CS549 Distributed Information Systems

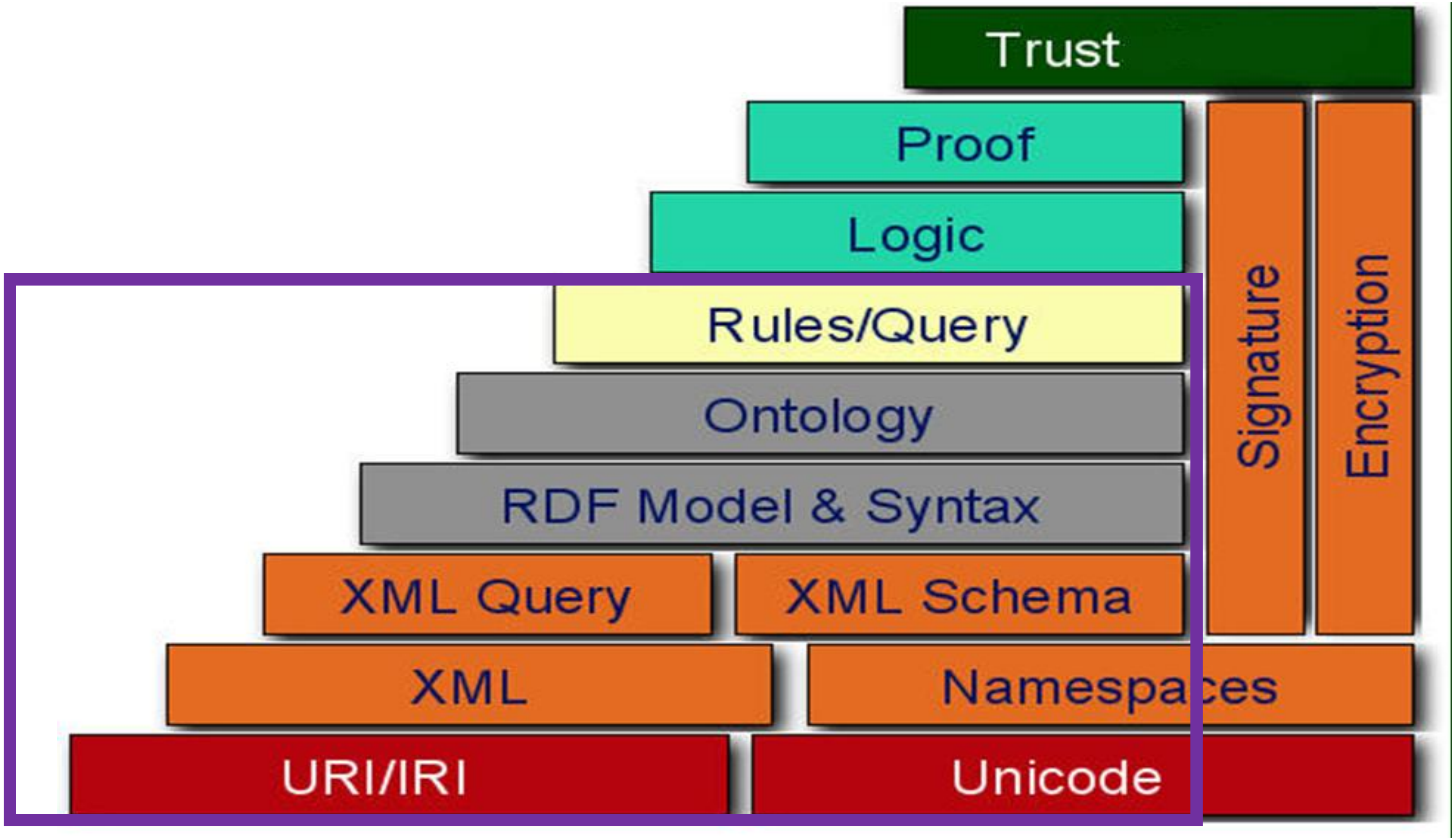
Lecture 10: Semantic Web - Summary

Session learning outcomes:

The learning outcomes of the session are to recap the concepts that have been considered for Distributed Systems. In particular:

- Ontology Engineering
- Data Models
  - Relation
  - XML
  - RDF
- Ontology Engineering
- SPARQL
- Description Logic

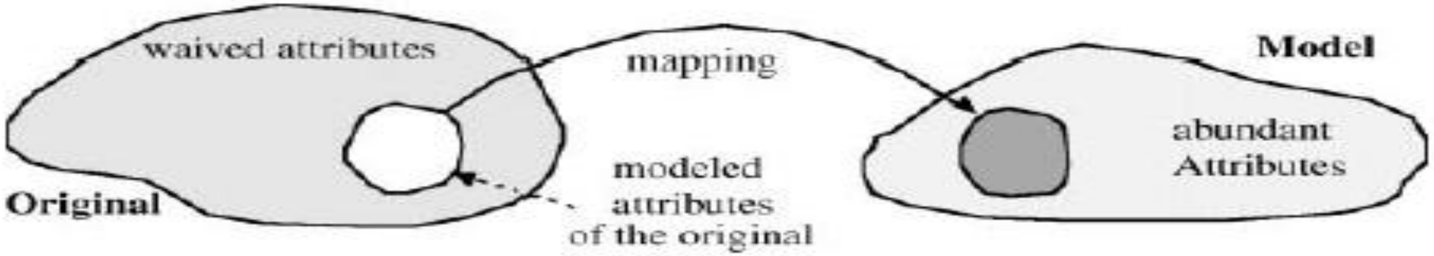
Semantic web stack
Where do we fit?



Systems

- A system can be defined as:
- A set of interrelated components that function together to achieve some outcome

Model Criteria



- Mapping criterion
  - The model maps to an original object (system, phenomenon, etc.)
- Reduction criterion
  - Not all ‘properties’ of the original are mapped onto the model
- Pragmatic criterion
  - A model must serve some purpose

Data Models

Relational Schema

- Each relation has attributes (headings) which are the relation schema

Before

1	2
RAM	20.00
Keyboard	2.50
Mouse	1.50
Printer	5.00

After

Attributes

Item	Price
RAM	20.00
Keyboard	2.50
Mouse	1.50
Printer	5.00

←Schema

Tuples

- By defining schema, we have more expressivity.
- Each attribute has a **domain** which includes all possible values for the considered attribute. E.g., Price domain is **float**

What does XML look like

Books

Title	Author	year
Java	John	1999
Pascal	Sara	1980
Basic	Mary	1975
Oracle	Emad	1999
….	….	

Relation

```
<Books>

    <Book>
        <Title>      Java        </Title>
        <Author>          John        </Author>
        <Year>            1999        </year>
    </Book>
    …
    …
    …
    <Book>
        <Title>      Oracle          </Title>
        <Author>          Emad
          </Author>
        <Year>            1999         </Year>
    </Book>
    ….

    ….
</ Books>
```

XML document

The Anatomy of XML Document

XML Declaration

Processing instruction

```
<?xml version:"1.0"?>

<?xml-stylesheet type="text/xsl" href="template.xsl"?>
```

