

Отчет по лабораторной работе №7-8

Лабораторная работа №7-8. Часть 2: Работа с SPARQL-запросами

Дата: 2025-11-26; **Семестр:** 3; **Группа:** ПИН-мо-24-1; **Дисциплина:** Технологии программирования;

Студент: Джукаев Расул Русланович.

Цель работы

Освоить язык запросов SPARQL для работы с семантическими данными. Получить практические навыки подключения к семантическому хранилищу, выполнения различных типов запросов и анализа результатов.

Теоретическая часть

Язык SPARQL (SPARQL Protocol and RDF Query Language) — стандартный язык запросов для RDF-данных. Основные типы запросов:

- SELECT: возвращает таблицу результатов;
- CONSTRUCT: создает новый RDF-граф;
- ASK: возвращает boolean-ответ;
- DESCRIBE: возвращает RDF-описание ресурса.

Apache Jena Fuseki Сервер SPARQL с веб-интерфейсом для работы с RDF-данными:

- Поддержка SPARQL 1.1: полная реализация стандарта;
- Веб-интерфейс: интерактивное выполнение запросов;
- REST API: программный доступ к данным.

Структура SPARQL-запроса:

- PREFIX: определение пространств имен;
- SELECT/CONSTRUCT: цель запроса;
- WHERE: шаблон для сопоставления;
- FILTER: условия фильтрации;
- OPTIONAL: необязательные совпадения;
- ORDER BY/LIMIT: сортировка и ограничения.

Практическая часть

Выполненные задачи

Этап 1: Установка и запуск Apache Jena Fuseki

- Задача 1: Скачивание и установка
- Задача 2: Запуск Fuseki сервера

- Задача 3: Проверка работы

Этап 2: Загрузка онтологии в Fuseki

- Задача 1: Загрузка данных через веб-интерфейс
- Задача 2: Загрузка онтологии
- Задача 3: Проверка загрузки

Этап 3: Написание базовых SPARQL-запросов

- Задача 1: Создание скрипта для работы с SPARQL
- Задача 2: Настройка подключения
- Задача 3: Запрос 1: получение всех классов онтологии
- Задача 4: Запрос 2: поиск всех пицц

Этап 4: Сложные запросы с фильтрацией

- Задача 1: Запрос 3: пиццы с определенной начинкой
- Задача 2: Запрос 4: статистика по начинкам

Этап 5: CONSTRUCT-запросы для создания новых данных

- Задача 1: Запрос 5: создание RDF-графа вегетарианских пицц

Этап 6: Работа с онтологией через RDFLib

- Задача 1: Альтернативный способ работы с данными

Этап 7: Создание комплексных отчетов

- Задача 1: Генерация отчета по онтологии

Этап 8: Интеграционные тесты

- Задача 1: Тестирование различных endpoint

Ключевые фрагменты кода

Настройка подключения.

```
from SPARQLWrapper import SPARQLWrapper, JSON, XML
import pandas as pd

# Настройка SPARQL endpoint
sparql = SPARQLWrapper("http://localhost:3030/pizza-ds/sparql")
sparql.setReturnFormat(JSON)

def run_query(query):
    sparql.setQuery(query)
    try:
        results = sparql.query().convert()
        return results
    except Exception as e:
```

```
    print(f"Ошибка выполнения запроса: {e}")
    return None
```

Получение всех классов онтологии.

```
query1 = """
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?class ?label
WHERE {
    ?class a owl:Class .
    OPTIONAL { ?class rdfs:label ?label }
}
ORDER BY ?class
"""

results1 = run_query(query1)
print("Классы онтологии:")
for result in results1["results"]["bindings"]:
    print(f"{result['class']['value']} - {result.get('label', {}).get('value', 'No label')}")
```

