



# CSE 488: Ontologies and the Semantic Web

*Project Submission*

## Countries Ontology

*Submitted to:*

*Dr. Ensaf Hussein Mohamed*

*Eng. Dina Amr*

*Submitted By:*

*Yomna Hussien Mohamed - 18P5794*

*Sherif Ahmed Naiem - 18P6546*

*Reem Khaled Elsayed Aboushama - 18Q9822*

*Omar Mohamed Lotfy - 18P5606*

*Mohamed Sayed awwad - 18p7298*

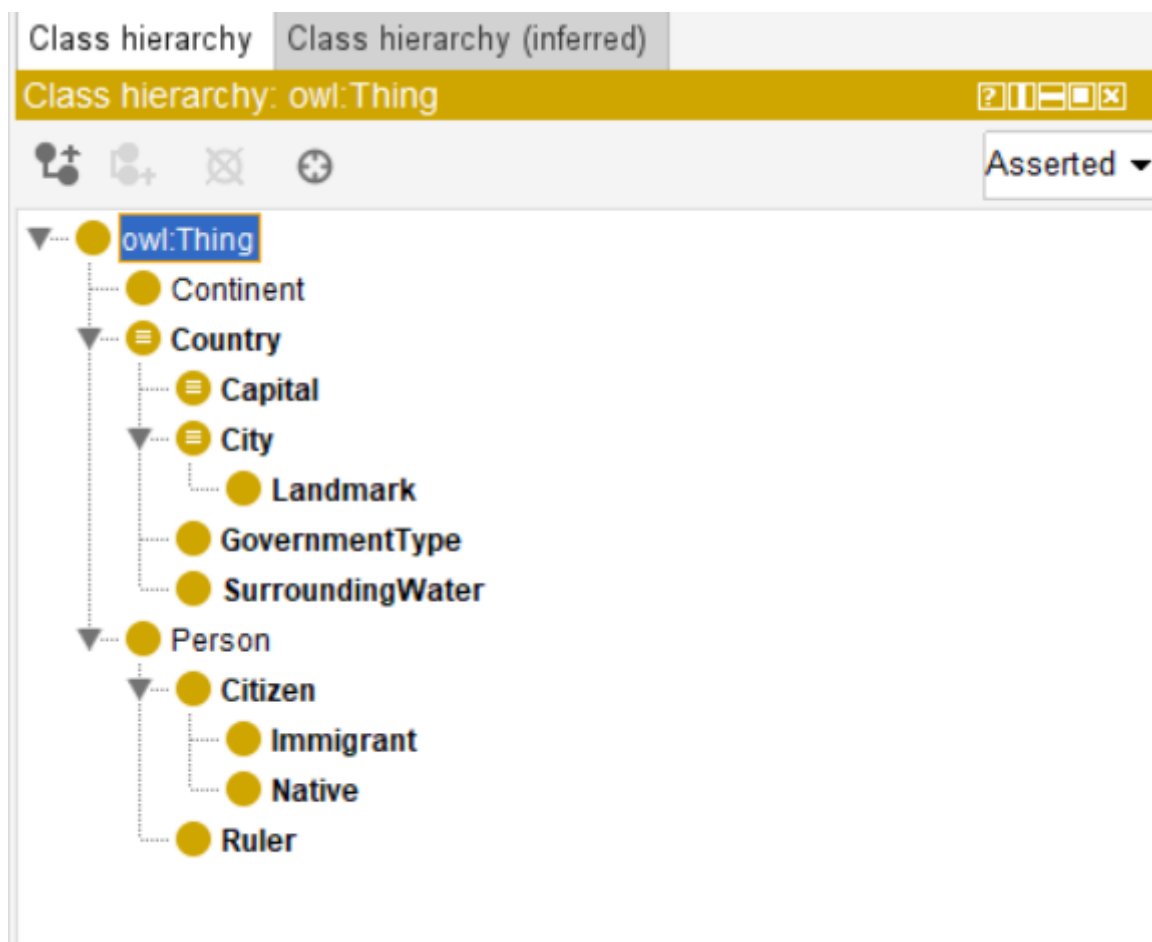
## Table of Contents

I. Ontology .....	3
Classes.....	3
Object properties .....	4
constraints .....	8
Data properties .....	9
Individuals.....	12
Visualization .....	13
Ontograph .....	13
II. PROTÉGÉ SPARQL .....	16
➤ Query 1: .....	16
➤ Query 2: .....	16
➤ Query 3: .....	17
➤ Query 4: .....	17
➤ Query 5: .....	18
➤ Query 6: .....	19
➤ Query 7: .....	20
➤ Query 8: .....	21
➤ Query 9: .....	21
III. JENA QUERIES .....	23
➤ Query 1: .....	23
➤ Query 2: .....	24
➤ Query 3: .....	25
➤ Query 4: .....	26
➤ Query 5: .....	26
➤ Query 6: .....	28
➤ Query 7: .....	29
➤ Query 8: .....	30

# I. Ontology

The countries ontology is designed to represent information about countries, cities, landmarks, rulers, government types, immigrants, and surrounding waters. The ontology defines various object properties such as "hasCity," "hasGovernmentType," "hasImmigrants," "hasLandmark," "hasNativeResidents," "hasRuler," "isCapitalOf," "isInContinent," and "isSurroundedBy" to establish relationships between entities. It also includes data properties like "address," "hasYearBuilt," "rulertype," and "watertype" to capture specific information about landmarks, rulers, and surrounding waters. The ontology further defines classes such as "Capital," "City," "Continent," "Country," "GovernmentType," "Immigrant," "Landmark," "Native," "Person," and "Ruler" to classify different types of entities. By utilizing this ontology, you can organize and query data related to countries, their attributes, and relationships in a structured and semantic manner.

## Classes



## Object properties

- [hasCity](#)
- [hasGovernmentType](#)
- [hasImmigrants](#)
- [hasLandmark](#)
- [hasNativeResidents](#)
- [hasRuler](#)
- [isCapitalOf](#)
- [isInContinent](#)
- [isSurroundedBy](#)

---

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

### Object Property: hasCity

#### Domains (1)

- Country

#### Ranges (1)

- City

#### Usage (7)

- Australia **hasCity** Sydney
- China **hasCity** Shanghai
- Egypt **hasCity** Giza
- France **hasCity** Marseille
- Germany **hasCity** Hamburg
- India **hasCity** Mumbai
- United\_States **hasCity** New\_York

---

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

### Object Property: hasGovernmentType

#### Domains (1)

- Country

#### Ranges (1)

- GovernmentType

#### Usage (8)

- Country **= hasGovernmentType exactly** 1 GovernmentType
- Australia **hasGovernmentType** Federal\_Parliamentary\_Constitutional\_Monarchy
- China **hasGovernmentType** Communist\_Party-led\_State
- Egypt **hasGovernmentType** Republic
- France **hasGovernmentType** Semi-Presidential\_Republic
- Germany **hasGovernmentType** Federal\_Republic
- India **hasGovernmentType** Federal\_Parliamentary\_Democratic\_Republic
- United\_States **hasGovernmentType** Democracy

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: hasImmigrants

### Domains (1)

- Country

### Ranges (1)

- Immigrant

### Usage (7)

- Australia **hasImmigrants** 300
- China **hasImmigrants** 450
- Egypt **hasImmigrants** 700
- France **hasImmigrants** 250
- Germany **hasImmigrants** 102
- India **hasImmigrants** 450
- United\_States **hasImmigrants** 700

- 
- X

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: hasLandmark

### Domains (1)

- City

### Ranges (1)

- Landmark

### Usage (8)

- City **hasLandmark** max 3 Landmark
- Giza **hasLandmark** The\_Great\_Pyramids
- Hamburg **hasLandmark** Brandenburg\_Gate
- Marseille **hasLandmark** Eiffel\_Tower
- Mumbai **hasLandmark** Taj\_Mahal
- New\_York **hasLandmark** Statue\_of\_Liberty
- Shanghai **hasLandmark** Great\_Wall\_of\_China
- Sydney **hasLandmark** Sydney\_Opera\_House

