CSC343 Worksheet 9 Solution

## June 28, 2020

1. **Exercise 11.1.1:**

a)

Notes:

- Semistructured data
  - serves as a model suitable for **databases integration**, that is, for describing the data contained in two or more databases that contain similar data with different schemas
  - It serves as the underlying model for notations such as XML, to be taken up in Section 2, that are being used to share information on the web.
- Semistructured Data Representation
  - is a collection of nodes

Table

Tuple (w/ abbreviated name as label)

-
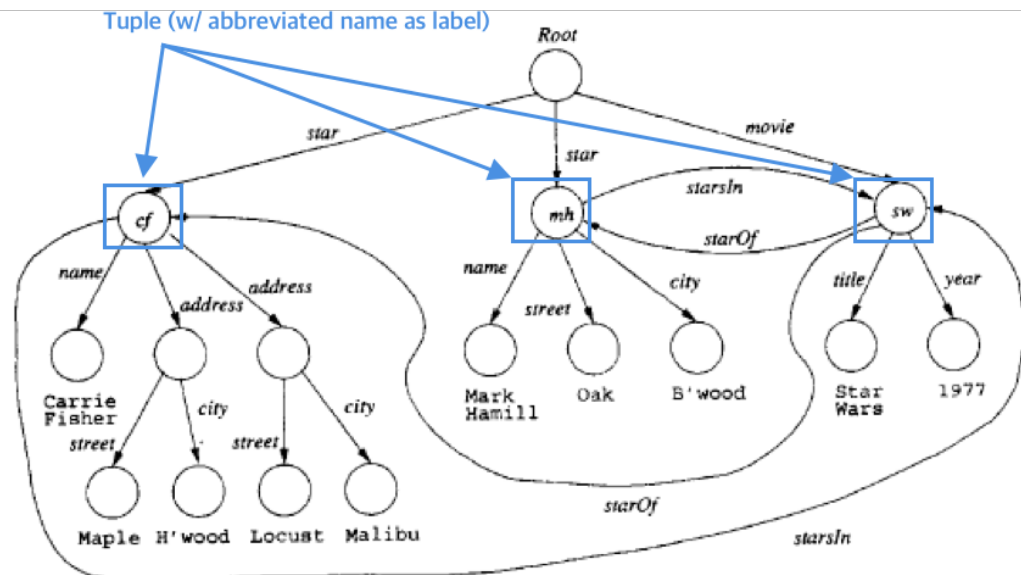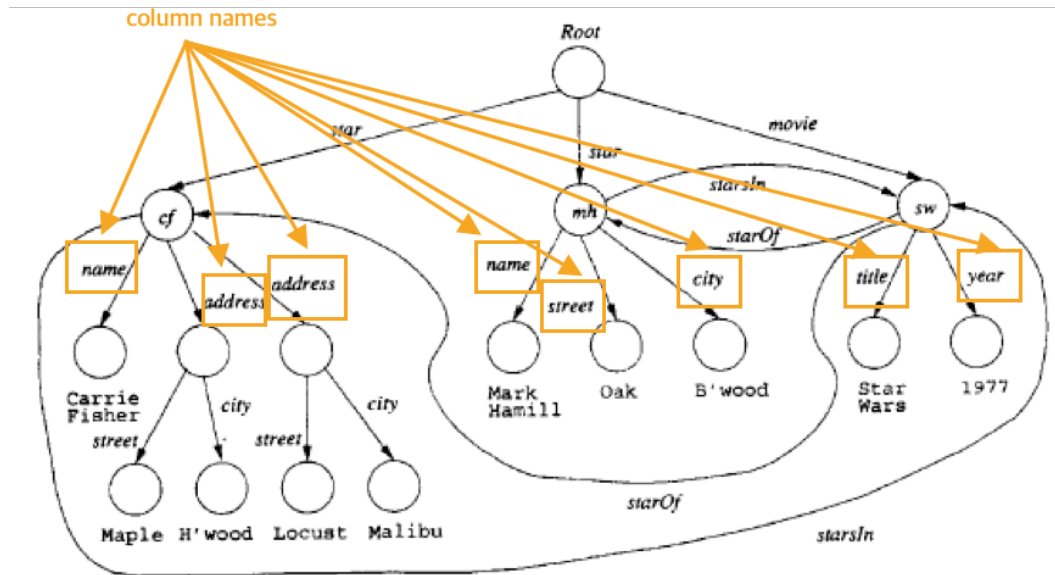
b)

c)

2.

