**Лабораторная работа № 8.**

**Разработка SPARQL-запросов к наборам RDF- данных в Web**

*Цели работы*:

1) изучение основных элементов RDF(S)- онтологий;

2) создание набора RDF- данных с использованием системы Protégé.

3) изучение языков запросов SPARQL и SPARQL Update к RDF-данным;

4) получение навыков работы с сервером Apache Jena Fuseki для выполнения SPARQL-запросов.

Порядок выполнения работы:

1) Скачать и установить систему Protégé для разработки онтологий (*если не установлена*):

<https://protege.stanford.edu/>

2) Изучить основы теории онтологий (если не изучены). В определенной мере можно использовать источники [1-3].

3) Изучить языки запросов SPARQL и SPARQL Update к RDF-данным. Для изучения языков рекомендуется книга [4];

4) Скачать и установить последнюю версию сервера Apache Jena Fuseki:

<https://jena.apache.org/documentation/fuseki2/>

Данный сервер работает как Web-сервис, выполняющий SPARQL-запросы. Для формирования SPARQL-запросов и просмотра результатов их вычислений используется Web-браузер (например, Chrome). При работе в локальном варианте для загрузки человеко-машинного интерфейса в адресной строке браузера надо ввести:

<http://localhost:3030>

5) Взять онтологию семейных отношений из предыдущей лабораторной работы, созданную в системе Protégé. Удалить из нее все SWRL-правила и аксиомы (если есть). Запомнить в формате RDF/XML. В таком виде онтология содержит только классы, свойства и индивиды, и представляет RDF-онтологию (или RDF-набор данных).

6) Разработать на языке SPARQL-запросы для нахождения всех родственников, определенных в онтологии.

Пример SPARQL-запроса для поиска женщин среди множества людей.

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX lab5: <IRI онтологии>

SELECT ?x

WHERE {

?x rdf:type lab5:Female.

?x rdf:type lab5:Person

}

*Примечание*: URI онтологии определяется из Protégé.

7) Выполнить SPARQL-запросы на сервере Apache Jena Fuseki. Проанализировать полученные результаты.

8) Добавить RDF-базу данных новые данные (новых людей), удалить и модифицировать некоторые данные с использованием языка SPARQL Update. Проанализировать полученные результаты.

Литература

1. Горшков С. Введение в онтологическое моделирование. – ТриниДата, 2016. – 166 с. <https://trinidata.ru/files/SemanticIntro.pdf>

2. Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы. Учебное пособие. – Казань, Москва, 2006. – 157 с.

<http://window.edu.ru/resource/722/41722/files/ot_2006_posobie.pdf>

3. Балашова И.Ю., Прошкина Е.Н. Онтологический инжиниринг в проектировании интеллектуальных информационных систем. Учебное пособие, Пенза: ПГУ, 2019. – 156 с. <https://yadi.sk/i/9aaabTSaYiyfeA>

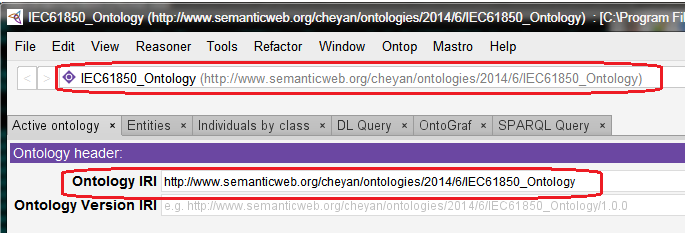
4. DuCharme B. Learning SPARQL. - O’Reilly, 2011. – 256 p. (на английском)