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: hasNativeResidents

### Domains (1)

- Country

### Ranges (1)

- Native

### Usage (7)

- Australia **hasNativeResidents** 600000
- China **hasNativeResidents** 900000
- Egypt **hasNativeResidents** 5462700
- France **hasNativeResidents** 400000
- Germany **hasNativeResidents** 10000000
- India **hasNativeResidents** 670000
- United\_States **hasNativeResidents** 5000000

•

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: hasRuler

### Domains (1)

- Country

### Ranges (1)

- Ruler

### Usage (7)

- Country  $\equiv$  **hasRuler** exactly 1 Ruler
- Australia **hasRuler** Elizabeth\_II
- China **hasRuler** Xi\_Jinping
- Egypt **hasRuler** Abdel\_Fattah\_el-Sisi
- Germany **hasRuler** Angela\_Merkel
- India **hasRuler** Ram\_Nath\_Kovind
- United\_States **hasRuler** Joe\_Biden

•

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: isCapitalOf

### Domains (1)

- Capital

### Ranges (1)

- Country

### Usage (8)

- Capital  $\equiv$  isCapitalOf exactly 1 Country
- Beijing isCapitalOf China
- Berlin isCapitalOf Germany
- Cairo isCapitalOf Egypt
- Canberra isCapitalOf Australia
- New\_Delhi isCapitalOf India
- Paris isCapitalOf France
- Washington\_DC isCapitalOf United\_States

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: isInContinent

### Domains (1)

- Country

### Usage (8)

- Country  $\equiv$  isInContinent exactly 1 Continent
- Australia isInContinent Oceania
- China isInContinent Asia
- Egypt isInContinent Africa
- France isInContinent Europe
- Germany isInContinent Europe
- India isInContinent Asia
- United\_States isInContinent North\_America

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

## Object Property: isSurroundedBy

### Domains (1)

- Country

### Ranges (1)

- SurroundingWater

### Usage (8)

- Country  $\equiv$  isSurroundedBy max 4 SurroundingWater
- Australia isSurroundedBy Pacific\_Ocean
- China isSurroundedBy Yellow\_Sea
- Egypt isSurroundedBy Red\_Sea
- France isSurroundedBy Mediterranean\_Sea
- Germany isSurroundedBy North\_Sea
- India isSurroundedBy Arabian\_Sea
- United\_States isSurroundedBy Atlantic\_Ocean

constraints

each country has exactly 1 ruler

each city has max landmarks of 3

each country is in exactly 1 continent

each capital is capital of exactly 1 country

each country is surrounded max by 4 water bodies

each country has exactly 1 gov type



## Data properties

---

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals  
Datatypes Clouds

### Data Property: address

#### Domains (1)

- Landmark

#### Ranges (1)

- xsd:string

#### Usage (7)

- Brandenburg\_Gate **address** "Brandenburger Tor  
Pariser Platz  
10117 Berlin  
Germany"(xsd:string)
- Eiffel\_Tower **address** "Champ de Mars, 5 Avenue Anatole France, 75007 Paris,  
France"(xsd:string)
- Great\_Wall\_of\_China **address** "Huairou, China"(xsd:string)
- Statue\_of\_Liberty **address** "Liberty Island, New York, NY 10004, United States"  
(xsd:string)
- Sydney\_Opera\_House **address** "Bennelong Point, Sydney NSW 2000, Australia"  
(xsd:string)
- Taj\_Mahal **address** "Dharmapuri, Forest Colony  
Tajganj, Agra, Uttar Pradesh 282001"(xsd:string)
- The\_Great\_Pyramids **address** "Al Haram, Nazlet El-Semman  
Giza Governorate  
Egypt"(xsd:string)

OWL HTML inside

## Data Property: hasYearBuilt

### Domains (1)

- Landmark

### Ranges (1)

- xsd:string

### Usage (6)

- Brandenburg\_Gate **hasYearBuilt** "1791 CE"(xsd:string)
- Eiffel\_Tower **hasYearBuilt** "1889 CE"(xsd:string)
- Great\_Wall\_of\_China **hasYearBuilt** "7th century BCE"(xsd:string)
- Sydney\_Opera\_House **hasYearBuilt** "1973 CE"(xsd:string)
- Taj\_Mahal **hasYearBuilt** "1653 CE"(xsd:string)
- The\_Great\_Pyramids **hasYearBuilt** "2560 BCE"(xsd:string)

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals  
Datatypes Clouds

## Data Property: rulertype

### Domains (1)

- Ruler

### Ranges (1)

- xsd:string

### Usage (6)

- Abdel\_Fattah\_el-Sisi **rulertype** "President"(xsd:string)
- Angela\_Merkel **rulertype** "Chancellor"(xsd:string)
- Emmanuel\_Macron **rulertype** "President"(xsd:string)
- Joe\_Biden **rulertype** "President"(xsd:string)
- Ram\_Nath\_Kovind **rulertype** "President"(xsd:string)
- Xi\_Jinping **rulertype** "President"(xsd:string)

OWL HTML inside

Ontologies Classes Object Properties [Data Properties](#) Annotation Properties Individuals  
Datatypes Clouds

## Data Property: watertype

### Domains (1)

- SurroundingWater

### Ranges (1)

- xsd:string

### Usage (7)

- Arabian\_Sea **watertype** "sea"(xsd:string)
- Atlantic\_Ocean **watertype** "ocean"(xsd:string)
- Mediterranean\_Sea **watertype** "sea"(xsd:string)
- North\_Sea **watertype** "sea"(xsd:string)
- Pacific\_Ocean **watertype** "ocean"(xsd:string)
- Red\_Sea **watertype** "sea"(xsd:string)
- Yellow\_Sea **watertype** "sea"(xsd:string)

OWL HTML inside

# Individuals

countriesOnto (http://www.semanticweb.org/reemk/ontologies/countriesOnto) : [D:\semester 10\Ontologies\OntologiesProject\CountriesOnto\countriesOnto.owl]

File Edit View Reasoner Tools Refactor Window Help

countriesOnto (http://www.semanticweb.org/reemk/ontologies/countriesOnto)

Active ontology: countriesOnto

Individuals: Sydney\_Opera\_House

Annotations Usage

Usage: Sydney\_Opera\_House

Show: this different

Found 10 uses of Sydney\_Opera\_House

- Sydney
- Sydney hasLandmark Sydney\_Opera\_House

Sydney\_Opera\_House

- Sydney\_Opera\_House hasYearBuilt "1973 CE"
- Sydney\_Opera\_House Type Landmark
- Sydney\_Opera\_House address "Bennelong Point, Sydney NSW 2000, Australia"

Description: Sydney\_Opera\_House

Types

- Landmark

Same Individual As

Different Individuals

Property assertions: Sydney\_Opera\_House

Object property assertions

Data property assertions

- hasYearBuilt "1973 CE"
- address "Bennelong Point, Sydney NSW 2000, Australia"

Negative object property assertions

Negative data property assertions

Reasoner active Show Inferences

countriesOnto (http://www.semanticweb.org/reemk/ontologies/countriesOnto) : [D:\semester 10\Ontologies\OntologiesProject\CountriesOnto\countriesOnto.owl]

File Edit View Reasoner Tools Refactor Window Help

countriesOnto (http://www.semanticweb.org/reemk/ontologies/countriesOnto)

Active ontology: countriesOnto

Individuals: China

Annotations Usage

Usage: China

Show: this different

Found 20 uses of China

- Beijing
- Beijing isCapitalOf China

China

- China hasImmigrants 450
- China isinContinent Asia
- China isSurroundedBy Yellow\_Sea
- China hasGovernmentType Communist\_Party-led\_State
- China hasRuler Xi\_Jinping
- China hasCity Shanghai
- China Type Country
- Individual: China
- China hasNativeResidents 900000

