
Работа с онтология за пици

Симеон Христов

6MI3400191

SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI



Курсов проект по
Базы от знания

Факултет по математика и информатика
Софийски университет

Оценен от: ас. Мелания Бербатова

Юни 2023

Съдържание

1	Описание на онтологията	3
1.1	Класове	3
1.2	Свойства	3
1.2.1	Свойства на обектите	3
1.2.2	Свойства на данните	3
1.3	Визуализация на онтологията	3
2	Принципи и методи за извършване на логически извод в GraphDB	5
3	Заявки и резултати	6
3.1	Заявки върху класове	6
3.2	Заявки върху екземпляри	9
3.3	Заявки, с повече от един концепт и повече от един FILTER	14
3.4	Заявки, демонстриращи логически извод	18
3.5	Заявки, с операции върху множества	22
3.6	Заявки с функции за запазване на стойност в параметър .	24
3.7	Заявки, с агрегиращи функции	27
4	Използвани технологии	30
5	Източници	30

1 Описание на онтологията

1.1 Класове

Онтологията съдържа йерархия от класове, описваща видове пици, тяхните основи, възможни гарнитюри и степента на пикантност на тези гарнитюри. Основните класове са *Food*, *Pizza*, *PizzaTopping*, *PizzaBase* и *Spiciness*. Онтологията също така включва примитивни и дефинирани класове, както и обединени класове. Използва се оригиналната версия на онтологията, без промени, така, както е дефинирана [тук](#).

1.2 Свойства

1.2.1 Свойства на обектите

Дефинираните свойства на обектите са:

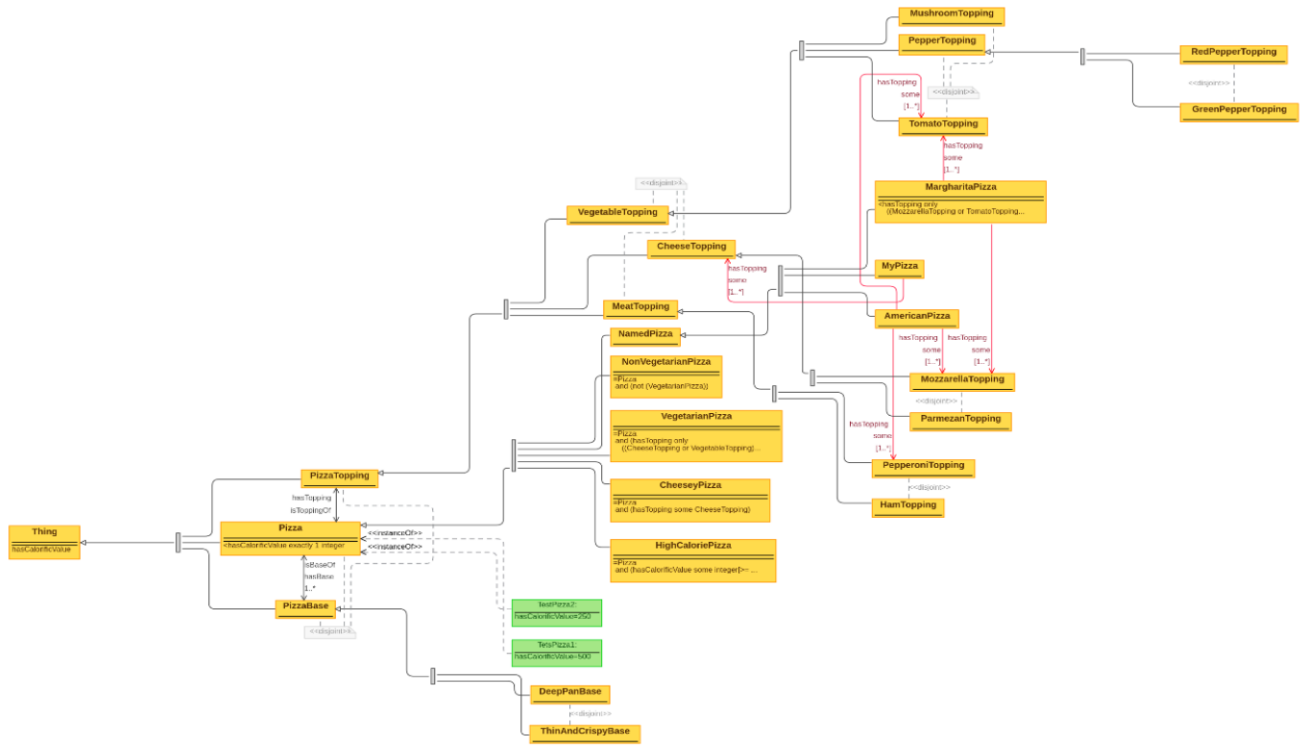
- *hasBase*: свързва пица с нейната основа.
- *isBaseOf*: обратно свойство на *hasBase*.
- *hasCountryOfOrigin*: свързва обект с неговата страна на произход.
- *hasIngredient*: свързва храна с друга храна, която е нейна съставка.
- *isIngredientOf*: обратно свойство на *hasIngredient*.
- *hasSpiciness*: свързва гарнитюра с нейното ниво на пикантност.
- *hasTopping*: свързва пица с нейните гарнитюри.
- *isToppingOf*: обратно свойство на *hasTopping*.

