# 1 Background

## 1.1 Semantic web and Ontology Introduction

**Semantic Web (Web 3.0:**

* **make Internet data machine-readable**.
* **based on Ontology**
  + **defines** **concepts** **and relationships** **(terms) to describe** **an** **area of concern**.
  + **Vocabularies**: **classify** **terms** **used in** **a particular application**, **characterize** **possible relationships**, **define possible constraints**
  + **Ontology** **understood as data model.**

Reference: <https://arxiv.org/pdf/1006.4567.pdf>

## 1.2 Triple

**A triple**[[1]](#footnote-1) is a **set of three entities** that **codifies a statement about semantic data** **in** the form of **subject–predicate–object** **expressions** (e.g., "Bob is 35", or "Bob knows John"). This format **enables** knowledge to be represented in a **machine-readable** way.

A triple can be **represented** in multiple formats[[2]](#footnote-2) such **as RDF/XML, RDFa, Notation3[[3]](#footnote-3), and Turtle[[4]](#footnote-4).**

Typical **turtle** representation:

Subject predicate1 object1;

predicate2 object2;

predicate3 object3, object4.

For example, in the following triple

<<http://www.example.org/index.html>> <hasCreator> "John Smith" .

* Subject is the web page **http://www.example.org/index.html**
* Predicate is the **hasCreator** property
* Object is the String **"John Smith"**.

## 1.3 Ontology components

**Ontology** includes **three main components**: **concepts** (**or classes**), **instances**, and **properties**

A **Concept** (also known as a **class** or a **term**) is an **abstract group, set, or collection of objects**. It is the **fundamental element of** the **domain** and usually **represents** a **group** or **class** **whose members share common properties**. This component is **represented** in **hierarchical graphs**, such that it looks **similar** to **object-oriented systems**. The concept is represented by a “super-class”, representing the higher class or so-called “parent class”, and a “subclass” which represents the subordinate or so-called “child class”.

**Individuals** (**instances** or **objects**) are the **basic, "ground level" components** of an ontology. The individuals in an ontology may **include** **concrete objects** such as people, animals, tables, automobiles, molecules, and planets. Strictly speaking, an ontology need not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology.

**Individuals** have properties, to store their information. There are **two types of properties**:

**Data property**

Data properties connect individuals with literals. In some knowledge representation systems, functional data properties are called attributes. Examples of data properties are: hasName, hasPhoneNumber, etc.

**Object properties**

Object properties connect two individuals. Example of object properties are isTaughtBy, supervises, etc.

Each property has domains and ranges

* **Property domain**s: specify which classes that the property belongs to
* **Property range**: specify the possible value of the property

Example:

:age a owl:DatatypeProperty;

rdfs:domain foaf:Person;

rdfs:range xsd:nonNegativeInteger .

:isTaughtBy a owl:ObjectProperty;

rdfs:domain :Course;

rdfs:range :AcademicStaffMember;

rdfs:subPropertyOf :involves.

## 1.4 Vietnamese Tourism ontology

Vietnamese tourism ontology is a unified and common representation of data in tourism domain. It involves many subjects of history and culture as a natural feature of tourism.

Link download: <https://drive.google.com/drive/folders/1QGoKV4hWqQzi2LGo8T8Fu30LzZPc3nwL?usp=sharing>

