# Life Expectancy and Health Indicators Analysis

(Semantic Technologies)

## *Author: Abhishek Bargujar*

***Professor: Dr. Franconi Enrico***

|  |
| --- |
| Table of Contents |
| 1. Introduction |
| 2. Data Sources |
| 3. SQL Schema and Data Import |
| 4. RDF and Ontology |
| 5. Visualization |
| 6. Flask Application for SPARQL Queries |
| 7. Conclusion |
| 8. Appendices |

# 1. Introduction

This report provides an in-depth analysis of life expectancy and various health indicators using data sourced from the World Health Organization and Kaggle. The project aims to create a comprehensive data processing pipeline that transforms the raw data into both SQL and RDF formats, facilitating efficient querying and analysis.

By leveraging structured and semantic data, this report explores the integration of different data sources, processing techniques, and tools to generate meaningful insights. The data is standardized and processed using SQL for relational storage and RDF for semantic querying, allowing for flexible data interaction and analysis.

A significant part of the project involves using Ontop, a powerful OBDA (Ontology-Based Data Access) tool, to generate RDF mappings and materialize data into RDF format. Ontop enables the seamless integration of SQL databases with RDF, supporting complex queries and enhancing data interoperability.

The project also includes the development of a web-based application using Flask, which provides a user-friendly interface for performing SPARQL queries on the RDF datasets. This application demonstrates the practical utility of combining traditional relational databases with modern semantic technologies to enhance data accessibility and analysis capabilities.

### DATA SOURCES

1. **WHO National Life Expectancy Data (who\_life\_exp.csv)**
   * **Source:** [Kaggle Life Expectancy Dataset](https://www.kaggle.com/datasets/mmattson/who-national-life-expectancy)
   * **Description:** This dataset contains information about life expectancy at the national level. It includes various health indicators such as life expectancy, adult mortality, infant deaths, alcohol consumption, BMI, and many other factors influencing life expectancy.
2. **Relay World Health Statistics (RELAY\_WHS.csv)**
   * **Source:** [WHO Country Data](https://data.who.int/countries/380)
   * **Description:** This dataset includes various health statistics for different countries. It covers a range of indicators, such as healthcare access, disease prevalence, and demographic data.
3. **Modified Relay World Health Statistics (relay\_whs\_modified.csv)**
   * **Source:** Generated from the original RELAY\_WHS.csv
   * **Description:** This file is a modified version of the RELAY\_WHS.csv dataset. It includes a combined indicator column created by merging IND\_NAME and DIM\_SEX columns for easier querying and analysis.

### Modification Process

The modification process involved transforming the RELAY\_WHS.csv file to create relay\_whs\_modified.csv. This transformation was done using the provided Python script (whl\_relay.py). Below is a detailed explanation of the script and the modification logic.

### 3. SQL Schema and Data Import

#### 3.1 SQL Tables

The project utilizes two main SQL tables to store and manage the life expectancy and health indicators data:

1. **life\_expectancy Table**
   * **Description:** This table stores detailed life expectancy and various health-related metrics for different countries. It includes fields such as country name, country code, region, year, and numerous health indicators like adult mortality, infant deaths, alcohol consumption, BMI, hepatitis, measles, and access to basic water, among others.
   * **Key Fields:**
     + Country: Name of the country.
     + country\_code: ISO code of the country.
     + region: Region of the country.
     + Year: Year of the data entry.
     + Life\_Expectancy: Life expectancy in years.
     + Adult\_Mortality, Infant\_Deaths, bmi, alcohol, etc.: Various health indicators.
2. **relay\_whs Table**
   * **Description:** This table stores health statistics from the World Health Organization (WHO) for different countries. It includes fields for the country, indicator name, sex dimension, value of the indicator, and a combined indicator column for easier querying.
   * **Key Fields:**
     + Country: Name of the country.
     + IND\_NAME: Name of the health indicator.
     + DIM\_SEX: Sex dimension (e.g., Male, Female).
     + Value: Value of the health indicator.
     + Indicator: A combined column created by merging IND\_NAME and DIM\_SEX for unique identification.

