

# Курсов проект по Бази от знания

## Работа с база от знания за книги

проект тип А

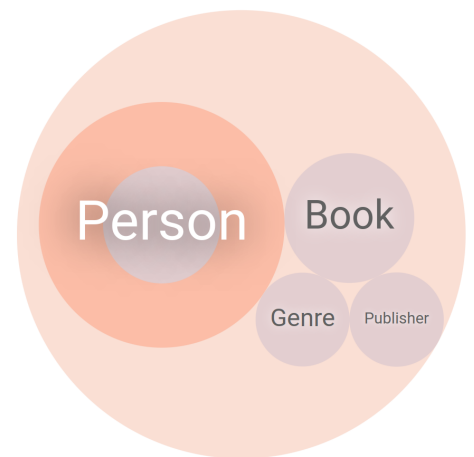
Изготвил: Васил Николаев Василев,  
ф.н. 3М13400253,  
спец. Изкуствен Интелект

## 1. Идеята на проекта е:

- 1.1. Да се предостави възможност на потребителя да извлича информация за книги
- 1.2. Да се предостави възможност на потребителя да класифицира книги в различни жанрове
- 1.3. Да може да се взимат всички книги на даден автор
- 1.4. Да може да намира поредица от книги, както и тяхната последователност
- 1.5. Да се извежда информация за книги като брой страници или година на издаване
- 1.6. Да се извеждат всички книги от даден жанр
- 1.7. Да се извеждат статистики за автори/издателства

## 2. Кратко описание на данните

Данните се описват от 4 основни класа - Book, Author, Genre, Publisher. Author е подклас на dbo:Person.



```
bko:Book a rdfs:Class ;
    rdfs:isDefinedBy <http://example.org/book/ontology#> ;
    rdfs:label "book" ;
    rdfs:subClassOf owl:Thing .

bko:Genre a rdfs:Class ;
    rdfs:isDefinedBy <http://example.org/book/ontology#> ;
    rdfs:label "genre" ;
    rdfs:subClassOf owl:Thing .

bko:Author a rdfs:Class ;
    rdfs:isDefinedBy <http://example.org/book/ontology#> ;
    rdfs:label "author" ;
    rdfs:subClassOf dbo:Person .

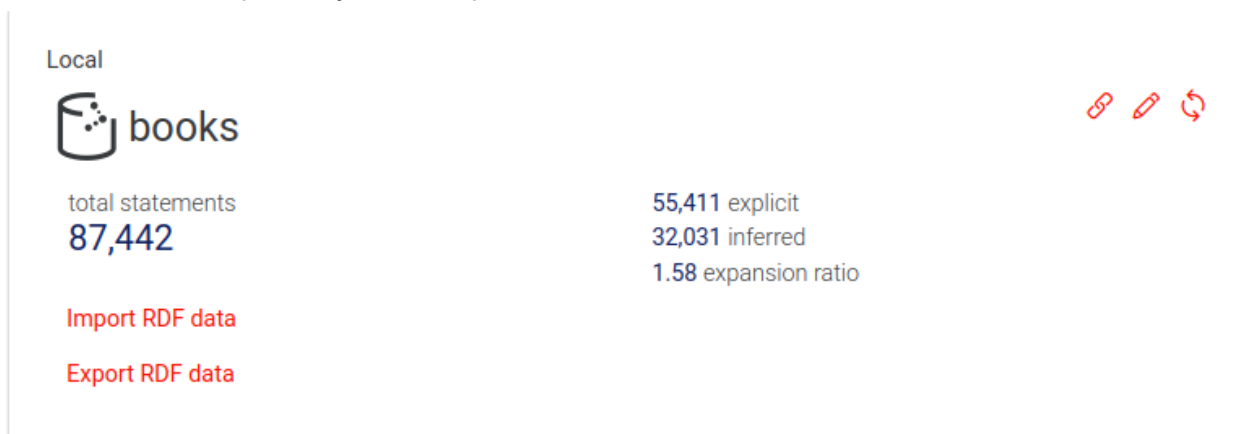
bko:Publisher a rdfs:Class ;
    rdfs:isDefinedBy <http://example.org/book/ontology#> ;
    rdfs:label "publisher" ;
    rdfs:subClassOf owl:Thing .
```

Всяка книга се описва от своето **име, жанр, автор, издателство, брой страници, дата**, на която е издадена и ако е част от поредица, **книгата която я следва** (може да се вземе и предхождащата я книга).

Основните пространства от имена, дефинирани и използвани в онтологията:

```
PREFIX bko: <http://example.org/book/ontology#>
PREFIX bkr: <http://example.org/book/resource#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

Статистика за repository-то в GraphDB след наливане на данните:



За малко повече информация върху класовете и property-тата в онтологията, може да разгледаме следните заявки:

- Заявка за извличане на всички класове в онтологията

```
1 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2
3 SELECT ?class WHERE {
4     ?class a rdfs:Class .
5 }
```

Тази заявка се пуска без да е включен reasoning-a в GraphDB с цел да се изведат само важните за онтологията класове:

	class
1	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
2	<a href="http://example.org/book/ontology#Author">http://example.org/book/ontology#Author</a>
3	<a href="http://example.org/book/ontology#Genre">http://example.org/book/ontology#Genre</a>
4	<a href="http://example.org/book/ontology#Publisher">http://example.org/book/ontology#Publisher</a>

- Може да се види, че Author е подклас на dbo:Person със следната заявка (взимайки всички подкласове на Person):

```

1 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT ?subclass
4 WHERE {
5     ?subclass rdfs:subClassOf dbo:Person .
6 }

```

В този случай резултатът е само един и той е класът за Автори на книги:

	subclass
1	<a href="http://example.org/book/ontology#Author">http://example.org/book/ontology#Author</a>

- Можем да вземем всички property-та, в които Book е domain, тъй като това е главния клас в онтологията. В заявката са включени и property-тата, в които Person е domain.