Description: China

Types

- Country
- City

Same Individual As

Different Individuals

Property assertions: China

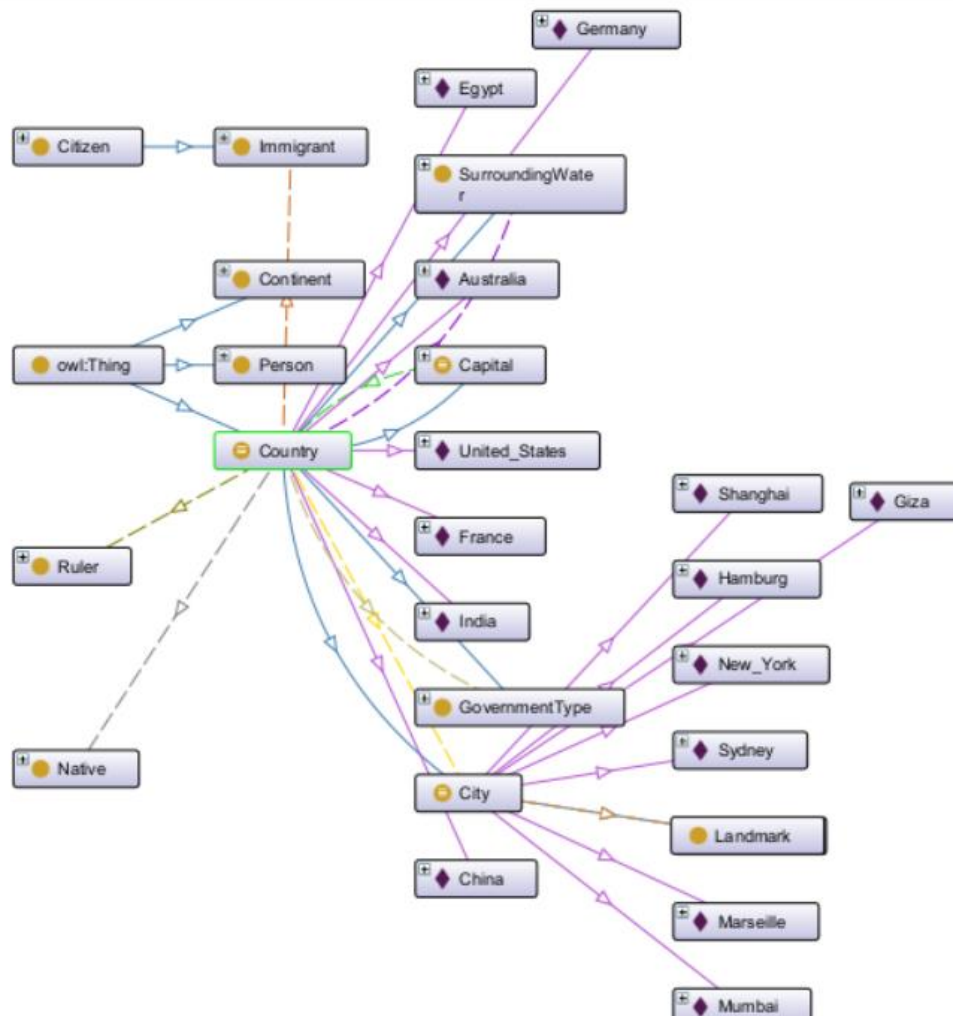
Object property assertions

- hasImmigrants 450
- isinContinent Asia
- isSurroundedBy Yellow\_Sea
- hasGovernmentType Communist\_Party-led\_State
- hasRuler Xi\_Jinping
- hasCity Shanghai
- hasNativeResidents 900000

