Ontalogies, R.DF and OWL

* Define Web Resource, URI, namespace.

=> Web Resource:

A Web Resource is anything that can be identified and referes as Web Page, a Fragment of an XML Document.

A Resource allows for the representation and description of virtually anything that can be identified.

=> URI:

URI stands for Uniform Resource Identifier which is string of character that are unambiguously identifies a perticula resources.

URIs can be URLs that can be accessed by Human or software agents.

URIs are used to uniquely identify elements like classes, properties and individuals.

=> Namespace :

Namespace are a way of avoid using Long URIs repeatedly.

This is especially useful in XML and RDF where element and properties might belong to different ontologies.

✳ RDF with Example :

=> RDF stands for Resource Description Framework which is used to represent information on the web.

It is particularly useful for expressing metadata about web resources.

=> RDF Syntax or Structure :

RDF is used Triplet Structure for describe the information.

(i) Subject: Resource or Thing being described.

(ii) Predicate: The Property or Relationship expresed about subject.

(iii) Object: Another Resources Related to the subject.

In RDF, Subject and Predicate are identified by URIs.

An RDF Graph is a visual representation of set of RDF Tuple where,

Node: Represent Subject or Object

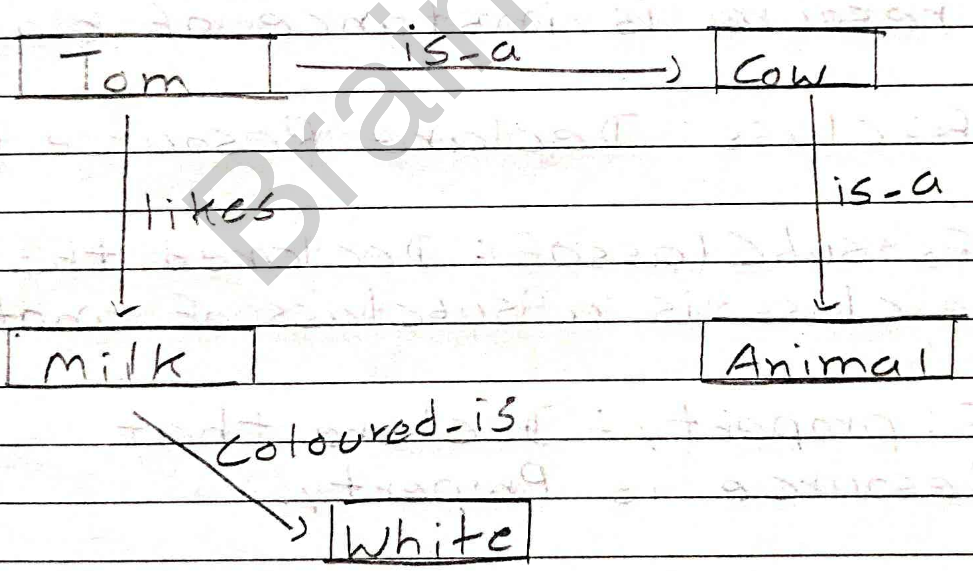
Edges: Represent Predicates

RDF is also use Schema and OWL to define more complex relationship between their Facts.

Ex. Tom is a Cow.
Cow is a Animal
Tow likes Milk.
Milk coloured is white.

=> For, Every statement,

Subject	Object	Predicate
Tom	Cow	is_a
Cow	Animal	is_a
Tom	Milk	likes
Milk	White	coloured_is



RDF Graph

* RDFS : RDF Schema

=) RDFS is a semantic extension of RDF that provides mechanism to describe the structure and constaint of RDF data.

It allows the define of classes, properties, and relationship between them.

=) Syntax of RDFs :

1 rdf : type : Used to state that a resource is instance of class

2 rdfs: class : Declare Resource is Class

3 rdfs : subClassof : Declared that one class is a subclass of another

4 rdf : property : Declare that Resource is Property.

5 rdfs : subPropertyof : Used to declared one property is a subproperty of another

6 rdfs : domain : Declare the Domain of property

7 'rdfs:range' : Declare the range of a property.

Ex. Library Management

Classes:
:Book , :Author

Properties: :WrittenBy

(1) Declare the Classes:

:Book rdf:type rdfs:Class.
:Author rdf:type rdfs:Class.

(2) Declare Property

:WrittenBy rdf:type rdf:property.
:WrittenBy rdfs:domain :Book.
:WrittenBy rdfs:range :Author.

(3) Add Instance:

:Book1 rdf:type:Book.
:Author1 rdf:type :Authour.
:Book1 :WrittenBy :Author1.

* OWL:

=> OWL stands For Web Ontology Language which is designed for creating and sharing ontologies on the sementic web.

It extends RDF and RDFS by adding more expressiveness by alowing them to define complex relationships.

OWL comes in three sublanguages

(i) OWL Lite: Simplified version For user needing to classification hierachies.