1.2.2 Свойства на данните

В използваната онтология не са дефинирани свойства на данните.

1.3 Визуализация на онтологията

Онтологията е представена визуално на [Фигура 1](#).



Фигура 1: Визуализация на онтологията *pizza.owl*.

2 Принципи и методи за извършване на логически извод в GraphDB

GraphDB съхранява експлицитни и имплицитни твърдения (твърдения, изведени от експлицитни твърдения). Когато експлицитно твърдение се премахне от базата знания, всички имплицитни твърдения, които са изведени от него също трябва да бъдат премахнати.

Премахването на експлицитни твърдения се постига чрез анулиране на изведените твърдения, които вече не могат да бъдат извлечени. Един подход е да се поддържа информация за проследяване за всяко твърдение - обикновено списъкът с твърдения, които могат да бъдат изведени от това твърдение. Списъкът се изгражда по време на извода, тъй като правилата се прилагат и твърденията, изведени от правилата, се добавят към списъците на всички твърдения, които са задействали изводите. Недостатъкът на тази техника е, че информацията за веригата расте по-бързо от подразбираното затваряне. Друг подход е да се извърши обратен извод. Този вид извод не изисква информация за веригата, тъй като той по същество я преизчислява. За тази цел се използва флаг за всяко твърдение, така че алгоритъмът да може да открие кога даден възел е бил посетен преди това и по този начин да избегне безкрайна рекурсия.

Алгоритъмът, използван в GraphDB, работи по следния начин:

1. Поставя се флаг „посетен“ към всички твърдения (по подразбиране стойността е *не*).
2. Създава се списък *L* и в него се съхраняват твърденията, които трябва да се премахнат.
3. Всяко твърдение в *L*, което не е посетено все още, се маркира като посетено и към него се прилагат правилата за прав извод.
4. Ако няма повече непосетени твърдения в *L*, тогава КРАЙ.
5. Всички изведени твърдения се добавят в списък *L1*.
6. За всеки елемент в *L1* се прави проверката:

- (а) Ако твърдението е чисто имплицитно (ако е и имплицитно, и експлицитно не се счита за чисто имплицитно), се маркира като изтрито и се проверява дали се поддържа от други твърдения. Методът `isSupported()` използва заявки, които съдържат предпоставките на правилата и променливите на правилата са предварително обвързани с помощта на въпросния оператор. Това означава, че методът `isSupported()` започва от проекцията на заявката и след това проверява дали заявката ще върне резултати (поне един), т.е. този метод извършва обратен извод.
- (б) Ако резултатът е върнат от която и да е заявка (всяко правило е представено от заявка) в `isSupported()`, тогава това твърдение все още може да бъде извлечено от други твърдения в базата знания и затова не се изтрива.
- (в) Ако всички заявки не върнат резултати, тогава това твърдение вече не може да бъде извлечено от други твърдения, така че състоянието му остава „изтрито“ и броячът на твърденията се актуализира.