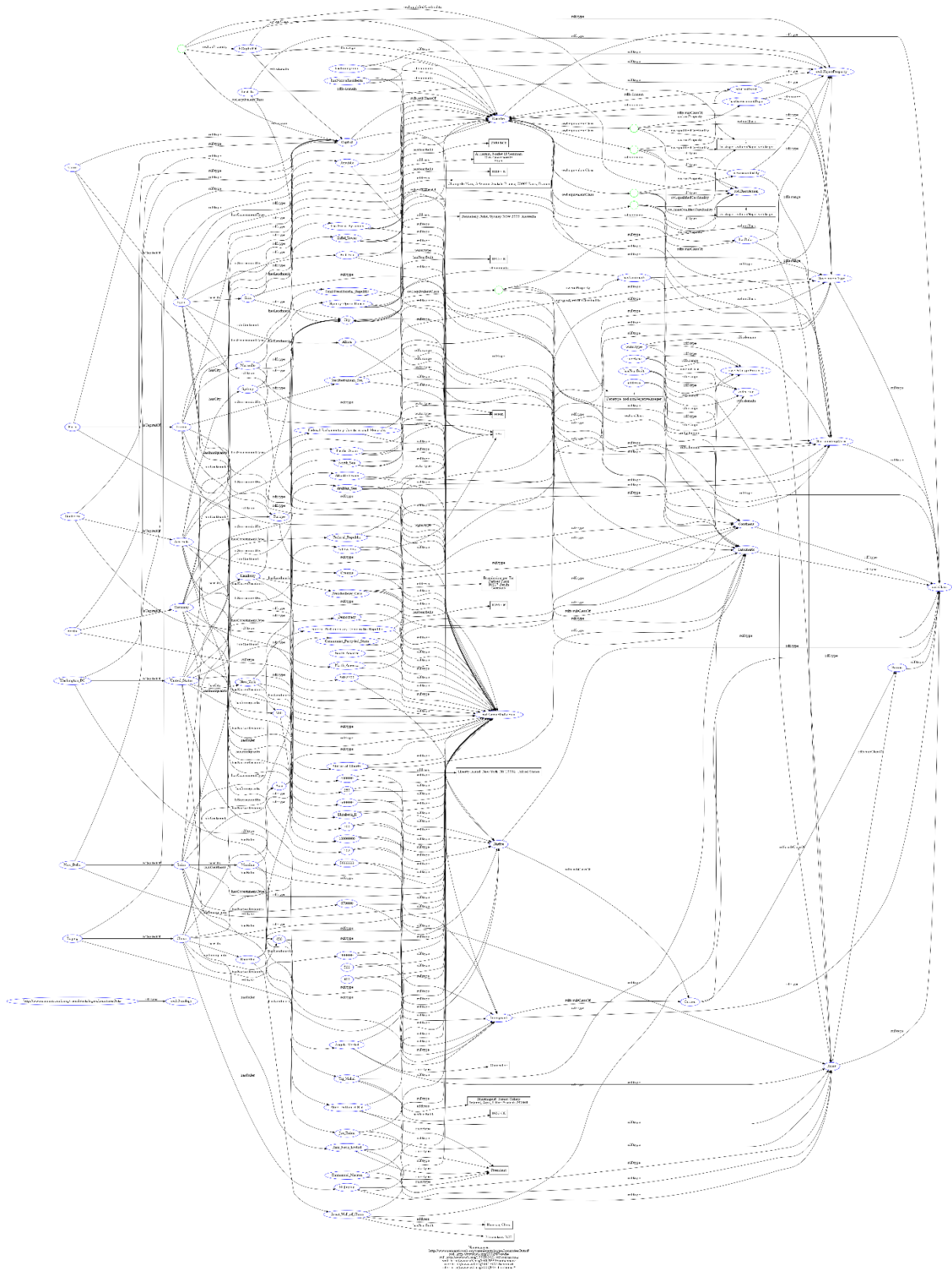
Data property assertions

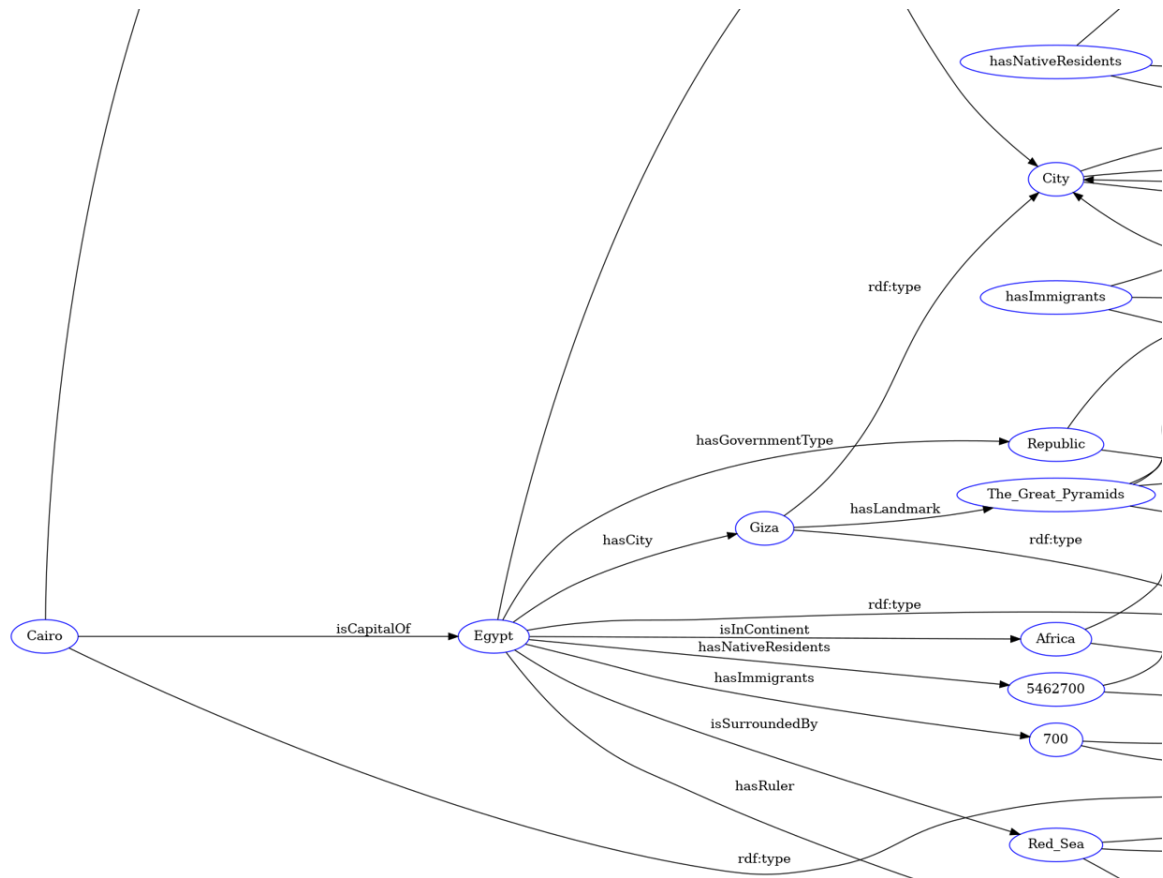
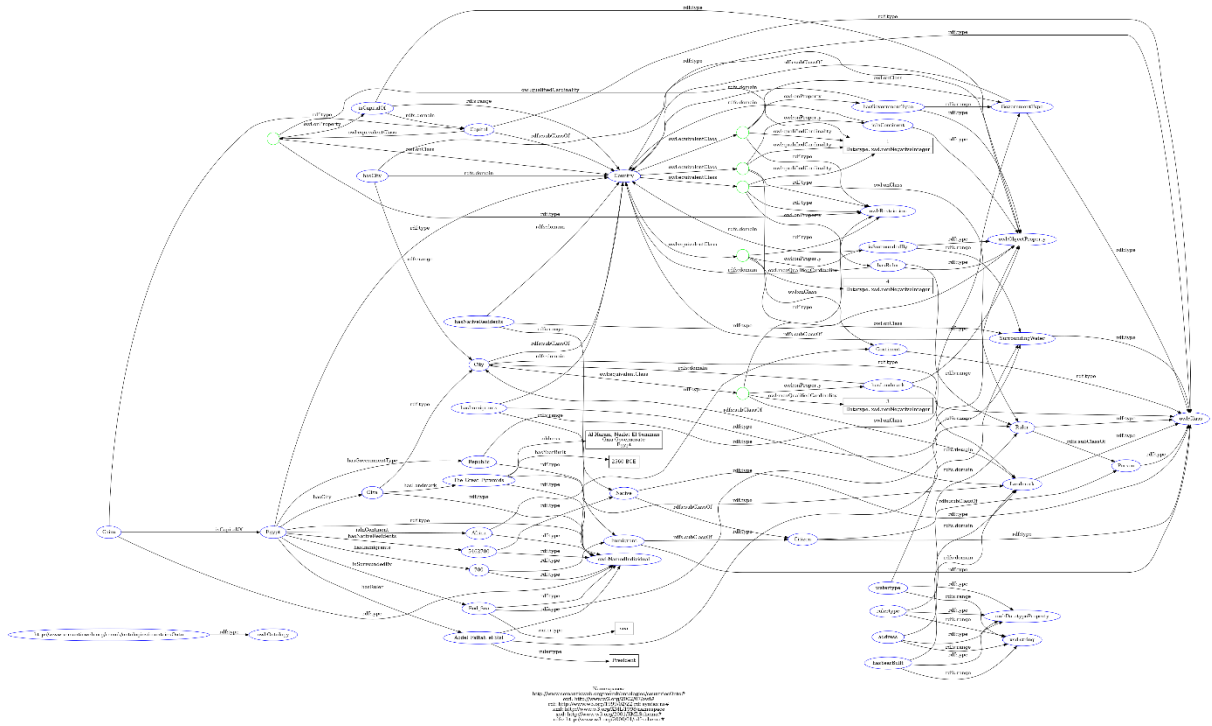
Reasoner active Show Inferences

## Ontograph



## RDF graph





## II. PROTÉGÉ SPARQL

### ➤ Query 1:

List all countries name, their capitals, government type, current ruler name, and OPTIONAL the surrounding water

#### • In Protégé SPARQL

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

SELECT ?country ?capital ?governmentType ?ruler ?water
WHERE {
    ?country uri:isInContinent ?continent.
    ?capital uri:isCapitalOf ?country.
    ?country uri:hasGovernmentType ?governmentType.
    ?country uri:hasRuler ?ruler.
    OPTIONAL { ?country uri:isSurroundedBy ?water. }
} ORDER BY ?country
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>
```

```
SELECT ?country ?capital ?governmentType ?ruler ?water
WHERE {
    ?country uri:isInContinent ?continent.
    ?capital uri:isCapitalOf ?country.
    ?country uri:hasGovernmentType ?governmentType.
    ?country uri:hasRuler ?ruler.
    OPTIONAL { ?country uri:isSurroundedBy ?water. }
}
```

} ORDER BY ?country

Execute

?country	?capital	?governmentType	?ruler	?water
uri:Australia	uri:Canberra	uri:Federal_Parliamentary_Constitutional_Monarchy	uri:Elizabeth_II	
uri:China	uri:Beijing	uri:Communist_Party-led_State	uri:Xi_Jinping	
uri:Egypt	uri:Cairo	uri:Republic	uri:Abdel_Fattah_el-Sisi	uri:Mediterranean_Sea
uri:Egypt	uri:Cairo	uri:Republic	uri:Abdel_Fattah_el-Sisi	uri:Red_Sea
uri:France	uri:Paris	uri:Semi-Presidential_Republic	uri:Emmanuel_Macron	uri:Mediterranean_Sea
uri:France	uri:Paris	uri:Semi-Presidential_Republic	uri:Emmanuel_Macron	uri:English_Channel
uri:France	uri:Paris	uri:Semi-Presidential_Republic	uri:Emmanuel_Macron	uri:Bay_of_Biscay
uri:Germany	uri:Berlin	uri:Federal_Republic	uri:Angela_Merkel	uri:Baltic_Sea
uri:Germany	uri:Berlin	uri:Federal_Republic	uri:Angela_Merkel	uri:North_Sea
uri:India	uri:New_Delhi	uri:Federal_Parliamentary_Democratic_Republic	uri:Ram_Nath_Kovind	uri:Arabian_Sea
uri:United_States	uri:Washington_DC	uri:Democracy	uri:Joe_Biden	uri:Atlantic_Ocean

### ➤ Query 2:

#### • In Protégé SPARQL

Find the landmarks that were built at year > 1700 and show which cities and countries they belong to ordered by the year build in descending order.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>
SELECT ?landmarks ?city ?country ?year
WHERE {
    ?country uri:isInContinent ?continent.
    ?country uri:hasCity ?city.
    ?city uri:hasLandmark ?landmarks.
}
```



```

?landmarks uri:hasYearBuilt ?year
FILTER(?year > 1700).

