2/21/2022

# Watt-Tour Ontology.

Big Data Coursework-1

F21BD- Stage 2 Report

## Submitted To:


Dr Alasdair Gray
A.J.G.Gray@hw.ac.uk
Course Instructor


## Submitted From:


Group 2
Jiancheng Zhang H00341619
Shreyas Arunesh H00357095
Shonan Gomes H00383160

## Table of Contents

# I. Group Contribution

Everybody in the group contributed equally in this coursework and all the assumptions and discussions illustrated in this report are the results of mutual conclusion and knowledge sharing among the team members.

**Table 1. Group contribution summary**

| HWID | Name | Contributions | Total Contribution level | Signature |
|---|---|---|---|---|
| H00341619 | Jiancheng Zhang | Task -1 Task-2 Task -3 | 100 % | 张佳程 |
| H00357095 | Shreyas Arunesh | Task -1 Task-2 Task -3 | 100% | Shreyas Arunesh |
| H00383160 | Shonan Gomes | Task -1 Task-2 Task -3 | 100 % |  |

# II. Introduction:

This report illustrates the design and development of the Watt- tour Ontology which represent the need for the Watt tour. This ontology is aspired to offer a good data sharing practice to tourists by keeping an up-to-date information about the tour plans of their destination cities. The designed Watt Tour ontology offers variety of tour plans for the tourists to covers number of cities to travel and places that will be visited in that city. The number of cities and countries is proportional to the number of days the tourist wishes to opt.

The ontology was designed using the Protégé software and the queries was developed using SPARQL Protocol and RDF (Resource Description Framework) Query Language in Apache Jena Fuseki server.

The Watt Tour can be imported from the following Web Address:
URL : [https://github.com/ShreyasArunesh/Watt-Tour-Ontology-/blob/main/F21BD-CW-1-Group-2.owl](https://github.com/ShreyasArunesh/Watt-Tour-Ontology-/blob/main/F21BD-CW-1-Group-2.owl)

# III. Watt Tour Design Discussions:

## 3.1 Different classes for Cities and Countries:

Our individual Ontologies had 2 different approaches. Having city as a subclass of country and having a separate class for cities and countries covered in the tour. As shown in the Figure 1: Separated City and country class and Figure 2: City subclass of country.

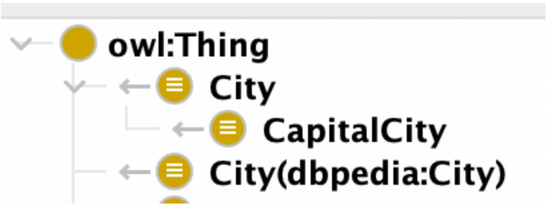


**Figure:1 Separated City and Country class**

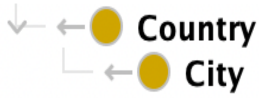


**Figure:2 Inherited classes**

We noted that each class might have multiple individual and the properties of all the instances of Country cannot be inherited to the subclass city. This inheritance property is not satisfied in the 2nd type of approach. For Example: City: Delhi can be a capital of the Country: India but City: Beijing cannot be capital of the same Country: India. Therefore, we decided to go with the 1st Approach of separating the city and country classes.
Note that the CapitalCity is considered as the sub-class class for City which implies all the properties of class City is also inherited to its subclass CapitalCity. For Example: Delhi is the capital of a country India which is also a City, i.e., it can inherit all the properties from the parent class City.

## 3.2 Watt Tour having subclasses called Name and Description and tour packages as a subclass of tour plans:

Two of our Individual ontologies had different ways of knowledge representation for the tour Packages offered in Watt tour. This is illustrated in Figure 3: Approach (1) and Figure 4: Approach (2). One of the individual ontologies did not capture this requirement of extension of number of cities by number of days. In the first approach the Name and description is subclass of the class Watt tour and tour packages can be inherited form the class TourPlan. In Approach (2), the country name is taken as a subclass which can be taken as individuals for the class city and country. Both the approaches can be validated.

We decided to finalise the approach (1) because:

- Approach 1 provided us to have a good data sharing practice among the classes.
- This gives flexibility for tourists to choose any Package and cities of that country for their destination rather than having a fixed cities and country name of that package. Ex: Tourists can choose to either travel to 3 cities in Country: china or travel 3 cities in Country: India in One day package.
  All the city and country name are provided in the individual along with the capital city of that country.
- The number of cities to be covered in that package can be restricted by adopting the complex class expressions of having min 3 cities in one day TourPackage and min of 4 cities with max 6 cities in 2 day TourPackage and min 5 cities with max 9 cities in 3 day TourPackage.