<https://yadi.sk/d/1VjzMHzqMa-wXw>

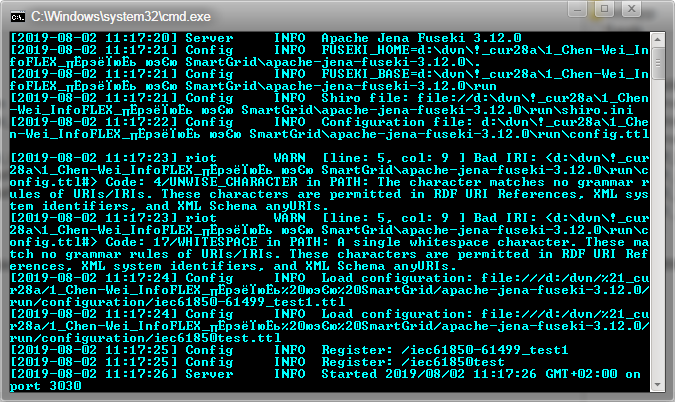
**ПРИЛОЖЕНИЕ 1**

**Работа со SPARQL-сервером Apache Jena Fuseki**

1. Создать RDF-онтологию (семейных отношений) и запомнить ее в формате RDF/XML (файл с расширением .owl). Запомнить ее идентификатор IRI. Где его можно найти, показано ниже (поле Ontology IRI):



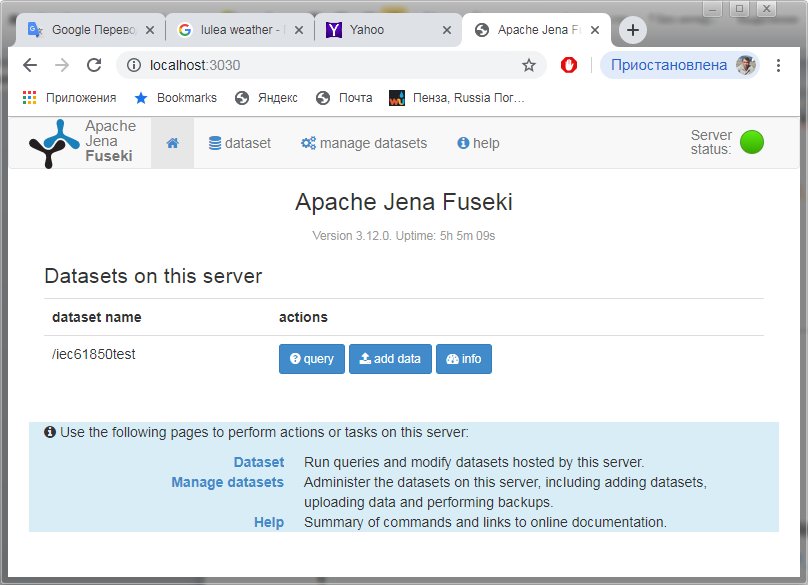
2) Запустить SPARQL-сервер *Apache Jena Fuseki* с помощью командного файла *fuseki-server.bat*.



3) Скопировать файл c RDF-онтологией в каталог сервера *Apache Jena Fuseki*.

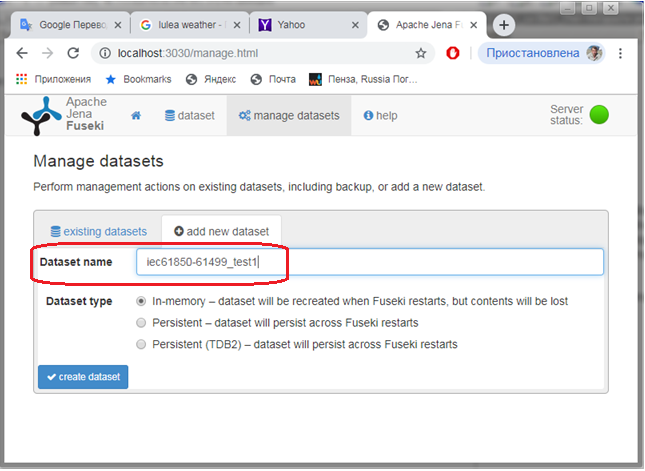
4) Запустить Web-браузер (например, Chrome) и ввести в его адресной строке URL SPARQL-сервера: <http://localhost:3030>

В браузере появится начальная страница сервера *Apache Jena Fuseki*:

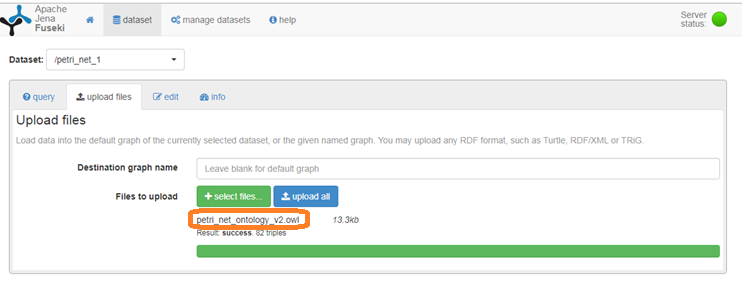


Таким образом, в данном случае SPARQL-сервер и клиент (Web-браузер) будут работать на одном компьютере.

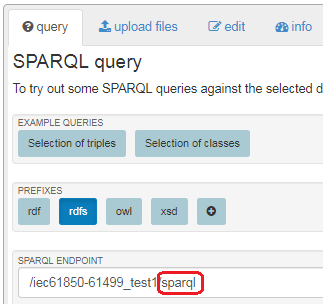
5) Используя Web-интерфейс браузера, создайте новый набор данных, придумав его имя (например, *iec61850-61499\_test1*).



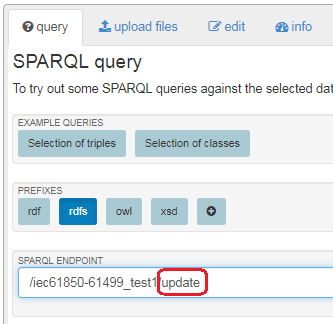
6) Загрузите ранее созданную RDF-онтологию (файл с расширением .owl) в созданный набор данных.



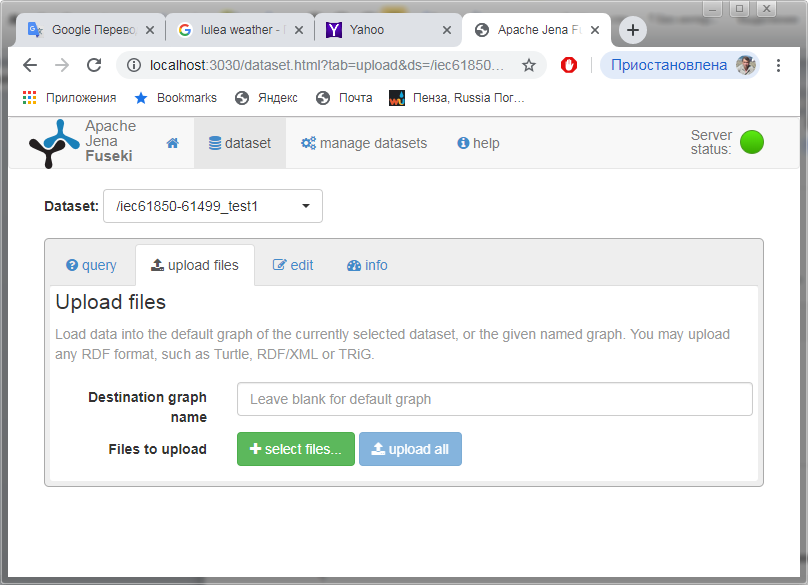
7) Сейчас система готова выполнять ваши SPQRQL-запросы. Чтобы выполнить SPARQL-запрос на выборку данных, установите режим “sparql”:



Чтобы выполнить SPARQL-запрос на изменение данных, установите режим “update”:



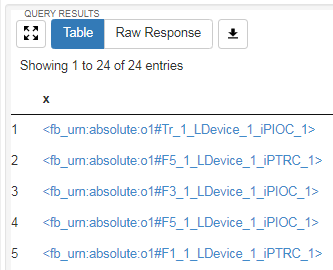
Измененные данные можно выгрузить в файл.



8) Пример запроса на выборку данных:



Результат выполнения запроса:



9) Пример запроса на изменение данных:

