SPARQL Lab Report

Timesheet

Exploration

Task	Time Spent (hh:mm)	Comments
Exploring predicates for continents	00:25	- Took some time to find the correct predicate for continents - Had to investigate sensible methods to remove sub continents and other entities which were classified as continent, but do not relate to the meaning of continent in Mondial.
Exploring predicates for countries	00:30	- Had many former countries, and unrecognised countries - Faced issues with namespaces and choice of predicates - Needed to investigate how to remove unrecognised and defunct countries.
Exploring predicates for cities	00:25	- Faced issues with namespaces and choice of predicates - Needed to get cities based on valid, current countries
Exploring predicates for provinces	01:25	- Could not retrieve provinces using the same predicates from cities and countries. - Had to further investigate how a province could be classified in dbpedia. - Some provinces included clerical regions, such as religious diocese, this required further investigation for removal.

Task	Time Spent (hh:mm)	Comments
Exploring predicates for organizations	01:20	- Dbpedia defines organizations differently to Mondial, organizations in dbpedia included schools and clubs - Research showed that organizations in Mondial are either political or economical organizations. This required further investigation on how these organizations can be specifically filtered.
Exploring predicates for languages	01:15	- Countries did not have one common property for spoken languages, this required further investigation of countries to identify the variations of how spoken languages are represented for each country.

Retrieval

Task	Time Spent (hh:mm)	Comments
Retrieving predicates for continents	00:45	- Most continents had predicates for Area. - Missing values for area was allowed, as I could not find a path to any other predicates which could represent area of continents.
Retrieving predicates for countries	01:25	- Countries did not use consistent predicates to respresent the same attribute. - Further work was needed to get possible candidate values for some attributes. - There were some missing values, in such instances a comprimise was made to either allow or fill-in a missing value.
Retrieving predicates for cities	01:55	- Faced some problems in retrieving a province for the city. - Some predicates were missing, e.g. population, latitude, longitude and elevation.

Task	Time Spent (hh:mm)	Comments
Retrieving predicates for provinces	02:00	- Most of the time was taken to find a way to get a capital city for provinces, escpecially where such values were missing, I had to decide on the best way to impute a value. - Took time to handle missing values for population and area. - Used existing filters from exploration to remove clerical/religious administrative regions.
Retrieving predicates for organizations	01:45	- Difficulty in retrieving formation/established dates, there wasn't a consistent way that organizations represented these values. - Information about the headquarter city/country/province were difficult to retrieve, most of it were missing and I did not have enough time to find alternative predicates that represent this information.
Retrieving predicates for languages	01:30	- Most of the time was spent on calculating the percentage of speakers. No langauges had a percentage of speakers, so it required a hacky way of calculating a percentage. - Some languages did not have a property for the number of speakers. I did not have enough time to find other properties which may refer to the number of speakers.

1 Continent

1.1 Exploration

```
SELECT DISTINCT ?continent
WHERE {
    ?continent rdf:type dbo:Continent.
    ?continent rdfs:label ?continentN.
    FILTER(lang(?continentN) = "en")
    BIND(STR(?continentN) AS ?cName)
```

```
    FILTER(?cName in ("North America", "South America", "Oceania", "Asia", "Europe", "Africa"))
}
```

For the Continent concept, the goal of the exploration was to identify the continents present in DBpedia and confirm their relevance to the Mondial database.

Retrieving all continent entities of the class `dbo:Continent` showed that DBpedia classified sub-continents and continental regions as part `dbo:Continent` ontology. Moreover, DBpedia holds multiple continents with the same name as separate continents, e.g. Australasia, and Oceania. Another conflict was that Mondial does not distinguish between North America or South America, rather, it's classified together as America. DBpedia did not present any results where both North America and South America were classified together as a continent.

Due to these reasons, filters on the continent name were used to retrieve URIs of continents similar to those stored in the Mondial database.

1.2 Retrieval

```
SELECT DISTINCT ?Name ?Area
WHERE {
    ?continent rdf:type dbo:Continent.
    ?continent rdfs:label ?continentN.
    FILTER(lang(?continentN) = "en")
    BIND(STR(?continentN) AS ?Name)
    FILTER(?Name in ("North America", "South America", "Oceania", "Asia", "Europe", "Africa"))
    OPTIONAL{?continent dbo:areaTotal ?AreaTotal}
    BIND(?AreaTotal / 1000000 AS ?Area) #Convert m^2 to km^2
}
```

The goal of the retrieval query was to build on top of the exploratory query to retrieve continent names and the area in squared kilometers, as seen in the Mondial database. From exploring the URI pages for all the continents, it was found that Europe did not have a predicate for the toal area, all the other continent had a predicate: textttdbo:areaTotal. This demonstrates one of the challenges of working with schema-less graph databases, where properties are not uniformly applied across all instances of a particular type (e.g., dbo:Continent). Such inconsistencies can limit the accuracy and comprehensiveness of the dataset. To handle this limitation, the retrieval query uses the OPTIONAL keyword, ensuring that the absence of dbo:areaTotal for Europe does not cause the query to omit true continents.

