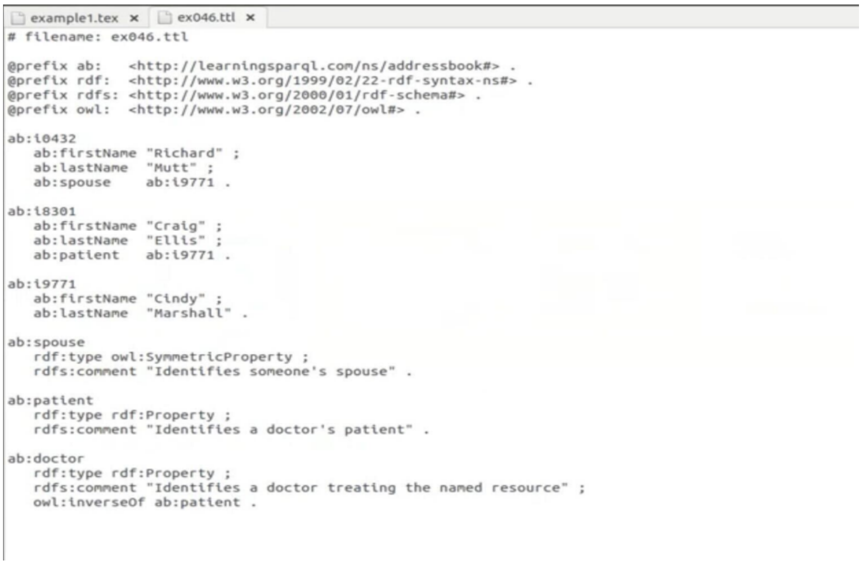
TR-102

MASTERING THE SEMANTIC WEB

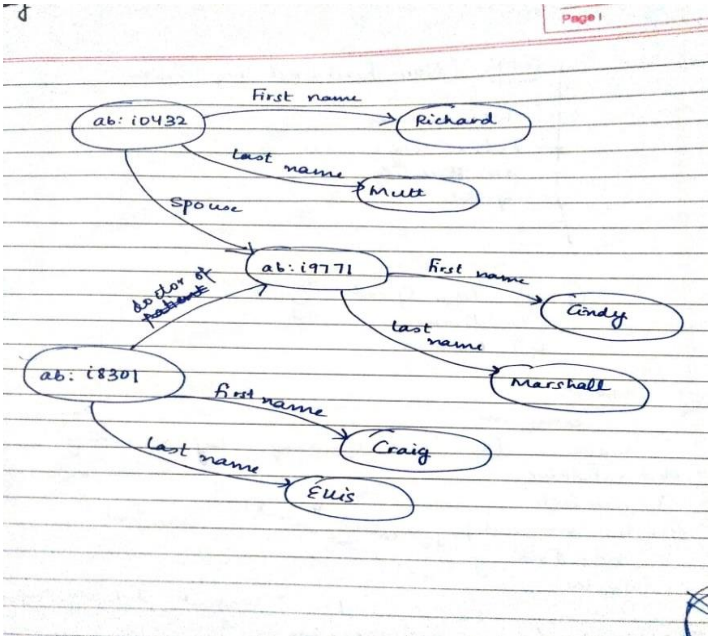
DAY-7

- **Task**: **Read the following turtle file and draw the RDF graph related to it.**



```
# filename: ex046.ttl

@prefix ab:   <http://learningsparql.com/ns/addressbook#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .

ab:i0432
   ab:firstName "Richard" ;
   ab:lastName  "Mutt" ;
   ab:spouse    ab:i9771 .

ab:i8301
   ab:firstName "Craig" ;
   ab:lastName  "Ellis" ;
   ab:patient   ab:i9771 .

ab:i9771
   ab:firstName "Cindy" ;
   ab:lastName  "Marshall" .

ab:spouse
   rdf:type owl:SymmetricProperty ;
   rdfs:comment "Identifies someone's spouse" .

ab:patient
   rdf:type rdf:Property ;
   rdfs:comment "Identifies a doctor's patient" .

ab:doctor
   rdf:type rdf:Property ;
   rdfs:comment "Identifies a doctor treating the named resource" ;
   owl:inverseOf ab:patient .
```

Ans:



❖ Introduction to Turtle files

Turtle (Terse RDF Triple Language) is a popular and user-friendly syntax for writing RDF (Resource Description Framework) data. It is designed to be more readable and writable for humans compared to other serialization formats like RDF/XML.