Comments

```
<!--  File name: Bibliography.xml  -->
```

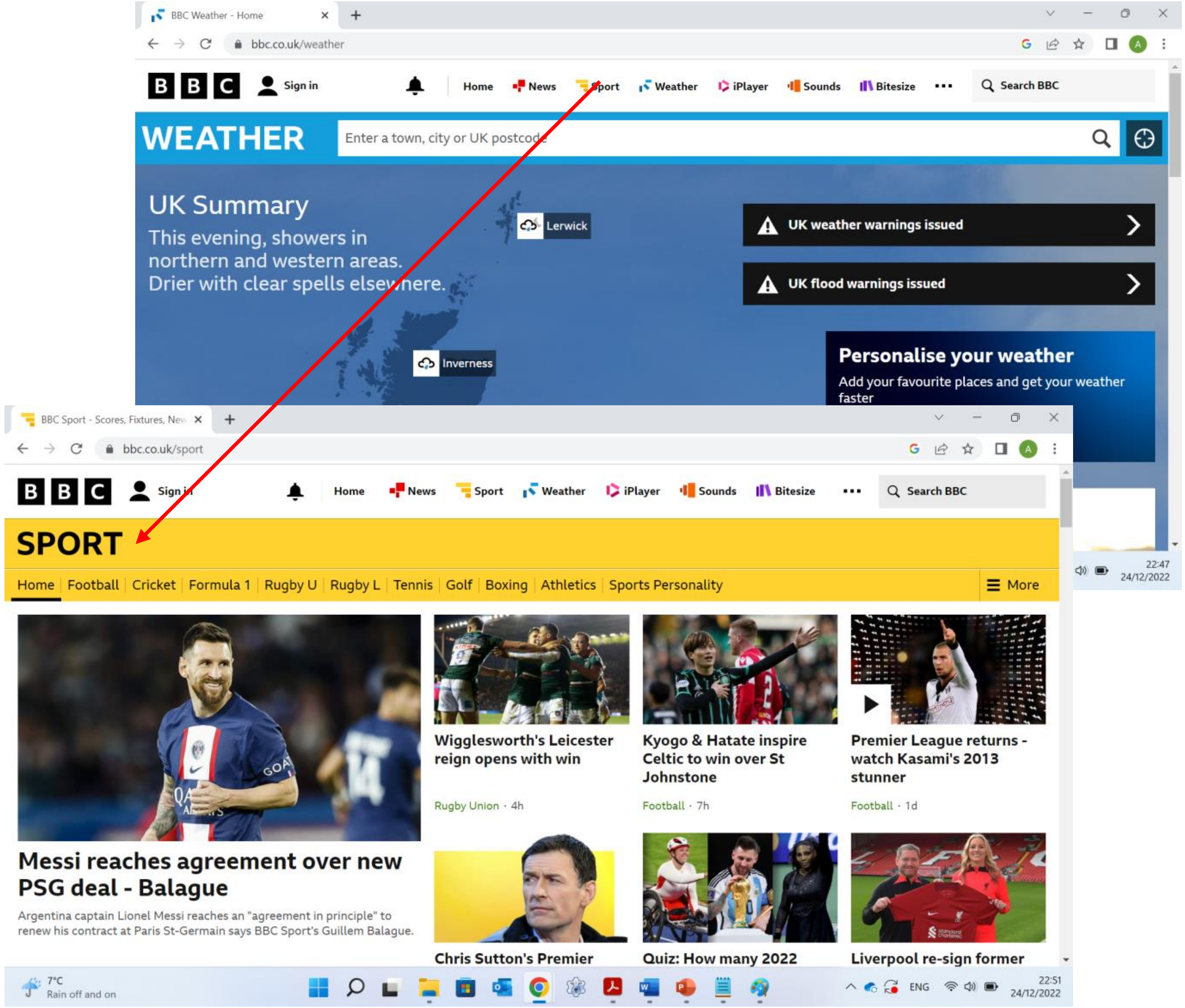
Attribute

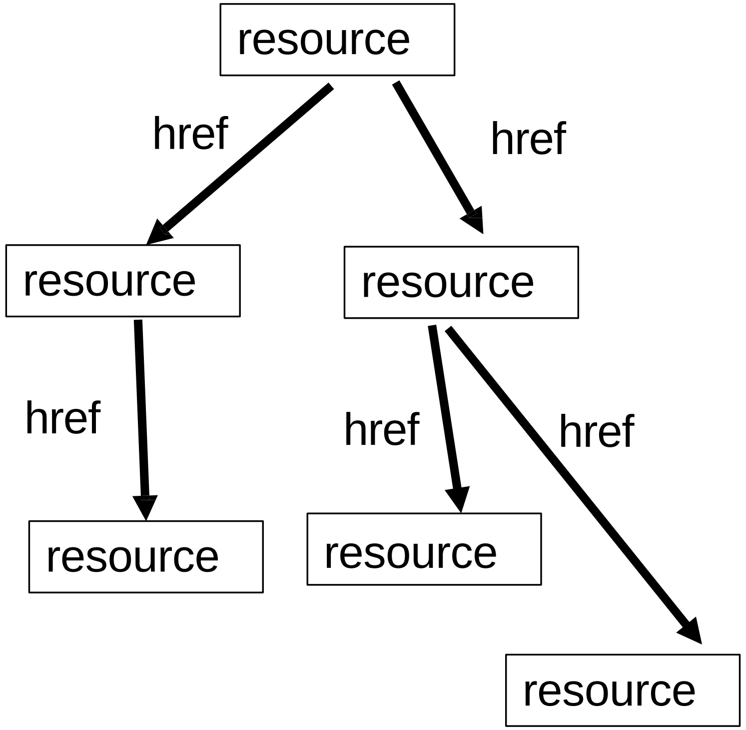
Root or document element

Elements nested Within root element

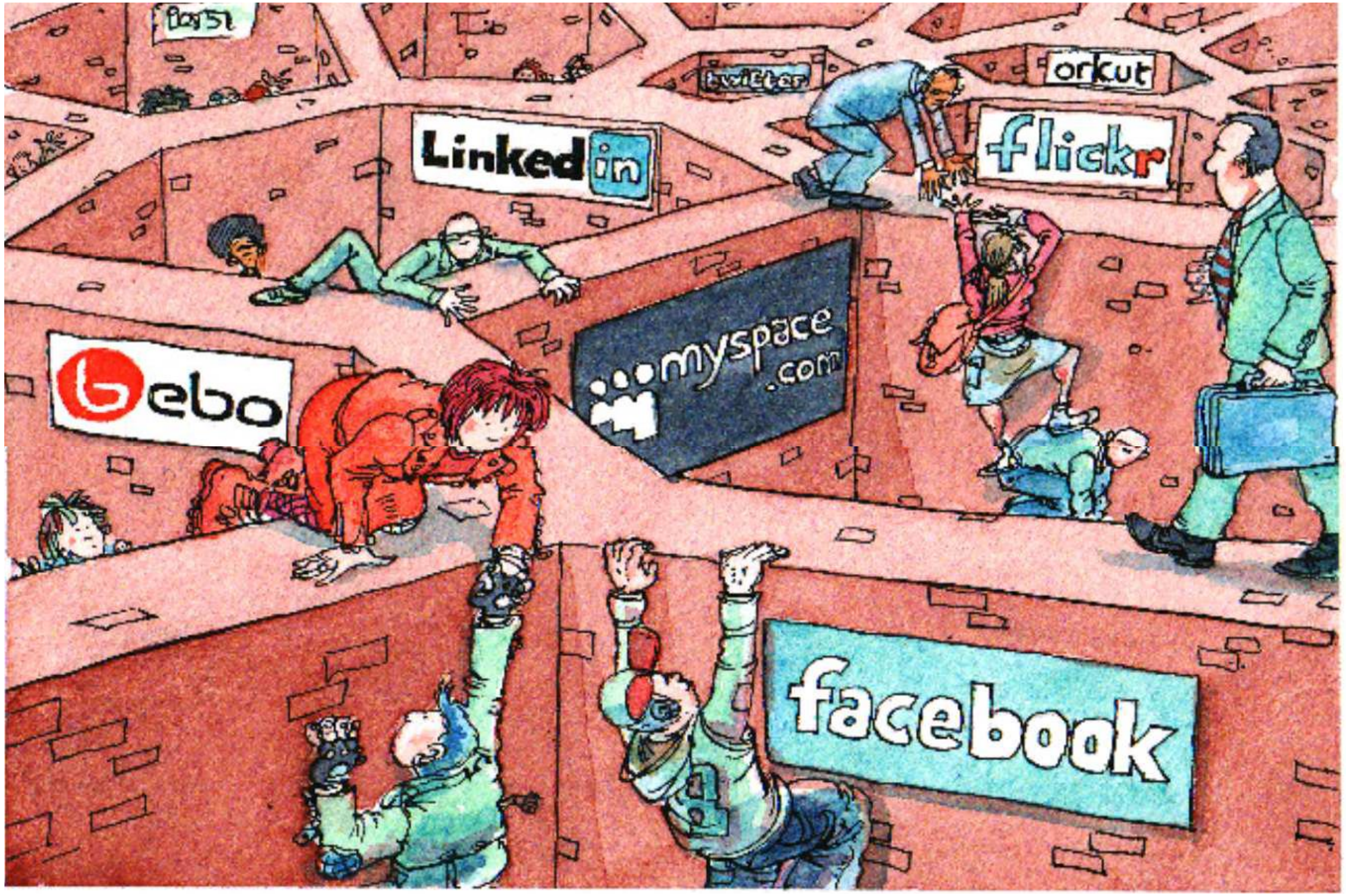
```
<Bibliography>
        <Book ISBN="1-111-122">
                <Title>      Java        </Title>
                <Author>     John        </Author>
                <Year>       1998        </Year>
        </Book>
        .
        .
        <Book>
                <Title>      Oracle      </Title>
                <Author>     Emad        </Author>
                <Year>       1999        </Year>
        </Book>
</Bibliography>
```