The query is effective to an extent for retreiving continents relevant to Mondial. It's able to retrieve both the continent name and the total area (where available), using strict filters and language constraints to ensure minimal and relevant data retrieval. Although it supresses the absence of continent area, further work could be done to sum up areas of countries within continents, to get a reasonable estimate. This does however mean that the query may become

more complex and less efficient, as it would require the traversal of additional relationships and the addition of more triple patterns. It would also require more time to investigate a path between continent and country.

2 Country

2.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT
    ?country
WHERE {
    ?country rdf:type dbo:Country.
    ?country dcterms:subject dbc:Member_states_of_the_United_Nations.
}
```

The goal of this query was to identify and retrieve country URIs in DBpedia and ensure alignment with those in Mondial. When retrieving all entities of the dbo:Country class, it was found that it included relevant countries, but also defunct and unrecognised countries. A suitable approach to remove incorrect countries was to find countries which are members of the UN. This would ensure that defunct or unrecognised countries would not be retrieved. A limitation of this however, was that in Mondial overseas territories (e.g. Réunion) and administrative regions (e.g. Hong Kong) of countries are classed as a separate country. Due to time constraints retrieving countries based on UN membership was the most viable option.

2.2 Retrieval

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT
    ?Name
    COALESCE(STR(?isoCode), ?Name) AS ?Code
    (SAMPLE(?capitalCity) AS ?Capital)
    (SAMPLE(?ProvinceValue) AS ?Province)
    (xsd:integer(COALESCE(?cPop, ?popEst, ?popCen)) AS ?Population)
    (xsd:integer(SAMPLE(?cArea) / 1000) AS ?Area)
WHERE {
    ?country rdf:type dbo:Country.
    ?country dcterms:subject dbc:Member_states_of_the_United_Nations.
    ?country rdfs:label ?countryN.
    OPTIONAL{?country dbo:iso31661Code ?isoCode.}
    FILTER(lang(?countryN) = "en")
    BIND(STR(?countryN) AS ?Name)
    FILTER(?Name != "Member states of the United Nations")
    OPTIONAL{?country dbo:populationTotal ?cPop}
    OPTIONAL{?country dbp:populationEstimate ?popEst}
    OPTIONAL{?country dbp:populationCensus ?popCen}
    OPTIONAL{?country dbo:area ?cArea}
```

```
    {
        ?country dbo:capital ?capitalC.
    } UNION {
        ?country dbp:capital ?capitalC.
    }
    ?capitalC rdfs:label ?capitalCN.
    FILTER(lang(?capitalCN) = "en")
    BIND(STR(?capitalCN) AS ?capitalCity)
    OPTIONAL{
        ?capitalC dbo:subdivision ?capitalSubdivisionEntity.
        ?capitalSubdivisionEntity rdfs:label ?capitalSubdivisionN.
        FILTER(lang(?capitalSubdivisionN) = "en")
        BIND(STR(?capitalSubdivisionN) AS ?capitalSubdivision)}
    BIND(COALESCE(?capitalSubdivision, ?Name) AS ?ProvinceValue)
}
GROUP BY ?Name ?isoCode ?cPop ?popEst ?popCen
```

The retrieval query builds on the exploratory query to extract key country attributes relevant to the Mondial database: Name, code, capital, province of the capital, population, and area.

To account for graph database inconsistencies, the query handles cases where certain properties are missing. For instance, while dbo:capital is used to identify the capital city for many countries, some rely on dbp:capital (e.g. Denmark), so a union is applied to retrieve capital names consistently. Similarly, ISO codes (?isoCode) are not available for all countries; in such cases, the country name is used as a fallback to maintain data completeness. Area values are also not consistently identifiable, and the query accounts for their absence without omitting the corresponding countries.

Population data was also missing for some countries, in such instances, estimates and census data were used. These various predicates were used to retrieve potential population values and coalesced to minimise missing data. Some predicates also had multiple values for population, so aggregation and sampling were used to prevent multiple rows of the same country.