The subclasses of Tour plans are not disjointed as this may share the individuals of each other classes.

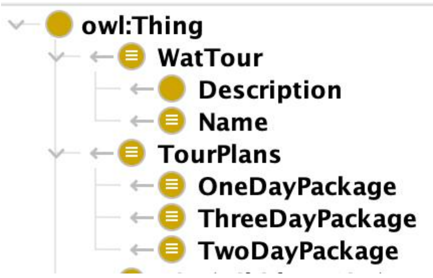


**Figure 3: Approach 1**


**Figure 4: Approach 2**

## 3.3 Including Tourists and Places covered in the knowledge representation of watt tour:

The details of the tourists involved in Watt tour is Tourists are the subclass of Person and Tourists have subclasses AdultsTourists and KidsTourists. Places has subclasses Kids attractions and Adult attractions. Figure 5 bellow lustrates the class hierarchy of Tourists and Places. The complex class restrictions in these classes are:

- Every Person can register to exactly one WattTour Package.
- Every Person is attracted to some places covered in watt tour cities.
- KidsTourists are attracted to only kids' attractions which can be Museums and Parks.
- Adult tourists are attracted to either KidsAttractions or Adult attractions. Some of the adult attractions considered in this ontology are Temples and Clubs. The name of these attractions can be considered as an individual.

Note that the age limit is not considered in this ontology for tourists to be classify as adults and kids.

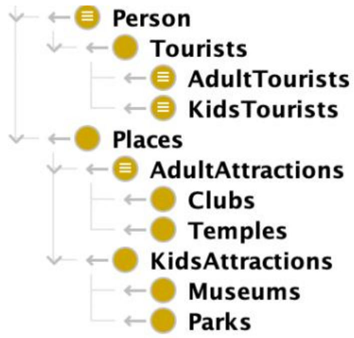


**Figure 5: Class Hierarchy for Person and Places in watt tour.**

## 3.4 Implementing Complex Relations among the classes:

The class hierarchy in our Group Ontology contains complex relations between the classes which aids in creating a good data sharing technique in the Ontology. Some of the complex restrictions are Intersection class, Union class. This describes an anonymous class that contains the individuals that are members of the class for Ex: Human and the class Male [1]

**Intersection class** is implemented for 2 or more classes using AND operator. In the Ontology as shown in the Figure 6 below, The class AdultAttractions are Union with the Kidstourists and AdultTourists.This implies that AdultAttractions is not Attracted by kid tourists AND is attracted only by AdultTourists. This shows the efficient data sharing practice among the union classes.


**Figure 6: Intersection Classes implementation.**

**Union class** is implemented using OR operator which is the combination of two or more classes using described in the Figure 7 bellow, WattTour class can either have OneDayPackage OR TwoDayPackage OR ThreeDayPackage tour classes.

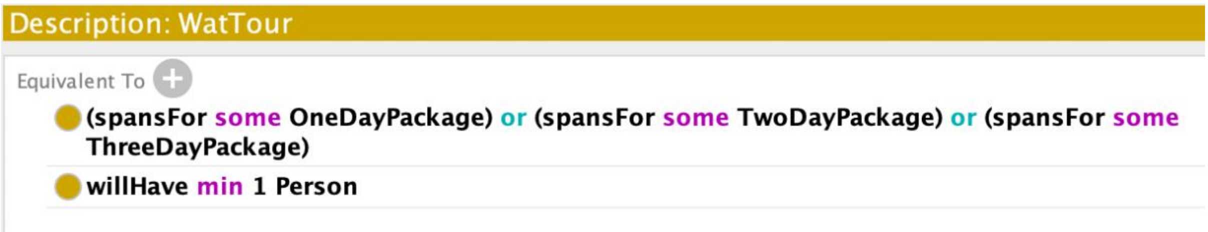


**Figure 7: Union class description Implementation.**

## 3.5 Implementing Qualifier Restrictions among the classes:

In the group ontology, we decided to implement the Qualifier restrictions among the classes as per the instructions suggested by Rector et.al in their research.
Qualifier restrictions are divided into: [2]

1. A qualifier- which is either the existential quantifier (some), or the universal quantifier (only).
2. A property, along which the restriction acts.
3. A filler that is a class description.

The quantifier effectively constrains the relationships in which a specific individual participates for that individual. It accomplishes this by specifying the existence of at least one type of relationship or limiting the types of relationships that can exist (if they exist).

**Some of the Qualifiers implemented in our Ontology are:**

**Cardinality restrictions:** These are used to talk about the number of relationships that an individual may participate in for a given property. For instance, MIN, MAX, ATLEAST
As described in the Figure 8 bellow, the ThreeDayPackage tour should cover MIN 7 Cities and MAX 9 cities in the tour Plans this limits the assertions among the classes.