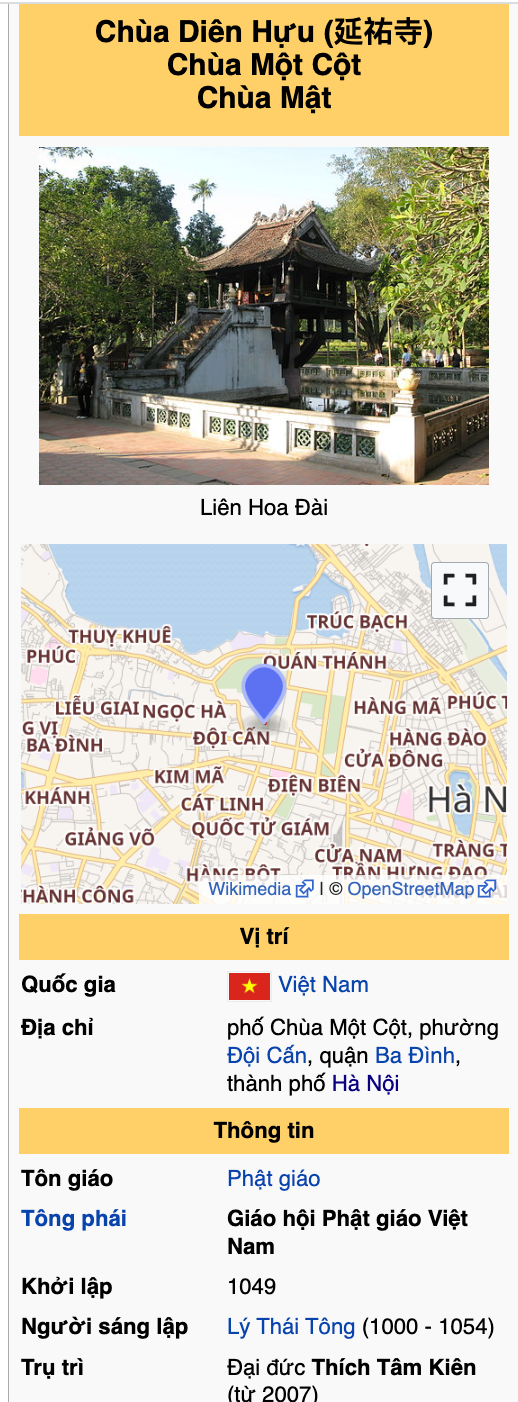
The ontology can be opened with **protege** software <https://protege.stanford.edu/>

## 1.5 Websites to collect Tourism data

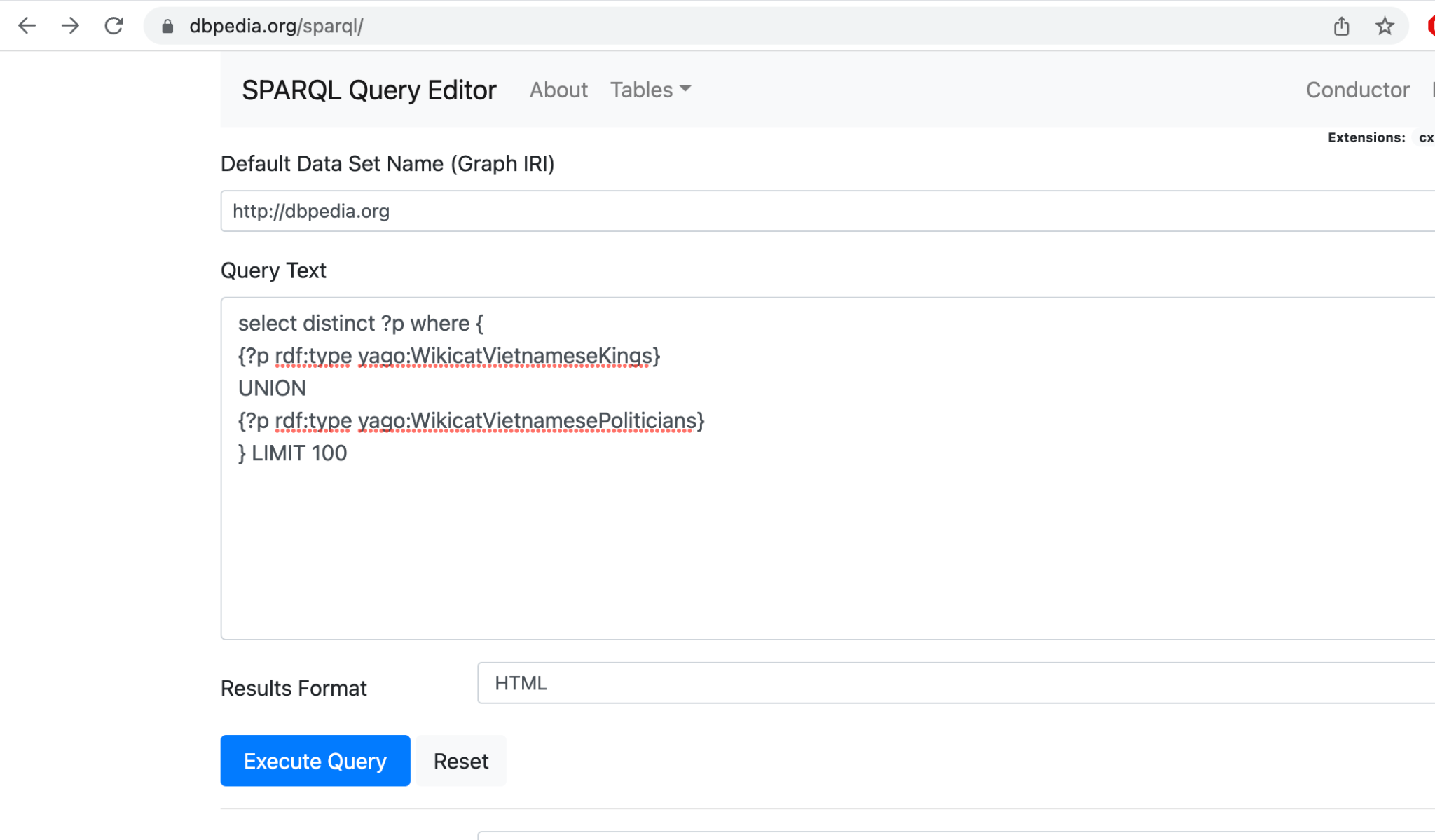
There are numerous websites where we can collect tourism data. Examples are Wikipedia, Wikidata.org, DBpedia.org, and the websites of the department of culture and tourism in many cities and provinces in Vietnam.