By grouping data and using functions like SAMPLE, the query efficiently handles duplicates and inconsistencies. The query is effective to an extent where it uses various candidate values for missing populations and country codes. However from a scalability perspective, the inclusion of multiple OPTIONAL clauses and unions introduces complexity, potentially increasing execution time as the number of triples grows.

3 City

3.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
```

```
SELECT DISTINCT ?country ?city
WHERE {
    ?country rdf:type dbo:Country;
        dcterms:subject dbc:Member_states_of_the_United_Nations.
    ?city rdf:type dbo:City;
        dbo:country ?country.
}
```

The goal of this query was to identify city entities in DBpedia and establish their association with countries, and ensuring relevance to the Mondial database. By limiting countries to UN member countries, the query excludes cities located in defunct or unrecognised countries. Each city is linked to a country through the dbo:country predicate, ensuring proper hierarchical alignment.

A limitation of this approach is that cities in overseas territories (e.g. Réunion) or administrative regions (e.g. Hong Kong) are omitted, potentially leading to incomplete coverage. Additionally, variations in how DBpedia classifies cities and their hierarchical relations with countries may introduce inconsistencies with how the Mondial database defines cities. However, this approach provides a robust starting point for retrieving cities relevant to the Mondial database while maintaining efficiency by focusing only on recognised UN member countries.

3.2 Retrieval

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?Name ?Country SAMPLE(?ProvinceValue) AS ?Province
?Population SAMPLE(?Lat) AS ?Latitude
SAMPLE(?Long) AS ?Longitude SAMPLE(?Elev) AS ?Elevation
WHERE {
    ?countryN rdf:type dbo:Country;
        dcterms:subject dbc:Member_states_of_the_United_Nations;
        rdfs:label ?countryLbl.
    OPTIONAL{?countryN dbo:iso31661Code ?isoCode.}
    ?cityN rdf:type dbo:City;
        dbo:country ?countryN;
        rdfs:label ?cityLbl.
    FILTER(lang(?countryLbl) = "en" && lang(?cityLbl) = "en")
    BIND(COALESCE(STR(?isoCode), STR(?countryLbl)) AS ?Country)
    BIND(STR(?cityLbl) AS ?Name)
    OPTIONAL{
        ?cityN dbo:subdivision ?subDiv.
        ?subDiv rdfs:label ?subDivLbl .
        FILTER(lang(?subDivLbl) = "en")
        BIND(STR(?subDivLbl) AS ?citySubDiv)}
    BIND(COALESCE(?citySubDiv, ?Country) AS ?ProvinceValue)
    OPTIONAL{?cityN geo:lat ?Lat}
    OPTIONAL{?cityN geo:long ?Long}
    OPTIONAL{?cityN dbp:elevationM ?Elev}
```

```
    OPTIONAL{?cityN dbp:populationTotal ?Population}
}
GROUP BY ?Country ?Name ?Population
```

The retrieval query involved finding city name, country, province of the city, population, longitude, latitude and elevation.

The query builds on the exploration query, ensuring that cities are part of UN countries, ensuring that the countries column would not have missing values. The query uses existing statements from previous concept queries to hande missing values. One notable feature of the query is the use of SAMPLE for handling optional data, such as the province, latitude, longitude, elevation, and population. This ensures that the query returns only one value per city for each attribute, even when multiple values exist in DBpedia.

The query is effective to an extent as it is able to retrieve known cities such as New York, Manchester, Paris, etc. and can handle missing values such as population, and geographical properties, some of which are also missing in Mondial. However, as with most concepts in DBpedia, the definition of cities differs to that of Mondial. The predicate dbo:City includes a wider variety of urban settlements, ranging from larger muncipalities to smaller towns, which are not necessarily considered cities in the Mondial database. Further investigation could have been carried out to identify patterns that filters out smaller towns and settlements.

The inclusion of multiple optional statements can increase the complexity. As more optional relationships (e.g., subdivisions or population estimates) are added, the query may take longer to execute, especially on larger datasets, since it needs to explore multiple potential connections for each city. This can slow down performance if the dataset is large, as each city is being linked to multiple properties.