Web 1.0 : The syntactic Web – browsing





So, world wide web is a huge collection of data but it lacks integration….. How to improve current Web?

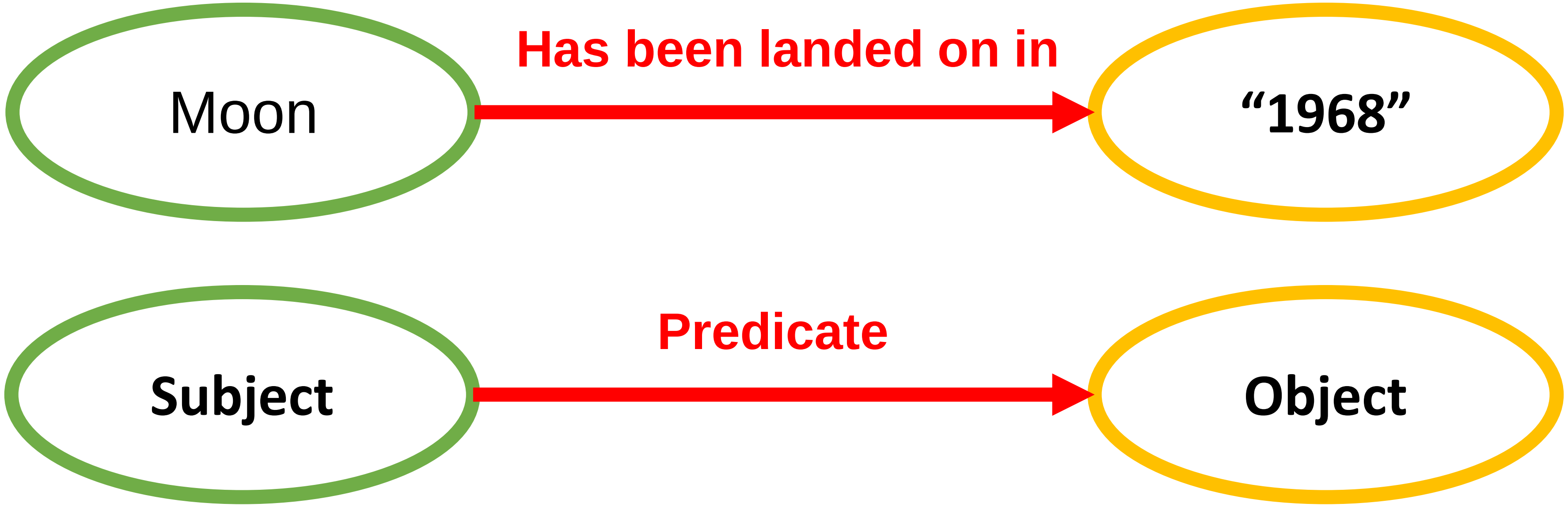


Scientific American, May 2001:

A Web in which data and information is machine readable

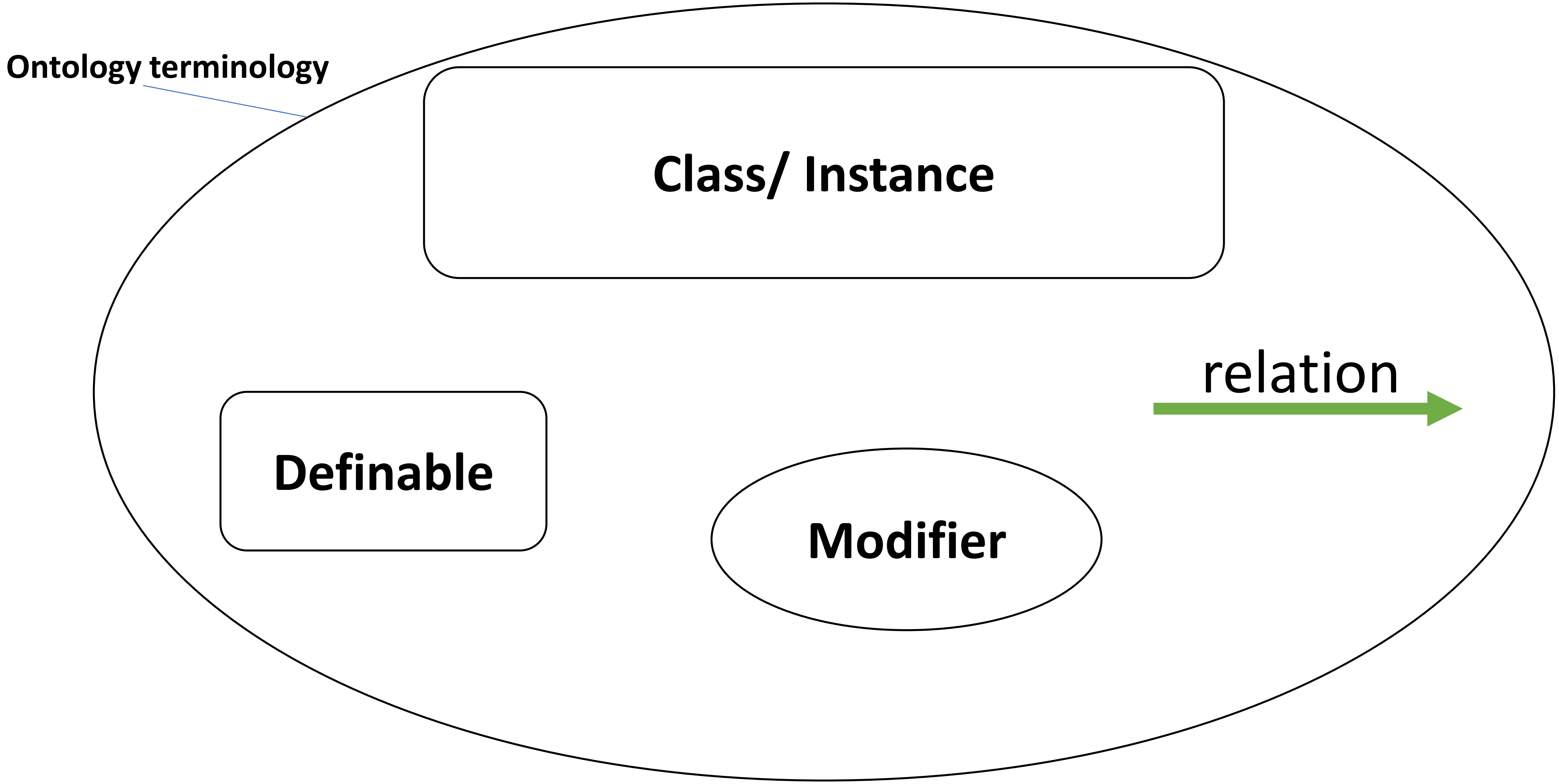
RDF: Representing Simple facts of Knowledge

