

Laboratorio 2 – SPARQL

Objetivos:

1. Entender el lenguaje y protocolo de consulta SPARQL.
2. Ser capaz de hacer consultas SPARQL mediante un formulario Web, mediante una petición HTTP, y mediante consultas federadas.

Recursos necesarios:

- GraphDB en Ubuntu (Usuario/password: Isi/Isi) o servidor Google Cloud.

1.- Introducción

SPARQL (SPARQL Protocol and RDF Query Language) es el lenguaje de consulta de RDF. Incluye el lenguaje mismo y un protocolo para ejecutarlo mediante HTTP, permitiendo la federación de cualquier servicio que implemente SPARQL, aumentando su interoperabilidad con respecto a otros paradigmas como SQL.

2.- SPARQL básico

Cargar archivo RDF **Museoak.rdf** en GraphDB, en el Named Graph <http://data.abd.ehu.eus/museoak>. Para ejecutar consultas en GraphDB, ir a la pestaña SPARQL:



Crea un archivo .rq para cada una de las siguientes consultas SPARQL:

- ¿Cuál sería la consulta SPARQL para obtener todos los triples de un grafo concreto? (**SPARQL-1.rq**)
- ¿Cuál sería la consulta SPARQL para saber qué museo tiene trabajadores cuyo nombre empieza por "Mi" y más de 800 trabajadores? (**SPARQL-2.rq**)

- ¿Cuál sería la consulta SPARQL para saber dónde se sitúa el museo en el que trabaja Aitor Labajo? (SPARQL-3.rq)

3.- Named Graphs en SPARQL

RDF, a diferencia de SQL, permite codificar los datos y los metadatos sobre esos datos usando el mismo lenguaje. Para añadir metadatos sobre el Named Graph de museos, ejecuta una consulta INSERT que añada el siguiente triple¹: `<http://data.abd.ehu.eus/museoak> dcat:theme "museos"`.

- ¿Cuál sería la consulta SPARQL para saber qué Named Graphs almacena GraphDB? (SPARQL-4.rq)
- ¿Cuál sería la consulta SPARQL para saber qué dataset trata sobre museos y que información contiene sobre Aitor Labajo? (SPARQL-5.rq)

4.- SPARQL mediante HTTP(S)

SPARQL, además de un lenguaje de consulta, es un protocolo para ejecutar consultas sobre HTTP (A diferencia de SQL). Crea un Shell Script (.sh) que contenga un línea de cURL que, mediante el método HTTP POST, ejecute una consulta SPARQL (La URL del endpoint SPARQL de GraphDB sigue la sintaxis `http://IP_HOST:PUERTO/repositorios/NOMBRE_REPOSITORIO`). (SPARQL.sh)

5.- Consultas federadas en SPARQL

SPARQL, a diferencia de SQL y al ser un estándar W3C, permite ejecutar consultas federadas uniendo diferentes SPARQL endpoints. Para conseguir la federación basta con referenciar la URL del SPARQL endpoint "externo" en la consulta, mediante el keyword **SERVICE**.

¿Cuál sería la consulta SPARQL para saber quién es el alcalde de la ciudad en la que se sitúa el museo Gugenheim, aunque esa información no esté en **Museoak.rdf**? Tendrás que ejecutar la consulta desde tu GraphDB local y añadir la URL del endpoint de DBpedia (<http://dbpedia.org/sparql>). Además, tendrás que asegurarte de que hay un enlace mediante el predicado **owl:sameAs** en tus datos, que apunte a los datos de DBpedia, como se pedía en el laboratorio de RDF. (SPARQL-6.rq)

Ahora vamos a combinar nuestro SPARQL endpoint local con nuestro SPARQL endpoint remoto (Google Cloud). En el GraphDB local, crea un repositorio que se llame **abd-local** y sube el archivo **ejemplo1-2.ttl** al grafo **urn:local**. En el GraphDB remoto, crea un repositorio que se llame **abd-remoto** y sube el archivo **ejemplo2-2.ttl** al grafo **urn:remoto**. ¿Cómo puedes combinar ambos SPARQL endpoints en una consulta ejecutada en el local, que solo pueda ser respondida mediante la combinación de los dos endpoints? (SPARQL-7.rq)