#### 3.2 Data Import Instructions

To populate these tables with data from the provided CSV files, follow these steps:

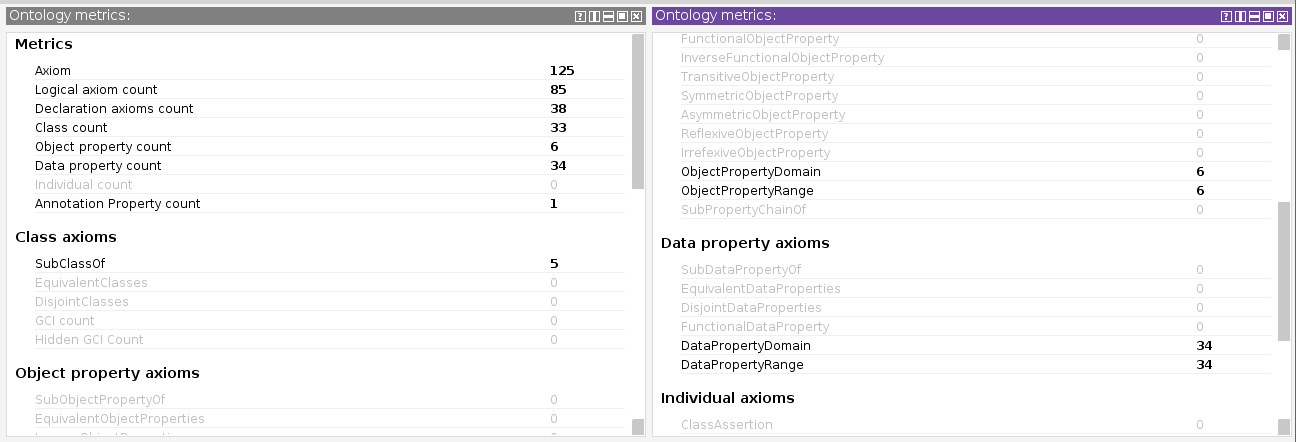
1. **Ensure Accessibility:** Make sure the CSV files (who\_life\_exp.csv and RELAY\_WHS.csv) are accessible from your system.
2. **Data Import Commands:** Use the PostgreSQL psql client to execute the data import commands. The \copy command is used to copy data from the CSV files into the respective tables.
3. **Combining Columns:** For the relay\_whs table, an additional step is required to create the Indicator column by concatenating IND\_NAME and DIM\_SEX fields.

#### Ontology

The ontology developed for this project provides a structured framework for representing and analysing life expectancy and health indicators data. It uses standard vocabularies and namespaces to organize and interlink the data, defining entities, properties, and relationships between them. This ontology enables detailed analyses and meaningful insights into the domain. In the following sections, we explore each part of the ontology separately.

#### 0.1 Life Expectancy Ontology

The main ontology, referred to as Life Expectancy, encompasses the life expectancy and health indicators graph. This section analyses the structure of the dataset, its namespaces, classes, properties, and their relationships. The RDF dataset under analysis is referred to as the Life Expectancy dataset. It contains information related to health and life expectancy data and is accessible through the base URI provided in the project setup.



##### Dataset Description

The dataset is defined as an OWL ontology with the following properties:

* **rdf**

: Specifies that the dataset is an ontology.

* **dcterms**

: Indicates the last modification date of the dataset.

* **rdfs**

: Provides a comment stating that the dataset was created by the project team.

* **owl**

: Imports multiple ontologies, including specifications, schemas, and vocabularies related to health and demographics.

* **owl**

: Indicates the version of the dataset.

##### Classes and Properties

The ontology defines various classes, object properties, and datatype properties. Some notable classes include:

* : A class representing life expectancy data.
* : A class representing different health indicators.

The dataset also defines various properties, including:

* : Links a country to its life expectancy data.
* : Links a country to various health indicators.

##### Limitations and Recommendations

To further improve the dataset, it is recommended to provide more extensive documentation, including examples and usage guidelines for each class and property. Additionally, clarifying the relationships between imported ontologies and the dataset's classes and properties would enhance understanding and interoperability with other related datasets.

#### 0.2 Health Indicators Ontology

The dataset contains information about various health indicators related to the project. The analysis focuses on the structure of the dataset, including its namespaces, properties, and the information provided for each indicator. The RDF dataset under investigation is a collection of information about health indicators serialized in RDF. The dataset's main resource is identified by the base URI provided in the project setup.