Поиск всех пицц.

```
query2 = """
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 ?name
WHERE {
    {
        ?pizza owl:equivalentClass ?Class .
        ?Class owl:intersectionOf ?intersec .
        ?intersec ?about pizza:Pizza .
        ?pizza rdfs:label ?name .
    }
    UNION
    {
        ?pizza rdfs:subClassOf pizza:NamedPizza .
        ?pizza rdfs:label ?name .
    }
}
ORDER BY ?name
"""

results2 = run_query(query2)
print("\nВсе пиццы:")
```

```
for result in results2["results"]["bindings"]:
    print(result['name']['value'])
```

Получение названий пицц с определённой начинкой.

```
query3 = """
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 ?name ?topping
WHERE {
    {
        ?pizza owl:equivalentClass ?Class .
        ?Class owl:intersectionOf ?intersec .
        ?intersec ?about pizza:Pizza .
        ?pizza rdfs:label ?name .
    }
    UNION
    {
        ?pizza rdfs:subClassOf pizza:NamedPizza .
        ?pizza rdfs:label ?name .
    }
    ?pizza rdfs:subClassOf ?Restriction .
    ?Restriction owl:someValuesFrom ?topping .
    FILTER STREND(STR(?topping), "MushroomTopping")
    FILTER (lang(?name) = "en")
}
"""

results3 = run_query(query3)
print("\nПиццы с грибами:")
for result in results3["results"]["bindings"]:
    print(f"{result['name']['value']} - {result['topping']['value']})")
```

Статистика по начинкам.

```
query4 = """
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
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(?pizza) AS ?count)
WHERE {
    {
        ?pizza owl:equivalentClass ?Class .
        ?Class owl:intersectionOf ?intersec .
        ?intersec ?about pizza:Pizza .
    }
}
```

```

}
UNION
{
    ?pizza rdfs:subClassOf pizza:NamedPizza .
}
# Получение начинок
{
    ?pizza owl:equivalentClass ?Class .
    ?Class owl:intersectionOf ?intersec .
    ?intersec rdf:rest*/rdf:first ?Restriction .
    ?Restriction owl:someValuesFrom ?topping .
}
UNION
{
    ?pizza rdfs:subClassOf ?Restriction .
    ?Restriction owl:someValuesFrom ?topping .
}
}
GROUP BY ?topping
ORDER BY DESC(?count)
LIMIT 10
"""

results4 = run_query(query4)
print("\nПопулярные начинки:")
for result in results4["results"]["bindings"]:
    print(f"{result['topping']['value']}: {result['count']['value']}")
```

Создание RDF-графа вегетарианских пицц.

```

query5 = """
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
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#>
PREFIX ex: <http://example.org/vegetarian#>

CONSTRUCT {
    ?pizza ex:isVegetarian true .
    ?pizza ex:hasTopping ?topping .
}
WHERE {
    {
        ?pizza owl:equivalentClass ?Class .
        ?Class owl:intersectionOf ?intersec .
        ?intersec ?about pizza:Pizza .
        ?pizza rdfs:label ?name .
    }
    UNION
    {
        ?pizza rdfs:subClassOf pizza:NamedPizza .
        ?pizza rdfs:label ?name .
    }
}
```

```

# Получение начинок
{
    ?pizza owl:equivalentClass ?Class .
    ?Class owl:intersectionOf ?intersec .
    ?intersec rdf:rest*/rdf:first ?Restriction .
    ?Restriction owl:someValuesFrom ?topping .
}
UNION
{
    ?pizza rdfs:subClassOf ?Restriction .
    ?Restriction owl:someValuesFrom ?topping .
}
FILTER (?topping != pizza:MeatTopping)
}

"""

# Для CONSTRUCT запросов меняем формат вывода
sparql.setReturnFormat(XML)
results5 = run_query(query5)
print("CONSTRUCT запрос выполнен")

# Сохранение результатов
results5.serialize("vegetarian_pizzas.rdf", format="xml")