**Figure 8: Qualifier Restrictions implemented in ontology**

## 3.6 Defining Individuals Property and Restrictions:

The Individual properties and Restrictions property assertions can be made using SameAs and DifferentFrom assertion which defines the exact characters of that Individual for that Instance of classes.
The Figure 9 bellow describes the Individual Properties and restrictions in the ontology, the individual London is the Instance of the class CapitalCIty which is also a subclass of class City as Explained above in the section 4.1. This individual is same as the Existing Wikidata called Wikidata:London and is different Individual from the rest of the Class: CapitalCIty Instances. This implies that the CapitalCity might have multiple instances, but all the properties asserted to those instances are different from each other. Figure bellow represents the individual property like "Same Individual As" and "Different Individual As" for the City London which is also a capital city of the country United Kingdom (This will be inferred through Reasoner).

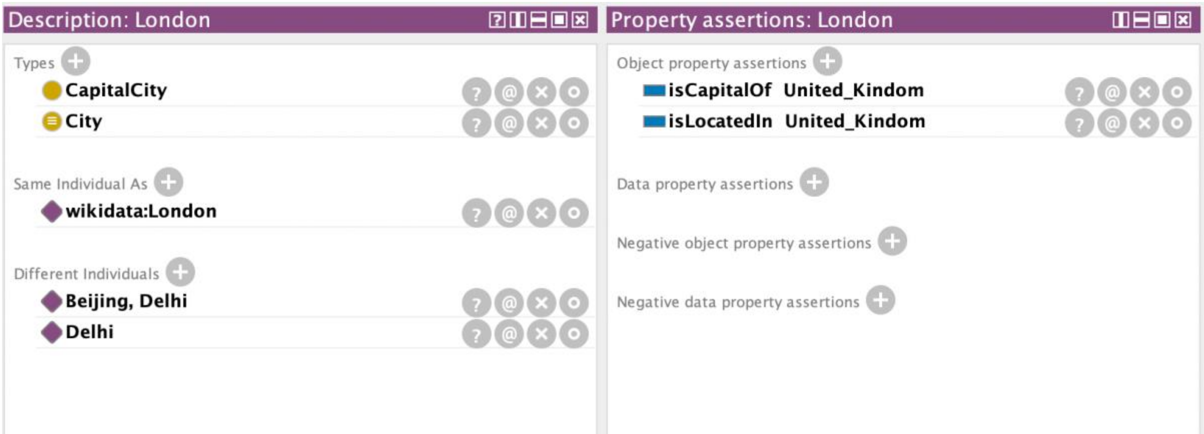


**Figure 9: Individual Property and Restrictions in Ontology.**

### 3.7 Fine tuning of the Group Ontology:

- We incorporated the good practice of naming the IRI's Human Readable and Human Understandable to make this ontology. Two of our ontologies had opaque Identifiers for Individuals but we finalised by using correct identifiers for our Group Ontology.

- Changed all the IRI links consistent to make querying easier. For instance, as suggested in the coursework specifications, we changed all the IRI to the form as follows: Wikidata: http://www.wikidata.org/entity/

  We have also linked the IRI to a GitHub repository.

- While linking the external Wikidata knowledge graph, two of our individual ontologies had imported the wrong individual which was found while extracting the knowledge in the Querying. For instance We had linked Bermingham (Which is the common family name in the country) but Birmingham is the Wikidata knowledge graph which was supposed to be imported.
- Keeping words and things separated in the ontology, that is not confusing entities with its name or identity.
- Finally documentation of all the classes and data, object properties by giving appropriate definitions for all the classes.

## IV. Requirements Criteria:

This section describes the met requirement specification of the final ontology. Table 1 below lustrates the concise view on the requirements satisfied by the proposed ontology

**Table 1: Major Requirements of Watt Tour and Classes Satisfying the requirements for Task -1**

| **Coursework Requirements** | **Classes satisfied** | **Explanation** |
|---|---|---|
| Tours which visit several cities. | City | This class stores the name of the cities Watt-Tour Offers |
| Tours that extend to number of days. | Tour Plans | This class offers the packages offered by watt tour. |
| Name of the tour | Tour Plans | This class stores the name of the tour Ex: Gold, Silver and Platinum. |
| Description of the tour visited | Description | This class captures the brief description of the tour plans. |
| Name of the country the city located in. | Country. | This class captures the destination country name. |