➢ Basics of a Turtle File

A Turtle file consists of statements that describe RDF triples. Each triple is composed of a subject, predicate, and object. Here's a basic structure:

- **Subject**: The resource being described.
- **Predicate**: The property or attribute of the subject.
- **Object**: The value or another resource that the predicate points to.

➢ Key Elements of a Turtle File

▪ Prefixes

a. **Purpose**: To simplify URIs, Turtle allows the use of prefixes, which are shorthand for longer URIs. This makes the file more readable.
b. **Syntax**: Prefixes are defined using the @prefix keyword, followed by a prefix label and the full URI enclosed in angle brackets (<>).
c. **Example**:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/> .
```

▪ Triples

a. **Structure**: Each triple consists of a subject, predicate, and object, ending with a period (.).
b. **Example**:

```
ex:book1 dc:title "Example Book" .
ex:book1 dc:creator "John Doe" .
```

▪ Comments

a. **Syntax**: Comments start with the # character and extend to the end of the line.
b. **Example**:

```
# This is a comment
```

➢ Detailed Example of a Turtle File

Below is a complete Turtle file example, illustrating the use of prefixes, triples, and comments:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/> .

# Define a book with a title and a creator
ex:book1 dc:title "The Catcher in the Rye" ;
        dc:creator "J.D. Salinger" .

# Define another book with a title and a creator
ex:book2 dc:title "1984" ;
        dc:creator "George Orwell" .
```

- **Understanding the Example**

a. **Prefixes**:

i. dc: is a prefix for Dublin Core elements, commonly used for metadata.
ii. ex: is a custom prefix for the example namespace.

b. **Triples**:

i. ex:book1 dc:title "The Catcher in the Rye": Describes the title of book1.
ii. ex:book1 dc:creator "J.D. Salinger": Describes the creator of book1.
iii. The same pattern is followed for ex:book2.

c. **Comments**:

i. Comments provide explanations or notes, which are ignored during processing.

➢ Advantages of Turtle

- **Readability**: Turtle's concise syntax makes it easier for humans to read and write compared to other RDF serialization formats.
- **Prefixes**: Simplify URIs, making the data more compact and easier to manage.
- **Support**: Widely supported by RDF tools and libraries, facilitating interoperability and data exchange.

Turtle files are an essential part of the RDF ecosystem, enabling the effective representation and exchange of semantic data on the web. By understanding the basics of Turtle syntax and structure, you can efficiently work with RDF data in a human-readable format.

❖ Turtle file depicting flight details

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

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

@prefix ex: <http://www.example.org> .

ex:Airline_1
```

```
    rdf:type ex:Airline;
    ex:name "IndiGo";
    ex:headquarters "Gurgaon";
    ex:iataCode "6E" .

ex:Airline_2
    rdf:type ex:Airline;
    ex:name "Air India";
    ex:headquarters "Delhi";
    ex:iataCode "AI" .

ex:Airline_3
    rdf:type ex:Airline;
    ex:name "AIX Connect";
    ex:headquarters "Bangalore";
    ex:iataCode "I5" .

ex:Airline_4
    rdf:type ex:Airline;
    ex:name "Akasa Air";
    ex:headquarters "Mumbai";
    ex:iataCode "QP" .


ex:flight_123
    rdf:type ex:flight;
    ex:flightNumber "6E 2193";
    ex:departureAirport ex:ABC;
    ex:arrivalAirport ex:DEF;
    ex:departureTime "2024-06-24 T19:30:00"^^xsd:dateTime;
    ex:arrivalTime "2024-06-24 T20:30:00"^^xsd:dateTime;
    ex:operatedBy ex:Airline_1 .

ex:ABC
    rdf:type ex:Airport;
    ex:name "Shaheed Bhagat Singh International Airport";
    ex:iataCode "IXC";
    ex:location "Chandigarh,Punjab" .

ex:DEF
    rdf:type ex:Airport;
    ex:name "Indira Gandhi International Airport";
    ex:iataCode "DEL";
    ex:location "New Delhi,Delhi" .
```