

Country Ontology

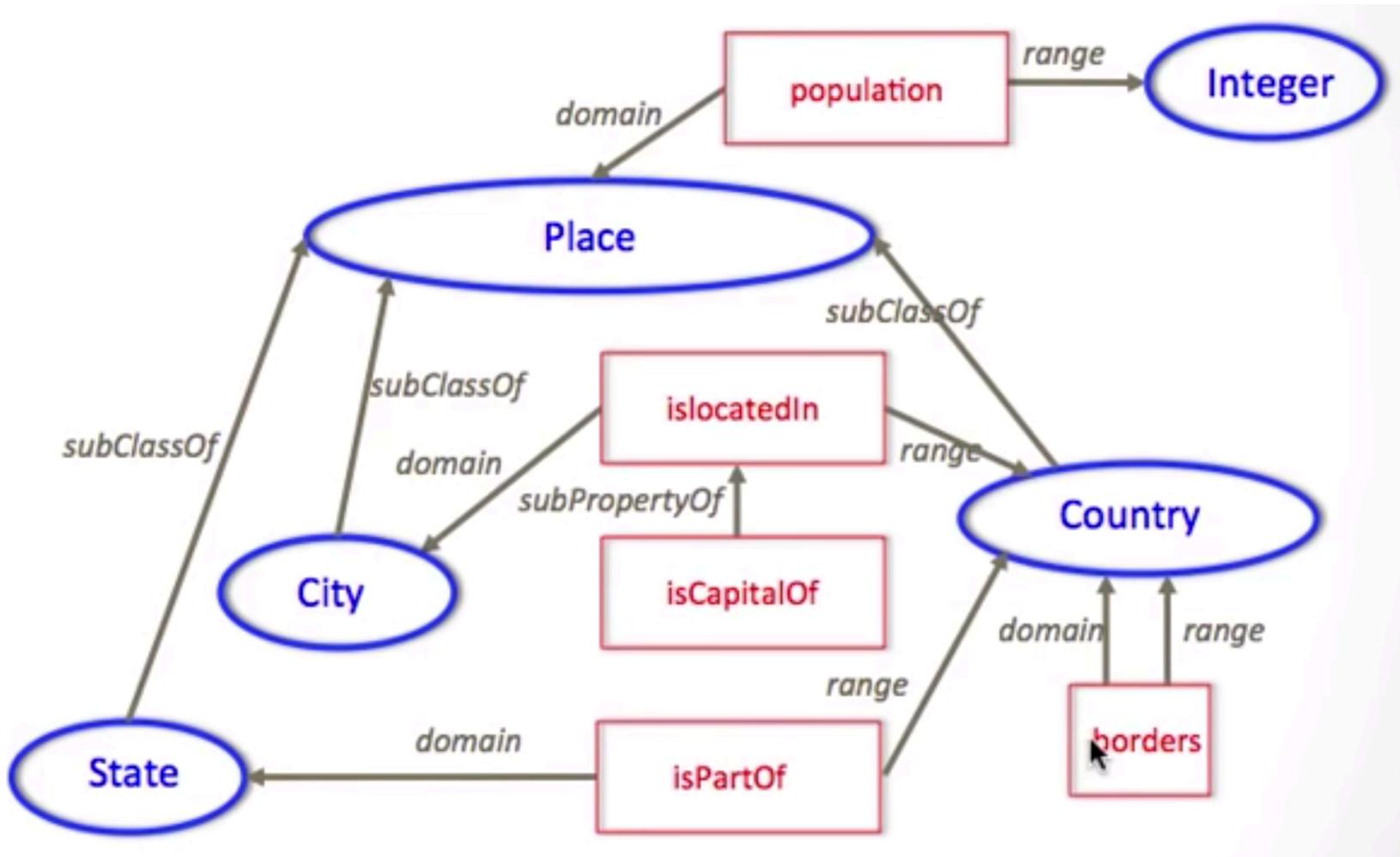
Reference:

<https://www.youtube.com/playlist?list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j>

Create Classes and Properties

- <https://www.youtube.com/watch?v=MbauHV2-XYw&list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j&index=1>

Country Ontology



Define Property Types and Axioms

<https://www.youtube.com/watch?v=qVOQLDTNsSg&list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j&index=2>

Symmetric Property

2. Would you describe `geo:borders` as a transitive property or as a symmetric property? If yes, write the appropriate definition in Turtle.

X

`geo:borders` is not a transitive property (if country A borders country B, and country B borders country C, then it's not necessary that A borders C)

`geo:borders` is a symmetric property (if country A borders country B, then country B also borders A)

`geo:borders`

a

`owl:SymmetricProperty`.