| | | |
|---|---|---|
| Capital city of that country. | CapitalCity | This subclass of City captures the capital of the country |
| Mapping Cities to external knowledge graph | Dbpedia:Cities | This is the dbpedia knowledge which is equivalent to Watt: city. |
| Mapping Countries to external knowledge graph | Dbpedia:Countries | This is the dbpedia knowledge which is equivalent to Watt: country. |
| Mapping Instances to external knowledge graph | Wikidata:CounrtyName | This is the existing wikidata knowledge that captures the country names |
| Mapping person to external knowledge graph. | Foaf:Person | This is the foaf:Person that shares the properties of Watt:Person to store tourists involved in WattTour. |
| Mapping the data property number of days to external knowledge graph | Time: days<br>Time: day | This is the time:days that shares the. Data property with watt:numberofdays and the date and time of the travel. |

**Table 2 – All the queries satisfies the Requirement Criteria.**

| Tasks | Requirements | Explanation |
|---|---|---|
| Query 1 | Provide the name for the cities in your ontology in two non-English languages. | Fully satisfied. Figure 10 illustrates the expected output |
| Query 2 | Information about cities:<br>Size (area)<br>Population size<br>Time zone offsets (cities may have more than one)<br>Official website<br>Number of twinned cities | Fully satisfied. Figure 11 illustrates the expected output. |
| Query 3 | Provide the number of art galleries, museums, and botanical gardens in the city. | Fully satisfied. Figure 12.1 Figure 12.2.1, Figure 12.2.2 , Figure 12.2.3 illustrates the expected output. |

## V. Ontology Mapping Table:

In this section Table 3 will give a brief overview on the criteria for selecting a particular ontology. And Table 4 will provide details on mapping of Watt Tour ontology with the existing ontology.

**Table 3: Overview of Selection of Existing Ontology.**

| SL.No | Target Ontology | Summary | Selection Criteria |
|---|---|---|---|
| 1. | Foaf: Person | Defines the recourse who is executing a particular action | Selected to represent people involved in Watt tour. |
| 2. | Dbpedia: Country | This defines the Countries and its properties. | Selected to represent the countries covered in Watt tour. |
| 3. | Dbpedia: City | This defines the Cites and its properties | Selected and expanded to represent the Cities that are covered in watt tour and to share the properties. |
| 4. | Time: days | length of, or element of the length of, a temporal extent expressed in days [6] | Used to represent the length of each watt Tour package. |
| 5. | Time: day | Represents any specific representation of a calendar day from any calendar. | Selected to represent the data property instance of days. |
| 6. | Wikidata: (individuals) | Defines the all the properties and labels of that particular city and country. | Represented to express the name of the cities and countries covered in watt tour. |

**Table 4: Mapping Table of Watt tour Ontology.**

| Mapping ID | Source Entity | Mapping Property | Target Entity |
|---|---|---|---|
| M1 | Watt: Person | owl:equivalentClass | Foaf: Person |
| M2 | Watt: Country | Skos: related | Dbpedia: Country |
| M3 | Watt: City | Skos: related | Dbpedia: City |
| M4 | Watt: numberofDays | Owl:equivalentProperty | Time: days |
| M5 | Time:day | Skos: broader | Time: day |
| M6 | Watt: individual_Cities(Ex: Mysore) | Owl:equivalentClass | Wikidata: (individuals) |

**M1** This has the mapping property of equivalent class between the source entity Watt:Person and Target entity: Person. Kindly note that the domain is defined more broadly in the foaf:Person class compared to watt:Person. We can see the foaf:Person is the subclass of foaf:Agent and the properties of foaf:Person are equivalent to the properties of watt:Person such as the "projectPerson" which can be related to Tourists in the project watt Tour.

**M2 and M3** These two classes are equivalent class with the existing dbpedia data: country and city but the class city and country in Watt Tour is effectively and abstract class with the subclass of capitalCity which differs from the dppedia data where each capitalcity is instantiated. Owing to these differences in the patterns of both the ontologies, we decided to give an appropriate definitive mapping called skos:related.

**M4** The number of days covered in the watt tour has an equivalent data property to define the total number of days each watt tour packages can extend to. We can note that the equivalent property of the source entity of the existing knowledge graph time:days has the domain generaDurationDescription and range:Decimal. Hence, we decided to use this explicitly mapped property in watt tour as owl: equivalentProperty.

**M5** Time:Day encompasses a variety of GeneralDateAndTimeDescription domain which has a broader mapping to watt tour to define the day and time of each watt tour packages. This is also a general data property instance of Days used to define the number of days for watt tour. That is why we used the skos: broader.

**M6** Existing wikidata ontology defines the equivalent class for the individuals of the watt tour. For example, City Mysore has all the properties defined in the wikidata:Mysore like (coordinate location, labels, population size and tourist attractions etc…). this is coined as an Equivalent class to the instances of the watt tour. Also, this was necessary for the querying in the task-2 for labels to retrieve from these wikidata using rdf:label property.

# VI. Task 2 Queries:

We provided the shared latest information covered in Watt Tour ontology by using SPARQL query to extract information about the watt tour. These queries can either be executed in Apache Jena Fuseki or in Protégé software. After exploring the various issues in Protégé, we decided to stick to Apache Jena Fuseki server since we got an httpclient jar not defined exception when tried executing our queries in Protégé.
This was done by uploading the watt tour ontology to the Apache Jena server through the PRIFIX defined bellow. We noted that for federated queries, it was necessary to implement some optimisation for our queries since it makes the queries much longer than expected (Particularly for Query 3). This optimisation was made to avoid time run out exception.

Note the following PRIFIX used for the 3 queries:

PREFIX this: <https://github.com/ShreyasArunesh/Watt-Tour-Ontology-/blob/main/F21BD-CW1.owl/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: http://www.wikidata.org/prop/direct/

## 1) Query 1

The task in here was to provide the names of the cities covered in watt tour ontology. In the Query it can be noted that we have languages Chinese and Hindi as non-English languages. Here ?cities represents IRI label of cities in Watt tour Ontology and ?wiki_cities is the IRI label of the cities in Wikidata. Figure 10 bellow illustrates the output results of Query -1.

**SPARQL query:**

```
SELECT ?cities ?wiki_cities (GROUP_CONCAT(?label; SEPARATOR=", ") as ?labels)
WHERE
{
  ?cities rdf:type this:City.
  ?wiki_cities owl:sameAs ?cities.

  SERVICE <http://query.wikidata.org/sparql> {
    ?wiki_cities rdfs:label ?label.
    FILTER (lang(?label) = "zh" || lang(?label) = "hi")
  }
}
GROUP BY ?cities ?wiki_cities
ORDER BY ?cities
```

## Results:

QUERY RESULTS

Table Raw Response

Showing 1 to 9 of 9 entries

Search: Show 50 entries

| | cities | wiki_cities | labels |
|---|---|---|---|
| 1 | this:Beijing | wd:Q956 | "बीजिंग, 北京市" |
| 2 | this:Birmingham | wd:Q2256 | "बर्मिंघम, 伯明翰" |
| 3 | this:Changsha | wd:Q174091 | "चांगशा, 长沙市" |
| 4 | this:Chengdu | wd:Q30002 | "成都市, चेंगदू" |
| 5 | this:Delhi | wd:Q1353 | "德里, दिल्ली" |
| 6 | this:London | wd:Q84 | "लंदन, 倫敦" |
| 7 | this:Manchester | wd:Q18125 | "曼彻斯特, मैन्चेस्टर" |
| 8 | this:Mumbai | wd:Q1156 | "मुम्बई, 孟买" |
| 9 | this:Mysore | wd:Q10086 | "मैसूर, 迈索尔" |

Showing 1 to 9 of 9 entries

**Figure 10: Output Results of Query 1**

## 2) Query 2

The second query was to populate information box for watt tour ontology containing following data:

- Size (area)
- Population size
- Time zone offsets (cities may have more than one)
- Official website
- Number of twinned cities

Here, we can add OPTIONAL to all the triples in federated query, but this will lead to time run out exception. Therefore, in our query, we just used OPTIONAL for time zone and website property. To make result more readable, we had to convert result to string by using STR() method.

**SPARQL query:**

```
SELECT ?cities
        ?wiki_cities
        (CONCAT(STR(?area), " square kilometre") AS ?area_new)
        (STR(?popluation) AS ?number_of_people)
        (GROUP_CONCAT(DISTINCT ?name; SEPARATOR=", ") as ?names)
        ?website
        (STR(COUNT(?twinned_city)) as ?number_of_twinned_cities)
WHERE
{
  ?cities rdf:type this:City.
  ?wiki_cities owl:sameAs ?cities.

  SERVICE <http://query.wikidata.org/sparql> {
    ?wiki_cities wdt:P2046 ?area.
    ?wiki_cities wdt:P1082 ?popluation.
    OPTIONAL {?wiki_cities wdt:P421/wdt:P373 ?name.}
    OPTIONAL {?wiki_cities wdt:P856 ?website.}
    ?wiki_cities wdt:P190 ?twinned_city.
  }

}
GROUP BY ?cities ?wiki_cities ?area ?popluation ?website
ORDER BY ?cities
```

## Results:

In the Figure 11 bellow, illustrates the output results of Query 2.

Please note that London and Birmingham have two time zone, one is Greenwich Mean Time, another is British Summer Time. And the time zone for Mumbai and Mysore is null. Also, Mysore does not have an official website.

QUERY RESULTS

Table | Raw Response

Showing 1 to 9 of 9 entries

Search: Show 50 entries

| | cities | wiki_cities | area_new | number_of_people | names | website | number_of_twinned_ci |
|---|---|---|---|---|---|---|---|
| 1 | this:Beijing | wd:Q956 | "16410.54 square kilometre" | "21893095" | "UTC+08:00" | <http://www.beijing.gov.cn> | "65" |
| 2 | this:Birmingham | wd:Q2256 | "267.77 square kilometre" | "1137100" | "UTC±00:00, UTC+01:00" | <http://www.birmingham.gov.uk/> | "20" |
| 3 | this:Changsha | wd:Q174091 | "11815.96 square kilometre" | "10047914" | "UTC+08:00" | <http://www.changsha.gov.cn/> | "6" |
| 4 | this:Chengdu | wd:Q30002 | "14334.78 square kilometre" | "20937757" | "UTC+08:00" | <http://www.chengdu.gov.cn> | "41" |
| 5 | this:Delhi | wd:Q1353 | "1484 square kilometre" | "26495000" | "UTC+05:30" | <http://delhi.gov.in/> | "11" |
| 6 | this:London | wd:Q84 | "1572 square kilometre" | "8908081" | "UTC±00:00, UTC+01:00" | <https://www.london.gov.uk> | "64" |
| 7 | this:Manchester | wd:Q18125 | "115.6 square kilometre" | "547627" | "UTC±00:00" | <https://cms.manchester.gov.uk/> | "13" |
| 8 | this:Mumbai | wd:Q1156 | "603 square kilometre" | "15414288" | | <https://portal.mcgm.gov.in/> | "13" |
| 9 | this:Mysore | wd:Q10086 | "132000000 square kilometre" | "3001127" | | | "4" |

Showing 1 to 9 of 9 entries

**Figure 11: Outputs Results for Query 2**

## 3) Query 3

This task was to find all the botanical gardens, museums, and art galleries in a city. The IRI for the type are Q167346, Q63982911, and Q1007870. Note that the Q63982911 is for all types of museums.
However, there is no certain property to indicate which city the facility is located in. We can use either coordinate location (P625) or located in the administrative territorial entity (P131), but neither of these provided consistency in extracting the information about the city since there can be an enormous difference between cities. Hence, we decided to use schema:description property because many facilities are contained in the city label. Although we cannot guarantee that this provides a perfectly accurate result, this is consistent for many cities in the world.
In the query, we used regular expression to find substring of city label in the description. The issue for this is that regular expression is a time-consuming operation, and quite easy to have a time run out exception in federated query.

Therefore, we suggested two ways of solving this, first is to separate them into three federated queries, for botanical garden, museum, and art gallery respectively; second is to use country (P17) property to have a pre-filter operation, so that we only apply regular expression for those facilities in the same country with the cities, which can significantly decrease the data size for regular expression filtering.
Moreover, between federated queries, there is an AND operation implicitly used in our query. To solve this, we can add UNION keyword between subqueries, but this will lead to server error. We can also use OPTIONAL in each federated query, where means the city can have zero or many facilities.

### SPARQL query for task-3

```
SELECT ?cities
        ?wiki_cities
        (STR(COUNT(DISTINCT ?botanical_garden)) as ?number_of_botanical_garden)
        (STR(COUNT(DISTINCT ?museum)) as ?number_of_museum)
        (STR(COUNT(DISTINCT ?art_gallery)) as ?number_of_art_gallery)
WHERE
{
  ?cities rdf:type this:City.
  ?wiki_cities owl:sameAs ?cities.

  SERVICE <http://query.wikidata.org/sparql> {
    ?wiki_cities wdt:P17 ?wiki_country.
    ?wiki_cities rdfs:label ?label.
    OPTIONAL{
      ?botanical_garden wdt:P31 wd:Q167346.
      ?botanical_garden wdt:P17 ?wiki_country.
      ?botanical_garden schema:description ?BG_description.

      FILTER (lang(?BG_description) = "en")
      FILTER (lang(?label) = "en")
      FILTER REGEX(?BG_description, ?label, "i")
    }
  }
```

```
  SERVICE <http://query.wikidata.org/sparql> {
?wiki_cities wdt:P17 ?wiki_country.
    ?wiki_cities rdfs:label ?label.
    OPTIONAL{
      ?museum wdt:P31/wdt:P31 wd:Q63982911.
      ?museum  wdt:P17 ?wiki_country.
      ?museum schema:description ?M_description.

      FILTER (lang(?M_description) = "en")
      FILTER (lang(?label) = "en")
      FILTER REGEX(?M_description, ?label, "i")
    }
  }
  SERVICE <http://query.wikidata.org/sparql> {
    ?wiki_cities wdt:P17 ?wiki_country.
    ?wiki_cities rdfs:label ?label.
OPTIONAL{
      ?art_gallery wdt:P31 wd:Q1007870.
      ?art_gallery wdt:P17 ?wiki_country.
      ?art_gallery schema:description ?AG_description.

      FILTER (lang(?AG_description) = "en")
      FILTER (lang(?label) = "en")
      FILTER REGEX(?AG_description, ?label, "i")
    }
  }
}
GROUP BY ?cities ?wiki_cities
ORDER BY ?cities
```