- Consider the fact “ Moon has been landed on in 1968 “



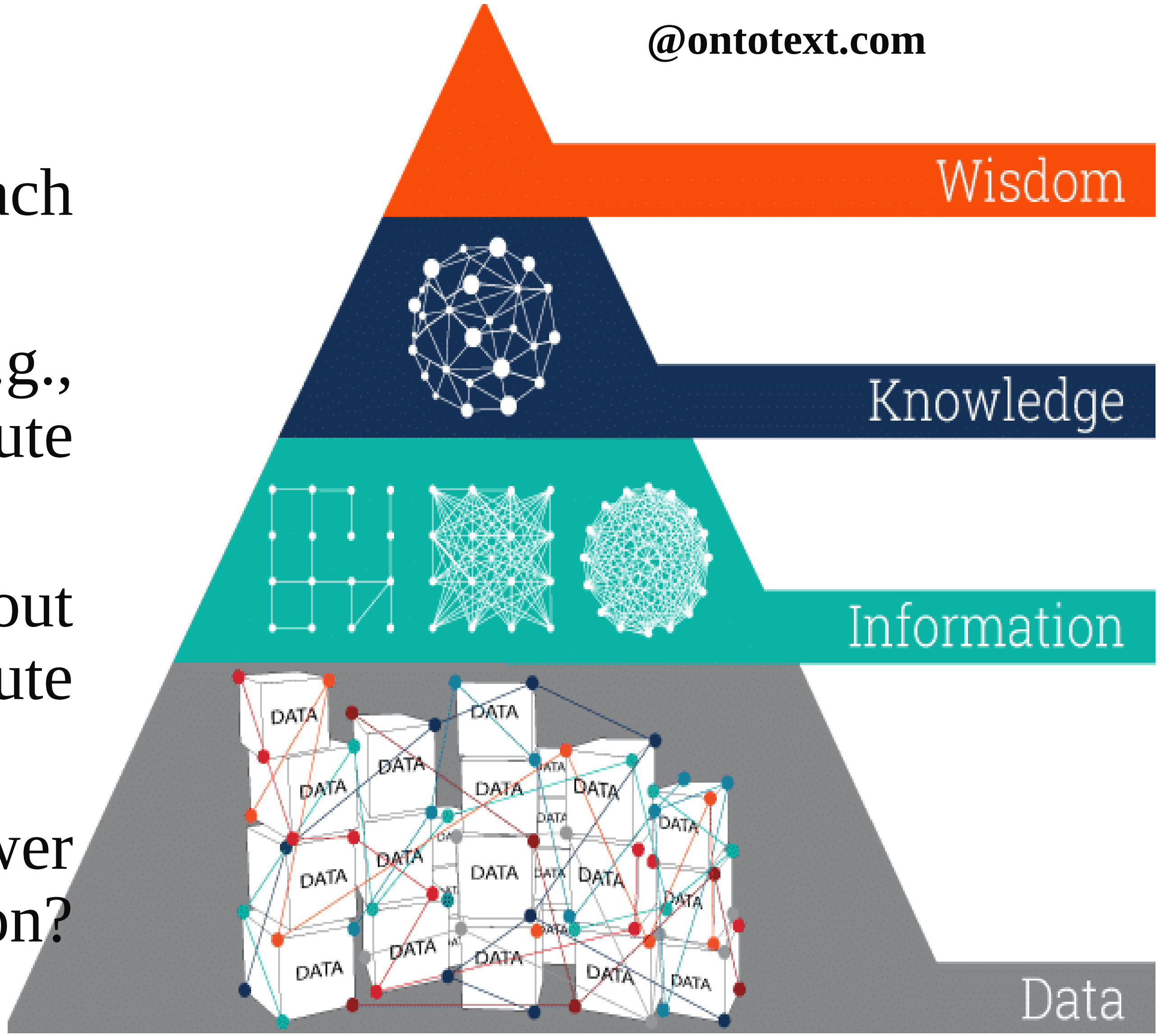
- Where the object is some value for the predicate
- So by using directed graph, we can represent facts of knowledge as Triple (Subject, Predicate, Object)

Ontology Structure : Terminology



DIKW Pyramid

- We have data at the bottom.
- This data, when it is related to each other forms information.
- We can amylase this information, e.g., locating patterns, then we constitute knowledge.
- When understanding principles about these patterns then we constitute wisdom
- Wisdom can be used to answer question like What is the best option? OR to predict.



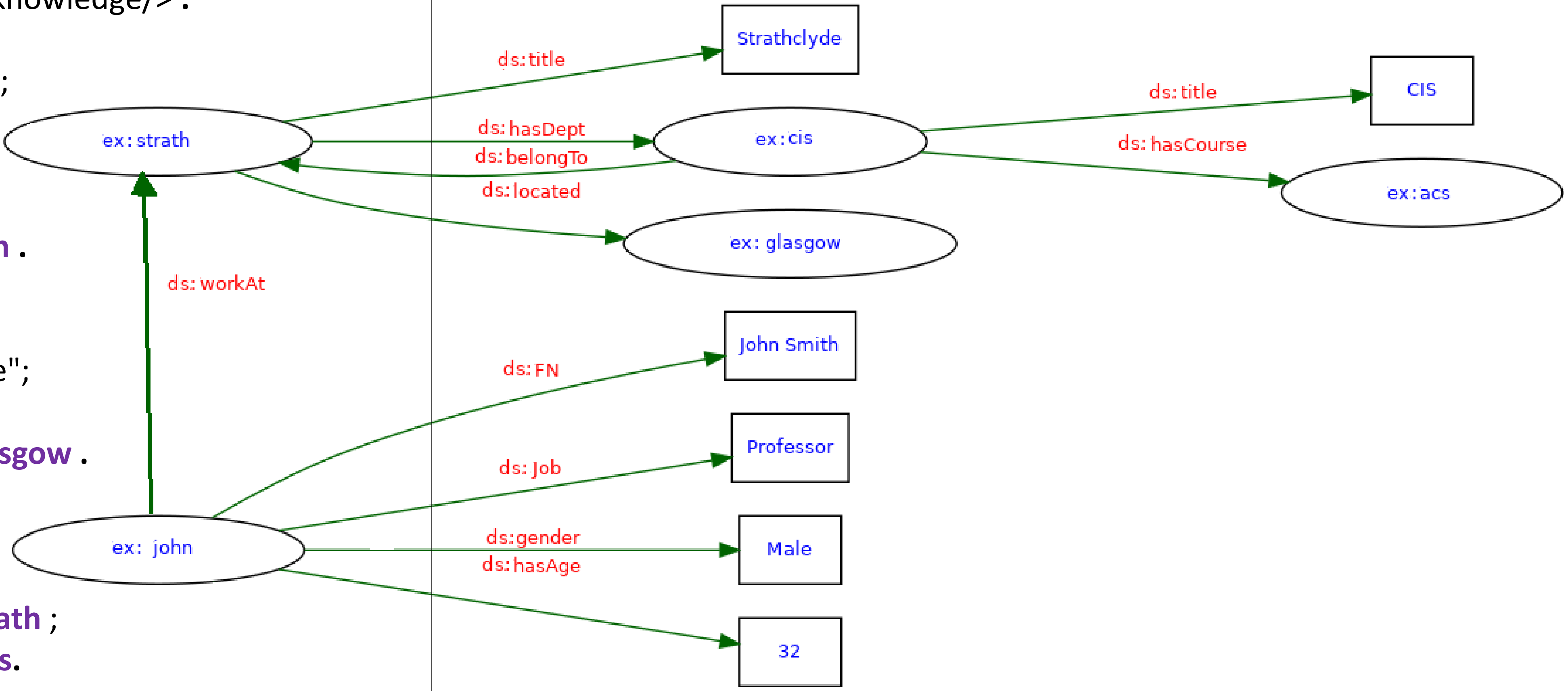
OWL : Web Ontology Language

- **OWL**: **is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things.**

```
<owl:Class rdf:about="#Triangle">
        <rdfs:subClassOf rdf:resource="#Shape"/>
   <owl:equivalentClass>
     <owl:Restriction>
       <owl:onProperty rdf:resource="#hasAngles"/>
       <owl:qualifiedCardinality rdf:datatype="XMLSchema#nonNegativeInteger">
            3
        </owl:qualifiedCardinality>
       <owl:onClass rdf:resource="#Angle"/>
     </owl:Restriction>
   </owl:equivalentClass>
 </owl:Class>
```