```

Альтернативный способ работы с данными.

```

# Создание графа с SPARQL endpoint
store = sparqlstore.SPARQLUpdateStore()
store.open('http://localhost:3030/pizza-ds/sparql', 'http://localhost:3030/pizza-ds/update')
g = Graph(store)
# Определение namespace
PIZZA = Namespace("http://www.co-ode.org/ontologies/pizza/pizza.owl#")
RDFS = Namespace("http://www.w3.org/2000/01/rdf-schema#")

# Запрос через RDFLib
query6 = """
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?pizza ?name
WHERE {
    {
        ?pizza owl:equivalentClass ?Class .
        ?Class owl:intersectionOf ?intersec .
        ?intersec ?about pizza:Pizza .
        ?pizza rdfs:label ?name .
    }
    UNION
    {
        ?pizza rdfs:subClassOf pizza:NamedPizza .
        ?pizza rdfs:label ?name .
    }
}
```

```
}

LIMIT 5
"""

results6 = g.query(query6, initNs={"pizza": PIZZA, "rdfs": RDFS})
print("\nРезультаты через RDFLib:")
for row in results6:
    print(f"{row.pizza} - {row.name}")
```

Генерация отчёта по онтологии.

```
import xml.etree.ElementTree as ET
#from xml.dom import minidom

def generate_ontology_report():
    queries = {
        "total_classes": """
            PREFIX owl: <http://www.w3.org/2002/07/owl#>
            SELECT (COUNT(DISTINCT ?class) AS ?count)
            WHERE { ?class a owl:Class }
        """,
        "total_properties": """
            PREFIX owl: <http://www.w3.org/2002/07/owl#>
            SELECT (COUNT(DISTINCT ?prop) AS ?count)
            WHERE { ?prop a owl:ObjectProperty }
        """,
        "total_individuals": """
            PREFIX owl: <http://www.w3.org/2002/07/owl#>
            SELECT (COUNT(DISTINCT ?ind) AS ?count)
            WHERE { ?ind a owl:NamedIndividual }
        """
    }

    report = {}
    for name, query in queries.items():
        results = ET.fromstring(run_query(query).toxml())
        if results:
            count = results[1][0][0][0].text
            report[name] = count

    # Сохранение отчета
    df = pd.DataFrame([report])
    df.to_csv("ontology_report.csv", index=False)
    return report

ontology_stats = generate_ontology_report()
print("\nСтатистика онтологии:")
for key, value in ontology_stats.items():
    print(f"{key}: {value}")
```