```

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 ?property ?domainClass WHERE {
5     VALUES ?domainClass {bko:Book dbo:Person}
6     ?property rdfs:domain ?domainClass .
7 }
8

```

Получаваме следните резултати:

	property	domainClass
1	<a href="http://example.org/book/ontology#followedBy">http://example.org/book/ontology#followedBy</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
2	<a href="http://example.org/book/ontology#releaseDate">http://example.org/book/ontology#releaseDate</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
3	<a href="http://example.org/book/ontology#hasAuthor">http://example.org/book/ontology#hasAuthor</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
4	<a href="http://example.org/book/ontology#hasGenre">http://example.org/book/ontology#hasGenre</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
5	<a href="http://example.org/book/ontology#publishedBy">http://example.org/book/ontology#publishedBy</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
6	<a href="http://example.org/book/ontology#numberOfPages">http://example.org/book/ontology#numberOfPages</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
7	<a href="http://example.org/book/ontology#following">http://example.org/book/ontology#following</a>	<a href="http://example.org/book/ontology#Book">http://example.org/book/ontology#Book</a>
8	<a href="http://example.org/book/ontology#nationality">http://example.org/book/ontology#nationality</a>	<a href="http://dbpedia.org/ontology/Person">http://dbpedia.org/ontology/Person</a>

Свойството `nationality` е важно за примерните заявки, които са показани.

Информацията, която се показва за свойството `following` във визуализацията на онтологията:

<

following

This book is following another in a series

following

Types:

rdf:Property

owl:TransitiveProperty

RDF Rank:

0

rdfs:comment

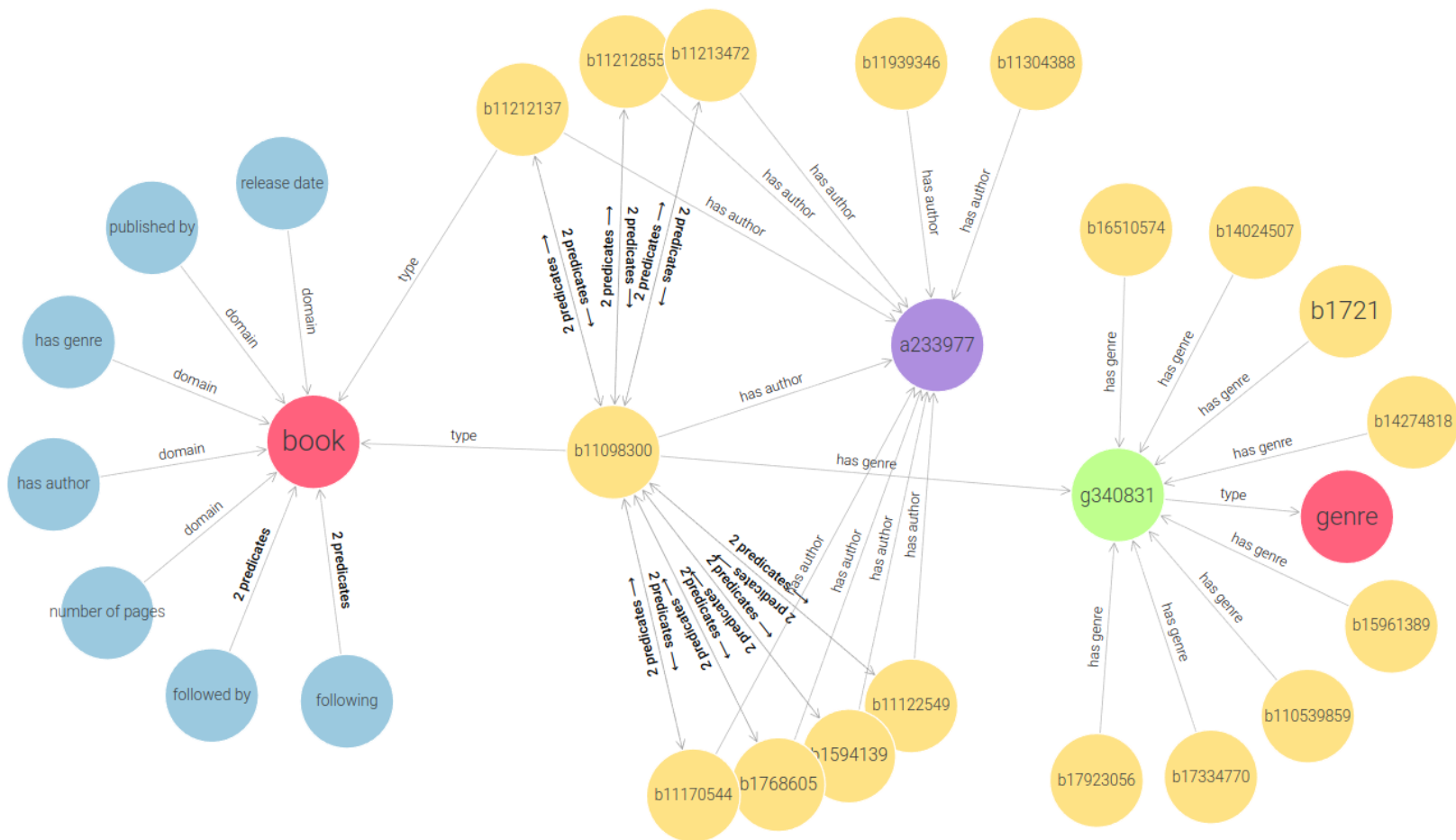
This book is following another in a series

rdfs:label

following

Може да се види, че това свойство е транзитивно. Същото важи и за `followedBy`. Тази информация е важна за следващите заявки, свързани с reasoning-а, защото на база на тях, както и на наследяването между класовете `Author` и `Person` се извличат нови знания. В примерите в точка 3.2. са дадени конкретни примери за това с описание.

Част от визуализацията на графа:



### 3. Заявки върху данните

#### 3.1. Заявки върху индивиди

- Нека първо извлечем всички книги на J.K.Rowling:

```
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 PREFIX dbp: <http://dbpedia.org/property/>
5 SELECT ?bookName WHERE {
6     ?book a bko:Book .
7     ?book bko:hasAuthor/dbp:name "J. K. Rowling" ;
8         dbp:title ?bookName .
9 }
```

Резултатите, които се получават:

	bookName
1	"The Casual Vacancy"
2	"Harry Potter and the Philosopher's Stone"
3	"Harry Potter and the Chamber of Secrets"
4	"Harry Potter and the Goblet of Fire"
5	"Harry Potter and the Order of the Phoenix"
6	"Harry Potter and the Deathly Hallows"
7	"Harry Potter and the Half-Blood Prince"
8	"Harry Potter and the Prisoner of Azkaban"
9	"Harry Potter prequel"
10	"The Cuckoo's Calling"
11	"The Silkworm"
12	"Career of Evil"
13	"Lethal White"
14	"Hogwarts: An Incomplete and Unreliable Guide"
15	"Short Stories from Hogwarts of Power, Politics and Pesky Poltergeists"
16	"Short Stories from Hogwarts of Heroism, Hardship and Dangerous Hobbies"
17	"Troubled Blood"
18	"Harry Potter and the Cursed Child"
19	"The Christmas Pig"
20	"The Ink Black Heart"

- Освен да се търсят книги на даден автор, може да се намери и авторът на книга, както и неговата националност. Например за книгата "The physics of sorrow":

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?authorName ?nationality WHERE {
4     ?book a bko:Book .
5     ?book dbp:title "The physics of sorrow" ;
6         bko:hasAuthor ?author .
7     ?author dbp:name ?authorName ;
8         bko:nationality ?nationality .
9 }

```

	authorName	nationality
1	"Georgi Gospodinov"	"Bulgaria"

- Може да се вземат и жанровете на дадена книга - например за "Harry Potter and the Philosopher's Stone":

```

1 PREFIX bkr: <http://example.org/book/resource#>
2 PREFIX bko: <http://example.org/book/ontology#>
3 PREFIX dbp: <http://dbpedia.org/property/>
4 SELECT ?genre WHERE {
5     ?book dbp:title "Harry Potter and the Philosopher's Stone" .
6     ?book bko:hasGenre/dbp:name ?genre .
7 }

```

се получават следните резултати:

	genre
1	"adventure fiction"
2	"fantasy"
3	"young adult fiction"



- Нека видим всички автори в онтологията и техните националности:

```
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 }
```

Показани са част от резултатите, тъй като както може да се види, заявката връща 3147 резултата

Filter query results		Showing results from 1 to 1,000 of 3,147. Query took 0.1s, minutes ago	
	authorName		nationality
1	"Émile Durkheim"		"France"
2	"Arthur Rimbaud"		"France"
3	"Nicolas Sarkozy"		"France"
4	"Voltaire"		"France"
5	"Andreas Capellanus"		"France"
6	"Chrétien de Troyes"		"France"
7	"Jean Bodel"		"France"
8	"Béroul"		"France"

С малко модифициране на предната заявка може да видим автори от кои държави има в онтологията:

1	"France"
2	"Bulgaria"
3	"Germany"
4	"United Kingdom"

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

- Може да се видят всички книги, написани от български автори:

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?bookName ?authorName WHERE {
4     ?book bko:hasAuthor ?author ;
5         dbp:title ?bookName .
6     ?author bko:nationality "Bulgaria" ;
7         dbp:name ?authorName .
8 }
9

```

Като резултат от заявката може да се види, че в онтологията има 12 такива книги:

Filter query results		Showing results from 1 to 12 of 12. Query took 0.1s, minutes ago.	
	bookName		authorName
1	"Powers of Horror"		"Julia Kristeva"
2	"Auto-da-Fé"		"Elias Canetti"
3	"The physics of sorrow"		"Georgi Gospodinov"
4	"Under the Yoke"		"Ivan Vazov"
5	"A Newspaperman?"		"Ivan Vazov"
6	"The Outcasts"		"Ivan Vazov"
7	"The Destruction of the Thracian Bulgarians in 1913"		"Lyubomir Miletich"
8	"Primer with Various Instructions"		"Petar Beron"
9	"Mammoth"		"Stanislav Stratiev"
10	"Poem of the Plougher"		"Hristo Gorov"
11	"The Iron Candlestick"		"Dimitar Talev"
12	"Notes from Hell"		"Nikolay Yordanov"

- Търсене на книги по жанр - може да се вземат всички книги, които принадлежат към някакъв конкретен жанр (в примера фентъзи), заедно с техните автори и тяхната националност.

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3
4 SELECT ?bookTitle ?authorName ?nationality WHERE {
5     ?book bko:hasGenre/dbp:name "fantasy" ;
6         dbp:title ?bookTitle ;
7         bko:hasAuthor ?author .
8     ?author dbp:name ?authorName ;
9         bko:nationality ?nationality .
10 }

```

В случая отново са показани част от резултатите, защото както може да се види заявката връща общо 837 резултата.