(ii) OWL DL: Ensuring all conclusions can be computed.

(iii) OWL Full: Allowing For Full RDF compatibility.

=> OWL Description Logic Notation:

1 Class Disjointness: Declares that two classes do not share any instances.

Ex. Classes : A and B => Disjoint

OWL Syntax : A owl:disjointWith :B

2 Functional Constraint: Declares that a property can have at most one value for subject.

Ex Property : 'hasSSN'
Constraint: Person can have only one 'SSN'

OWL Syntax :
:Person rdf:type owl:Class.
:hasSSN rdf:type owl:Datatype Property ;
rdf:type owl:Functional-Property.

3 Intentional class Definition: Defines a class by specifying and sufficient condition for membership.

Ex. Class: 'Parent'
Define: Person has at least one child.

OWL Syntax :
:Parent rdf:type owl:Class.
:hasChild rdf:type owl:Object-Property.

```
:Parent owl:equivalentClass
[
   rdf:type owl:Restriction;
   owl:onProperty :hasChild;
   owl:someValuesFrom :
        Person
].
```

4 Class Union : Create a new class that is union of two or more class.

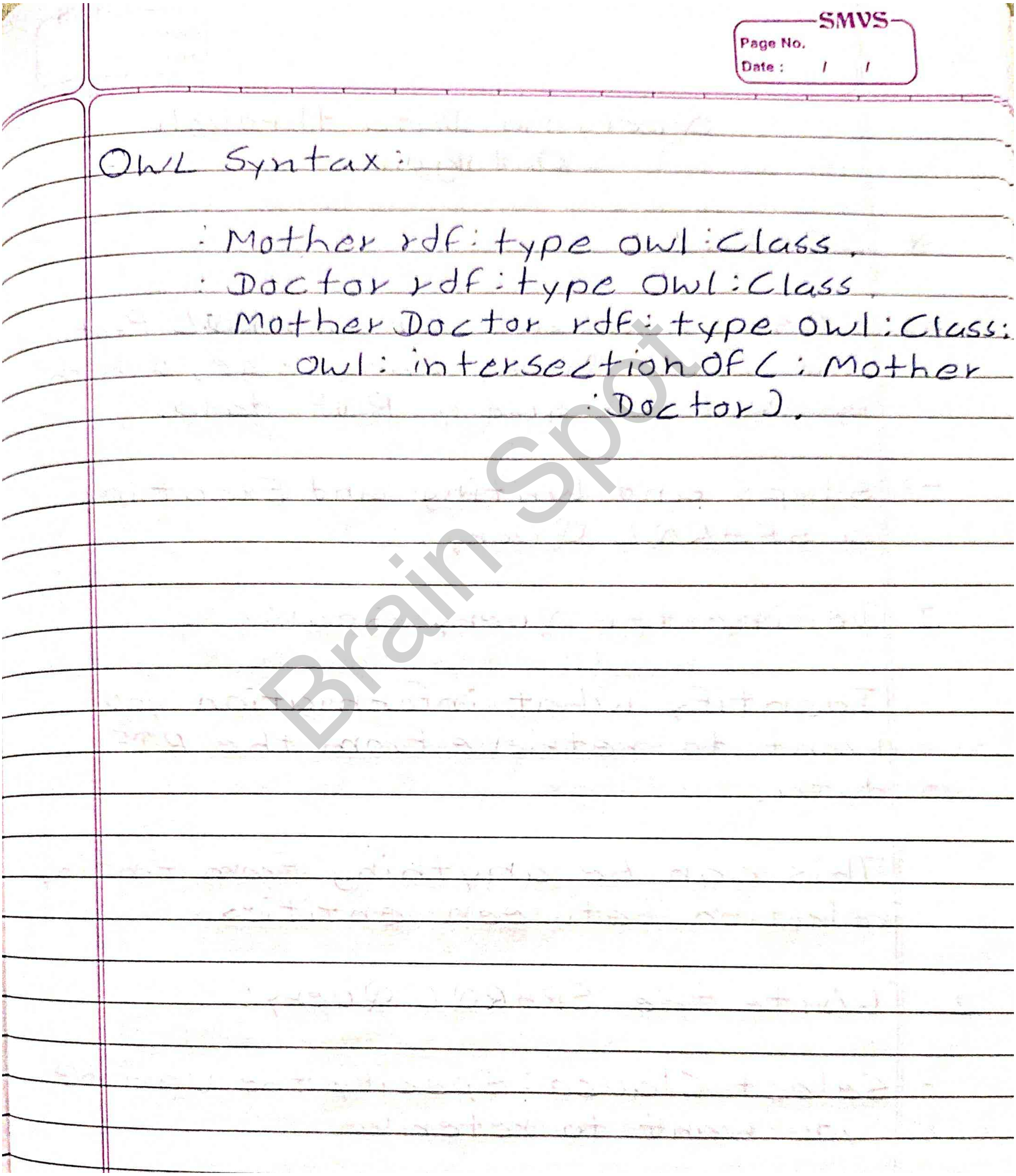
Ex. Classes : Doctor, Engineer
Union : Professional class