## Results of task-3

Please note that, the query will keep running up to 30 minutes when executed in Apache Jena Fuseki. Although the query satisfies the requirement specifications in the coursework, there were few glitches found in the results while working on this Query. Some of them are:

- Results of art gallery disappeared.
- Individual of City- Mysore disappeared (Found in Figure 12.1 bellow)

QUERY RESULTS

Table | Raw Response

Showing 1 to 7 of 7 entries

Search: Show 50 entries

| | cities | wiki_cities | number_of_botanical_garden | number_of_museum | number_of_art_gallery |
|---|---|---|---|---|---|
| 1 | this:Beijing | wd:Q956 | "1" | "14" | "0" |
| 2 | this:Birmingham | wd:Q2256 | "2" | "10" | "0" |
| 3 | this:Chengdu | wd:Q30002 | "0" | "4" | "0" |
| 4 | this:Delhi | wd:Q1353 | "0" | "11" | "0" |
| 5 | this:London | wd:Q84 | "2" | "111" | "0" |
| 6 | this:Manchester | wd:Q18125 | "2" | "19" | "0" |
| 7 | this:Mumbai | wd:Q1156 | "0" | "1" | "0" |

Showing 1 to 7 of 7 entries

**Figure 12.1 : Output results for Query 3 using regular expression filtering.**

## 3.1) Federated Queries Results of Task 3

To overcome this glitch, we divided this query into multiple sections of federated query and executed each section of the federated query individually i.e. one federated query at a time. By doing this we successfully got the expected results as illustrated in the figure 12.2.1, Figure 12.2.2 and Figure 12.2.3.
Here in the output the value Zero is obtained if there are no botanical garden or art gallery is found in that Wikidata city.

QUERY RESULTS

Table | Raw Response

Showing 1 to 9 of 9 entries

Search: Show 50 entries

| | cities | wiki_cities | number_of_botanical_garden |
|---|---|---|---|
| 1 | this:Beijing | wd:Q956 | "1" |
| 2 | this:Birmingham | wd:Q2256 | "2" |
| 3 | this:Changsha | wd:Q174091 | "0" |
| 4 | this:Chengdu | wd:Q30002 | "0" |
| 5 | this:Delhi | wd:Q1353 | "0" |
| 6 | this:London | wd:Q84 | "2" |
| 7 | this:Manchester | wd:Q18125 | "2" |
| 8 | this:Mumbai | wd:Q1156 | "0" |
| 9 | this:Mysore | wd:Q10086 | "0" |

Showing 1 to 9 of 9 entries

**Figure 12.2.1: Federated Query output for Number of botanical garden in the cities**

QUERY RESULTS

Table | Raw Response

Showing 1 to 9 of 9 entries

Search: Show 50 entries

| | cities | wiki_cities | number_of_museum |
|---|---|---|---|
| 1 | this:Beijing | wd:Q956 | "14" |
| 2 | this:Birmingham | wd:Q2256 | "10" |
| 3 | this:Changsha | wd:Q174091 | "0" |
| 4 | this:Chengdu | wd:Q30002 | "4" |
| 5 | this:Delhi | wd:Q1353 | "11" |
| 6 | this:London | wd:Q84 | "111" |
| 7 | this:Manchester | wd:Q18125 | "19" |
| 8 | this:Mumbai | wd:Q1156 | "1" |
| 9 | this:Mysore | wd:Q10086 | "0" |