```

```

}ORDER BY DESC (?year)

```

Snap SPARQL Query

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>
SELECT ?landmarks ?city ?country ?year
WHERE {
    ?country uri:isInContinent ?continent.
    ?country uri:hasCity ?city.
    ?city uri:hasLandmark ?landmarks.
    ?landmarks uri:hasYearBuilt ?year
    FILTER(?year > 1700).
}

```

}ORDER BY DESC (?year)

Execute

?landmarks	?city	?country	?year
uri:national_museum_of_egypt	uri:Giza	uri:Egypt	1990
uri:Sydney_Opera_House	uri:Sydney	uri:Australia	1973
uri:oriental_pearl_lower	uri:Shanghai	uri:China	1900
uri:Eiffel_Tower	uri:Marseille	uri:France	1889
uri:Statue_of_Liberty	uri:New_York	uri:United_States	1876
uri:Brandenburg_Gate	uri:Hamburg	uri:Germany	1791

### ➤ Query 3:

List all counties name and their surrounding water that has surrounding water that has type sea

- In Protégé SPARQL

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

```

```

SELECT ?country ?water
WHERE {
    ?country uri:isInContinent ?continent.
    ?country uri:isSurroundedBy ?water.
    ?water uri:watertype "sea".
}

```

```

} ORDER BY ?country

```

Snap SPARQL Query

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

```

```

SELECT ?country ?water
WHERE {
    ?country uri:isInContinent ?continent.
    ?country uri:isSurroundedBy ?water.
    ?water uri:watertype "sea".
}

```

} ORDER BY ?country

Execute

?country	?water
uri:Egypt	uri:Mediterranean_Sea
uri:Egypt	uri:Red_Sea
uri:France	uri:Mediterranean_Sea
uri:Germany	uri:Baltic_Sea
uri:Germany	uri:North_Sea
uri:India	uri:Arabian_Sea

### ➤ Query 4:

List all countries and their capitals that don't have surrounding water ordered by capital name.

- In Protégé SPARQL

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

```

```

SELECT ?country ?capital
WHERE {
    ?country uri:isInContinent ?continent.
    ?capital uri:isCapitalOf ?country.
    MINUS {?country uri:isSurroundedBy ?water.}
} ORDER BY ?capital

```

Snap SPARQL Query

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

SELECT ?country ?capital
WHERE {
    ?country uri:isInContinent ?continent.
    ?capital uri:isCapitalOf ?country.
    MINUS {?country uri:isSurroundedBy ?water.}
} ORDER BY ?capital

```

Execute

?country	?capital
uri:China	uri:Beijing
uri:Australia	uri:Canberra

### ➤ Query 5:

List cities that have “a” in their name and exist in Europe or the cities that ends with “ai” and exist in Asia (FILTER REGEX)

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>
SELECT ?city ?continent
WHERE {
    {?country uri:isInContinent ?continent.
    ?country uri:hasCity ?city.
    FILTER REGEX(str(?city), "ai$", "i").
    FILTER(?continent = uri:Asia).}

    UNION

    {?country uri:isInContinent ?continent.
    ?country uri:hasCity ?city.
    FILTER REGEX(str(?city), "a", "i").
    FILTER(?continent = uri:Europe).}
}

```

Snap SPARQL Query	
<pre> PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX uri: &lt;http://www.semanticweb.org/reemk/ontologies/countriesOnto#&gt; SELECT ?city ?continent WHERE {   {?country uri:isInContinent ?continent.    ?country uri:hasCity ?city.    FILTER REGEX(str(?city), "ai\$", "i").    FILTER(?continent = uri:Asia).}    UNION    {?country uri:isInContinent ?continent.    ?country uri:hasCity ?city.    FILTER REGEX(str(?city), "a", "i").    FILTER(?continent = uri:Europe).} }</pre>	
Execute	
?city	?continent
uri:Shanghai	uri:Asia
uri:Mumbai	uri:Asia
uri:Marseille	uri:Europe
uri:Hamburg	uri:Europe

### ➤ Query 6:

List countries, and their ruler that has government type democracy or republic or ruler is queen.

- In Protégé SPARQL

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

SELECT ?country ?ruler ?rulerType ?governmentType
WHERE {
  {?country uri:isInContinent ?continent.
   ?country uri:hasGovernmentType ?governmentType
   FILTER(?governmentType = uri:Republic || ?governmentType =
uri:Democracy).
   ?country uri:hasRuler ?ruler.
   ?ruler uri:rulertype ?rulerType. }

  UNION

  {?country uri:isInContinent ?continent.
   ?country uri:hasGovernmentType ?governmentType.
   ?country uri:hasRuler ?ruler.
   ?ruler uri:rulertype ?rulerType.
   ?ruler uri:rulertype "Queen".}

} ORDER BY ?country
```

Snap SPARQL Query				
<pre> PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX uri: &lt;http://www.semanticweb.org/reemk/ontologies/countriesOnto#&gt;  SELECT ?country ?ruler ?rulerType ?governmentType WHERE {   {?country uri:isInContinent ?continent.    ?country uri:hasGovernmentType ?governmentType    FILTER(?governmentType = uri:Republic    ?governmentType = uri:Democracy).    ?country uri:hasRuler ?ruler.    ?ruler uri:rulertype ?rulerType. }    UNION    {?country uri:isInContinent ?continent.    ?country uri:hasGovernmentType ?governmentType.    ?country uri:hasRuler ?ruler.    ?ruler uri:rulertype ?rulerType.    ?ruler uri:rulertype "Queen".}  } ORDER BY ?country </pre>				
Execute				
?country	?ruler	?rulerType	?governmentType	
uri:Australia	uri:Elizabeth_II	Queen	uri:Federal_Parliamentary_Constitutional_Monarchy	
uri:Egypt	uri:Abdel_Fattah_el-Sisi	President	uri:Republic	
uri:United_States	uri:Joe_Biden	President	uri:Democracy	

### ➤ Query 7:

List all cities and their landmarks that are in Europe or North America ordered by the city in ascending order

- In Protégé SPARQL

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

SELECT ?city ?landmarks
WHERE {
  {?country uri:isInContinent ?continent
   FILTER(?continent = uri:Europe).
   ?country uri:hasCity ?city.
   ?city uri:hasLandmark ?landmarks.}

  UNION

  {?country uri:isInContinent ?continent
   FILTER(?continent = uri:North_America).
   ?country uri:hasCity ?city.
   ?city uri:hasLandmark ?landmarks.}

} ORDER BY ?city

```

Snap SPARQL Query:	
<pre> PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX uri: &lt;http://www.semanticweb.org/reemk/ontologies/countriesOnto#&gt;  SELECT ?city ?landmarks WHERE {   {?country uri:isInContinent ?continent   FILTER(?continent = uri:Europe).   ?country uri:hasCity ?city.   ?city uri:hasLandmark ?landmarks.}    UNION    {?country uri:isInContinent ?continent   FILTER(?continent = uri:North_America).   ?country uri:hasCity ?city.   ?city uri:hasLandmark ?landmarks.}  } ORDER BY ?city </pre>	
Execute	
?city	?landmarks
uri:Hamburg	uri:Brandenburg_Gate
uri:Marseille	uri:Eiffel_Tower
uri:New_York	uri:times_square
uri:New_York	uri:Statue_of_Liberty

### ➤ Query 8:

List the countries, government type ,population, and their ruler that has ruler type president and population >10 million and order countries in descending order according to their population