##### Dataset Description

The dataset is classified as a dctype

and has various properties:

* **dcterms**

: Indicates the last modification date of the dataset.

* **schema**

: Specifies the licensing terms for the dataset.

* **dcterms**

: Provides a brief overview of the dataset.

* **owl**

: Indicates that the dataset imports other ontologies.

* **owl**

: Mentions the version of the dataset and its creator.

##### Classes and Properties

The dataset provides information about various health indicators. Each indicator is represented as a schema

and has the following properties:

* **rdfs**

: Provides the name or label of the indicator.

* **schema**

: Specifies the description of the indicator.

##### Limitations and Recommendations

To enhance the usability of the dataset, it is recommended to provide more detailed properties for each health indicator, such as their definitions, measurement units, and relevance. Expanding the dataset to include more health indicators would increase its value and provide a more comprehensive resource for research.

#### 0.3 Ontology Metrics

The ontology metrics provide a quantitative overview of the ontology's structure and complexity. These metrics include the number of axioms, classes, properties, and individuals defined in the ontology. The figure below illustrates the detailed metrics of the ontology.

The metrics provide insights into the structure and complexity of the ontology, highlighting the following key points:

* **Axiom Count**: The total number of axioms defined in the ontology.
* **Class Count**: The number of classes present in the ontology.
* **Object Property Count**: The number of object properties, which define relationships between classes.
* **Data Property Count**: The number of data properties, which define attributes of classes.
* **Individual Count**: The number of individual instances defined in the ontology.

These metrics help in understanding the richness and depth of the ontology, providing a snapshot of its structural elements.

### 0.4 Ontology Overview

The ontology screenshot provides an overview of the entire ontology, showcasing the various classes, properties, and relationships defined within the dataset. This visual representation helps in understanding the hierarchical structure and the interconnections between different entities in the ontology.

The ontology is structured into several main components:

* **Classes**: Represent the main entities within the ontology, such as countries, health indicators, and life expectancy data.
* **Object Properties**: Define relationships between different classes, such as a country having a specific life expectancy or a health indicator.
* **Data Properties**: Define attributes of the classes, such as the value of life expectancy or the measurement of a health indicator.

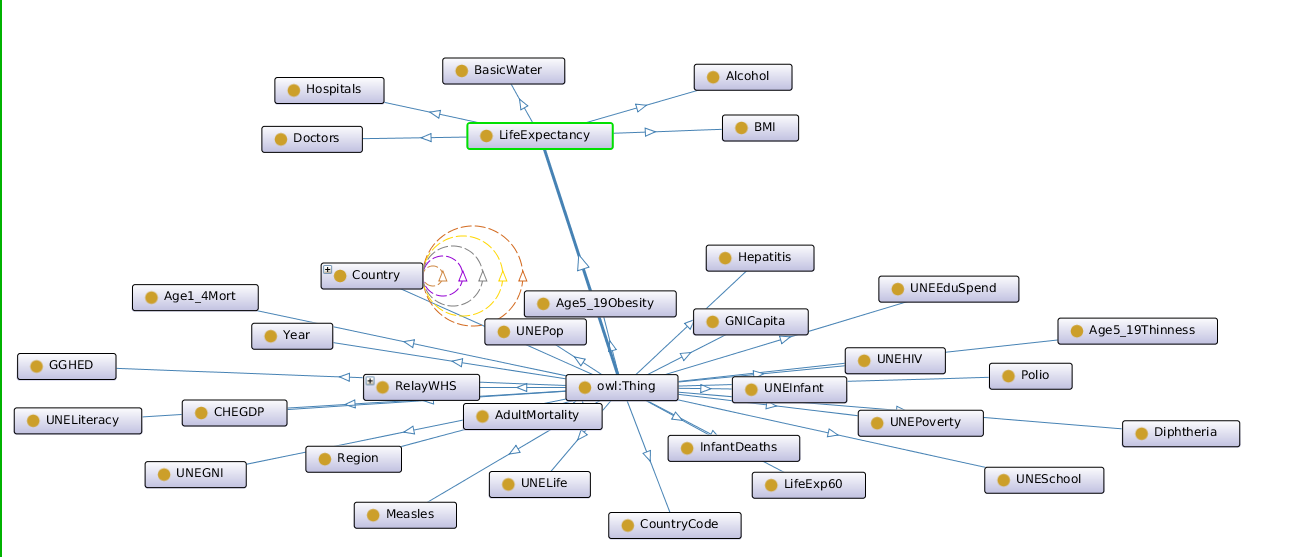
This comprehensive structure allows for detailed analysis and querying of the dataset, facilitating the extraction of meaningful insights and data-driven decision-making.