Filter query results			Showing results from 1 to 837 of 837. Query took 0.1s, m		
	bookTitle	authorName	nationality		
1	"Adventures of the Rat Family"	"Jules Verne"	"France"		
2	"An Antarctic Mystery"	"Jules Verne"	"France"		
3	"The Sea Serpent"	"Jules Verne"	"France"		
4	"Twenty Thousand Leagues Under the Sea"	"Jules Verne"	"France"		
5	"The Mysterious Island"	"Jules Verne"	"France"		
6	"Le Secret de Wilhelm Storitz"	"Jules Verne"	"France"		
7	"Master Zacharius"	"Jules Verne"	"France"		
8	"The Wolf Leader"	"Alexandre Dumas"	"France"		
9	"Her Majesty, The Cat"	"Bernard Werber"	"France"		
10	"The last Fay"	"Honoré de Balzac"	"France"		
11	"Continuidad de los parques"	"Julio Cortázar"	"France"		

### 3.2. Заявки, които демонстрират reasoning-a в GraphDB

Нека първо разгледаме заявка, която взима всички хора, които са автори на книга.

Тази заявка няма да върне нито един резултат, когато е пусната без reasoning. Това е така, защото в онтологията всички, които са писали книги са от клас Author. В случая нямаме знание, че има обект, който е от клас dbo:Person и да е автор на книга.

bko:Author обаче е подклас на dbo:Person. За това ако изпълним същата заявка, но този път с reasoning се получават следните резултати (на фигурата са обхванати 8 от общо 3125 резултата):

```
1 PREFIX dbo: <http://dbpedia.org/ontology/>
2 PREFIX bko: <http://example.org/book/ontology#>
3 PREFIX dbp: <http://dbpedia.org/property/>
4 SELECT DISTINCT ?personName WHERE {
5     ?person a dbo:Person ;
6             dbp:name ?personName .
7     ?book bko:hasAuthor ?person .
8 }
```

Filter query results		Showing results from 1 to 1,000 of 3,125. Query took 0.1s, moments ago.	
	personName		
1	"Émile Durkheim"		
2	"Arthur Rimbaud"		
3	"Nicolas Sarkozy"		
4	"Voltaire"		
5	"Andreas Capellanus"		
6	"Chrétien de Troyes"		
7	"Jean Bodel"		
8	"Béroul"		

Друг такъв пример е със заявка, която извлича за всяка книга всички книги, които я следват. Тук е важно да се отбележи, че bko:followedBy е **транзитивно свойство**.

Тоест ако имаме книга А, която е следвана от книга В, която пък от своя страна е следвана от книга С, то от книга А ще следва С. В това се изразява и разликата в това дали примерната заявка ще бъде изпълнена с или без reasoning. Когато е изпълнена без reasoning, получаваме следните резултати:

```

1 PREFIX dbp: <http://dbpedia.org/property/>
2 PREFIX bko: <http://example.org/book/ontology#>
3 select ?bookTitle ?author ?followingBookTitle where {
4     ?book dbp:title ?bookTitle ;
5         bko:followedBy ?followingBook ;
6         bko:hasAuthor/dbp:name ?author .
7     ?followingBook dbp:title ?followingBookTitle .
8 }
```

Showing results from 1 to 1,000 of 1,073. Query took 0.1s, minutes ago.

Докато            ако            е            изпълнена            с            reasoning            имаме

Showing results from 1 to 1,000 of 4,487. Query took 0.1s, moments ago.

Тоест когато сме пуснали reasoning, базата извлича нови знания.

За по-конкретен пример нека разгледаме книгата "Harry Potter and the Philosopher's Stone". Нека има следната заявка, която взима всички книги, които следват от нея:

```
1 PREFIX dbp: <http://dbpedia.org/property/>
2 PREFIX bko: <http://example.org/book/ontology#>
3 SELECT ?followingBookTitle WHERE {
4     ?book dbp:title "Harry Potter and the Philosopher's Stone" ;
5         bko:followedBy ?followingBook .
6     ?followingBook dbp:title ?followingBookTitle .
7 }
8
```

Книгата е пряко следвана само от “Harry Potter and the Chamber of Secrets”.