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

```

```

SELECT ?country ?governmentType ?ruler ?population
WHERE {
  ?country uri:isInContinent ?continent.
  ?country uri:hasGovernmentType ?governmentType.
  ?country uri:hasPopulation ?population.
  ?country uri:hasRuler ?ruler.
  ?ruler uri:rulertype "President".
  FILTER(?population > 10000000).
} ORDER BY DESC (?population)

```

Snap SPARQL Query

PREFIX owl: <http://www.w3.org/2002/07/owl#>

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

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>

SELECT ?country ?governmentType ?ruler ?population

WHERE {

?country uri:isInContinent ?continent.

?country uri:hasGovernmentType ?governmentType.

?country uri:hasPopulation ?population.

?country uri:hasRuler ?ruler.

?ruler uri:ruletype "President".

FILTER(?population > 10000000).

}

ORDER BY DESC (?population)

Execute

?country	?governmentType	?ruler	?population
uri:China	uri:Communist_Party-led_State	uri:Xi_Jinping	1400000000
uri:India	uri:Federal_Parliamentary_Democratic_Republic	uri:Ram_Nath_Kovind	1408000000
uri:Egypt	uri:Republic	uri:Abdel_Fattah_el-Sisi	1000000000
uri:France	uri:Semi-Presidential_Republic	uri:Emmanuel_Macron	670000000
uri:United_States	uri:Democracy	uri:Joe_Biden	331900000

### ➤ Query 9:

Which countries has how many landmarks.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>
```

```
SELECT ?country (COUNT(?landmarks) AS ?landmarks_count)
WHERE {
    ?country uri:isInContinent ?continent.
    ?country uri:hasCity ?city.
    ?city uri:hasLandmark ?landmarks.
} GROUP BY ?country
ORDER BY DESC (?landmarks_count)
```

Snap SPARQL Query

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/reemk/ontologies/countriesOnto#>
```

```
SELECT ?country (COUNT(?landmarks) AS ?landmarks_count)
WHERE {
    ?country uri:isInContinent ?continent.
    ?country uri:hasCity ?city.
    ?city uri:hasLandmark ?landmarks.
}
```

```
} GROUP BY ?country
ORDER BY DESC (?landmarks_count)
```

Execute

?country		?landmarks_count
uri:Egypt	3	
uri:United_States	2	
uri:China	2	
uri:Australia	1	
uri:India	1	
uri:France	1	
uri:Germany	1	

### III. JENA QUERIES

#### ➤ Query 1:

List all countries' name, their capitals, current ruler name, and the surrounding water.

- In Jena SPARQL

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
```

```
SELECT (str(?countryName) as ?label0) (str(?capitalName) as ?label1)  
(str(?rulerName) as ?label2) (str(?waterName) as ?label3)
```

```
WHERE {  
    ?country uri:belongsTo ?continent.  
    ?country uri:hasCountryName ?countryName.  
    ?country uri:hasCapital ?capitalName.  
    ?country uri:hasRuler ?ruler.  
    ?ruler uri:hasRulerName ?rulerName.  
    ?country uri:surroundedBy ?water.  
    ?water uri:hasWaterName ?waterName.  
}  
OrderBY?countryName
```

#### SPARQL Query:

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
```

```
SELECT (str(?countryName) as ?label0) (str(?capitalName) as ?label1) (str(?rulerName) as ?label2) (str(?waterName) as ?label3)
```

```
WHERE {  
    ?country uri:belongsTo ?continent.  
    ?country uri:hasCountryName ?countryName.  
    ?country uri:hasCapital ?capitalName.  
    ?country uri:hasRuler ?ruler.  
    ?ruler uri:hasRulerName ?rulerName.  
    ?country uri:surroundedBy ?water.  
    ?water uri:hasWaterName ?waterName.  
}  
OrderBY?countryName
```

Submit

#### Query Result:

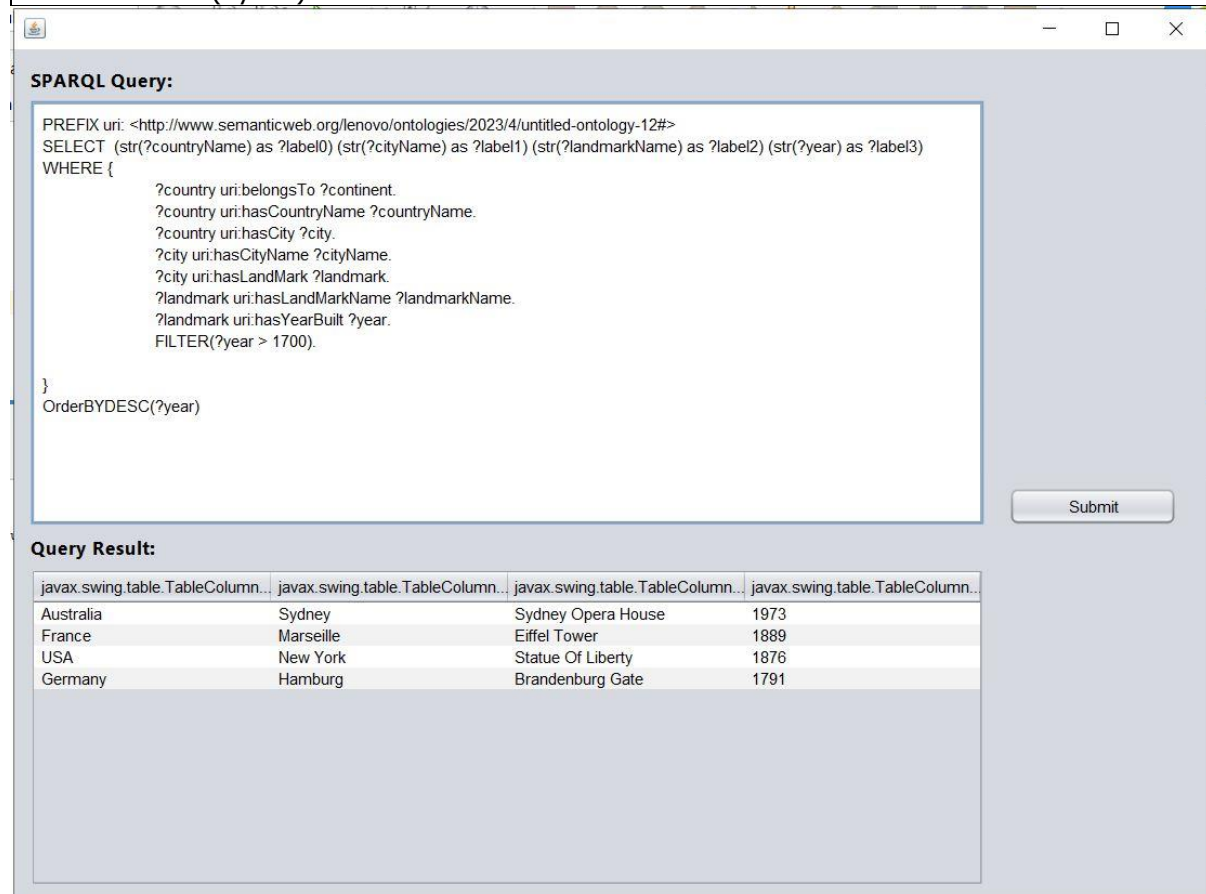
javax.swing.table.TableColumn...	javax.swing.table.TableColumn...	javax.swing.table.TableColumn...	javax.swing.table.TableColumn...
Australia	Canberra	Elizabeth II	Pacific Ocean
China	Beijing	Xi Jinping	Yellow Sea
Egypt	Cairo	Abel Fattah El Sisi	Mediterranean Sea
Egypt	Cairo	Abel Fattah El Sisi	Red Sea
France	Paris	Emmanuel Macron	Mediterranean Sea
Germany	Berlin	Angela Merkel	North Sea
India	New Delhi	Ram Nath Kovind	Arabian Sea

## ➤ Query 2:

- In Jena SPARQL

Find the landmarks that were built at year > 1700 and show which cities and countries they belong to ordered by the year build in descending order.

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
SELECT (str(?countryName) as ?label0) (str(?cityName) as ?label1)
(str(?landmarkName) as ?label2) (str(?year) as ?label3)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:hasCity ?city.
    ?city uri:hasCityName ?cityName.
    ?city uri:hasLandMark ?landmark.
    ?landmark uri:hasLandMarkName ?landmarkName.
    ?landmark uri:hasYearBuilt ?year.
    FILTER(?year > 1700).
}
OrderBYDESC(?year)
```



The screenshot shows a web application for executing SPARQL queries. It features a text area for the query, a 'Submit' button, and a 'Query Result' section displaying the results in a table.