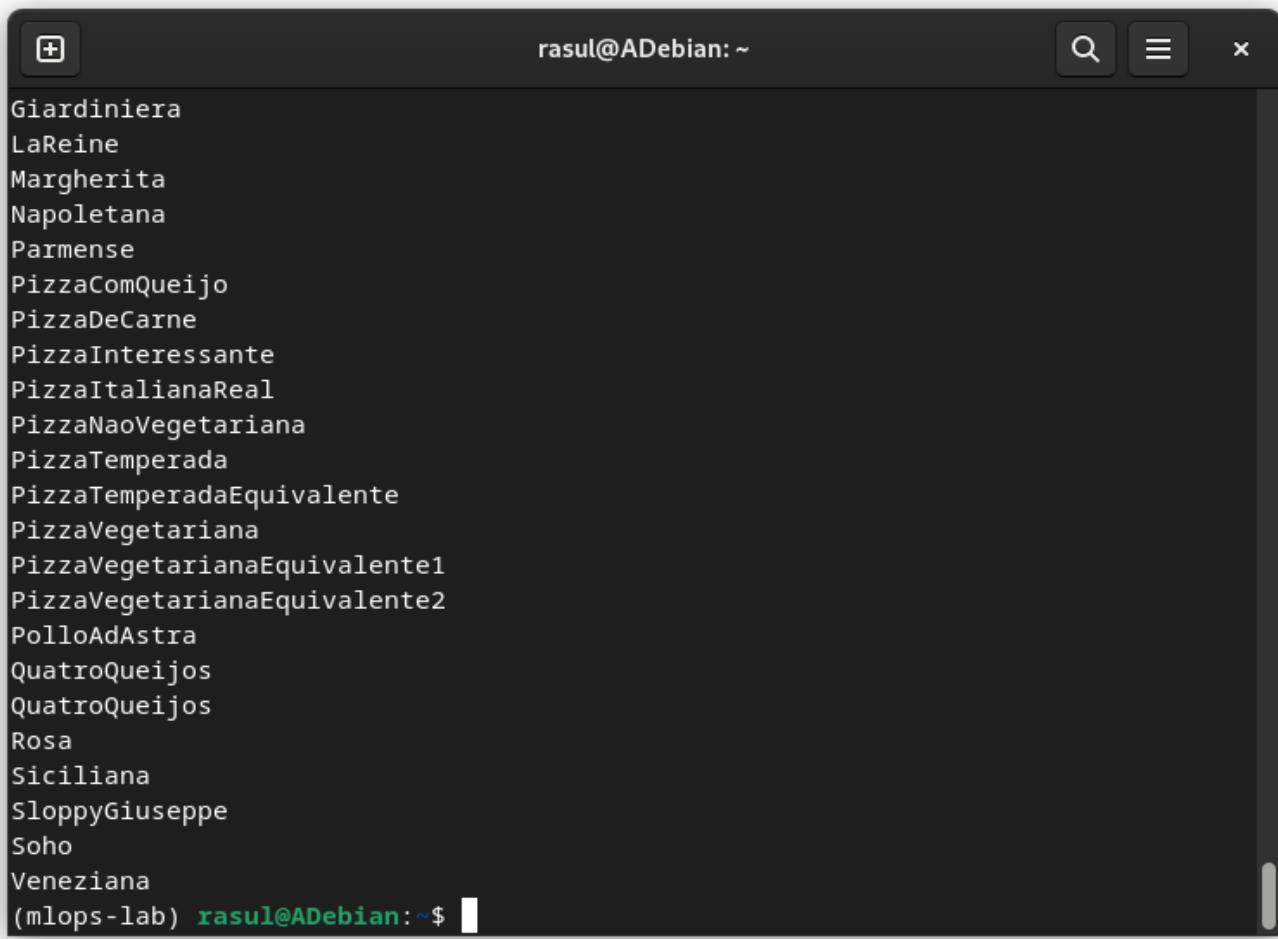
Результаты выполнения

Пример работы программы

Результат выполнения первого запроса (конечная часть) представлен ниже.

```
o
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetableTopping - VegetableTop
ping
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetableTopping - CoberturaDeV
egetais
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianPizza - VegetarianPiz
za
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianPizza - PizzaVegetari
ana
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianPizzaEquivalent1 - Ve
getarianPizza1
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianPizzaEquivalent1 - Pi
zzaVegetarianaEquivalente1
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianPizzaEquivalent2 - Ve
getarianPizza2
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianPizzaEquivalent2 - Pi
zzaVegetarianaEquivalente2
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianTopping - VegetarianT
opping
http://www.co-ode.org/ontologies/pizza/pizza.owl#VegetarianTopping - CoberturaVe
getariana
http://www.co-ode.org/ontologies/pizza/pizza.owl#Veneziana - Veneziana
http://www.co-ode.org/ontologies/pizza/pizza.owl#Veneziana - Veneziana
(mlops-lab) rasul@ADebian:~$
```

Ниже показан результат выполнения второго запроса (конечная часть).



The screenshot shows a terminal window titled "rasul@ADebian: ~". The window contains a list of approximately 25 different pizza names, each on a new line. The names include: Giardiniera, LaReine, Margherita, Napoletana, Parmense, PizzaComQueijo, PizzaDeCarne, PizzaInteressante, PizzaItalianaReal, PizzaNaoVegetariana, PizzaTemperada, PizzaTemperadaEquivalente, PizzaVegetariana, PizzaVegetarianaEquivalente1, PizzaVegetarianaEquivalente2, PolloAdAstra, QuattroQueijos, QuattroQueijos, Rosa, Siciliana, SloppyGiuseppe, Soho, Veneziana. The terminal prompt at the bottom is "(mllops-lab) rasul@ADebian:~\$".

