages	
Irène	62	Short Length	
No Exit	65	Short Length	
L'Ordre du discours	86	Short Length	
Le Diable au corps	137	Short Length	
Le Bergsonisme	119	Short Length	

Заявка: само книги между 150 и 450 страници

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?pages ?basedOnPages WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:numberOfPages ?pages .
9     FILTER(?pages ≥ 151 && ?pages ≤ 450)
10    BIND(IF(?pages ≤ 150, "Short Length", IF(?pages ≤ 450, "Medium Length", "Long Length"))) as ?basedOnPages)
11 }
```

Резултат: 156 резултата

Chart Config Showing results from 0 to 156 of 156. Query took 0.1s, moments ago.

bookName	pages	basedOnPages
L'Ecosaise	204	Medium Length
Master of the World	317	Medium Length
Paris in the Twentieth Century	216	Medium Length
A Happy Death	230	Medium Length

Заявка: само книги над 450 страници

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?pages ?basedOnPages WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:numberOfPages ?pages .
9     FILTER(?pages ≥ 451)
10    BIND(IF(?pages ≤ 150, "Short Length", IF(?pages ≤ 450, "Medium Length", "Long Length"))) as ?basedOnPages)
11 }
```

Резултат: 45 резултата

Chart Config Showing results from 0 to 45 of 45. Query took 0.1s, moments ago.

bookName	pages	basedOnPages
Journey to the End of the Night	623	Long Length
Madness and Civilization	672	Long Length
Histoire naturelle du Sénégal	624	Long Length
Capital and Ideology	1,197	Long Length
The Unknown Revolution	690	Long Length

Aggregation Functions

Както знаем, агрегиращите функции (като COUNT, SUM, AVG, MIN, MAX) са полезни за изчисляване на статистики, събиране на данни и получаване на обобщена информация. В нашия случай, можем да ги използваме, за да преброим колко автори са от дадена националност, колко произведения притежава даден автор, изчисляване на средната дължина на книгите на даден автор и т.н. Това би било изключително полезно при съставянето на някакви статистики.

Нека покажем как се работи с тези функции чрез няколко примера.

1. COUNT – да преброим колко са авторите от дадена националност

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?nationality (COUNT(?author) as ?number) WHERE {
6     ?author a bko:Author ;
7             bko:nationality ?nationality .
8 }
9 GROUP BY ?nationality
```

Резултат:

Chart Config		Showing results from 0 to 4 of 4. Query took 0.1s, moments ago.	
nationality		number	
France		849	
Bulgaria		10	
Germany		475	
United Kingdom		1,816	

Сега да преброим колко произведения притежава всеки автор, който е французин.

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?authorName (COUNT(?book) as ?number) WHERE {
6     ?book a bko:Book .
7     ?book bko:hasAuthor ?author .
8     ?author bko:nationality "France" ;
9             dbp:name ?authorName .
10 }
11 GROUP BY ?authorName
```

Резултат: 848 резултата

Chart Config

Showing results from 0 to 848 of 848. Query took 0.1s, moments ago.

authorName	number
Émile Durkheim	2
Arthur Rimbaud	4
Nicolas Sarkozy	1
Voltaire	51
Andreas Capellanus	1

Да изведем топ 3 на издателите, които имат най-много издадени книги.

Заявка:

query × ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?publisherName (COUNT(?book) as ?total_books) WHERE {
6     ?book a bko:Book .
7     ?book bko:publishedBy ?publisher .
8     ?publisher dbp:name ?publisherName .
9 }
10 GROUP BY ?publisherName
11 ORDER BY DESC(?total_books)
12 LIMIT 3
```

Резултат:

Chart Config

Showing results from 0 to 3 of 3. Query took 0.1s, moments ago.

publisherName	total_books
HarperCollins	164
Hodder & Stoughton	120
Macmillan Publishers	101

- SUM – да преброим общия брой страници на всички произведения на даден автор, който е от френски произход

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?authorName (SUM(?pages) as ?total_pages) WHERE {
6     ?book a bko:Book .
7     ?book bko:hasAuthor ?author ;
8           bko:numberOfPages ?pages .
9     ?author bko:nationality "France" ;
10            dbp:name ?authorName .
11 }
12 GROUP BY ?authorName
```

Резултат: 93 резултата

Chart Config		Showing results from 0 to 93 of 93. Query took 0.1s, moments ago.	
authorName		total_pages	
Voltaire		266	
Jules Verne		533	
Jean-Paul Sartre		65	
Albert Camus		230	
Louis-Ferdinand Céline		623	
Michel Foucault		1,355	

- AVG – да изведем средната дължина на произведенията на всеки автор, който е от български произход