**SPARQL Query:**

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
SELECT (str(?countryName) as ?label0) (str(?cityName) as ?label1) (str(?landmarkName) as ?label2) (str(?year) as ?label3)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:hasCity ?city.
    ?city uri:hasCityName ?cityName.
    ?city uri:hasLandMark ?landmark.
    ?landmark uri:hasLandMarkName ?landmarkName.
    ?landmark uri:hasYearBuilt ?year.
    FILTER(?year > 1700).
}
OrderBYDESC(?year)
```

**Submit**

**Query Result:**

javax.swing.table.TableColumn...	javax.swing.table.TableColumn...	javax.swing.table.TableColumn...	javax.swing.table.TableColumn...
Australia	Sydney	Sydney Opera House	1973
France	Marseille	Eiffel Tower	1889
USA	New York	Statue Of Liberty	1876
Germany	Hamburg	Brandenburg Gate	1791



### ➤ Query 3:

List all counties name and their surrounding water that has surrounding water that has type sea

- In Jena SPARQL

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
```

```
SELECT (str(?countryName) as ?label0) (str(?waterName) as ?label1)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:surroundedBy ?water.
    ?water uri:hasWaterName ?waterName.
    FILTER REGEX(?waterName,"sea$","i")
}
OrderBY?countryName
```

The screenshot shows a web application window titled "SPARQL Query:". It contains a text area with the SPARQL query and a "Submit" button. Below the query area, the "Query Result:" section displays a table with two columns. The first column contains country names, and the second column contains the names of the surrounding seas. The results are as follows:

Country	Surrounding Water
China	Yellow Sea
Egypt	Mediterranean Sea
Egypt	Red Sea
France	Mediterranean Sea
Germany	North Sea
India	Arabian Sea

#### ➤ Query 4:

List all countries and their capitals that don't have surrounding water type sea ordered by capital name and show the water name.

- In Jena SPARQL

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
SELECT (str(?countryName) as ?label0) (str(?capitalName) as ?label1)
(str(?waterName) as ?label2)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:hasCapital ?capitalName.
    ?country uri:surroundedBy ?water.
    ?water uri:hasWaterName ?waterName.
    MINUS {
        ?water uri:hasWaterName ?waterName.
        FILTER REGEX(?waterName,"sea$", "i").}
    }
    OrderBY?capitalName
```

The screenshot shows the Jena SPARQL Query interface. The query is pasted into the 'SPARQL Query:' field. Below the query field is a 'Submit' button. The 'Query Result:' section displays a table with three columns: 'Country Name', 'Capital Name', and 'Water Name'. The results are as follows:

Country Name	Capital Name	Water Name
Australia	Canberra	Pacific Ocean
USA	Washington DC	Atlantic Ocean

#### ➤ Query 5:

List cities that have “a” in their name and exist in Europe or the cities that ends with “ai” and exist in Asia (FILTER REGEX)

- In Jena SPARQL

PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>

SELECT (str(?cityName) as ?label0) (str(?continentName) as ?label1)

WHERE {

```
{?country uri:belongsTo ?continent.
?continent uri:hasContinentName ?continentName.
?country uri:hasCity ?city.
?city uri:hasCityName ?cityName
FILTER REGEX(?cityName,"ai$","i").
FILTER (?continentName = "Asia").}
```

UNION

```
{?country uri:belongsTo ?continent.
?continent uri:hasContinentName ?continentName.
?country uri:hasCity ?city.
?city uri:hasCityName ?cityName
FILTER REGEX(?cityName,"a","i").
FILTER (?continentName = "Europe").}
```

}

#### SPARQL Query:

```
SELECT (str(?cityName) as ?label0) (str(?continentName) as ?label1)
WHERE {
    {?country uri:belongsTo ?continent.
    ?continent uri:hasContinentName ?continentName.
    ?country uri:hasCity ?city.
    ?city uri:hasCityName ?cityName
    FILTER REGEX(?cityName,"ai$","i").
    FILTER (?continentName = "Asia").}

    UNION

    {?country uri:belongsTo ?continent.
    ?continent uri:hasContinentName ?continentName.
    ?country uri:hasCity ?city.
    ?city uri:hasCityName ?cityName
    FILTER REGEX(?cityName,"a","i").
    FILTER (?continentName = "Europe").}
}
```

Submit

#### Query Result:

javax.swing.table.TableColumn@9914c6b	javax.swing.table.TableColumn@4cbb0f13
Mumbai	Asia
Shanghai	Asia
Hamburg	Europe
Marseille	Europe

### ➤ Query 6:

List all cities and their landmarks that are in Europe or North America ordered by the city in ascending order

- In Jena SPARQL

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
SELECT (str(?cityName) as ?label0) (str(?landmarkName) as ?label1)
WHERE {
    {?country uri:belongsTo ?continent.
    FILTER(?continent = uri:Europe).
    ?country uri:hasCity ?city.
    ?city uri:hasCityName ?cityName.
    ?city uri:hasLandMark ?landmark.
    ?landmark uri:hasLandMarkName ?landmarkName.}
    UNION
    {?country uri:belongsTo ?continent.
    FILTER(?continent = uri:North_America).
    ?country uri:hasCity ?city.
    ?city uri:hasCityName ?cityName.
    ?city uri:hasLandMark ?landmark.
    ?landmark uri:hasLandMarkName ?landmarkName.}
}ORDERBY?cityName
```

The screenshot shows a web-based SPARQL query interface. The query is entered in a text area and is identical to the one in the previous block. A 'Submit' button is located to the right of the query area. Below the query area, the 'Query Result:' section displays a table with two columns. The first column contains city names, and the second column contains landmark names. The results are ordered by city name in ascending order.

City	Landmark
Hamburg	Brandenburg Gate
Marseille	Eiffel Tower
New York	Statue Of Liberty