При изпълнение на заявката се получава следния резултат:

Filter query results		Showing results from 1 to 1 of 1. Query took 0.1s, minutes ago.	
		followingBookTitle	
1	"Harry Potter and the Chamber of Secrets"		

Това е така защото в оригиналната онтология единствено се знае, че книгата "Harry Potter and the Philosopher's Stone" е следвана от “Harry Potter and the Chamber of Secrets”. На базата на това знание и на транзитивното свойство на предиката bko:followedBy морат да се извлекат нови знания. За това при изпълнение на заявката с reasoning получаваме всички книги от поредицата за Хари Потър, които са след "Harry Potter and the Philosopher's Stone":

Filter query results		Showing results from 1 to 6 of 6. Query took 0.1s, moments ago.	
		followingBookTitle	
1	"Harry Potter and the Chamber of Secrets"		
2	"Harry Potter and the Goblet of Fire"		
3	"Harry Potter and the Order of the Phoenix"		
4	"Harry Potter and the Deathly Hallows"		
5	"Harry Potter and the Half-Blood Prince"		
6	"Harry Potter and the Prisoner of Azkaban"		

Също така в онтологията има и свойството bko:following, което указва да се вземат всички предхождащи книги. В оригиналната онтология няма знания дали съществува книга, която да предшества “Harry Potter and the Deathly Hallows”. За това при изпълнение на следната заявка без reasoning не се получават никакви резултати.

```
1 PREFIX dbp: <http://dbpedia.org/property/>
2 PREFIX bko: <http://example.org/book/ontology#>
3 SELECT ?previousBookTitle WHERE {
4     ?book dbp:title "Harry Potter and the Deathly Hallows" ;
5         bko:following ?previousBook .
6     ?previousBook dbp:title ?previousBookTitle .
7 }
```

В случая обаче bko:following е inverse property на bko:followedBy. За това ако заявката бъде изпълнена с reasoning се връщат следните 6 резултата:

Filter query results		Showing results from 1 to 6 of 6. Query took 0.1s, moments ago.	
	previousBookTitle		
1	"Harry Potter and the Philosopher's Stone"		
2	"Harry Potter and the Chamber of Secrets"		
3	"Harry Potter and the Goblet of Fire"		
4	"Harry Potter and the Order of the Phoenix"		
5	"Harry Potter and the Half-Blood Prince"		
6	"Harry Potter and the Prisoner of Azkaban"		

### 3.3. Заявки, които използват повече от 1 концепт и имат FILTER - примери за такива заявки са:

- Заявка, която намира всички книги, които са fantasy/adventure fiction и не са написани от J. K. Rowling

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?title ?authorName ?genreLabel
4 WHERE {
5     ?book bko:hasGenre ?genre ;
6         bko:hasAuthor ?author ;
7         dbp:title ?title .
8     ?genre dbp:name ?genreLabel .
9     ?author dbp:name ?authorName
10    FILTER (?genreLabel IN ("fantasy", "adventure fiction"))
11    FILTER (?author != "J. K. Rowling")
12 }
```

И резултатите които се получават - общо 1026, като са показани част от тях:

Filter query results		Showing results from 1 to 1,000 of 1,026. Query took 0.1s, moments ago.	
	title	authorName	genreLabel
1	"Adventures of the Rat Family"	"Jules Verne"	"fantasy"
2	"An Antarctic Mystery"	"Jules Verne"	"fantasy"
3	"The Sea Serpent"	"Jules Verne"	"fantasy"
4	"Twenty Thousand Leagues Under the Sea"	"Jules Verne"	"fantasy"
5	"The Mysterious Island"	"Jules Verne"	"fantasy"
6	"Le Secret de Wilhelm Storitz"	"Jules Verne"	"fantasy"
7	"Master Zacharius"	"Jules Verne"	"fantasy"
8	"The Wolf Leader"	"Alexandre Dumas"	"fantasy"

- Заявка, която намира всички автори от България или Франция и техни книги, които не са исторически

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT DISTINCT ?authorName ?bookTitle
4 WHERE {
5     ?book bko:hasGenre ?genre ;
6         bko:hasAuthor ?author ;
7         dbp:title ?bookTitle .
8     ?genre dbp:name ?genreLabel .
9     ?author dbp:name ?authorName ;
10         bko:nationality ?nationality .
11    FILTER (?nationality IN ("Bulgaria", "France"))
12    FILTER (?genreLabel != "historical fiction")
13 }
```

Тази заявка връща общо 1675 резултата, като тук са показани първите 12, където се вижда, че има книги от български и френски писатели:

	authorName	bookTitle
1	"Julia Kristeva"	"Powers of Horror"
2	"Elias Canetti"	"Auto-da-Fé"
3	"Georgi Gospodinov"	"The physics of sorrow"
4	"Ivan Vazov"	"A Newspaperman?"
5	"Ivan Vazov"	"The Outcasts"
6	"Lyubomir Miletich"	"The Destruction of the Thracian Bulgarians in 1913"
7	"Petar Beron"	"Primer with Various Instructions"
8	"Stanislav Stratiev"	"Mammoth"
9	"Hristo Gorov"	"Poem of the Plougher"
10	"Nikolay Yordanov"	"Notes from Hell"
11	"Émile Durkheim"	"Suicide"
12	"Émile Durkheim"	"The Rules of Sociological Method"

- Заявка, която връща всички книги от серията "Harry Potter and the ..." с книги, които я следват (ако има такива), които не са "young adult fiction".

```

1 PREFIX dbp: <http://dbpedia.org/property/>
2 PREFIX bko: <http://example.org/book/ontology#>
3 SELECT DISTINCT ?bookTitle ?followingBook (GROUP_CONCAT(?genreName; separator=", ") as ?genres) WHERE {
4     ?book a bko:Book ;
5         bko:hasGenre/dbp:name ?genreName ;
6         dbp:title ?bookTitle .
7     ?genre dbp:name ?genreName .
8     OPTIONAL { ?book bko:followedBy/dbp:title ?followingBook }
9     FILTER ( REGEX(?bookTitle, "^Harry Potter and the .*" ) )
10    FILTER NOT EXISTS { ?book bko:hasGenre/dbp:name "young adult fiction" . }
11 } GROUP BY ?bookTitle ?followingBook

