


Nombre: Irene**Apellidos: Francisco López****Ejercicio 1:** Habiendo descargado los ficheros .rdf que hay en ágora, realizar lo siguiente:

1. Convertir los tres RDF en .ttl.
2. Unificar los tres ficheros .ttl en uno único.
3. Importar el fichero .ttl en algún entorno donde poder realizar consultas SPARQL.

Para cada query en SPARQL, pegar el Código y la captura de pantalla con el resultado.

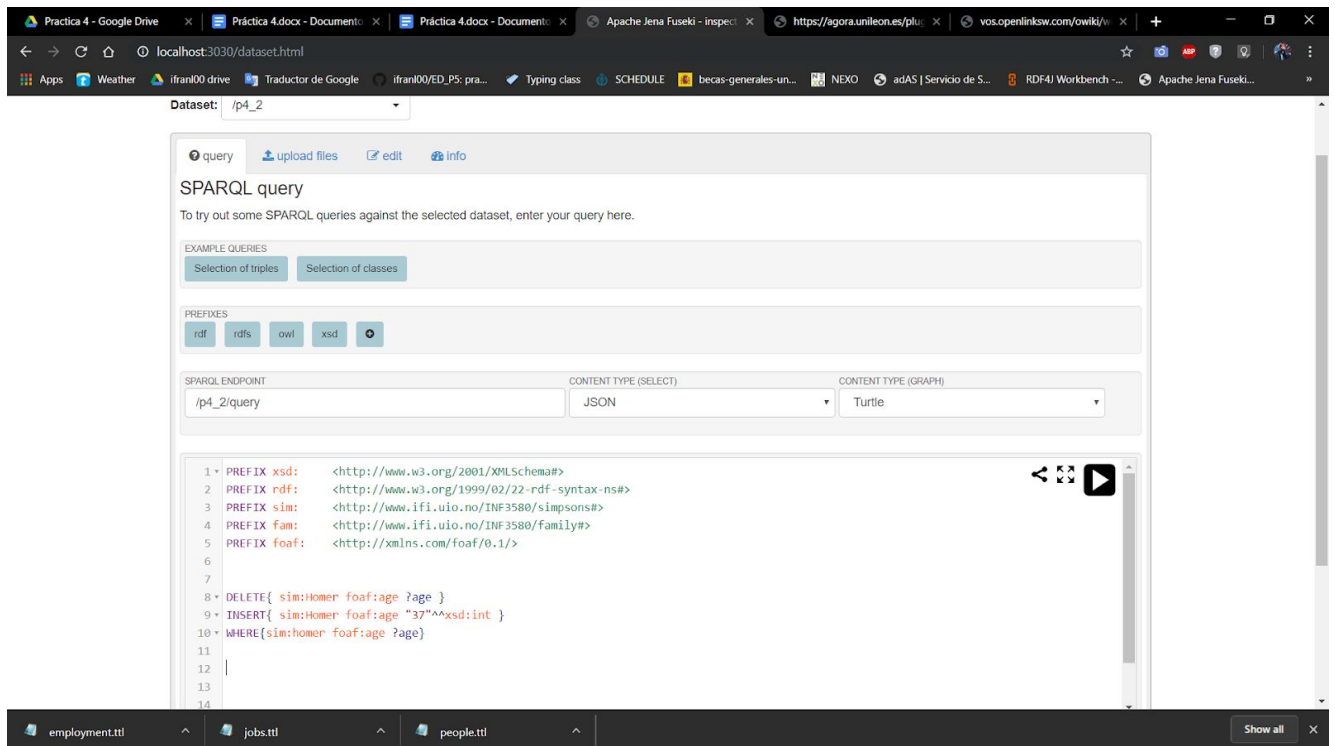
Mostrar los nombres completos (nombre + apellido) de todas las personas menores de 30 años que tienen un trabajo de investigación en orden ascendente de edad.																			
SPARQL query	Screenshot of the result set																		
<pre>PREFIX f: <http://example.org#> PREFIX e: <http://jyu.fi/employment#> PREFIX j: <http://jyu.fi/jobs#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> SELECT ?firstName ?surName ?age ?person ?job WHERE{ ?person f:firstName ?firstName. ?person f:surName ?surName. ?person xsd:age ?age. ?person e:worksAs ?job. ?job rdf:type j:ResearchJob FILTER (?age<30) } ORDER BY ?age</pre>	 <p>Showing 1 to 2 of 2 entries</p> <table><tr><th></th><th>firstName</th><th>surName</th><th>age</th><th>person</th><th>job</th></tr><tr><td>1</td><td>"Mary"</td><td>"Popens"</td><td>"24"^^xsd:integer</td><td>f:mary</td><td>j:seniorResearcher</td></tr><tr><td>2</td><td>"Jane"</td><td>"Joli"</td><td>"26"^^xsd:integer</td><td>f:jane</td><td>j:juniorResearcher</td></tr></table>		firstName	surName	age	person	job	1	"Mary"	"Popens"	"24"^^xsd:integer	f:mary	j:seniorResearcher	2	"Jane"	"Joli"	"26"^^xsd:integer	f:jane	j:juniorResearcher
	firstName	surName	age	person	job														
1	"Mary"	"Popens"	"24"^^xsd:integer	f:mary	j:seniorResearcher														
2	"Jane"	"Joli"	"26"^^xsd:integer	f:jane	j:juniorResearcher														