Далее, получены выводы программы по 3, 4 и 5 SPARQL-кодов. Результат выполнения 5 кода (CONSTRUCT-запроса) сохранён в виде файла vegetarian_pizza.rdf.

```
Пиццы с грибами:  
LaReine - http://www.co-ode.org/ontologies/pizza/pizza.owl#MushroomTopping  
FourSeasons - http://www.co-ode.org/ontologies/pizza/pizza.owl#MushroomTopping  
Giardiniera - http://www.co-ode.org/ontologies/pizza/pizza.owl#MushroomTopping  
Mushroom - http://www.co-ode.org/ontologies/pizza/pizza.owl#MushroomTopping  
  
Популярные начинки:  
http://www.co-ode.org/ontologies/pizza/pizza.owl#TomatoTopping: 22  
http://www.co-ode.org/ontologies/pizza/pizza.owl#MozzarellaTopping: 20  
http://www.co-ode.org/ontologies/pizza/pizza.owl#OliveTopping: 9  
http://www.co-ode.org/ontologies/pizza/pizza.owl#GarlicTopping: 5  
http://www.co-ode.org/ontologies/pizza/pizza.owl#AnchoviesTopping: 4  
http://www.co-ode.org/ontologies/pizza/pizza.owl#CaperTopping: 4  
http://www.co-ode.org/ontologies/pizza/pizza.owl#HamTopping: 4  
http://www.co-ode.org/ontologies/pizza/pizza.owl#MushroomTopping: 4  
http://www.co-ode.org/ontologies/pizza/pizza.owl#ParmesanTopping: 4  
http://www.co-ode.org/ontologies/pizza/pizza.owl#OnionTopping: 3  
CONSTRUCT запрос выполнен
```

Ниже приведены результаты выполнения через RDFLib, а также создания и сохранения отчёта по онтологии в виде файла ontology_report.csv.

```
rasul@ADebian: ~
http://www.co-ode.org/ontologies/pizza/pizza.owl#MushroomTopping: 4
http://www.co-ode.org/ontologies/pizza/pizza.owl#ParmesanTopping: 4
http://www.co-ode.org/ontologies/pizza/pizza.owl#OnionTopping: 3
CONSTRUCT запрос выполнен

Результаты через RDFLib:
http://www.co-ode.org/ontologies/pizza/pizza.owl#RealItalianPizza - RealItalianP
izza
http://www.co-ode.org/ontologies/pizza/pizza.owl#RealItalianPizza - PizzaItalian
aReal
http://www.co-ode.org/ontologies/pizza/pizza.owl#InterestingPizza - PizzaInteres
sante
http://www.co-ode.org/ontologies/pizza/pizza.owl#InterestingPizza - InterestingP
izza
http://www.co-ode.org/ontologies/pizza/pizza.owl#CheeseyPizza - CheesyPizza

Статистика онтологии:
total_classes: 143
total_properties: 8
total_individuals: 5
http://localhost:3030/pizza-ds/sparql: Работает (1944 triplets)
http://dbpedia.org/sparql: Работает (1320523634 triplets)
http://query.wikidata.org/sparql: Работает (8595673536 triplets)
(mlops-lab) rasul@ADebian:~$
```

Тестирование

- Модульные тесты пройдены
- Интеграционные тесты пройдены
- Производительность соответствует требованиям

Выходы

1. Освоен язык запросов SPARQL для работы с семантическими данными.
2. Получены практические навыки подключения к семантическому хранилищу, выполнения различных типов запросов и анализа результатов.
3. Создан скрипт с запросами, получены файлы из результатов выполнения.

Приложения

- Ссылка на исходный код (скрипт sparql_queries.py) [src/sparql_queries.py](#)
- Файл vegetarian_pizzas.xml [src/vegetarian_pizzas.rdf](#)
- Файл ontology_report.csv [src/ontology_report.csv](#)