For example, using Wikipedia, you can parse the data represented in the infobox.

<https://vi.wikipedia.org/wiki/Ch%C3%B9a_M%E1%BB%99t_C%E1%BB%99t>



Using the following query with the DBPedia SPARQL endpoint, we can roughly get famous people in Vietnam. Visit <https://www.w3.org/TR/rdf-sparql-query/> to learn SPARQL.



select distinct ?p where {

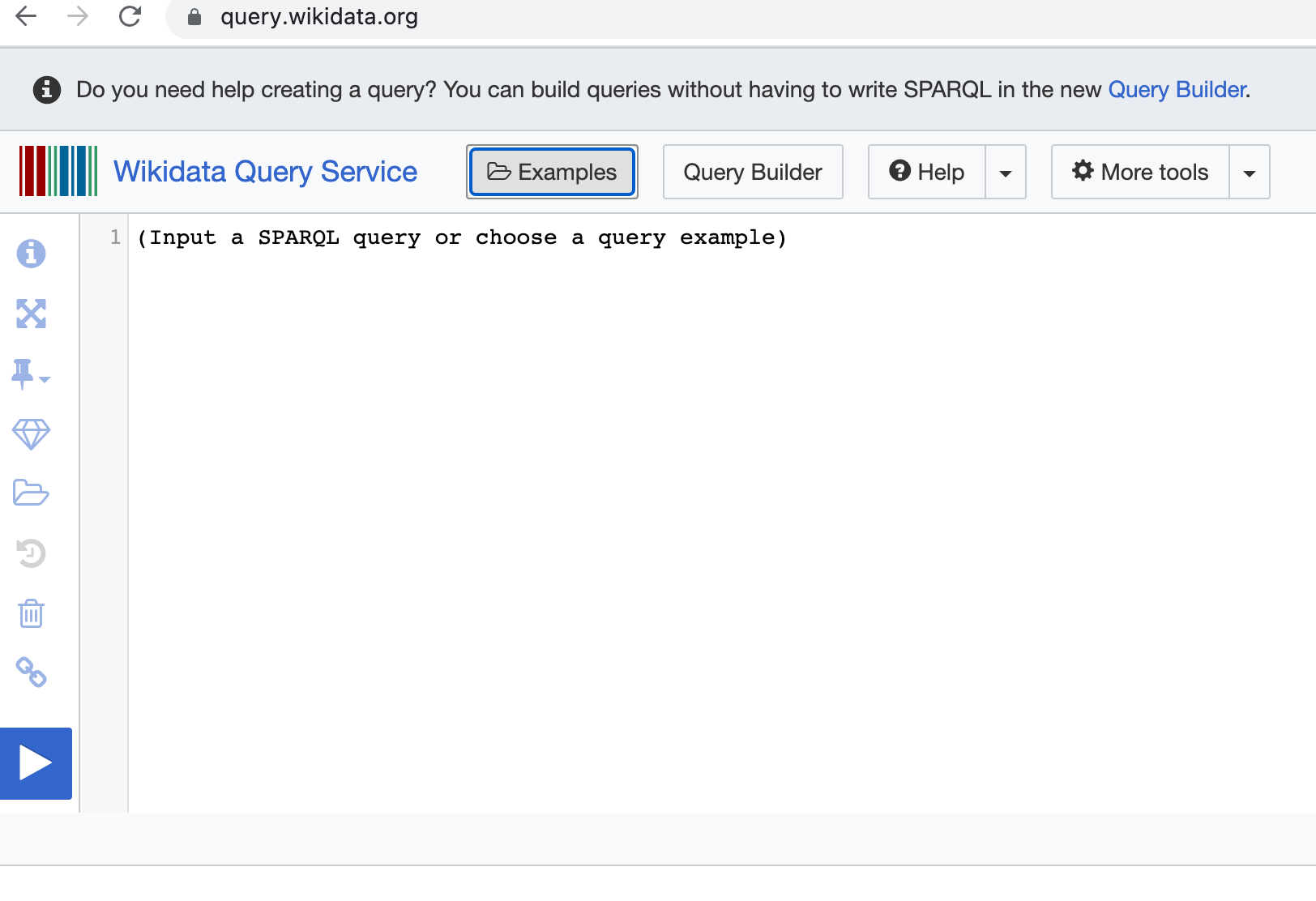
{?p rdf:type yago:WikicatVietnameseKings}