Showing 1 to 9 of 9 entries

**Figure 12.2.2: Federated query output for Number of museum in the cities**

QUERY RESULTS

Table | Raw Response

Showing 1 to 9 of 9 entries

Search: Show 50 entries

| | cities | wiki_cities | number_of_art_gallery |
|---|---|---|---|
| 1 | this:Beijing | wd:Q956 | "0" |
| 2 | this:Birmingham | wd:Q2256 | "0" |
| 3 | this:Changsha | wd:Q174091 | "0" |
| 4 | this:Chengdu | wd:Q30002 | "0" |
| 5 | this:Delhi | wd:Q1353 | "1" |
| 6 | this:London | wd:Q84 | "45" |
| 7 | this:Manchester | wd:Q18125 | "0" |
| 8 | this:Mumbai | wd:Q1156 | "1" |
| 9 | this:Mysore | wd:Q10086 | "0" |

Showing 1 to 9 of 9 entries

**Figure 12.2.3: Federated query output for Number of Art Gallery in the city.**

# VII. Task3: Advantages and Disadvantages of Watt tour with External Knowledge Graph:

The tourism ontology which illustrated by Daramola, O et.al [8] has high detailed representation of all the aspects of tourism industry. Its far broader and deeper compared to watt tour ontology. For example, It provides series of classes for tourism ontology workflow which is not illustrated in watt tour ontology.
But the Tourism ontology [8] doesn't show the tour plans such as the cities covered and the places that are attracted to the individual persons. Our ontology offers something different in this aspect.
As suggested by Noy, et.al [9] on their research "There is no single correct ontology for any domain. Creative ontology design is the potential applications of the ontology and the designer's understanding and view of the domain will undoubtedly affect ontology design choices." [9]

Following are some of main advantages and disadvantages of linking our ontology to an external knowledge graph.

## 1) Advantages

### 1) Integrated access to data

The essential strength of Linking the ontology to External Data is its capacity to link geographically dispersed data and enable integrated data access. The navigation of information sources gets more refined because of this. [4]

### 2) Data Enrichment

Data in knowledge bases can now be easily supplemented using Linked Data to the external knowledge graph. Thanks to technological advancements, we have been able to enrich our data by simply linking to data that is already available somewhere on the Internet. We were able to avoid re-entering data and repeating our efforts. [5]

### 3) Independency from specific data format

Linking data has fundamentally altered the way we exchange, retrieve, and combine data. RDF is the universal and uniform data format for all material released as Linked Data on the Web. As a result, data mixing has become simple. [4]

### 4) Data sharing

Data sharing is now simple, which it was never previously. Data sharing has become more easier because of linked data technologies [5].

## 2) Disadvantages

1) **Missing semantics**

Link data lacks semantics, which is one of its main concerns. For example, consider the triple: http://example.org/abu rdf:type >. Based on this statement, abu' is a 'bank'. 'bank' can either mean a financial institution, or a riverbank. [4]

2**) Data quality**

Linked data suffers from several data quality problems for instance, issues, such as inconsistency, representational, accuracy, conciseness, the quality of the data, and the accuracy of the data.

3) **Social trust**

The positive side of Linked Data is that it is a community effort. Through community participation, the vision of Linked Data or Data Web will come to pass in the near future. Even though Linked Data is used in various applications, it still lacks social trust. The addition of provenance information to Linked Data will help to overcome this lack.[5]

4)**Data reuse**

The publication of linked data as part of the Linked Data Cloud (LDC) does not automatically make it usable. The data needs to be described. It should have appropriate metadata [3]

5**)Disruption if disconnected from the web.**

The data from the external knowledge graph will not be available in our ontology if it is not connected to the internet. The ontology will therefore be missing the external data which can be accessible only if it is connected to the web.

# VIII. References:


1. Deliyska, B. and Stoenchev, N., 2010, October. Ontology-Based Model Of Firm Competitiveness. In *AIP Conference Proceedings* (Vol. 1293, No. 1, pp. 271-278). American Institute of Physics.
2. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H. and Wroe, C., 2004, October. OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 63-81). Springer, Berlin, Heidelberg.
3. https://enterprise-knowledge.com/whats-the-difference-between-an-ontology-and-a-knowledge-graph/
4. Jeff, H and Dezao, S Ontology Instance Linking: Towards Interlinked Knowledge Graphs
5. Dutta, B. (2014). Symbiosis between Ontology and Linked Data. Librarian, vol. 21, no. 2, pp. 15- 24.
6. https://www.w3.org/TR/owl-time/#time:days
7. https://www.w3.org/TR/owl-time/#time:day
8. Daramola, O., Adigun, M.O. and Ayo, C.K., 2009. Building an ontology-based framework for tourism recommendation services. *Building an Ontology-Based Framework for Tourism Recommendation Services*, pp.135-147.
9. Noy, Natalya F., and Deborah L. McGuinness. "Ontology development 101: A guide to creating your first ontology." (2001).