7. $L := L1$ и се преминава към стъпка 3.

3 Заявки и резултати

3.1 Заявки върху класове

Заявка 1, намираща всички гарнитури, които са *FishTopping*.

```
PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?topping
WHERE { ?topping rdfs:subClassOf pizza:FishTopping }
```

Резултатът от изпълнението е показан на [Фигура 2](#).

	topping	↕
1	pizza:AnchoviesTopping	
2	pizza:PrawnsTopping	
3	pizza:MixedSeafoodTopping	

Фигура 2: Резултат от изпълнение на заявка 1.

	pizza	
1	pizza:Capricciosa	
2	pizza:FourSeasons	
3	pizza:Napoletana	
4	pizza:Siciliana	

Фигура 3: Резултат от изпълнение на заявка 2.

Заявка 2, намираща всички пици, които имат гарнитура *AnchoviesTopping*.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?pizza
WHERE {
    ?pizza rdfs:subClassOf pizza:NamedPizza .
    ?pizza rdfs:subClassOf ?restriction .
    ?restriction owl:onProperty pizza:hasTopping .
    ?restriction owl:someValuesFrom pizza:AnchoviesTopping .
}

```

Резултатът от изпълнението е показан на [Фигура 3](#).

	name	
1	pizza:America	
2	pizza:England	
3	pizza:France	
4	pizza:Germany	
5	pizza:Italy	

Фигура 4: Резултат от изпълнение на заявка 3.

3.2 Заявки върху екземпляри

Заявка 3, намираща всички екземпляри.

```
PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?name
WHERE { ?name a owl:NamedIndividual . }
```

Резултатът от изпълнението е показан на [Фигура 4](#).

	name	string	substring
1	pizza:America	"http://www.co-ode.org/ontologies/pizza/pizza.owl#America"	"America"
2	pizza:England	"http://www.co-ode.org/ontologies/pizza/pizza.owl#England"	"England"
3	pizza:Italy	"http://www.co-ode.org/ontologies/pizza/pizza.owl#Italy"	"Italy"

Фигура 5: Резултат от изпълнение на заявка 4.

Заявка 4, намираща всички екземпляри, чиито имена започват с гласна буква ('a', 'e', 'i', 'o' или 'u').

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?name ?string ?substring
WHERE {
    ?name a owl:NamedIndividual .
    BIND(STR(?name) AS ?string) .
    BIND(SUBSTR(?string, 50) AS ?substring) .
    FILTER REGEX (?substring, "^(a|e|i|o|u)", "i") .
}

```

Резултатът от изпълнението е показан на [Фигура 5](#).

	countryName	numPizzas
1	pizza:America	"2"^^xsd:integer
2	pizza:Italy	"2"^^xsd:integer

Фигура 6: Резултат от изпълнение на заявка 5.

Заявка 5, намираща броя пици за всяка държава, за която са създадени пици.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?countryName (COUNT (DISTINCT ?name) AS ?numPizzas)
WHERE {
    ?name rdfs:subClassOf pizza:NamedPizza, ?restr .
    ?restr owl:onProperty pizza:hasCountryOfOrigin ;
           owl:hasValue ?countryName .
}
GROUP BY ?countryName

```

Резултатът от изпълнението е показан на [Фигура 6](#).

	countryName	pizzaName	spiciness
1	pizza:America	pizza:American	pizza:Mild
2	pizza:America	pizza:American	pizza:Medium
3	pizza:America	pizza:AmericanHot	pizza:Hot
4	pizza:America	pizza:AmericanHot	pizza:Mild
5	pizza:America	pizza:AmericanHot	pizza:Medium
6	pizza:Italy	pizza:Napoletana	pizza:Mild
7	pizza:Italy	pizza:Veneziana	pizza:Mild
8	pizza:Italy	pizza:Veneziana	pizza:Medium

Фигура 7: Резултат от изпълнение на заявка 6.

Заявка 6, намираща степента на пикантност на пиците за всяка държава.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?countryName ?pizzaName ?spiciness
WHERE {
    ?pizzaName rdfs:subClassOf pizza:NamedPizza,
                ?restrCountry,
                ?restrTopping .
    ?restrTopping owl:onProperty pizza:hasTopping ;
                  owl:someValuesFrom ?topping .
    ?topping rdfs:subClassOf ?restrToppingSpiciness.
    ?restrToppingSpiciness owl:onProperty pizza:hasSpiciness ;
                           owl:someValuesFrom ?spiciness .
    ?restrCountry owl:onProperty pizza:hasCountryOfOrigin ;
                  owl:hasValue ?countryName .
}