### 0.5 RDF and Ontology Integration

The RDF and ontology integration process involves transforming the life expectancy and health indicators data into RDF format, which facilitates semantic querying and interoperability. This process includes:

1. **Generating Direct Mapping Using Ontop CLI**: Creating the initial RDF mappings from the relational data.
2. **Converting OBDA Mapping to R2RML**: Ensuring the mappings conform to RDF standards.
3. **Materializing Data into RDF Format**: Finalizing the RDF datasets for use

**Visualizations:**

****

# 6. Flask Application for SPARQL Queries

#### 6.1 Overview

The project includes a Flask-based web application designed to provide a user-friendly interface for querying RDF datasets using SPARQL. This application allows users to interact with the life expectancy and health indicators data by submitting SPARQL queries through a web form and viewing the results directly on the webpage.

#### 6.2 Application Functionality

**Main Features:**

* **Web Interface:** The application features a simple and intuitive web interface where users can input their SPARQL queries.
* **Data Loading:** It loads the RDF datasets (life\_expectancy\_rdf\_dataset.ttl and relay\_whs\_rdf\_dataset.ttl) into memory for querying.
* **SPARQL Query Execution:** Upon receiving a query, the application executes it against the loaded RDF datasets.
* **Result Display:** The results of the SPARQL query are displayed in a tabular format on the web page.

#### 6.3 Technical Details

**Application Components:**

1. **Flask Framework:** The web application is built using Flask, a lightweight Python web framework that simplifies web development and enables the integration of various libraries and tools.
2. **RDFlib Library:** RDFlib is used to parse and query the RDF datasets. It supports SPARQL and provides a convenient interface for handling RDF data.

**Workflow:**

1. **Home Page:** The home page displays a form where users can enter their SPARQL queries.
2. **Form Submission:** When the form is submitted, the application processes the query and executes it against the RDF datasets.
3. **Query Execution:** The RDFlib library parses the RDF data and executes the SPARQL query.
4. **Result Rendering:** The results are rendered in an HTML table and displayed on the web page.

#### 6.4 Example Usage

**User Interaction:**

1. **Input Query:** The user enters a SPARQL query in the provided text area on the web page.
2. **Submit Query:** The user submits the query by clicking the submit button.
3. **View Results:** The application processes the query, executes it, and displays the results in a tabular format.



**SPARQL Query:**

In this section, we will execute a series of queries on the catalog ontology. Through this exercise, we aim to delve into the rich and varied information contained within the ontology, demonstrating the practical application of data retrieval and manipulation techniques.

* 1. Life Expectancy by Country and year

A screen shot of a computer

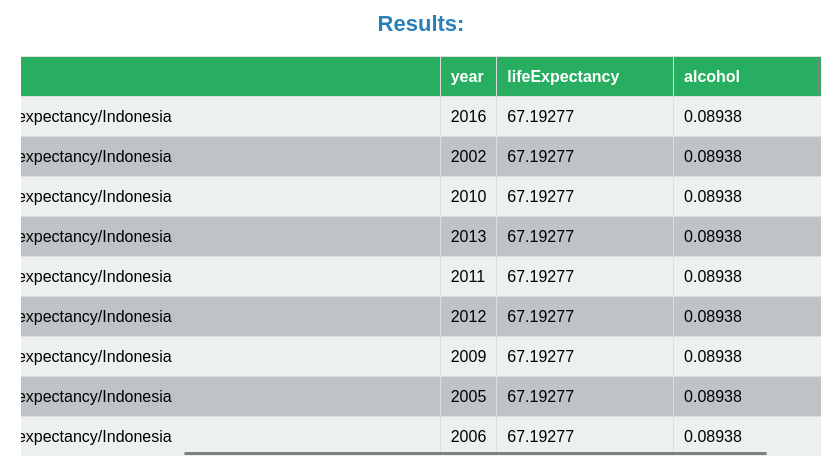
Description automatically generated



1. life\_expectancy\_values\_with\_associated\_countries\_and\_years\_and\_alcohol