SPARQL: Turtle document and its RDF graph

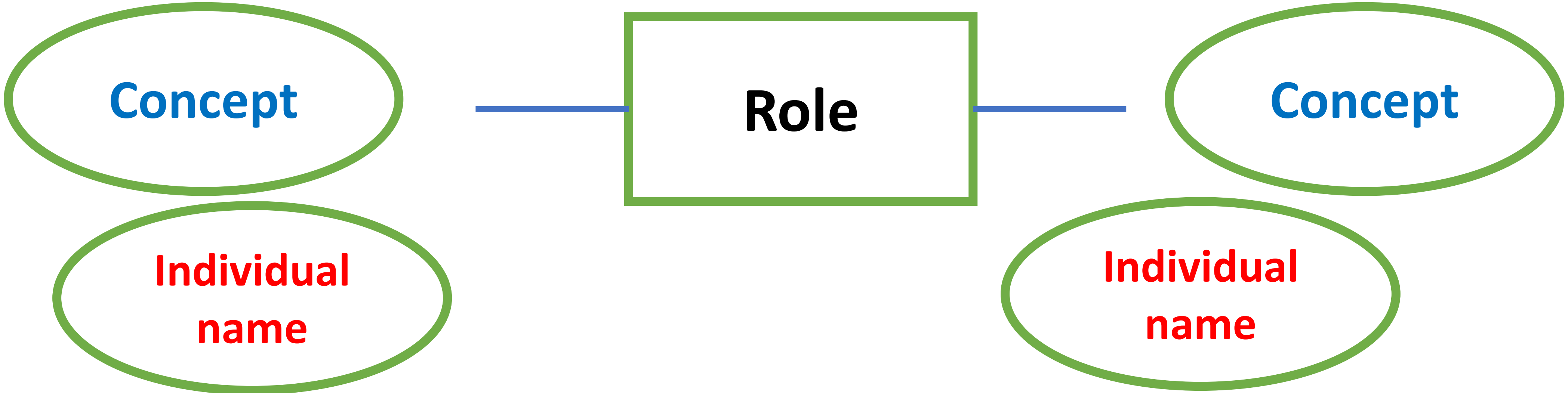
```
@prefix ex: <http://example.org/> .
@prefix ds: <http://knowledge/> .
ex:john
  ds:FN "John Smith" ;
  ds:Job "Professor" ;
  ds:gender "Male" ;
  ds:hasAge "32" ;
  ds:workAt  ex:Strath .

ex:strath
   ds:title "Strathclyde";
   ds:hasDept   ex:cis;
   ds:located     ex:glasgow .

ex:cis
   ds:title  "CIS" ;
   ds:belongTo  ex:strath ;
   ds:hasCourse ex:acs.
```



SELECT ?s ?p ?o
WHERE { ?s ?p ?o .}

Basic building blocks of DL ontologies

- DL allows modelling relationships between individuals in a domain of interest.
- To do so, DL ontologies represent knowledge by using three building blocks:
- **Concepts** to model a set of individuals
- **Roles** to model binary relations between the individuals, and
- **Individual names** to model single individuals in the domain.



DL knowledge representation

- When we use Description Logic (DL) to describe knowledge, we use this to capture some knowledge about the situation.
- It is important to note that to capture knowledge, we impose restriction.
- For example, consider a **Thing**. What knowledge we have about it?
- Impose restriction by saying that this thing is a **Living Thing**. The word Living is a restriction over Thing and carries knowledge.
- Similarly, **Plant Living Thing** is imposing more restriction and so more knowledge.

Types of axioms: (ABox) & (TBox)

- **Assertional (ABox) axioms:** These are one or more (set) of axioms that impose restriction on a given **individual**
- **Terminological (TBox) axioms:** These are one or more (set) of axioms that impose restriction on a given **Concept**

Conclusion

- Distributed information systems are important to develop systems that can be accessed remotely
- World Wide Web can be seen as a huge book (a lot of information)
- WWW requires to be integrated
- An Integrated WWW is a semantic web that allows both human and machines to understand.

Mock Exam Paper

Department of Computer and Information Sciences

CS549 Distributed Information Systems

Thursday 23rd March 2023 **9:30AM-11:30AM,**
Duration:2Hrs

Attempt ALL Questions

Calculators are NOT permitted

Question 1 **(15 marks)**

a. **(6marks)**

b. **(9 marks)**

Question 1. (15 marks)

- **(a)** Draw an entity relationship diagram (ERD) that represents the following scenario which allows a football club to check who played what: “Each player is assigned a unique number PN which uniquely identifies this player. Also, each match is identified by a match number MN. A player plays more than one match while a match is played by many players. Naturally, we want to store information about players name (pname) . Similarly, we want to store information about the match number (mn), match first team (mft), match second team(mst) and match date (md)”. **(6 marks)**

Question 1. (15 marks)

(b) Consider the three relational tables shown in Figure 1. These tables represent player, match, and player_match so that we can retrieve information about who plays what.

Player	
PN	**Pname**
01	John Smith
02	Tony Fiat
03	Martin Jonny
…	….

Player_Match	
PN	**MN**
01	01
01	02
01	03
02	01
..	…

Match			
MN	**MFT**	**MST**	**MD**
01	FFA	FFB	01/01/2000
02	FFA	FFC	15/01/2000
03	FFB	FFC	20/01/2000
…	….	….	

Figure 1: Sample relational tables

(i) Write SQL statement to create the table Player.

(ii) Write SQL statement to insert the first row of data (01,FFA, FFB,01/01/2000) into table Match.

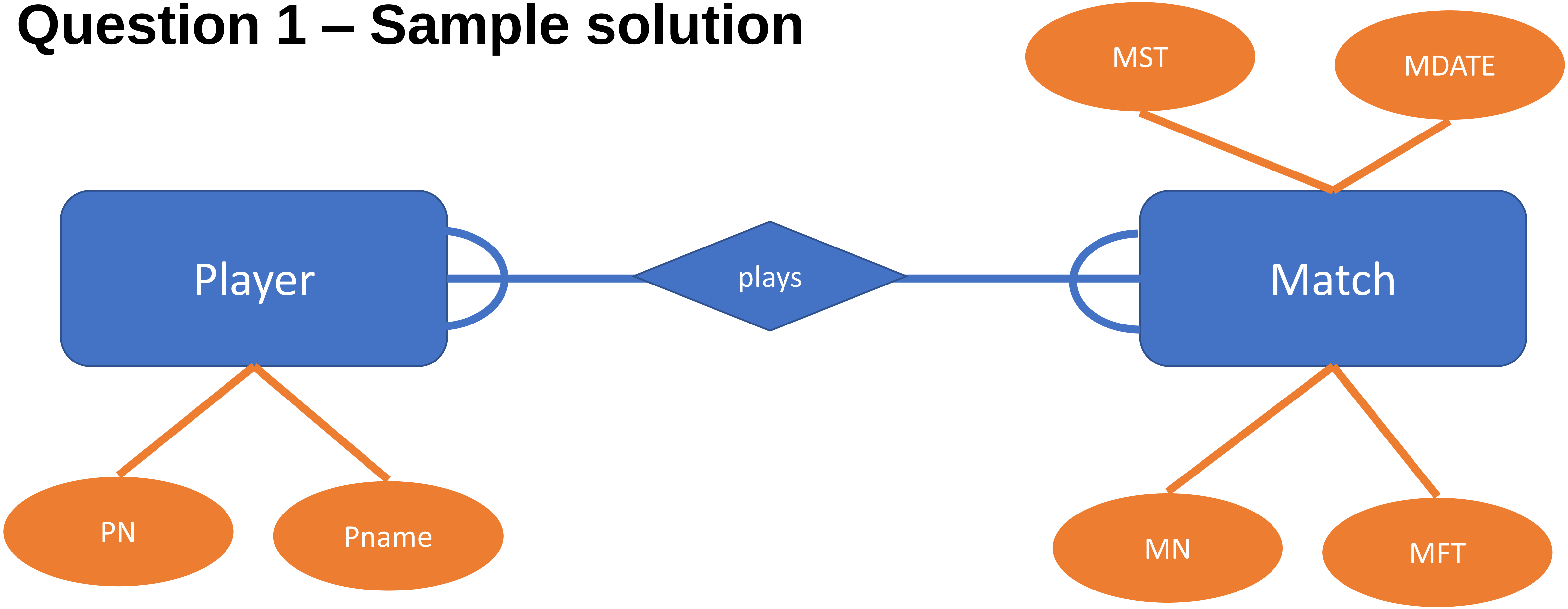
(iii) Write SQL statement to list names of players who played Match that has MN=03.