4 Province

4.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?Province
WHERE {
    ?Province a dbo:AdministrativeRegion .
    FILTER NOT EXISTS {
        ?Province a dbo:ClericalAdministrativeRegion .
    }
}
```

4.2 Retrieval

```
PREFIX dcterms: <http://purl.org/dc/terms/>
```

```
SELECT DISTINCT ?Name ?Country xsd:integer(?population) AS ?Population
    xsd:integer(SAMPLE(?totalArea) / 1000) AS ?Area ?Capital ?Name AS ?CapProv
WHERE {
    ?countryP rdf:type dbo:Country;
        dcterms:subject dbc:Member_states_of_the_United_Nations;
        rdfs:label ?countryName.
    OPTIONAL{?countryP dbo:iso31661Code ?isoCode}
    ?region a dbo:AdministrativeRegion;
        dbo:country ?countryP;
        rdfs:label ?regionName.
    FILTER NOT EXISTS {
        ?region a dbo:ClericalAdministrativeRegion.
    }
    OPTIONAL{?region dbo:areaTotal ?totalArea}
    OPTIONAL{?region dbo:populationTotal ?population}
    OPTIONAL{?region dbo:capital ?capitalCity.
        ?capitalCity rdfs:label ?capitalLabel.
        FILTER(lang(?capitalLabel) = "en")}
    OPTIONAL{?region dbo:largestCity ?largestCity.
        ?largestCity rdfs:label ?largestCityLabel.
        FILTER(lang(?largestCityLabel) = "en")}

    FILTER(lang(?countryName) = "en" && lang(?regionName) = "en")
    BIND(COALESCE(STR(?isoCode), STR(?countryName)) AS ?Country)
    BIND(STR(?regionName) AS ?Name)
    BIND(COALESCE(STR(?capitalLabel), STR(?largestCityLabel)) AS ?Capital)
}
GROUP BY ?Name ?Country ?population ?Capital
```

5 Organization

5.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?Organization
WHERE {
    ?Organization rdf:type dbo:Organisation;
        rdfs:label ?orgName;
        dcterms:subject ?category .
    FILTER(?category IN (
        dbc:Political_organizations, dbc:International_political_organizations,
        dbc:Economic_organizations, dbc:Trade_associations,
        dbc:Intergovernmental_organizations, dbc:International_economic_organizations,
        dbc:United_Nations))
    FILTER NOT EXISTS{
        ?Organization dcterms:subject ?excludeCategory.
        ?excludeCategory rdfs:label ?excludeLabel.
        FILTER(CONTAINS(LCASE(?excludeLabel), "school") ||
        CONTAINS(LCASE(?excludeLabel), "club"))
```

```
    }
}
```

5.2 Retrieval

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?Abbreviation ?Name
    SAMPLE(?HQCity) AS ?City SAMPLE(?HQCountry) AS ?Country
    SAMPLE(?HQProvince) AS ?Province SAMPLE(?Established) AS ?Established
WHERE {
    ?org rdf:type dbo:Organisation ;
        rdfs:label ?orgName .
    ?org dcterms:subject ?category .
    FILTER(?category IN (
        dbc:Political_organizations, dbc:International_political_organizations,
        dbc:Economic_organizations, dbc:Trade_associations,
        dbc:Intergovernmental_organizations, dbc:International_economic_organizations,
        dbc:United_Nations))
    FILTER NOT EXISTS{
        ?org dcterms:subject ?excludeCategory .
        ?excludeCategory rdfs:label ?excludeLabel .
        FILTER(CONTAINS(LCASE(?excludeLabel), "school") ||
            CONTAINS(LCASE(?excludeLabel), "club"))}
    OPTIONAL{?org dbp:abbreviation ?abbreviation.}
    OPTIONAL{?org dbp:nickname ?nickname.}
    OPTIONAL{?org dbp:establishedDate ?establishedDate. }
    OPTIONAL {?org dbp:formation ?formation .}
    OPTIONAL{?org dbo:foundingYear ?established.}
    OPTIONAL{?org dbo:headquarter ?hq .
        { ?hq rdf:type dbo:City .
            ?hq rdfs:label ?hqLabel . FILTER(lang(?hqLabel) = "en")
            OPTIONAL{?hq dbo:country ?hqCountry .
                ?hqCountry rdfs:label ?hqCountryName .
                OPTIONAL{?hqCountry dbo:iso31661Code ?isoCode}
                FILTER(lang(?hqCountryName) = "en") }
            OPTIONAL{?hq dbo:subdivision ?hqRegion .
                ?hqRegion rdfs:label ?hqRegionName .
                FILTER(lang(?hqRegionName) = "en") }
            OPTIONAL{?hq rdfs:label ?hqCityName .
                FILTER(lang(?hqCityName) = "en") }
        } UNION { ?hq rdf:type dbo:Country .
            OPTIONAL{?hq dbo:iso31661Code ?isoCode}
            ?hq rdfs:label ?hqCountryName .
            FILTER(lang(?hqCountryName) = "en")}}
    FILTER(lang(?orgName) = "en")
    BIND(STR(?orgName) AS ?Name)
    BIND(COALESCE(STR(?abbreviation), STR(?nickname)) AS ?Abbreviation)
    BIND(COALESCE(STR(?hqCityName)) AS ?HQCity)
    BIND(COALESCE(STR(?hqRegionName)) AS ?HQProvince)
    BIND(COALESCE(STR(?isoCode), STR(?hqCountryName), STR(?hqLabel)) AS ?HQCountry)
```