A computer screen with white text

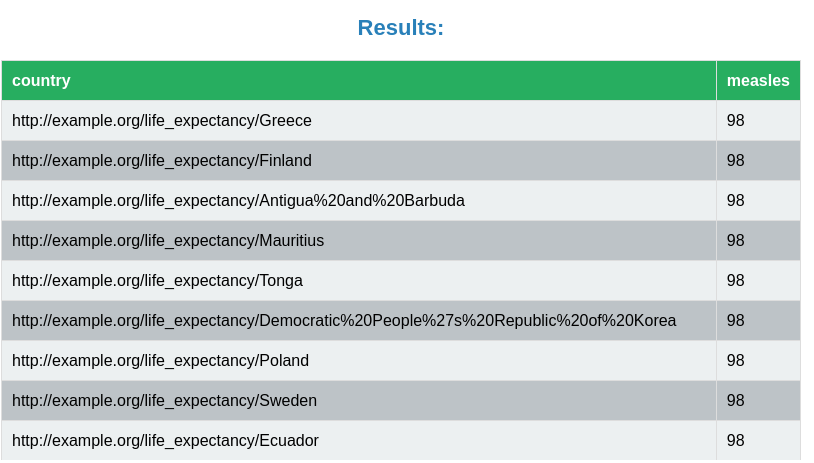
Description automatically generated



1. Retrieve\_Measles\_Cases\_by\_Country

A screen shot of a computer

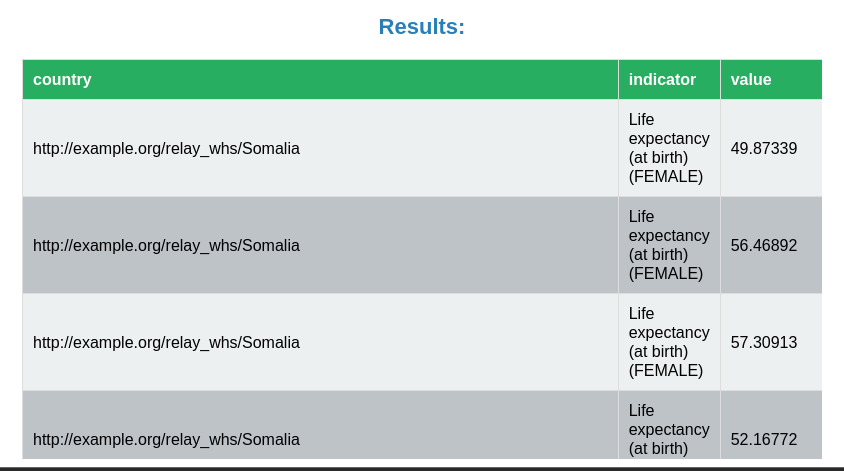
Description automatically generated



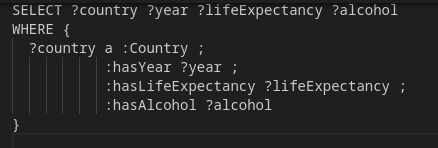
1. Indicator\_and\_their\_Values

A screen shot of a computer code

Description automatically generated



1. Life\_Expectancy\_Values\_With\_Associated\_Countries\_And\_Years\_And\_Alcohol



A table with numbers and numbers

Description automatically generated

1. Gni Per Capita by Country

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Hospitals\_by\_country

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### 7. Conclusion

In this project, we developed a robust and comprehensive system for analyzing life expectancy and various health indicators using datasets from the World Health Organization (WHO) and Kaggle. By transforming the raw data into both SQL and RDF formats, we enhanced the accessibility and interoperability of the data, enabling more sophisticated analyses and insights.