Functional Property

3. Which of the existing properties would you describe as functional? Give one example in Turtle.

`geo:borders` is not a functional property (a country may border more than one other countries)



All other properties are functional, e.g. `isLocatedIn` connects a city to a single country (a city may not be located in two countries)

`geo:isLocatedIn` a `owl:FunctionalProperty`.

Reflexive Property

4. Would you describe any of the properties as reflexive?

None of the existing properties are reflexive: `geo:borders` may not relate a country to itself (a country borders other countries only), while for the rest of the properties the domain and range are disjoint

Inverse Property

5. Give one example of a property that you would define in OWL2 as the inverse of one of the properties of the Schema. Write the definition in Turtle.

We can define `geo:containsState` as the inverse property of `geo:isPartOf` (if state A is part of country B, then B contains state A)

```
geo:containsState a owl:ObjectProperty;
owl:inverseOf geo:isPartOf.
```

Property Chain

6. Define `geo:nearbyCountry` as a property chain to connect a city to the neighboring countries of the country that the city is located in.

```
geo:nearbyCountry a owl:ObjectProperty;  
owl:propertyChainAxiom(geo:isLocatedIn geo:borders).
```

Define Class Axioms

<https://www.youtube.com/watch?v=RpM5f3OGqHo&list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j&index=3>

Disjoint Classes

7. Give one example of two or more classes of the RDF Schema that you would define in OWL2 as disjoint classes. Use the Turtle syntax to write the definition.

`geo:Country, geo:State and geo:City are disjoint classes.`

`I`

`geo:Country owl:disjointWith geo:State, geo:City.`

`geo:State owl:disjointWith geo:City.`

Complement Classes

8. Define in OWL2 two complement subclasses of `geo:Country`; one for European countries and another one for non-European countries, such that no member of one class can be member of the other.

```
geo:EuropeanCountry rdfs:subClassOf geo:Country.  
geo:NonEuropeanCountry rdfs:subClassOf geo:Country;  
owl:complementOf geo:EuropeanCountry.
```

Union Classes

9. Countries and states can both be described as regions. Define in OWL2
`geo:Region` as the union of the classes that represent countries and states.

```
geo:Region a owl:Class;  
owl:unionOf (geo:Country  
             geo:State).
```

Intersection Classes

10. Define in OWL2 `geo:NorthernEuropeanCountry` as the intersection of `geo:NorthernCountry` and `geo:EuropeanCountry`.

```
geo:NorthernEuropeanCountry a owl:Class;  
owl:intersectionOf ( geo:EuropeanCountry  
geo:NorthernCountry ).
```

Define Classes as Property Restrictions

<https://www.youtube.com/watch?v=2aJudF8AnBI&list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j&index=4>

Classes with Restrictions

11. A central European country is a European country that borders only European countries. Write the above definition in OWL2 as a universal restriction on **geo:EuropeanCountry** and **geo:borders**.

```
geo:CentralEuropeanCountry a owl:Class;
    rdfs:subClassOf geo:EuropeanCountry;
    rdfs:subClassOf [a owl:Restriction;
        owl:onProperty geo:borders;
        owl:allValuesFrom geo:EuropeanCountry].
```

Classes with Existential Restrictions

12. A Europe Neighbor Country is a non-European country that borders with at least one European country. Write the above definition in OWL2 as an existential restriction on `geo:NonEuropeanCountry` and `geo:borders`.

```
geo:EuropeNeighborCountry a owl:Class;
    rdfs:subClassOf geo:NonEuropeanCountry;
    rdfs:subClassOf [ a owl:Restriction;
        owl:onProperty geo:borders;
        owl:someValuesFrom geo:EuropeanCountry ].
```

Value Restrictions on Property

13. A German city is a city that is located in Germany. Write the above definition in OWL2 as a value restriction on `geo:City` and `geo:locatedIn`.

`geo:Germany` a `geo:EuropeanCountry`.

```
geo:GermanCity a owl:Class;  
rdfs:subClassOf geo:City;  
rdfs:subClassOf [a owl:Restriction;  
    owl:onProperty geo:isLocatedIn;  
    owl:hasValue geo:Germany].
```

Cardinality Restrictions on Property

14. An island country is a country that doesn't border any other country. Write the above definition in OWL2 as a cardinality restriction on `geo:Country` and `geo:borders`.

```
geo:IslandCountry a owl:Class;
    rdfs:subClassOf geo:Country;
    rdfs:subClassOf [a owl:Restriction;
        owl:onProperty geo:borders;
        owl:cardinality 0].
```

Adding Individuals

- [https://www.youtube.com/watch?v= 7MfDdsFePk&list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j&index=5](https://www.youtube.com/watch?v=7MfDdsFePk&list=PLD8uCWff9n-EG4KK2OAiPRSCPgNJXf49j&index=5)