3.

4. The difference is that UML must fit data into its schema, where as the semi structured data allows whatever schema information that is appropriate to be attached to data

Notees:

- Semi-structured Data
  - Is schemaless

- Is motivated primarily by its flexibility
- One could enter data at will, and attach to the data whatever schema information you felt was appropriate for that data.
- Makes query processing harder

- Structured Data

  - Is rigid framework into which data is placed.
  - Data must fit into schema
  - Fixed schema allows data to be organized with data structures that support efficient answering of queries
  - e.g. UML, E/R, Relational, ODL

5. a)
```xml
<? xml version = "1.0" encoding="utf-8" standalone = "yes">
<StarMovieData>
    <Star starID="cf" starredIn="sw">
        <Name>Carrie Fisher</Name>
        <Address>
            <Street>123 Maple St.</Street>
            <City>Hollywood</City>
        </Address>
        <Address>
            <Street>5 Locust Ln.</Street>
            <City>Malibu</City>
        </Address>
    </Star>
    <Star starID="mh" starredIn="sw">
        <Name>Mark Hamill</Name>
        <Street>456 Oak Rd.</Street>
        <City>Brentwood</City>
    </Star>
    <Movie movieID="sw" starsOf="cf", "mh">
        <Title>Star Wars</Title>
        <Year>1977</Year>
    </Movie>
    <MovieExec movieExecID="gl" execsIn="sw">
        <Name>George Lucas</Name>
        <Position>Director</Position>
    </MovieExec>
    <MovieExec movieExecID="gk" execsIn="sw">
        <Name>Gru Kurtz</Name>
        <Position>Produced By</Position>
    </MovieExec>
</StarMovieData>
```

- XML
  - is called *Extensible Markup Language*
  - is an example of semistructured data
- XML with and without a Schema

- has two different types

    1. Well-formed XML

        * allows to invent your own tags
        * corresponds very-similarly to semi-structured data

        Example:

```
1    <? xml version = "1.0" encoding="utf-8" standalone = "yes
     ">
2    <StarMovieData>
3        <Star>
4            <Name>Carrie Fisher</Name>
5            <Address>
6                <Street>123 Maple St.</Street>
7                <City>Hollywood</City>
8            </Address>
9            <Address>
10               <Street>5 Locust Ln.</Street>
11               <City>Malibu</City>
12           </Address>
13       </Star>
14       <Star>
15           <Name>Mark Hamill</Name>
16           <Street>456 Oak Rd.</Street>
17           <City>Brentwood</City>
18       </Star>
19       <Movie>
20           <Title>Star Wars</Title>
21           <Year>1977</Year>
22       </Movie>
23   </StarMovieData>
24
```

    2. Valid XML

        * Involes "Document Type Definition"
        * specifies allowable tags and gives a grammar for how they may be nested

- Attributes
    - is used to represent connections in a semistructured data graph

    Example:

```
1    <? xml version = "1.0" encoding="utf-8" standalone = "yes">
2    <StarMovieData>
3        <Star starID="cf" starredIn="sw">
4            <Name>Carrie Fisher</Name>
5            <Address>
6                <Street>123 Maple St.</Street>
7                <City>Hollywood</City>
```

```
 8              </Address>
 9              <Address>
10                  <Street>5 Locust Ln.</Street>
11                  <City>Malibu</City>
12              </Address>
13          </Star>
14          <Star starID="mh" starredIn="sw">
15              <Name>Mark Hamill</Name>
16              <Street>456 Oak Rd.</Street>
17              <City>Brentwood</City>
18          </Star>
19          <Movie starID="sw" starOf="cf", "mh">
20              <Title>Star Wars</Title>
21              <Year>1977</Year>
22          </Movie>
23      </StarMovieData>
24
```

- Namespaces
  - **Syntax:** xmlns:*name*:URI
  - Is similar to import numpy as np in python
  - Is used to distinguish tags coming from different sources, i.e. HTML