```

Нека изпълним заявката **без reasoning**, за да може за дадена книга да се върнат само книгите, които пряко я следват, а не всички, които са след нея в поредицата. Тази заявка връща следните 4 резултата:

Filter query results		Showing results from 1 to 4 of 4. Query took 0.1s, moments ago.	
	bookTitle	followingBook	genres
1	"Harry Potter and the Goblet of Fire"	"Harry Potter and the Order of the Phoenix"	"adventure fiction, fantasy"
2	"Harry Potter and the Order of the Phoenix"	"Harry Potter and the Half-Blood Prince"	"fantasy"
3	"Harry Potter and the Deathly Hallows"		"adventure fiction, fantasy"
4	"Harry Potter and the Cursed Child"		"fantasy, drama fiction"



- Заявка, която връща всички книги от български автори, които са издадени преди 1900-та година

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 SELECT ?bookTitle ?authorName ?date WHERE {
5     ?book a bko:Book ;
6         bko:hasAuthor ?author ;
7         dbp:title ?bookTitle ;
8         bko:releaseDate ?date FILTER (?date < "1900-01-01T00:00:00Z"^^xsd:date) .
9     ?author bko:nationality "Bulgaria" ;
10         dbp:name ?authorName .
11 }

```

От тази заявка се получават следните 3 резултата:

	bookTitle	authorName	date
1	"Under the Yoke"	"Ivan Vazov"	*1894-01-01T00:00:00Z^^xsd:date
2	"The Outcasts"	"Ivan Vazov"	*1884-01-01T00:00:00Z^^xsd:date
3	"Primer with Various Instructions"	"Petar Beron"	*1824-01-01T00:00:00Z^^xsd:date

### 3.4. Заявки, които използват операции върху множества

- Заявка, която намира всички автори, които не са писали книга в жанр трагедия  
Тук първо се взима множеството от всички автори и от него чрез оператора MINUS

се махат тези резултати за автори, които са писали книги в жанр "tragedy".

Заявката е конфигурирана така, че да връща първите 8 резултата за демонстрацията, в противен случай връща 3077.

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 PREFIX bkr: <http://example.org/book/resource#>
4 SELECT ?authorName
5 WHERE {
6     ?author a bko:Author ;
7         dbp:name ?authorName .
8     MINUS {
9         ?book bko:hasAuthor/dbp:name ?authorName ;
10             bko:hasGenre/dbp:name "tragedy" .
11     }
12 } limit 8

```

Filter query results Showing results from 1 to 8 of 8. Query took 0.1s, moments ago.

	authorName
1	"Émile Durkheim"
2	"Arthur Rimbaud"
3	"Nicolas Sarkozy"
4	"Andreas Capellanus"
5	"Chrétien de Troyes"
6	"Jean Bodel"
7	"Bérout"
8	"Bertrand de Bar-sur-Aube"

- Заявка, която дава всички книги, които са в жанр драма или са написани от Агата Кристи

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX bkr: <http://example.org/book/resource#>
3 PREFIX dbp: <http://dbpedia.org/property/>
4 SELECT DISTINCT ?bookTitle ?authorName WHERE {
5   { ?book bko:hasGenre/dbp:name "drama";
6         dbp:title ?bookTitle;
7         bko:hasAuthor/dbp:name ?authorName .}
8   UNION
9   { ?book bko:hasAuthor/dbp:name "Agatha Christie";
10     dbp:title ?bookTitle ;
11     bko:hasAuthor/dbp:name ?authorName .}
12 }

```

В тази заявка първо се взима множеството от книги, които имат жанр “drama” заедно с техните автори. След това резултатът се обединява с множеството от книги, които имат за автор “Agatha Christie”.

Резултатите от заявката - обхванати са само първите 8 от общо 319:

Filter query results		Showing results from 1 to 319 of 319. Query took 0.1s, minutes ago.	
	bookTitle		authorName
1	"Socrates"		"Voltaire"
2	"At sea"		"Sławomir Mrożek"
3	"The Police"		"Sławomir Mrożek"
4	"Tango"		"Sławomir Mrożek"
5	"The Undivine Comedy"		"Zygmunt Krasiński"
6	"Irydion"		"Zygmunt Krasiński"
7	"The Satin Slipper"		"Paul Claudel"
8	"Francillon"		"Alexandre Dumas fils"

### 3.5. Заявки, които използват функции, задаващи стойност

- Заявка, която задава за всички книги на български или френски автори етикет за това дали книгата е с повече или по-малко от 300 страници

Чрез VALUES се задават стойности "Bulgaria" и "France" като стойности на променливата ?country.

След това се взима броя страници на всички книги с автори от тези държави.

Заявката демонстрира използването на BIND(IF()) конструкцията в случая за етиктиране на резултатите

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?bookName ?authorName ?country ?numberOfPages ?isLongBook
4 WHERE {
5     VALUES ?country { "Bulgaria" "France" }
6     ?book a bko:Book ;
7         bko:numberOfPages ?numberOfPages ;
8         dbp:title ?bookName ;
9         bko:hasAuthor ?author .
10    ?author dbp:name ?authorName ;
11           bko:nationality ?country .
12    BIND(IF(?numberOfPages > 300, "Yes", "No") AS ?isLongBook)
13 }
```