```

Резултатът от изпълнението е показан на [Фигура 7](#).

	countryName	hotPizzaCount
1	pizza:America	"1"^^xsd:integer

Фигура 8: Резултат от изпълнение на заявка 7.

Заявка 7, намираща държавите, за които са създадени пици със степен на пикантност *Hot* и броя на тези пици.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT
  DISTINCT ?countryName
  (COUNT(DISTINCT ?pizzaName) AS ?hotPizzaCount)
WHERE {
  ?pizzaName rdfs:subClassOf pizza:NamedPizza,
                                     ?restrCountry,
                                     ?restrTopping .
  ?restrTopping owl:onProperty pizza:hasTopping ;
                 owl:someValuesFrom ?topping .
  ?topping rdfs:subClassOf ?restrToppingSpiciness .
  ?restrToppingSpiciness owl:onProperty pizza:hasSpiciness ;
                         owl:someValuesFrom pizza:Hot .
  ?restrCountry owl:onProperty pizza:hasCountryOfOrigin ;
                 owl:hasValue ?countryName .
}
GROUP BY ?countryName

```

Резултатът от изпълнението е показан на [Фигура 8](#).

	pizza	topping
1	pizza:American	pizza:MozzarellaTopping
2	pizza:American	pizza:PeperoniSausageTopping
3	pizza:American	pizza:TomatoTopping
4	pizza:AmericanHot	pizza:MozzarellaTopping
5	pizza:AmericanHot	pizza:PeperoniSausageTopping

Фигура 9: Частичен резултат от изпълнение на заявка 8.

3.3 Заявки, с повече от един концепт и повече от един FILTER

Заявка 8, намираща всички гарнитури на пиците, чиито имена са с повече от 5 символа.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?pizza ?topping
WHERE {
    ?pizza rdfs:subClassOf pizza:NamedPizza, ?restr .
    ?restr owl:onProperty pizza:hasTopping ;
           owl:someValuesFrom ?topping .
    FILTER NOT EXISTS {
        ?topping rdfs:subClassOf ?restrToppingSpiciness .
        ?restrToppingSpiciness owl:onProperty pizza:hasSpiciness;
                               owl:someValuesFrom pizza:Hot .
    }
    FILTER ((STRLEN(str(?pizza)) - 49) > 5)
}
ORDER BY ?pizza

```

Резултатът от изпълнението е обемен (93 реда). Затова към предадените файлове има и *csv* файлове. Резултатът от тази заявка се намира в *queryresult_q8.csv*. Първите 5 реда са показани на [Фигура 9](#).

	pizza	name
1	pizza:American	pizza:PeperoniSausageTopping
2	pizza:AmericanHot	pizza:PeperoniSausageTopping
3	pizza:FourSeasons	pizza:PeperoniSausageTopping
4	pizza:Capricciosa	pizza:HamTopping
5	pizza:LaReine	pizza:HamTopping
6	pizza:Parmense	pizza:HamTopping
7	pizza:Siciliana	pizza:HamTopping
8	pizza:PolloAdAstra	pizza:ChickenTopping

Фигура 10: Резултат от изпълнение на заявка 9.

Заявка 9, намираща пиците и гарнитурите им за всички пици, които имат вид месна гарнитура, чието име започва със съгласна буква, и нямат гарнитура от вид *HotSpicedBeefTopping*.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?pizza ?name
WHERE {
    ?name rdfs:subClassOf pizza:MeatTopping.
    BIND(STR(?name) AS ?string)
    BIND(SUBSTR(?string, 50 ) AS ?substring)
    FILTER(REGEX(?substring, "^[^(a|e|i|o|u)]", "i"))

    ?pizza rdfs:subClassOf pizza:NamedPizza, ?restrTopping.
    ?restrTopping owl:onProperty pizza:hasTopping;
                  owl:someValuesFrom ?name.
    FILTER NOT EXISTS {
        ?pizza rdfs:subClassOf pizza:NamedPizza, ?restrTopping.
        ?restrTopping owl:onProperty pizza:hasTopping;
                      owl:someValuesFrom pizza:HotSpicedBeefTopping.
    }
}

```

Резултатът от изпълнението е показан на [Фигура 10](#).

Заявка 10, намираща имената на всички пици, които не са от Америка и нямат гарнитура *TomatoTopping* и гарнитура *MushroomTopping*.

```
PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?pizzaName ?toppingName
WHERE {
    ?pizzaName rdfs:subClassOf pizza:NamedPizza, ?restrTopping .
    ?restrTopping owl:onProperty pizza:hasTopping ;
                  owl:someValuesFrom ?toppingName .

    FILTER NOT EXISTS {
        ?restrTopping owl:someValuesFrom pizza:TomatoTopping .
    }
    FILTER NOT EXISTS {
        ?restrTopping owl:someValuesFrom pizza:MushroomTopping .
    }
    FILTER NOT EXISTS {
        ?pizzaName rdfs:subClassOf ?restrCountry .
        ?restrCountry owl:onProperty pizza:hasCountryOfOrigin ;
                      owl:hasValue pizza:America .
    }
}
```

Резултатът от изпълнението е обемен (160 реда) и се намира в *query-result_q10.csv*. Първите 5 реда са показани на [Фигура 11](#).

	pizzaName	toppingName
1	pizza:Cajun	pizza:MozzarellaTopping
2	pizza:Cajun	pizza:OnionTopping
3	pizza:Cajun	pizza:PeperonataTopping
4	pizza:Cajun	pizza:PrawnsTopping
5	pizza:Cajun	pizza:TobascoPepperSauce

Фигура 11: Частичен резултат от изпълнение на заявка 10.

	pizzaName	1 ^A ₂
1	pizza:NamedPizza	
2	pizza:UnclosedPizza	

Фигура 12: Резултат от изпълнение на заявка 11 без логически извод.

3.4 Заявки, демонстриращи логически извод

Заявка 11, намираща имената на всички видове пици.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?pizzaName
WHERE { ?pizzaName rdfs:subClassOf pizza:Pizza. }
```

Резултатът от изпълнението без логически извод е показан на [Фигура 12](#). Резултатът от изпълнението с логически извод е показан на [Фигура 13](#).

	pizzaName		pizza-ontology-repository	en
1	pizza.American			
2	pizza.NamedPizza			
3	pizza.AmericanHot			
4	pizza.Cajun			
5	pizza.Capricciosa			
6	pizza.Caprina			
7	pizza.Florentina			
8	pizza.FourSeasons			
9	pizza.FruttiDiMare			
10	pizza.Giardiniera			
11	pizza.LaReine			
12	pizza.Margherita			
13	pizza.Mushroom			
14	pizza.Napoletana			
15	pizza.Parmense			
16	pizza.PolloAdAstra			
17	pizza.PrinceCarlo			
18	pizza.QuattroFormaggi			
19	pizza.Rosa			
20	pizza.Siciliana			
21	pizza.SloppyGiuseppe			
22	pizza.Soho			
23	pizza.UnclosedPizza			
24	pizza.Veneziana			

Фигура 13: Резултат от изпълнение на заявка 11 с логически извод.

	pizzaName	
No data available in table		

Фигура 14: Резултат от изпълнение на заявка 12 без логически извод.

	pizzaName	
1	pizza:Napoletana	
2	pizza:Veneziana	

Фигура 15: Резултат от изпълнение на заявка 12 с логически извод.