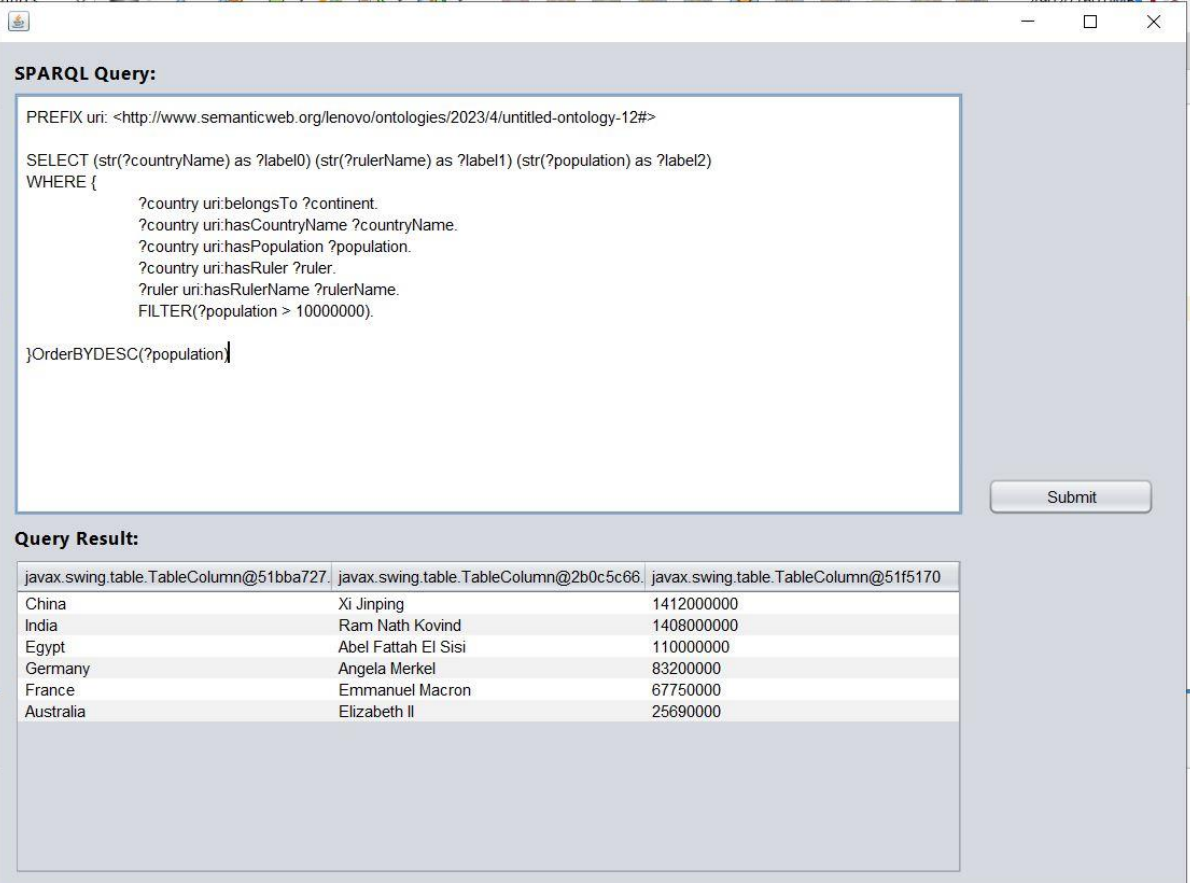
### ➤ Query 7:

List the countries, population, and their ruler and population >10 million and order countries in descending order according to their population.

- **In Jena SPARQL**

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>

SELECT (str(?countryName) as ?label0) (str(?rulerName) as ?label1) (str(?population) as ?label2)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:hasPopulation ?population.
    ?country uri:hasRuler ?ruler.
    ?ruler uri:hasRulerName ?rulerName.
    FILTER(?population > 10000000).
}OrderBYDESC(?population)
```



The screenshot shows a web-based SPARQL query interface. The query is entered in a text area and is identical to the one shown in the previous block. A 'Submit' button is located to the right of the query area. Below the query area, the 'Query Result:' section displays a table with three columns. The first column contains country names, the second column contains ruler names, and the third column contains population values. The results are ordered in descending order of population.

java.swing.table.TableColumn@51bba727	java.swing.table.TableColumn@2b0c5c66	java.swing.table.TableColumn@51f5170
China	Xi Jinping	1412000000
India	Ram Nath Kovind	1408000000
Egypt	Abel Fattah El Sisi	110000000
Germany	Angela Merkel	83200000
France	Emmanuel Macron	67750000
Australia	Elizabeth II	25690000

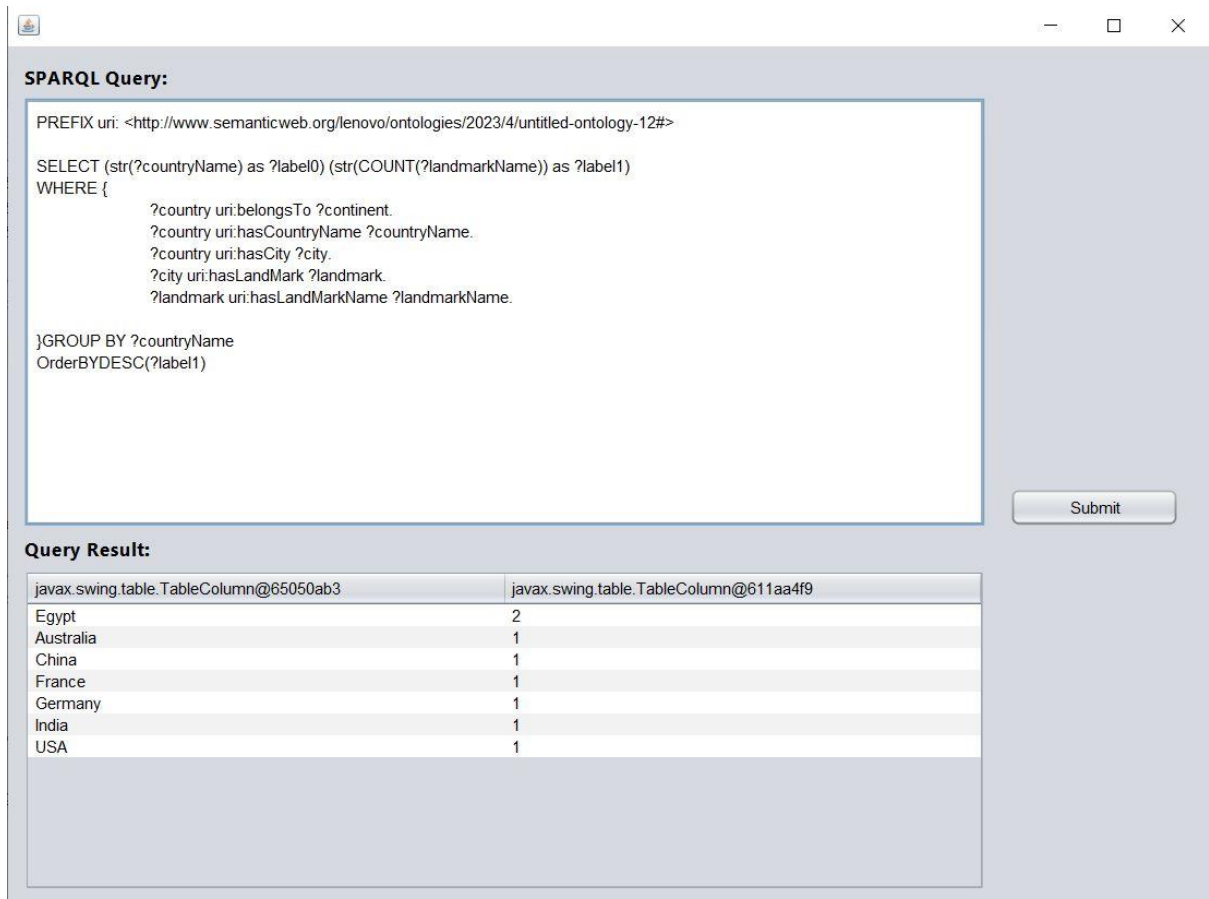
### ➤ Query 8:

Which countries has how many landmarks ordered in descending order of count.

- In Jena SPARQL

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>
```

```
SELECT (str(?countryName) as ?label0) (str(COUNT(?landmarkName)) as ?label1)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:hasCity ?city.
    ?city uri:hasLandMark ?landmark.
    ?landmark uri:hasLandMarkName ?landmarkName.
}GROUP BY ?countryName
OrderBYDESC(?label1)
```



The screenshot shows a web-based SPARQL query interface. The query is entered in a text area and is as follows:

```
PREFIX uri: <http://www.semanticweb.org/lenovo/ontologies/2023/4/untitled-ontology-12#>

SELECT (str(?countryName) as ?label0) (str(COUNT(?landmarkName)) as ?label1)
WHERE {
    ?country uri:belongsTo ?continent.
    ?country uri:hasCountryName ?countryName.
    ?country uri:hasCity ?city.
    ?city uri:hasLandMark ?landmark.
    ?landmark uri:hasLandMarkName ?landmarkName.
}GROUP BY ?countryName
OrderBYDESC(?label1)
```

Below the query area is a "Submit" button. The results are displayed in a table under the heading "Query Result:".

javax.swing.table.TableColumn@65050ab3	javax.swing.table.TableColumn@611aa4f9
Egypt	2
Australia	1
China	1
France	1
Germany	1
India	1
USA	1