И резултатите, които се получават - показани са първите 8 от общо 119:

	bookName	authorName	country	numberOfPages	isLongBook
1	"Powers of Horror"	"Julia Kristeva"	"Bulgaria"	"247"	"No"
2	"The physics of sorrow"	"Georgi Gospodinov"	"Bulgaria"	"344"	"Yes"
3	"Irène"	"Voltaire"	"France"	"62"	"No"
4	"L'Ecosaise"	"Voltaire"	"France"	"204"	"No"
5	"Master of the World"	"Jules Verne"	"France"	"317"	"Yes"
6	"Paris in the Twentieth Century"	"Jules Verne"	"France"	"216"	"No"
7	"No Exit"	"Jean-Paul Sartre"	"France"	"65"	"No"
8	"A Happy Death"	"Albert Camus"	"France"	"230"	"No"

- Заявка, която връща всички книги на J. K. Rowling и John Grisham, които са с жанр crime fiction или detective fiction и са издадени преди по-малко от 10 години

В променливата ?ageOfBook се калкулира разликата между сегашната година и годината на издаване на книгата.

Получават се следните 7 резултата:

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?bookTitle ?authorName ?genreName ?releaseDate ?ageOfBook WHERE {
4     VALUES ?genreName { "crime fiction" "detective fiction" }
5     VALUES ?authorName { "J. K. Rowling" "John Grisham" }
6     ?book a bko:Book ;
7         bko:releaseDate ?releaseDate ;
8         dbp:title ?bookTitle ;
9         bko:hasAuthor/dbp:name ?authorName ;
10        bko:hasGenre/dbp:name ?genreName
11        BIND(YEAR(NOW()) - YEAR(?releaseDate) AS ?ageOfBook)
12        FILTER(?ageOfBook < 10)
13 } ORDER BY ?ageOfBook

```

Filter query results						Showing results from 1 to 7 of 7. Query took 0.1s, moments ago.					
	bookTitle	authorName	genreName	releaseDate	ageOfBook						
1	"The Ink Black Heart"	"J. K. Rowling"	"crime fiction"	"2022-10-30T00:00:00Z"	"1"						
2	"The Ink Black Heart"	"J. K. Rowling"	"detective fiction"	"2022-10-30T00:00:00Z"	"1"						
3	"Troubled Blood"	"J. K. Rowling"	"crime fiction"	"2020-09-15T00:00:00Z"	"3"						
4	"Lethal White"	"J. K. Rowling"	"crime fiction"	"2018-09-18T00:00:00Z"	"5"						
5	"Camino Island"	"John Grisham"	"crime fiction"	"2017-06-06T00:00:00Z"	"6"						
6	"Career of Evil"	"J. K. Rowling"	"crime fiction"	"2015-10-20T00:00:00Z"	"8"						
7	"The Silkworm"	"J. K. Rowling"	"crime fiction"	"2014-06-19T00:00:00Z"	"9"						

### 3.6. Заявки с агрегиращи функции - могат да се извлекат различни статистики от онтологията като:

- За всяка държава от онтологията по колко автори и по колко книги има

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?nationality ?numOfAuthors (COUNT(?book) as ?numOfBooks) where {
4     ?author a bko:Author ;
5         bko:nationality ?nationality .
6     ?book bko:hasAuthor ?author .
7 }
8     SELECT ?nationality (COUNT (?author) as ?numOfAuthors) where {
9         ?author a bko:Author ;
10            bko:nationality ?nationality .
11     } GROUP BY ?nationality
12 }
13 } GROUP BY ?nationality ?numOfAuthors

```

В тази заявка се изпълнява първоначално една подзаявка, която взима за всички националности броя автори. Външната заявка брой броя на книгите.

Получават се следните резултати след изпълнение на заявката:

Filter query results				Showing results from 1 to 4 of 4. Query took 0.1s, moments ago.			
	nationality		numOfAuthors		numOfBooks		
1	"France"		"849"		"1716"		
2	"Bulgaria"		"10"		"12"		
3	"Germany"		"475"		"800"		
4	"United Kingdom"		"1816"		"5164"		

- Издателството с най-много издадени книги:

Filter query results			Showing results from 1 to 1 of 1. Query took 0.1s, moments ago.		
	publishingStudioName		countBook		
1	"HarperCollins"		"164"		

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?publishingStudioName (COUNT(?book ) as ?countBook) WHERE {
4     ?book a bko:Book ;
5           bko:publishedBy/dbp:name ?publishingStudioName .
6 }
7 GROUP BY ?publishingStudioName
8 ORDER BY DESC(?countBook)
9 LIMIT 1
```

- За всеки жанр в онтологията по колко книги има:

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?genreName (COUNT(?book) as ?countBook) WHERE {
4     ?book a bko:Book ;
5           bko:hasGenre/dbp:name ?genreName .
6 }
7 GROUP BY ?genreName
8 ORDER BY DESC(?countBook)
```

Може да се разпише заявка, подобна на горната, която да извежда броя книги, които са написани във всеки жанр.

Явно най-често книгите в онтологията са с жанр "science fiction". Общо има 614 жанра, като са сортирани според това по колко книги има от всеки жанр в намаляващ ред.