(9 marks)

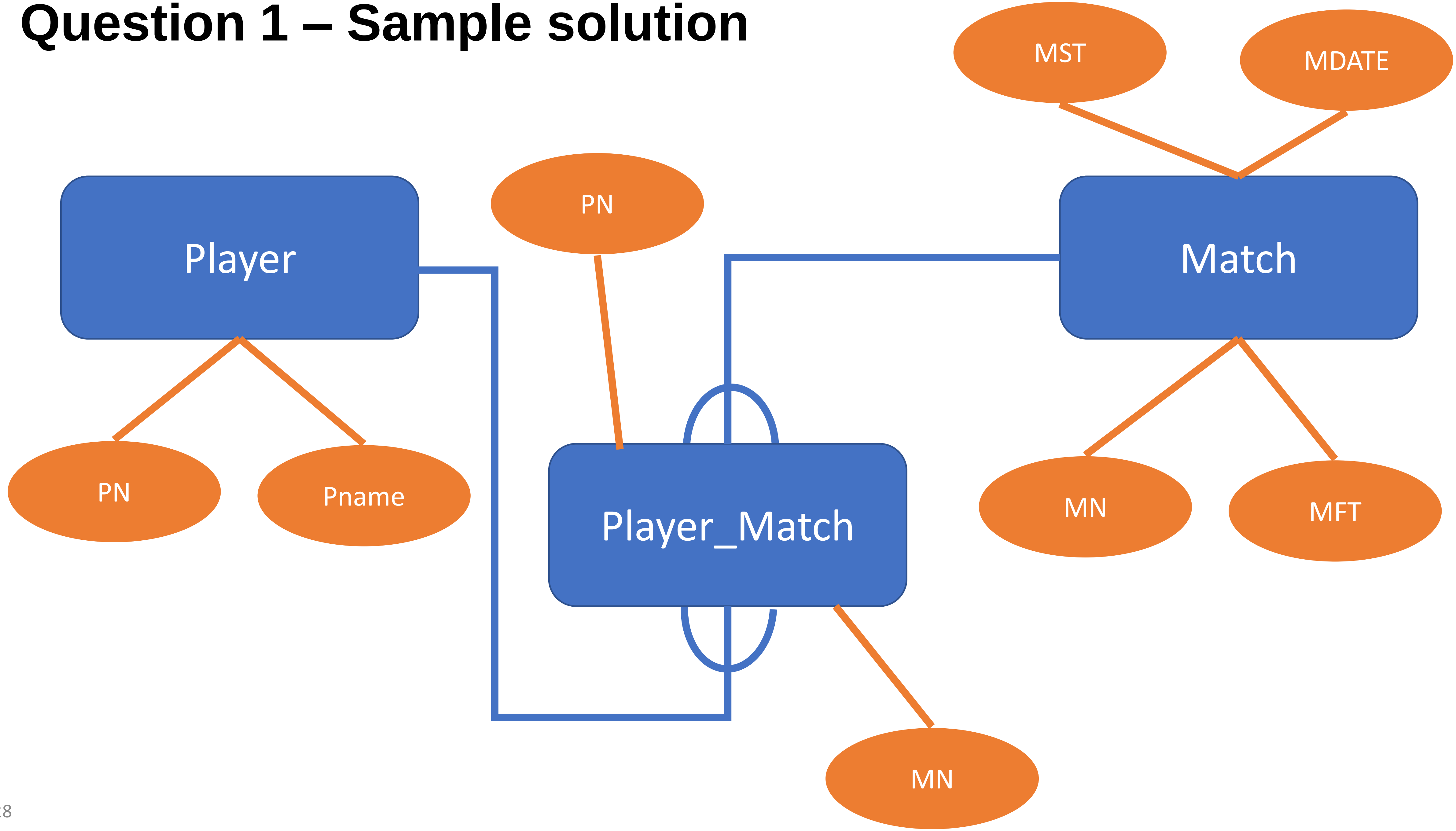
Question 1. (15 marks)

- **(a)** “Each **player** is assigned a unique **number PN** which uniquely identifies this player. Also, each match is identified by a **match** number MN. A player **plays** more than one match while a match is **played by** many players. Naturally, we want to store information about players name (**pname**) . Similarly, we want to store information about the match number (**mn**), match first team (**mft**), match second team(**mst**) and match date (**md**)”. **(6 marks)**

Question 1 – Sample solution



Question 1 – Sample solution



Question 1. (15 marks)

(b) Consider the three relational tables shown in Figure 1. These tables represent player, match, and player_match so that we can retrieve information about who plays what.

Player	
PN	**Pname**
01	John Smith
02	Tony Fiat
03	Martin Jonny
…	….

Player_Match	
PN	**MN**
01	01
01	02
01	03
02	01
..	…

Match			
MN	**MFT**	**MST**	**MD**
01	FFA	FFB	01/01/2000
02	FFA	FFC	15/01/2000
03	FFB	FFC	20/01/2000
…	….	….	

Figure 1: Sample relational tables

(i) Write SQL statement to create the table Player.

(ii) Write SQL statement to insert the first row of data (01,FFA, FFB,01/01/2000) into table Match.

(iii) Write SQL statement to list names of players who played Match that has MN=03.

(9 marks)

Question 1 – Sample solution

- **(b) (3 marks for each part)**

```
I. create TABLE player( PN INt, Pname varchar(10
   0), CONSTRAINT pk primary key (pn));
II.insert into test.match values(1, "FFA", "FFB"
   , "2000/01/01");
III.SELECT pnames from player, player_match
   WHERE (player.pn = player_match.pn) and
   (player_match.mn=3);
```

Question 2 (15 marks)

a. (3marks)

b. (12 marks)

Question 2. (15 marks)

- **(a)** Consider the **Match** table shown in Figure 1. Write an XML file that represents this table.

(3 marks)

```xml
<?xml version="1.0"?>
<matchtb>
   <match mn="01">
      <mft>FFA</mft>
      <mst>FFB</mst>
      <mdate> 01/01/2000</mdate>
  </match>
</matchtb>
```

Question 2. (15 marks)

- **(b)** Consider the sample XML file shown below in Figure 2:

```
<?xml version="1.0"?>
<matchtb>
   <match mn="01">
      <mft>FFA</mft>
      <mst>FFB</mst>
      <mdate> 01/01/2000</mdate>
  </match>
</matchtb>
```