Our system encompasses multiple components:

1. **Data Integration and Transformation:**
   * We imported and processed life expectancy and health indicators data from CSV files into SQL databases. This step included cleaning and modifying the datasets to ensure consistency and accuracy, as well as creating combined indicator columns for improved data querying.
2. **Ontology and RDF Mapping:**
   * Using Ontop CLI, we generated RDF datasets from the SQL data, leveraging ontology-based data access (OBDA) mappings. This conversion facilitated semantic querying and provided a structured framework for representing complex health data relationships.
3. **Web Application Development:**
   * We developed a Flask-based web application that allows users to perform SPARQL queries on the RDF datasets through a user-friendly interface. The application loads the RDF data into memory, executes user-submitted queries, and displays the results in a comprehensible tabular format.

The integration of SQL and RDF technologies in this project demonstrates the power and flexibility of combining traditional relational databases with modern semantic web standards. The RDF datasets enhance data interoperability and facilitate complex queries that would be difficult to execute with SQL alone.

**Key Achievements:**

* **Enhanced Data Accessibility:** The transformation of data into RDF format, combined with a web-based SPARQL query interface, makes it easier for users to access and analyze complex health data.
* **Improved Data Interoperability:** The use of standardized ontologies ensures that our datasets can interoperate with other datasets and systems, fostering greater collaboration and data sharing.
* **Flexible Data Analysis:** By providing both SQL and RDF representations of the data, we offer flexible options for querying and analyzing the data, catering to different user needs and expertise levels.

In conclusion, this project showcases an effective approach to integrating, transforming, and analysing health data. By leveraging the strengths of SQL for structured data storage and RDF for semantic querying, we created a versatile system that can support a wide range of data analysis workflows. This approach can be extended to other domains, demonstrating its potential for broader applications in data science and analytics.

# 8. Appendices

Appendix A: Python Code for Data Processing

import pandas as pd

# Load the CSV file

relay\_whs\_df = pd.read\_csv('RELAY\_WHS.csv')

# Rename columns for relay\_whs

relay\_whs\_df.rename(columns={

'GEO\_NAME\_SHORT': 'Country',

'AMOUNT\_N': 'Value'

}, inplace=True)

# Create a combined indicator column

relay\_whs\_df['Indicator'] = relay\_whs\_df.apply(lambda row: f"{row['IND\_NAME']} ({row['DIM\_SEX']})", axis=1)

# Save the modified CSV

relay\_whs\_df[['Country', 'Indicator', 'Value']].to\_csv('relay\_whs\_modified.csv', index=False)

Appendix B: SQL Scripts

SQL Script for Creating the life\_expectancy Table:

DROP TABLE IF EXISTS life\_expectancy;

CREATE TABLE life\_expectancy (

Country VARCHAR(100),

country\_code VARCHAR(10),

region VARCHAR(100),

Year INT,

Life\_Expectancy FLOAT,

life\_exp60 FLOAT,

Adult\_Mortality FLOAT,

Infant\_Deaths FLOAT,

age1\_4mort FLOAT,

alcohol FLOAT,

bmi FLOAT,

age5\_19thinness FLOAT,

age5\_19obesity FLOAT,

hepatitis FLOAT,

measles FLOAT,

polio FLOAT,

diphtheria FLOAT,

basic\_water FLOAT,

doctors FLOAT,

hospitals FLOAT,

gni\_capita FLOAT,

gghe\_d FLOAT,

che\_gdp FLOAT,

une\_pop FLOAT,

une\_infant FLOAT,

une\_life FLOAT,

une\_hiv FLOAT,

une\_gni FLOAT,

une\_poverty FLOAT,

une\_edu\_spend FLOAT,

une\_literacy FLOAT,

une\_school FLOAT

);

SQL Script for Creating the relay\_whs Table:

DROP TABLE IF EXISTS relay\_whs;

CREATE TABLE relay\_whs (

Country VARCHAR(100),

Value FLOAT,

Indicator VARCHAR(200)

);

#### Appendix C: RDF and Ontology Example

**Example Ontology:**

@prefix : <http://www.example.org/ontology#> .

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

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

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:Country a rdf:Class ;

rdfs:label "Country" ;

rdfs:comment "A country entity." .

:LifeExpectancy a rdf:Property ;

rdfs:domain :Country ;

rdfs:range xsd:float ;

rdfs:label "Life Expectancy" ;

rdfs:comment "Life expectancy of the country." .

Example RDF Data:

@prefix : <http://www.example.org/ontology#> .

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

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:Country\_1 a :Country ;

:name "Country A" ;

:LifeExpectancy 75.5 .