¿Cómo combinarías tu GraphDB local con otros dos GraphDB remotos de tus compañeros? (SPARQL-8.rq)

¹ PREFIX dcat: <<http://www.w3.org/ns/dcat#>>

6.- SPARQL en Jupyter Notebook

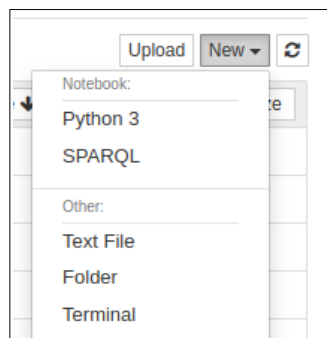
Jupyter Notebook es una herramienta muy potente para la ciencia de datos y machine learning, ya que permite ejecutar código (Kernels) de manera interactiva, reproducible y con documentación usable junto al código. También permite añadir visualizaciones interesantes. Un Jupyter Notebook esta típicamente compuesto por celdas de texto y de código, y las celdas de código pueden ser ejecutadas de manera interactiva, permitiendo la realización de un proceso complejo paso a paso siguiendo la documentación cómodamente.

- Instalar Jupyter Notebook (Requiere Python): **\$ pip install notebook**
- Instalar SPARQL Kernel: **\$ pip install sparqlkernel**
- Configurar SPARQL Kernel con usuario actual (Ej. Isi): **\$ jupyter sparqlkernel install --user \$USER**
- Ejecutar Jupyter Notebook en el path actual: **\$ jupyter notebook**

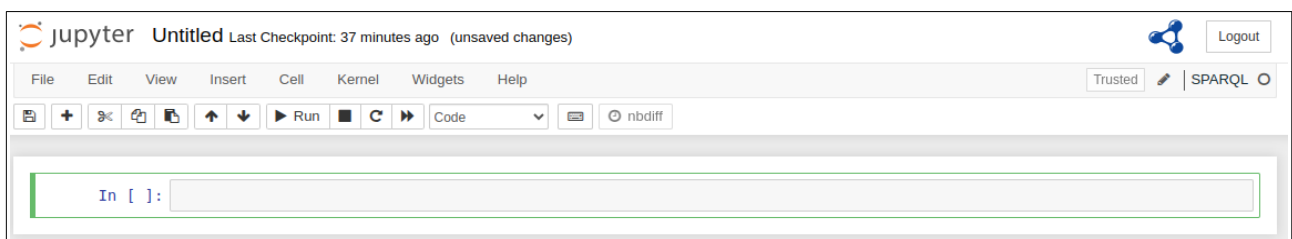
En la URL <http://localhost:8888/tree> debería aparecer esta web, con los directorios del path en el que se ha ejecutado Jupyter Notebook:



Pinchando en **New** debería aparecer la opción SPARQL:



Al seleccionar SPARQL aparece una pantalla nueva, que es un Kernel vacío:

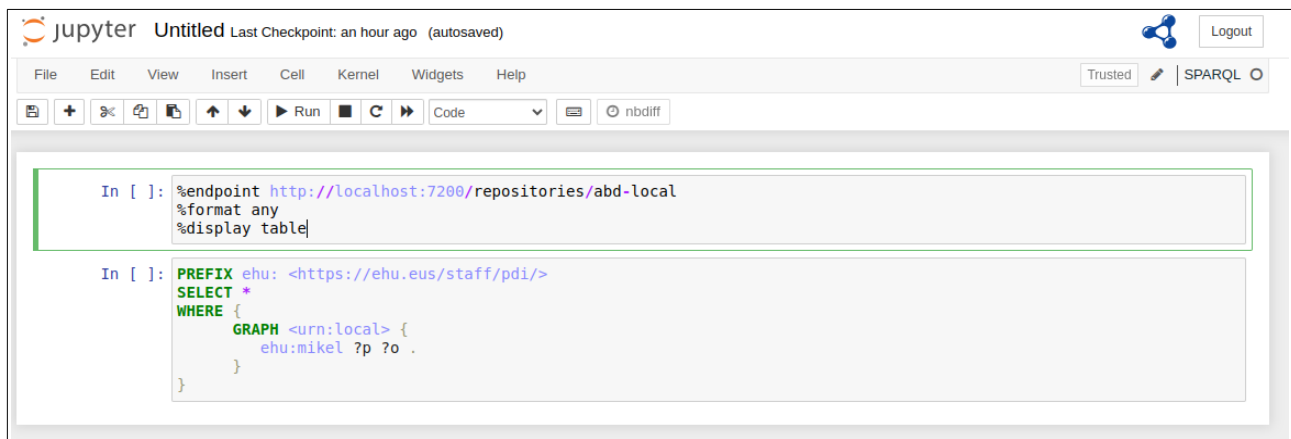


Administración de Bases de Datos 2024/2025

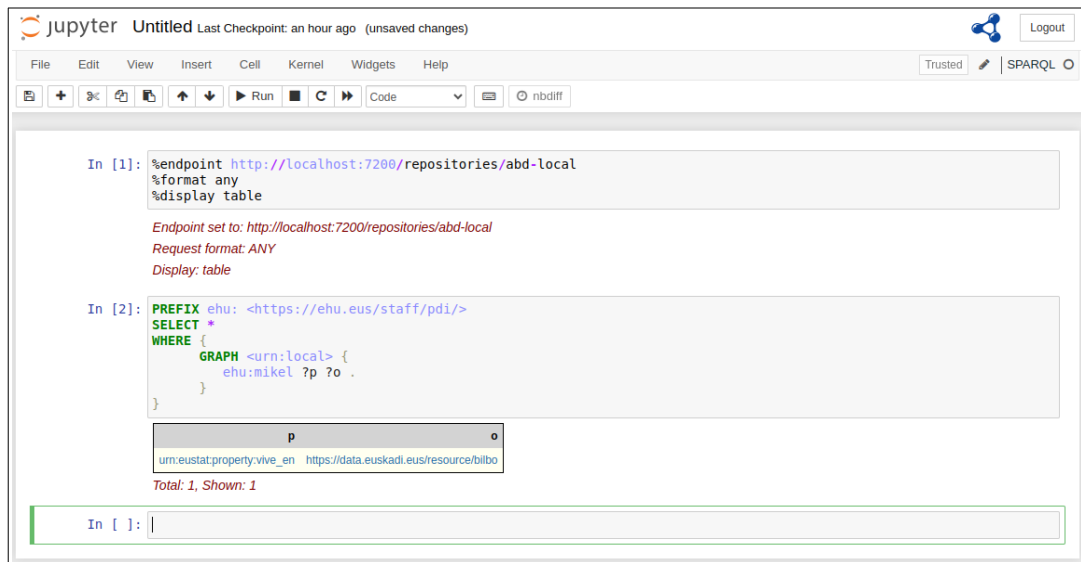
Para configurar el SPARQL endpoint tenemos que añadir una celda nueva con la siguiente información (la URL de tu GraphDB y el nombre del repositorio probablemente serán diferentes en tu caso):

```
%endpoint http://localhost:7200/repositories/abd-local
%format any
%display table
```

En la siguiente celda se puede poner la consulta SPARQL (**Insert; Cell Below**):



Para ejecutar seleccionar la primera celda y pinchar en **run**:



Reproduce el proceso con tus datos y tu GraphDB (Local o remoto) explorando diferentes opciones: <https://github.com/paulovn/sparql-kernel/blob/master/doc/magics.rst>

¿Como puedes configurar tu Jupyter Notebook para renderizar resultados en forma de Grafo?

Entrega:

Administración de Bases de Datos 2024/2025

- Archivos **.rq** con consultas SPARQL.
- Archivo **.sh** con ejecución CURL POST.
- Archivo Jupyter Notebook (**.ipynb**) con datos y servidor de estudiante, incluyendo visualización.