Filter query results			Showing results from 1 to 614 of 614. Query took 0.1s, minutes ago.		
	genreName		countBook		
1	"science fiction"		"835"		
2	"fantasy"		"811"		
3	"novel"		"626"		
4	"nonfiction"		"377"		
5	"historical fiction"		"360"		
6	"crime fiction"		"348"		
7	"comedy"		"307"		
8	"detective fiction"		"257"		

- За всеки автор по колко книги е написал и средно по колко страници

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?authorName (COUNT(?book) as ?numOfBooks) (AVG(?numOfPages) as ?avgNumOfPages) WHERE {
4     ?author a bko:Author ;
5             dbp:name ?authorName .
6     ?book bko:hasAuthor ?author .
7     OPTIONAL {
8         ?book bko:numberOfPages ?numOfPages .
9     }
10 } GROUP BY ?authorName
11 ORDER BY DESC(?numOfBooks)

```

Не всяка книга има брой страници, но те са включени с OPTIONAL, защото заявката брои броя книги. За смятане на средния брой страници е използвана функцията AVG.

Показани са резултати за първите 8 автора:

Filter query results		Showing results from 1 to 1,000 of 3,125. Query took 0.1s, moments ago.	
	authorName	numOfBooks	avgNumOfPages
1	"Agatha Christie"	"214"	"256"
2	"Jules Verne"	"77"	"266.5"
3	"Arthur Conan Doyle"	"74"	"305"
4	"Terry Pratchett"	"66"	"448.6666666666666666666666667"
5	"Voltaire"	"51"	"133"
6	"H. G. Wells"	"50"	"0.0"
7	"Piers Anthony"	"47"	"0.0"
8	"Arther C. Clark"	"44"	"273.5"

- Авторът с най-дългата поредица в онтологията

```

1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT ?authorName (MAX(?numOfBooksInSeries) as ?maxLengthOfSeries) WHERE {
4     ?author dbp:name ?authorName .
5     { SELECT ?author ?book (COUNT (?anotherBook) as ?numOfBooksInSeries) WHERE {
6         ?book a bko:Book ;
7             bko:hasAuthor ?author ;
8             bko:followedBy ?anotherBook .
9     } GROUP BY ?author ?book
10 }
11 }
12 GROUP BY ?authorName
13 ORDER BY DESC(?maxLengthOfSeries)
14 LIMIT 1

```

Тук се използва подзаявка, която смята за всеки автор броя на книгите в различните поредици. Във външната заявка се прави групиране по самите автори, като за всеки автор се взима тази поредица, която има най-много книги.

Резултатите се сортират и се връща авторът с най-дългата поредица от книги:

Filter query results			Showing results from 1 to 1 of 1. Query took 0.1s, minutes ago.		
	authorName		maxLengthOfSeries		
1	"Jules Verne"		"56"	xsd:integer	

Заявките с агрегиращи функции могат да се ползват не само за показване на статистики, но и за извличане на информация.

- Например може да се разпише заявка, която дава книгите на Жул Верн, Георги Господинов и J. K. Rowling с техните жанрове и първа дата на публикуване

```
1 PREFIX bko: <http://example.org/book/ontology#>
2 PREFIX dbp: <http://dbpedia.org/property/>
3 SELECT DISTINCT ?bookTitle ?authorName ?firstPublication (GROUP_CONCAT(?genreName; separator=", ") as ?genres) WHERE {
4     ?book dbp:title ?bookTitle ;
5         bko:releaseDate ?firstPublication .
6     OPTIONAL { ?book bko:hasGenre ?genre .
7         ?genre dbp:name ?genreName .
8     }
9     { SELECT ?book ?authorName (MIN(?publicationDate) as ?firstPublication) WHERE {
10         VALUES ?authorName { "Georgi Gospodinov" "Jules Verne" "J. K. Rowling" }
11         ?book bko:hasAuthor/dbp:name ?authorName
12         OPTIONAL { ?book bko:releaseDate ?publicationDate . }
13     } GROUP BY ?book ?authorName
14 } GROUP BY ?book ?bookTitle ?authorName ?firstPublication
```

Тук в подзаявката се взимат всички книги на съответните автори заедно с тяхната първа дата на публикуване.

Във външната заявка се взимат само тези инстанции в онтология за тези книги, които са с тази дата на публикуване (това се прави с цел да няма дублиране на жанрове). За тях се взимат всички жанрове на тези инстанции.

Filter query results					Showing results from 1 to 93 of 93. Query took 0.1s, moments ago.				
	bookTitle		authorName		firstPublication		genres		
1	"The physics of sorrow"		"Georgi Gospodinov"		"2011-01-01T00:00:00Z"	xsd:date	"novel"		
2	"Au XXIXe siècle. La journée d'un journaliste américain en 2889."		"Jules Verne"		"1889-01-01T00:00:00Z"	xsd:date	"futuristic fiction"		
3	"Clausius Bombarnac"		"Jules Verne"		"1892-01-01T00:00:00Z"	xsd:date	"comedy, adventure fiction, travel novel"		
4	"Foundling Mick"		"Jules Verne"		"1893-01-01T00:00:00Z"	xsd:date	"adventure fiction"		
5	"Michael Strogoff"		"Jules Verne"		"1876-01-01T00:00:00Z"	xsd:date	"historical fiction, adventure fiction, military fiction"		
6	"Master of the World"		"Jules Verne"		"1904-01-01T00:00:00Z"	xsd:date	"adventure fiction, science fiction, crime fiction"		
7	"Invasion of the Sea"		"Jules Verne"		"1905-01-01T00:00:00Z"	xsd:date	"adventure fiction"		
8	"Martin Paz"		"Jules Verne"		"1875-01-01T00:00:00Z"	xsd:date	"adventure fiction"		