UNION

{?p rdf:type yago:WikicatVietnamesePoliticians}

} LIMIT 100

Similarly, you can use the SPARQL endpoint from Wikidata



Many other websites provide data on tourism, but the data is unstructured. Examples are

* Hoidulich.net
* Khamphadisan.com
* http://lehoi.info/

2 Problem description

You need to understand the tourism ontology, collect tourism data from tourism websites, and store the data in files using the turtle[[5]](#footnote-5) format. Each topic needs to be stored in separate file.

You will get a high score if you be able to collect a huge amount of data on various topics and the data must be precise.

Example turtle representations:

@prefix vntourism: <http://www.semanticweb.org/minhn/ontologies/2021/0/vntourism#> .

vntourism:Cổ\_Loa\_Citadel rdf:type owl:NamedIndividual ,

vntourism:CitadelArchitecture ;

vntourism:chosenCapitalBy vntourism:An\_Dương\_Vương ,

vntourism:Ngô\_Quyền ,

vntourism:Ngô\_Xương\_Văn ;

vntourism:hasAdministrativeDivision vntourism:Hà\_Nội ;

vntourism:hasBuildBy vntourism:An\_Dương\_Vương ;

vntourism:hasBuildTime "3rd century"@en ;

vntourism:hasCountry vntourism:AuLac ;

vntourism:hasFestival vntourism:Lễ\_hội\_Cổ\_Loa ;

vntourism:hasHistoricalSiteLevel vntourism:SepcialNationalLevel ;

vntourism:hasPeriod vntourism:NgoDynasty ;

rdfs:label "Cổ Loa Citade"@en ,

"Thành Cổ Loa"@vn .

:Ly\_Thuong\_Kiet\_Working\_TimeLine

rdf:type owl:NamedIndividual, :TemporalEntity;

:hasBeginning [

rdf:type :Instant ;

:inTimePosition [

rdf:type :TimePosition ;

:hasJob :Kinh\_Phong\_Su\_Lý\_Thường\_Kiệt;

:year "1061"^^xsd:gYear

]

] ;

:hasDurationDescription [

rdf:type :GeneralDurationDescription ;

:year "8"^^xsd:gYear

] ;

:hasEnd [

rdf:type :Instant ;

:inTimePosition [

rdf:type :TimePosition ;

:year "1069"^^xsd:gYear

]

] .

1. <https://en.wikipedia.org/wiki/Semantic_triple> [↑](#footnote-ref-1)
2. <https://ontola.io/blog/rdf-serialization-formats/> [↑](#footnote-ref-2)
3. <https://www.w3.org/TeamSubmission/n3/> [↑](#footnote-ref-3)
4. ​​<https://www.w3.org/TR/turtle/> [↑](#footnote-ref-4)
5. ​​<https://www.w3.org/TR/turtle/> [↑](#footnote-ref-5)