Mostrar a todas las personas que aman a un investigador senior.	
SPARQL query	Captura de pantalla del conjunto de resultados

<pre> PREFIX f: <http://example.org#> PREFIX e: <http://jyu.fi/employment#> PREFIX j: <http://jyu.fi/jobs#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> SELECT ?person ?lover WHERE{ ?person f:loves ?lover. ?lover e:worksAs j:seniorResearcher } </pre>	<div>Showing 1 to 1 of 1 entries</div> <div> <div>person</div> <div>f:john</div> </div> <div> <div>lover</div> <div>f:mary</div> </div>
--	---

Ejercicio 2: Haciendo uso de del fichero simpsons.ttl que hay en ágora, importar este al repositorio, probar que funciona bien ejecutando `SELECT * WHERE { ?s ?p ?o }` y, a continuación, realizar los siguientes ejercicios:

2.1. Modificar el conjunto de datos haciendo que Homer Simpsons tenga 37 años.



2.2. Borrar la tripla de edad si alguien tiene 24 años.

2.3. Borrar a aquellas personas que tengan menos de 10 años.

2.4. Borrar todas las triplas de los Simpsons.

2.5. Insertar la siguiente tripla: `<http://example.org/jabenitez> rdfs:label "Jabenitez"@es, "Jabenitez"@fr`.

Ejercicio 3: Haciendo uso de RDF4J o de Fuseki

Teniendo estos datos en TriG:

```

@prefix ex: <http://www.example.com/ont#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

{
    ex:repository ex:createdBy ex:repOwner; ex:title "Rep_1".
}
ex:books
{
    ex:book_1 a ex:Science; ex:size "100"; ex:title "Science book 1".
    ex:book_2 a ex:Science; ex:size "1000"; ex:title "Science book 2".
    ex:book_3 a ex:Fantasy; ex:size "100"; ex:title "Fantasy book 1".
}

```

Describe la forma en que lo hizo y cree/realice una consulta que muestre todos los triples con el ID de gráfico correspondiente.

Fuseki:

[escriba aquí la descripción](#)

SPARQL query

Captura de pantalla del conjunto de resultados

[pega aquí la consulta SPARQL](#)

[pega aquí la captura de pantalla](#)

Usando la consulta de SPARQL Update crear un nuevo grafo con ID:

`http://www.example.com/ont#scBooks` en el repositorio. Crear/ejecutar consulta que devuelve "Graph exists" en caso de que exista un gráfico con id (`http://www.example.com/ont#scBooks`) existiendo en el repositorio.

SPARQL Update query

[pega aquí la consulta SPARQL](#)

SPARQL query

Captura de pantalla del conjunto de resultados

[pega aquí la consulta SPARQL](#)

[pega aquí la captura de pantalla](#)

Usando la consulta de SPARQL Update mueva todos los libros científicos (libros que pertenecen a la sección `ex:Science` class) del grafo (`http://www.example.com/ont#books`) al gráfico recién creado (`http://www.example.com/ont#scBooks`). Muestra el contenido del repositorio (todos los triples con su correspondiente Graph ID.)

SPARQL Update query

[pega aquí la consulta SPARQL](#)

SPARQL query

Captura de pantalla del conjunto de resultados

[pega aquí la consulta SPARQL](#)

[pega aquí la captura de pantalla](#)