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?authorName (AVG(?pages) as ?avg_length)
6 WHERE {
7     ?book a bko:Book .
8     ?book bko:hasAuthor ?author ;
9           bko:numberOfPages ?pages .
10    ?author bko:nationality "Bulgaria" ;
11            dbp:name ?authorName .
12 }
13 GROUP BY ?authorName
```

Резултат: 2 резултата

authorName	avg_length
Julia Kristeva	247
Georgi Gospodinov	344

Въпреки че в онтологията съществува информация за 10 различни български автора, нашата заявка връща само 2 резултата, защото само за тези двама автори съществува информация относно дължината на произведенията им.

За да се справим с този проблем, да изведем всички книги, независимо дали има информация за брой страници или не, можем да използваме OPTIONAL функцията. Сега ще изведем общия брой на книгите на всеки автор и средната дължина на всяка книга.

Заявка:

query × ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?authorName (COUNT(?book) as ?total_books) (ROUND(AVG(?pages) * 100) / 100 as ?avg_pages) WHERE {
6     ?book a bko:Book .
7     ?book bko:hasAuthor ?author .
8     ?author dbp:name ?authorName .
9     OPTIONAL {
10         ?book bko:numberOfPages ?pages .
11     }
12 }
13 GROUP BY ?authorName
14 ORDER BY DESC(?total_books)
```

Резултат: 3125 резултата – средната дължина е закръглена до втория знак след десетичната запетая

Chart Config Showing results from 0 to 1000 of 3125. Query took 0.2s, moments ago.

authorName	total_books	avg_pages
Agatha Christie	214	256
Jules Verne	77	266.5
Arthur Conan Doyle	74	305
Terry Pratchett	66	448.67
Voltaire	51	133
H. G. Wells	50	0.0
Piers Anthony	47	0.0
Arther C. Clark	44	273.5
Tanith Lee	42	0.0
P. G. Wodehouse	34	224

4. MIN/MAX – да изведем максималната дължина на произведение на всеки автор, който е с английски произход

Заявка:

```
query × ⊕
1 * PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT DISTINCT ?authorName (MAX(?pages) as ?longest)
6 * WHERE {
7     ?book a bko:Book .
8     ?book bko:hasAuthor ?author ;
9         bko:numberOfPages ?pages .
10    ?author bko:nationality "United Kingdom" ;
11           dbp:name ?authorName .
12 }
13 GROUP BY ?authorName
```

Резултат: 115 резултата

Chart Config		Showing results from 0 to 115 of 115. Query took 0.2s, moments ago.	
authorName	longest		
Christopher Tolkien	384		
J. R. R. Tolkien	384		
Philip Pullman	399		
George Orwell	86		
Joseph Conrad	375		
Robert Louis Stevenson	256		
Emma Goldman	277		
Ludwig Wittgenstein	75		
Robert Graves	368		
Caroline Lucas	208		

Заявки за логически извод

Reasoning-а позволява на потребителите да извличат скрита информация – например вместо да се извежда информация само за директни инстанции на даден клас, reasoning-а ни позволява да извеждаме информация и за под-класовете на съответен клас. Във всички заявки до тук сме използвали само включен reasoning. Ще дадем пример с 3 заявки към онтологията, която използваме за целите на проекта – с включен и изключен reasoning.

1. Може би най-добрият начин да покажем как точно работи reasoning-ът в нашата онтология е чрез извеждането на книгите, които са поредици. Всички книги, които са част от някаква поредица, притежават свойството followedBy, което „сочи“ към следващата книга в поредицата.

При изключен reasoning се извеждат само книгите, които пряко следват след дадена книга.

Заявка:

```
query × ⊕
1 * PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 * SELECT ?bookName ?followedByName WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:followedBy ?followingBook .
9     ?followingBook dbp:title ?followedByName .
10 }
```

Резултат: 1046 резултата

Chart Config

Showing results from 0 to 1000 of 1046. Query took 0.1s, moments ago.

bookName	followedByName
Yvain, the Knight of the Lion	Perceval, the Story of the Grail
Claudius Bombarnac	Foundling Mick
Foundling Mick	Captain Antifer
Michael Strogoff	Off on a Comet
Off on a Comet	The Child of the Cavern
Master of the World	Invasion of the Sea

При включен reasoning се извеждат всички книги, които следват след дадена книга. Така се включват не само преките наследници, но и тези по-надълбоко в йерархията.

Заявка:

query X +

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?followedByName WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:followedBy ?followingBook .
9     ?followingBook dbp:title ?followedByName .
10 }
```

Резултат: 4398 резултата

Chart Config

Showing results from 0 to 1000 of 4398. Query took 0.1s, moments ago.

bookName	followedByName
Yvain, the Knight of the Lion	Perceval, the Story of the Grail
Claudius Bombarnac	Foundling Mick
Claudius Bombarnac	Master of the World
Claudius Bombarnac	Invasion of the Sea
Claudius Bombarnac	Clovis Dardentor
Claudius Bombarnac	An Antarctic Mystery
Claudius Bombarnac	The Village in the Treetops
Claudius Bombarnac	The Sea Serpent
Claudius Bombarnac	Facing the Flag
Claudius Bombarnac	The Castaways of the Flag

2. Сега да изберем една поредица – например “The Sea Serpent”, за да покажем по-нагледно как работи reasoning-ът.

Заявка:

query X +

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?followedByName WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:followedBy ?followingBook .
9     FILTER(?bookName = "The Sea Serpent")
10     ?followingBook dbp:title ?followedByName .
11 }
```

Резултат:

- **Без reasoning:** 1 резултат

Получаваме само книгата, която следва директно след „The Sea Serpent”.

bookName	followedByName
The Sea Serpent	The Kip Brothers

Нека изведем книгата след “The Kip Brothers”.

query X ⊕

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?bookName ?followedByName WHERE {
6     ?book a bko:Book .
7     ?book dbp:title ?bookName ;
8         bko:followedBy ?followingBook .
9     FILTER(?bookName = "The Kip Brothers")
10    ?followingBook dbp:title ?followedByName .
11 }
```

- **Без reasoning:** 1 резултат

bookName	followedByName
The Kip Brothers	Traveling Scholarships

- **C reasoning:** 6 резултата

Chart Config Showing results from 0 to 6 of 6. Query took 0.1s, moments ago.

bookName	followedByName
The Kip Brothers	Master of the World
The Kip Brothers	Invasion of the Sea
The Kip Brothers	The Lighthouse at the End of the World
The Kip Brothers	The Golden Volcano
The Kip Brothers	Traveling Scholarships
The Kip Brothers	A Drama in Livonia

- **C reasoning:** 7 резултата

Получаваме всички книги, които са част от поредицата, и следват след „The Sea Serpent”.

bookName	followedByName
The Sea Serpent	Master of the World
The Sea Serpent	Invasion of the Sea
The Sea Serpent	The Lighthouse at the End of the World
The Sea Serpent	The Golden Volcano
The Sea Serpent	The Kip Brothers
The Sea Serpent	Traveling Scholarships
The Sea Serpent	A Drama in Livonia

- В нашите данни Author е подклас на класа Person. Следователно това също е още един хубав пример, с който да покажем как работи reasoning-а.

Ще разгледаме общо 4 случая:

- Използване на класа Person без reasoning:

Заявка:

```
query × ⊕
1 ▾ PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 ▾ SELECT ?authorName ?bookName WHERE{
6     ?book a bko:Book .
7     ?book bko:hasAuthor ?author ;
8         dbp:title ?bookName .
9     ?author a dbo:Person;
10         dbp:name ?authorName .
11 }
```

Резултат:

Заявката не извежда нищо, защото Author е подклас на Person (а всички автори в онтологията са от клас Author), а без reasoning не може да се постигне логически извод.

- Използване на класа Person с reasoning:

Резултат: 7660 резултата

Chart Config		Showing results from 0 to 1000 of 7660. Query took 0.1s, moments ago.	
authorName	bookName		
Émile Durkheim	Suicide		
Émile Durkheim	The Rules of Sociological Method		
Arthur Rimbaud	Le Bateau ivre		
Arthur Rimbaud	Poésies		
Arthur Rimbaud	Voyelles		
Arthur Rimbaud	Le Cœur supplicié		
Nicolas Sarkozy	La France pour la vie		
Voltaire	Zadig or Destiny		
Voltaire	Candide		
Voltaire	Zaire		
Voltaire	The White Bull		

С включен reasoning се извеждат всички автори от класа Author, които са наследници на класа Person.

с. Използване на класа Author без reasoning:

Заявка:

```
query × ⊕
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX dbo: <http://dbpedia.org/ontology/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 SELECT ?authorName ?bookName WHERE{
6   ?book a bko:Book .
7   ?book bko:hasAuthor ?author ;
8         dbp:title ?bookName .
9   ?author a bko:Author;
10          dbp:name ?authorName .
11 }
```

Резултат: 7660 резултата

Chart Config		Showing results from 0 to 1000 of 7660. Query took 0.1s, moments ago.	
authorName	bookName		
Émile Durkheim	Suicide		
Émile Durkheim	The Rules of Sociological Method		
Arthur Rimbaud	Le Bateau ivre		
Arthur Rimbaud	Poésies		
Arthur Rimbaud	Voyelles		
Arthur Rimbaud	Le Cœur supplicié		

Дори без включен reasoning, получаваме същия отговор като горе, защото всички автори са от клас Author, т.е. заявката търси всички директни инстанции, а не подкласове.

d. Използване на класа Author с reasoning:

Абсолютно същите резултати като в горната заявка. Няма значение в случая дали изпълняваме заявката с включен или изключен reasoning. Щеше да е от значение, ако например Author имаше подкласове като Novelist, Poet, Journalist, Playwriter и т.н.

Източници

1. SPARQL Tutorial – How to use FILTER, BIND and IF-ELSE in BIND:

https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

2. How to use ROUND in SPARQL:

<https://stackoverflow.com/questions/66296758/sparql-query-round-a-floating-number-to-third-digit-after-decimal-point>

3. Lectures