Заявка 12, намираща пиците, създадени в Италия.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?pizzaName
WHERE {
    ?pizzaName rdfs:subClassOf pizza:Pizza,
                                     ?restrCountry .
    ?restrCountry owl:onProperty pizza:hasCountryOfOrigin ;
                  owl:hasValue pizza:Italy .
}

```

Резултатът от изпълнението без логически извод е показан на [Фигура 14](#). Резултатът от изпълнението с логически извод е показан на [Фигура 15](#).

	pizza	name
No data available in table		

Фигура 16: Резултат от изпълнение на заявка 13 без логически извод.

	pizza	name
1	pizza:Cajun	pizza:TobascoPepperSauce

Фигура 17: Резултат от изпълнение на заявка 13 с логически извод.

Заявка 13, намираща имената на пиците и гарнитурите, които имат име на гарнитура, започващо със съгласна буква и завършващо с гласна.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?pizza ?name
WHERE {
    ?name rdfs:subClassOf pizza:PizzaTopping.
    BIND(STR(?name) as ?string) .
    BIND(SUBSTR(?string, 50) as ?substring) .
    FILTER REGEX (?substring, "^[^(a|e|i|o|u)]", "i") .
    FILTER REGEX (?substring, "(a|e|i|o|u)$", "i") .

    ?pizza rdfs:subClassOf pizza:NamedPizza,
                                     ?restrTopping .
    ?restrTopping owl:onProperty pizza:hasTopping ;
                  owl:someValuesFrom ?name .
}

```

Резултатът от изпълнението без логически извод е показан на [Фигура 16](#). Резултатът от изпълнението с логически извод е показан на [Фигура 17](#).

	pizzaName
1	pizza:Scilliana
2	pizza:SloppyGiuseppe
3	pizza:Soho
4	pizza:Cajun
5	pizza:Capricciosa
6	pizza:Gardiniera

Фигура 18: Резултат от изпълнение на заявка 14.

3.5 Заявки, с операции върху множества

Заявка 14, която намира всички пици, които имат имена с начална буква "S" или имат гарнитура *PeperonataTopping*. За да се постигнат докладваните резултати е необходимо да се използва логически извод.

```
PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?pizzaName
WHERE {
  {
    ?pizzaName rdfs:subClassOf pizza:Pizza ;
              rdfs:label ?label .
    FILTER regex(?label, "^S")
  }
  UNION
  {
    ?pizzaName rdfs:subClassOf pizza:Pizza,
                  ?restrictionTopping .
    ?restrictionTopping owl:onProperty pizza:hasTopping ;
                        owl:someValuesFrom pizza:PeperonataTopping .
  }
}
```

Резултатът от изпълнението е показан на [Фигура 18](#).

	name	label
1	rdf:Alt	
2	rdf:Bag	
3	rdf:Seq	
4	rdfs:ContainerMembershipProperty	
5	rdf:XMLLiteral	

Фигура 19: Частичен резултат от изпълнение на заявка 15.

Заявка 15, която намира всички класове и техните етикети. За да се постигнат докладваните резултати е необходимо да се използва логически ИЗВОД.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?name ?label
WHERE {
    ?name rdfs:subClassOf ?cls .
    OPTIONAL {
        ?name rdfs:label ?label .
    }
}

```

Резултатът от изпълнението е обемен (228 реда) и се намира в *query-result_q15.csv*. Първите 5 реда са показани на [Фигура 19](#).

3.6 Заявки с функции за запазване на стойност в параметър

Заявка 16, която намира имената и дължините на имената на всички пици с име.