OWL Syntax :

```
:Doctor rdf:type owl:class.
:Engineer rdf:type owl:class.
:Professional rdf:type Owl:Class;
    owl:unionof(:Doctor
                :Engineer).
```

5 Class Intersection : Create a class which is intersection of two or more classes.

Ex. Classes : Mother, Doctor
Intersection : MotherDoctor
Class



OwL Syntax:

```
:Mother rdf:type owl:Class .
:Doctor rdf:type Owl:Class .
:MotherDoctor rdf:type owl:Class;
    owl: intersectionOf ( :Mother
                          :Doctor ) .
```

* Explain Ontology with example.

=) An Ontology is a formal and explicit specification of a conceptualization within a specific domain of knowledge.

It provides a structured representation of the entities, concepts, relationships and constraints that exist within that domain.

-> Components of Ontology:

(i) Classes: Used to represent types of entities within the domain.

(ii) Properties: Used to represent relationships between entities.

(iii) Individuals: Refer to specific objects within the domain.

(iv) Axioms: Includes rules that defines the behavior and semantics of the ontology.

-> Purpose of Ontologies:

1 Knowledge Representation: Used to serve a formal representation of domain knowledge.

2 Semantic Interoperability: Used to Facilitate Interoperability by providing semantic for communicating and exchanging data between different system.

3 Reasoning: Ontologies support automated reasoning for allowing system to derive new knowledge.

-> Applications of Ontologies:

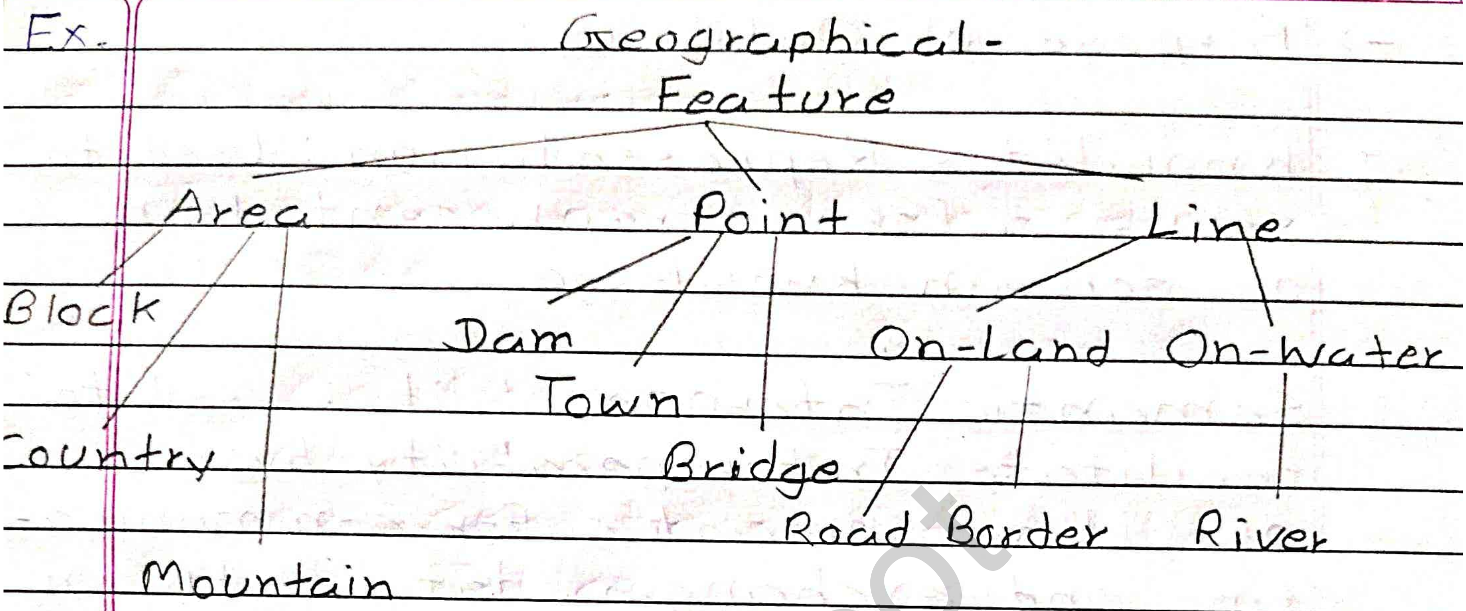
1 Used for Knowledge Management Systems.

2 Semantic Search Engines

3 Expert Systems and Decision Support systems.

4 Data Integration

Ex.


Geographical-
Feature
Area
Point
Line
Block
Country
Mountain
Dam
Town
Bridge
On-Land
On-Water
Road
Border
River