Figure 2: Sample XML file

I. Draw an XML tree that represents this XML file.

II. Write XPATH query to list the (**mn)** of the matches that has **mft** "FFA"

III. Write XQuery query to count the number of matches which have **mst** "FFB"

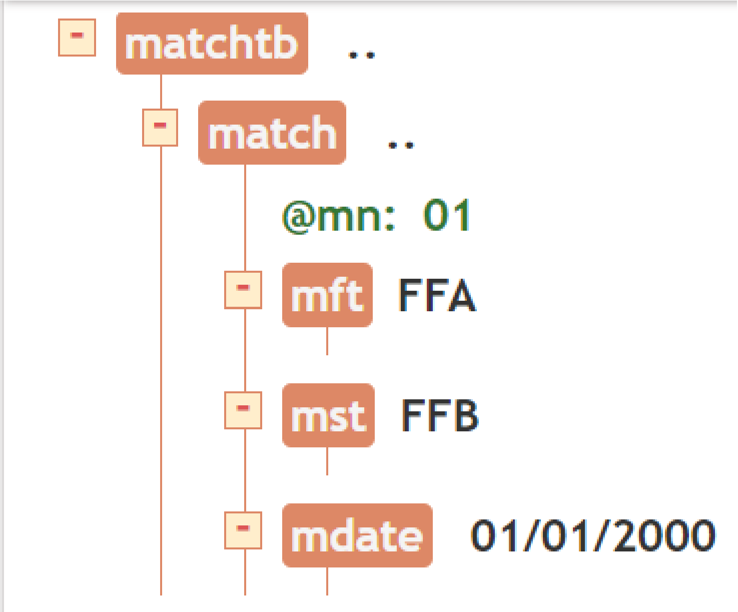
(12 marks)

Question 2. (15 marks)

- **(b)**

i. Draw an XML tree that represents this XML file.

```
<?xml version="1.0"?>
<matchtb>
   <match mn="01">
      <mft>FFA</mft>
      <mst>FFB</mst>
      <mdate> 01/01/2000</mdate>
  </match>
</matchtb>
```



ii. Write XPATH query to list the (**mn)** of the matches that has **mft** “FFA”

//match[mft="FFA"]/@mn

iii. Write XQuery query to count the number of matches which have **mst** “FFB”

let $x := //match return count ($x[mst="FFB"])

Question 3 (10 marks)

a. (3marks)

b. (3 marks)

c. (4 marks)

Question 3. (10 marks)

- **(a)** Draw RDF diagram to represent the relationships between the elements in these statements.
- (“MN01”, http://knowledge.db/played_by, “John_Smith”)
- (“John_Smith”, http:// knowledge. db/player_class, “A”)
- (“John_Smith”, http:// knowledge.db/playes_in, “FFA”)
- (“FFA”, http:// knowledge.db /has_player, “John_Smith”)

(3 marks)

Question 3. (10 marks)

- **(b)** Write The RDF file in Turtle Format that represent the graph the statements given in Part (Q3.a).

- **(“MN01”, http://knowledge.db/played_by, “John_Smith”)**
- **(“John_Smith”, http:// knowledge. db/player_class, “A”)**
- **(“John_Smith”, http:// knowledge.db/playes_in, “FFA”)**
- **(“FFA”, http:// knowledge.db /has_player, “John_Smith”)**

(3 marks)

Question 3. (10 marks)

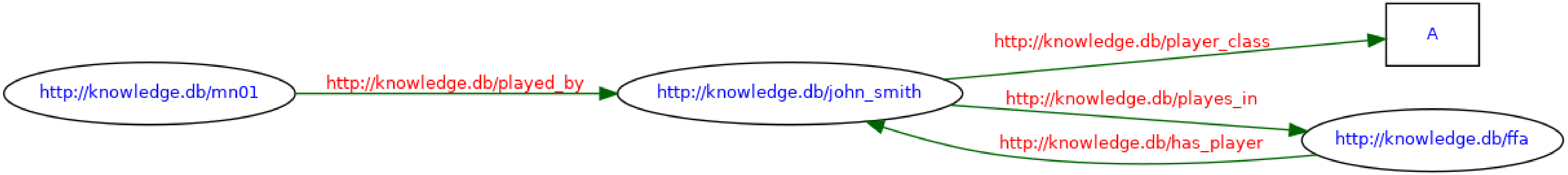
- **(c)** Using OWL, define the following class as A property restriction “Rectangle is a shape that has exactly 4 angles”

(4 marks)

Question 3 – Sample solution

- **(a)** Draw RDF diagram to represent the relationships between the elements in these statements.

- (“MN01”, http://knowledge.db/played_by, “John_Smith”)
- (“John_Smith”, http:// knowledge. db/player_class, “A”)
- (“John_Smith”, http:// knowledge.db/playes_in, “FFA”)
- (“FFA”, http:// knowledge.db /has_player, “John_Smith”) **(3 marks)**



Question 3 – Sample solution

- **(b)** Write The RDF file in Turtle Format that represent the graph the statements given in Part (Q3.a).

- (“MN01”, http://knowledge.db/played_by, “John_Smith”)
- (“John_Smith”, http:// knowledge. db/player_class, “A”)
- (“John_Smith”, http:// knowledge.db/playes_in, “FFA”)
- (“FFA”, http:// knowledge.db /has_player, “John_Smith”) **(3 marks)**

```
@prefix ex: <http://knowledge.db/> .
ex:mn01 ex:played_by ex:john_smith .
ex:john_smith ex:player_class "A" .
ex:john_smith ex:playes_in ex:ffa .
ex:ffa ex:has_player ex:john_smith .
```

Question 3 – Sample solution

(c) Using OWL, define the following class as A property restriction "Rectangle is a shape that has exactly 4 angles" **(4 marks)**

```
<owl:Class rdf:about="#Rectangle">
            <rdfs:subClassOf rdf:resource="#Shape"/>
     <owl:equivalentClass>
        <owl:Restriction>
           <owl:onProperty rdf:resource="#hasAngles"/>
           <owl:qualifiedCardinality rdf:datatype="XMLSchema#nonNegativeInteger">
                   4
             </owl:qualifiedCardinality>
           <owl:onClass rdf:resource="#Angle"/>
        </owl:Restriction>
     </owl:equivalentClass>
  </owl:Class>
```

Question 4 (20 marks)

a.	(6marks)
b.	(4 marks)
c.	(2 marks)
d.	(8 marks)

Question 4. (20 marks)

- Develop an ontology called SportOnt to represent the following sport-related concepts:

- "Player and Coach are Person. Football club and Basketball club are kinds of Club. Each club has a name. A coach works for a club and trains player. A player plays for a club. Each player has a position. A Basketballer is a player who plays only for a basketball club. Liverpool FC is an instance of a football club, Mohamed Salah is an instance of a player that plays for Liverpool club and has position forward. Jurgen Klopp is an instance of coach that works for Liverpool FC. MJordan is an instance of basketballer who plays for Chicago Bulls basketball club.

Question 4. (20 marks)

- **(a)** In your solution draw a table (in the format of the one shown below) that lists the standalone classes, modifiers, relations and definables, distinguishing between classes and instances. Indicate the hierarchies for classes. **(6 marks)**
-

Classes/ instances	Modifiers	Relation	Definable

Question 4. (20 marks)

(b) In your solution draw a table (in the format of the one shown below) that lists each relation type and its characteristics **(4 marks)**

Domain	Relation	Range	Object Property	Data Property	Symmetric	Transitive	Functional	Inverse Functional	Reflexive

Question 4. (20 marks)

(c) In your solution mention two classes that are subclasses

(2 marks)

(d) Draw the ontology graph that represents your ontology SportOnt (Don’t show instances on your ontology graph) **(8 marks)**

Question 4 – Sample solution

- (a)

Classes/ instances	Modifiers	Relation	Definable
Person **Player** (MohamedSalah inst) **Coach** (Jurgen Klopp inst) **Club** **FootballClub** (Liverpool FC inst) **BasketBallClub** **(Chicago Bull** inst) **Position** (forward inst)	**Name**	**worksFor** **playsFor** **trains** **hasPos** **hasName**	**Basketballer** (Mjordan inst)

Question 4. (20 marks)

(b) In your solution draw a table (in the format of the one shown below) that lists each relation type and its characteristics **(4 marks)**

Domain coach	Relation	Range	Object Property	Data Property	Symmetric	Transitive	Functional	Inverse Functional	Reflexive
~~Person~~	worksFor	Club	X				X		
Player	playsFor	Club	X				X		
Coach	trains	Player	X						
Player	hasPos	Position	X						
Club	hasName	Literal		X					

Question 4. (20 marks)