```
PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?pizza ?nameLength
WHERE {
    ?pizza rdfs:subClassOf pizza:NamedPizza;
           rdfs:label ?name.
    BIND(STRLEN(SUBSTR(STR(?pizza), 50)) AS ?nameLength)
}
ORDER BY ?nameLength ?pizza
```

Резултатът от изпълнението е показан на [Фигура 20](#).

	pizza	
1	pizza.Rosa	*4 ^{min} xsd:integer
2	pizza.Soho	*4 ^{min} xsd:integer
3	pizza.Cajun	*5 ^{min} xsd:integer
4	pizza.Caprina	*7 ^{min} xsd:integer
5	pizza.LaReine	*7 ^{min} xsd:integer
6	pizza.American	*8 ^{min} xsd:integer
7	pizza.Mushroom	*8 ^{min} xsd:integer
8	pizza.Parmense	*8 ^{min} xsd:integer
9	pizza.Siciliana	*9 ^{min} xsd:integer
10	pizza.Veneziana	*9 ^{min} xsd:integer
11	pizza.Florentina	*10 ^{min} xsd:integer
12	pizza.Margherita	*10 ^{min} xsd:integer
13	pizza.Napoletana	*10 ^{min} xsd:integer
14	pizza.AmericanHot	*11 ^{min} xsd:integer
15	pizza.Capricciosa	*11 ^{min} xsd:integer
16	pizza.FourSeasons	*11 ^{min} xsd:integer
17	pizza.Giardiniera	*11 ^{min} xsd:integer
18	pizza.PrinceCarlo	*11 ^{min} xsd:integer
19	pizza.FruttiDiMare	*12 ^{min} xsd:integer
20	pizza.PolloAdAstra	*12 ^{min} xsd:integer
21	pizza.UnclosedPizza	*13 ^{min} xsd:integer
22	pizza.SloppyGiuseppe	*14 ^{min} xsd:integer
23	pizza.QuattroFormaggi	*15 ^{min} xsd:integer

Фигура 20: Резултат от изпълнение на заявка 16.

	pizza	priceInDollars
1	pizza:Margherita	"Price not available"
2	pizza:LaReine	"Price not available"
3	pizza:VegetarianPizza	"Price not available"

Фигура 21: Резултат от изпълнение на заявка 17.

Заявка 17, която за пиците *Margherita*, *LaReine*, *VegetarianPizza* намира тяхната цена или показва текста *Price not available*, ако такава не е налична.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?pizza ?priceInDollars
WHERE {
  VALUES ?pizza {
    pizza:Margherita
    pizza:LaReine
    pizza:VegetarianPizza
  }
  OPTIONAL { ?pizza pizza:hasPrice ?price. }
  BIND(
    IF(BOUND(?price),
      ?price * "1.2"^^xsd:decimal,
      "Price not available"
    ) AS ?priceInDollars)
}

```

Резултатът от изпълнението е показан на [Фигура 21](#).

	topping	numPizzas
1	pizza:TomatoTopping	"22"^^xsd:integer
2	pizza:MozzarellaTopping	"21"^^xsd:integer
3	pizza:OliveTopping	"9"^^xsd:integer
4	pizza:GarlicTopping	"5"^^xsd:integer
5	pizza:AnchoviesTopping	"4"^^xsd:integer

Фигура 22: Резултат от изпълнение на заявка 18.

3.7 Заявки, с агрегиращи функции

Заявка 18, която за всички гарнитюри намира в колко пици се срещат те.

```

PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?topping (COUNT(DISTINCT ?pizza) AS ?numPizzas)
WHERE {
    ?pizza rdfs:subClassOf pizza:Pizza, ?restrTopping .
    ?restrTopping owl:onProperty pizza:hasTopping ;
                  owl:someValuesFrom ?topping .
}
GROUP BY ?topping
ORDER BY DESC(?numPizzas) ASC(?topping)

```

Резултатът от изпълнението е обемен (37 реда) и се намира в *query-result_q18.csv*. Първите 5 реда са показани на [Фигура 22](#).