```
    BIND(COALESCE(STR(?establishedDate), STR(?formation), STR(?established)) AS ?Established)
}
GROUP BY ?Name ?Abbreviation
```

6 Language

6.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?country ?language
WHERE {
    ?country rdf:type dbo:Country ;
        dcterms:subject dbc:Member_states_of_the_United_Nations.
    {
        ?country dbo:language ?language .
    } UNION {
        ?country dbo:officialLanguage ?language .
    }
}
```

Some countries have a language (e.g. Malaysia) property whilst some have an officialLanguage (e.g. South Africa) property.

6.2 Retrieval

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?Country ?Language
    ((SAMPLE(?LSpeakers) / ?Population) * 100) AS ?Percentage
WHERE {
    ?country rdf:type dbo:Country ;
        dcterms:subject dbc:Member_states_of_the_United_Nations ;
        rdfs:label ?countryName .
    OPTIONAL{?country dbo:iso31661Code ?isoCode}
    OPTIONAL{?country dbo:populationTotal ?population.}
    {
        ?country dbo:language ?language .
        ?language rdfs:label ?languageLabel .
        OPTIONAL{?language dbp:speakers ?speakers.}
    } UNION {
        ?country dbo:officialLanguage ?language .
        ?language rdfs:label ?languageLabel .
        OPTIONAL{?language dbp:speakers ?speakers.}
    }
    FILTER(lang(?languageLabel) = "en")
    FILTER(lang(?countryName) = "en")
    FILTER(datatype(?speakers) = xsd:integer || datatype(?speakers) = xsd:decimal)
    BIND(COALESCE(STR(?isoCode), STR(?countryName)) AS ?Country)
    BIND(STR(?languageLabel) AS ?Language)
    BIND(xsd:integer(?population) AS ?Population)
    BIND(xsd:integer(?speakers) AS ?LSpeakers)
}
GROUP BY ?Country ?Language ?Population
```

7 Ethnic Group

7.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?country ?ethnicGroup
WHERE {
    ?country rdf:type dbo:Country ;
        dcterms:subject dbc:Member_states_of_the_United_Nations ;
        rdfs:label ?countryName .
    ?country dbo:ethnicGroup ?ethnicGroup.
    ?ethnicGroup rdf:type dbo:EthnicGroup.
}
```

Some ethnic groups included religions.

7.2 Retrieval

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?Country ?Name
SAMPLE(xsd:decimal(?pop))/MAX(xsd:decimal(?population)) AS ?Percentage
WHERE {
    ?countryE rdf:type dbo:Country ;
    dcterms:subject dbc:Member_states_of_the_United_Nations ;
    rdfs:label ?countryName .
    OPTIONAL{?countryE dbo:iso31661Code ?isoCode}
    OPTIONAL{?countryE dbo:populationTotal ?population.}

    ?countryE dbo:ethnicGroup ?ethnicGroup.
    ?ethnicGroup rdf:type dbo:EthnicGroup;
        rdfs:label ?egLbl.
    FILTER(lang(?egLbl) = "en")
    FILTER(lang(?countryName) = "en")
    {
        ?ethnicGroup dbp:population ?pop.
    } UNION {
        ?ethnicGroup dbo:totalPopulation ?pop.
    }
    BIND(COALESCE(STR(?isoCode), STR(?countryName)) AS ?Country)
    BIND(STR(?egLbl) AS ?Name)
} GROUP BY ?Country ?Name
```

8 Religion

8.1 Exploration

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?country ?religion
WHERE {
```

```
    ?country rdf:type dbo:Country ;
        dcterms:subject dbc:Member_states_of_the_United_Nations ;
        rdfs:label ?countryName .
    ?country dbo:religion ?religion.
}
```