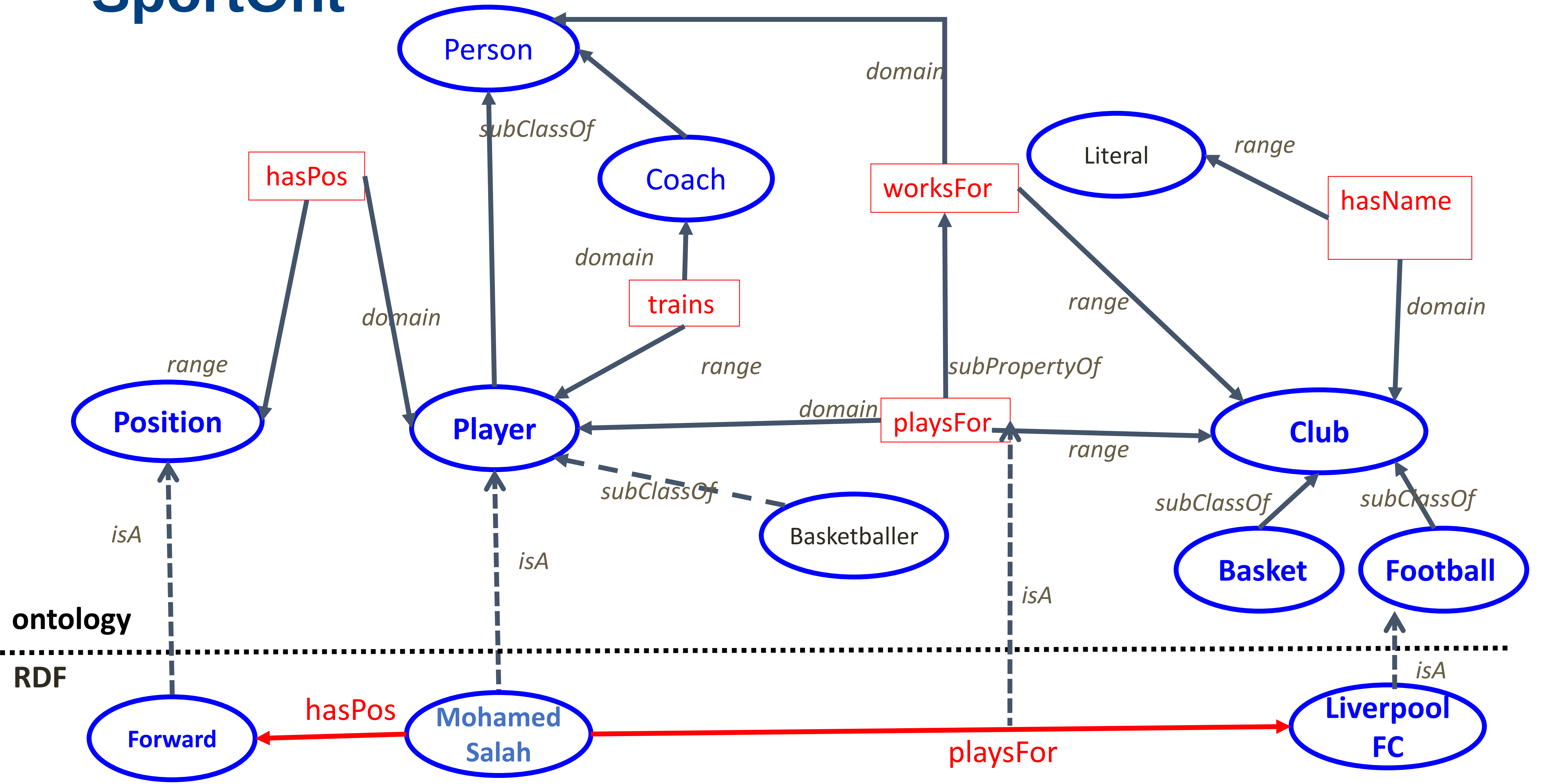
(c) In your solution mention two classes that are subclasses

(2 marks)

FootballClub is a subclass of Club
Player is subclass of Person

(d) Draw the ontology graph that represents your ontology SportOnt
(Don't show instances on your ontology graph) **(8 marks)**

SportOnt


Person
subClassOf
domain
Coach
hasPos
domain
trains
worksFor
Literal
range
hasName
range
domain
range
subPropertyOf
range
Position
Player
domain
playsFor
Club
range
subClassOf
Basketballer
subClassOf
subClassOf
isA
isA
Basket
Football
isA
ontology
RDF
isA
hasPos
Forward
Mohamed Salah
playsFor
Liverpool FC

Question 5 (10 marks)

a. (5marks)

b. (5marks)

Question 5. (10 marks)

- **(a)** Consider the RDF file in Turtle format shown in Figure 4 below. By using SPARQL, write a query that lists all the matches (as resources) which has been played by player ex:john_smith **(5 marks)**

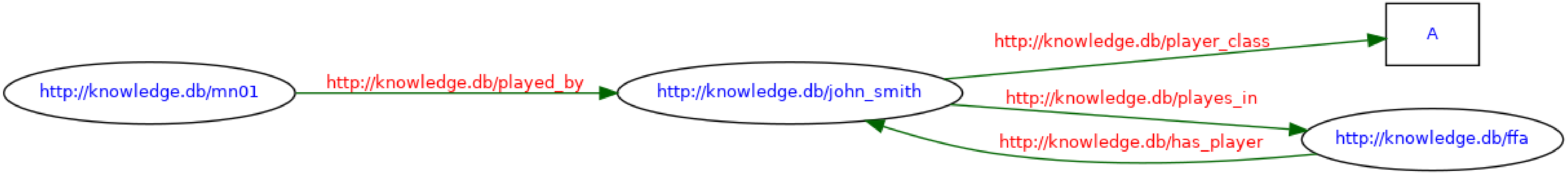
- **(b)** Consider the RDF file in Turtle format shown in Figure 4 below. By using SPARQL, write a query that lists all the players (as resources) who played match ex:mn01 and has “A” as a player class **(5 marks)**

```
@prefix ex: <http://knowledge.db/> .
ex:mn01 ex:played_by ex:john_smith .
ex:john_smith ex:player_class "A" .
ex:john_smith ex:playes_in ex:ffa .
ex:ffa ex:has_player ex:john_smith .
```

- **Figure 4:** Sample RDF file in Turtle

Question 5- Sample solution

- **(a)** it is optional to draw the RDF graph and then write the required query

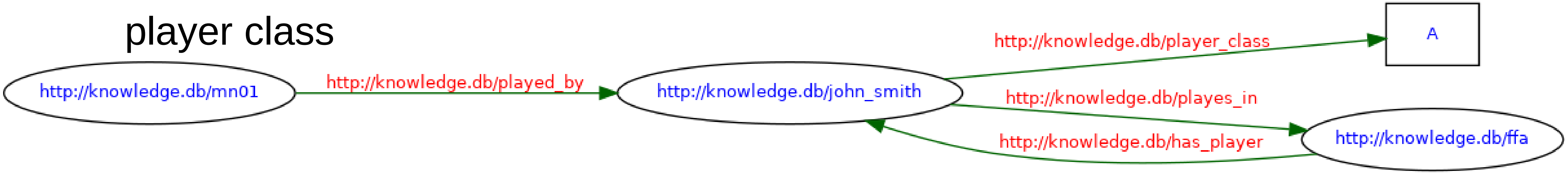


- The graph above represents the given RDF file. To answer Q5.a, we first identify the pattern to traverse through the graph (matches (as resources) which has been played by player ex:john_smith)

```
PREFIX ex: @prefix ex: <http://knowledge.db/#>.
SELECT ?s
WHERE {?s ex:played_by ex:john_smith}
```

Question 5- Sample solution

- **(b)** Similarly, we first identify the pattern to traverse through the graph (players (as resources) who played match ex:mn01 and has “A” as a player class




```
PREFIX ex: @prefix ex: <http://knowledge.db/#>.
SELECT ?s
WHERE {ex:mn01 ex:played_by ?s .
       ?s  ex:player_class  ?c .
       FILTER (?c = “A”) .}
```

END