Заявка 18, която за всички пици намира колко гарнитури имат те.

```
PREFIX pizza:<http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?pizza (COUNT(DISTINCT ?topping) AS ?numTopping)
WHERE {
    ?pizza rdfs:subClassOf pizza:NamedPizza, ?restrTopping .
    ?restrTopping owl:onProperty pizza:hasTopping ;
                  owl:someValuesFrom ?topping .
}
GROUP BY ?pizza
ORDER BY DESC(?numTopping) ASC(?pizza)
```

Резултатът от изпълнението е показан на [Фигура 23](#).

	pizza	Qun pizza-ontology-repository en
1	pizza:Gardiniera	*8 ^{xs} d:integer
2	pizza:Capricciosa	*7 ^{xs} d:integer
3	pizza:FourSeasons	*7 ^{xs} d:integer
4	pizza:PolloAdAstra	*7 ^{xs} d:integer
5	pizza:Siciliana	*7 ^{xs} d:integer
6	pizza:Veneziana	*7 ^{xs} d:integer
7	pizza:Cajun	*6 ^{xs} d:integer
8	pizza:Florentina	*6 ^{xs} d:integer
9	pizza:Soho	*6 ^{xs} d:integer
10	pizza:AmericanHot	*5 ^{xs} d:integer
11	pizza:LaReine	*5 ^{xs} d:integer
12	pizza:Napoletana	*5 ^{xs} d:integer
13	pizza:Parmense	*5 ^{xs} d:integer
14	pizza:PrinceCarlo	*5 ^{xs} d:integer
15	pizza:SloppyGiuseppe	*5 ^{xs} d:integer
16	pizza:Caprina	*4 ^{xs} d:integer
17	pizza:American	*3 ^{xs} d:integer
18	pizza:FruttiDiMare	*3 ^{xs} d:integer
19	pizza:Mushroom	*3 ^{xs} d:integer
20	pizza:Rosa	*3 ^{xs} d:integer
21	pizza:Margherita	*2 ^{xs} d:integer
22	pizza:QuattroFormaggi	*2 ^{xs} d:integer
23	pizza:UnclosedPizza	*1 ^{xs} d:integer

Фигура 23: Резултат от изпълнение на заявка 19.

4 Използвани технологии

1. [GraphDB](#) - за работа с онтологията
2. [webvowl](#) - за визуализация на онтологията

Списък на фигурите

1	Визуализация на онтологията <i>pizza.owl</i>	4
2	Резултат от изпълнение на заявка 1.	7
3	Резултат от изпълнение на заявка 2.	8
4	Резултат от изпълнение на заявка 3.	9
5	Резултат от изпълнение на заявка 4.	10
6	Резултат от изпълнение на заявка 5.	11
7	Резултат от изпълнение на заявка 6.	12
8	Резултат от изпълнение на заявка 7.	13
9	Частичен резултат от изпълнение на заявка 8.	14
10	Резултат от изпълнение на заявка 9.	15
11	Частичен резултат от изпълнение на заявка 10.	17
12	Резултат от изпълнение на заявка 11 без логически извод. .	18
13	Резултат от изпълнение на заявка 11 с логически извод. . .	19
14	Резултат от изпълнение на заявка 12 без логически извод. .	20
15	Резултат от изпълнение на заявка 12 с логически извод. . .	20
16	Резултат от изпълнение на заявка 13 без логически извод. .	21
17	Резултат от изпълнение на заявка 13 с логически извод. . .	21
18	Резултат от изпълнение на заявка 14.	22
19	Частичен резултат от изпълнение на заявка 15.	23
20	Резултат от изпълнение на заявка 16.	25
21	Резултат от изпълнение на заявка 17.	26
22	Резултат от изпълнение на заявка 18.	27
23	Резултат от изпълнение на заявка 19.	29

5 Източници

1. [Reasoning in GraphDB](#)
2. [SPARQL/Expressions and Functions](#)

3. SPARQL/Aggregate functions
4. SPARQL/FILTER