    Example:

    Retireving element *StarMovieData* from document **infolab.stanford.edu/movies**. Set md as the name of import

    ```
    1    <md:StarMovieData xmlns:md="http://infolab.stanford.edu/
         movies">
    2
    ```

b)
```
 1    <? xml version = "1.0" encoding="utf-8" standalone = "yes">
 2    <StarMovieData>
 3        <Star starID="cf" starredIn="sw">
 4            <Name>Carrie Fisher</Name>
 5            <Address>
 6                <Street>123 Maple St.</Street>
 7                <City>Hollywood</City>
 8            </Address>
 9            <Address>
10                <Street>5 Locust Ln.</Street>
11                <City>Malibu</City>
12            </Address>
13        </Star>
14        <Star starID="mh" starredIn="sw">
15            <Name>Mark Hamill</Name>
16            <Street>456 Oak Rd.</Street>
17            <City>Brentwood</City>
18        </Star>
19        <Movie movieID="sw" starsOf="cf", "mh">
20            <Title>Star Wars</Title>
21            <Year>1977</Year>
```

```
22          </Movie>
23          <Movie movieID="esb" starOf="cf", "mh">
24              <Title>Empire Strikes Back</Title>
25              <Year>1980</Year>
26          </Movie>
27          <Movie movieID="roj" starOf="cf", "mh">
28              <Title>Return of Jedi</Title>
29              <Year>1983</Year>
30          </Movie>
31      </StarMovieData>
32
```

c)
```
1       <? xml version = "1.0" encoding="utf-8" standalone = "yes">
2       <StarMovieData>
3           <Star starID="cf" starredIn="sw">
4               <Name>Carrie Fisher</Name>
5               <Address>
6                   <Street>123 Maple St.</Street>
7                   <City>Hollywood</City>
8               </Address>
9               <Address>
10                  <Street>5 Locust Ln.</Street>
11                  <City>Malibu</City>
12              </Address>
13          </Star>
14          <Star starID="mh" starredIn="sw">
15              <Name>Mark Hamill</Name>
16              <Street>456 Oak Rd.</Street>
17              <City>Brentwood</City>
18          </Star>
19          <Movie movieID="sw" starsOf="cf", "mh" movieIn="fx">
20              <Title>Star Wars</Title>
21              <Year>1977</Year>
22          </Movie>
23          <Movie movieID="esb" starOf="cf", "mh" movieIn="fx">
24              <Title>Empire Strikes Back</Title>
25              <Year>1980</Year>
26          </Movie>
27          <Movie movieID="roj" starOf="cf", "mh" movieIn="fx">
28              <Title>Return of Jedi</Title>
29              <Year>1983</Year>
30          </Movie>
31          <Studio studioID="fx" movieOf="esb", "roj", "sw">
32              <Name>Fox</Name>
33              <Address>Hollywood</Address>
34          </Studio>
35      </StarMovieData>
36
```

6. Consider the following relation Classes:

class	type	country	numGuns	bore	displacement
Bismarck	bb	Germany	8	15	42000
Iowa	bb	USA	9	16	46000
Kongo	bc	Japan	8	14	32000
North Carolina	bb	USA	9	16	37000
Renown	bc	Gt. Britain	6	15	32000
Revenge	bb	Gt. Britain	8	15	29000
Tennessee	bb	USA	12	14	32000
Yamato	bb	Japan	9	18	65000

(a) Sample data for relation **Classes**

```xml
<? xml version = "1.0" encoding="utf-8" standalone = "yes">
<shipsData>
    <Classes>
        <Class>Bismarck</Class>
        <Type>bb</Type>
        <Country>Germany</Country>
        <NumGuns>8</NumGuns>
        <Bore>15</Bore>
        <Displacement>42000</Displacement>
    </Classes>
    <Classes>
        <Class>Iowa</Class>
        <Type>bb</Type>
        <Country>USA</Country>
        <NumGuns>9</NumGuns>
        <Bore>16</Bore>
        <Displacement>46000</Displacement>
    </Classes>
    <Classes>
        <Class>Kongo</Class>
        <Type>bc</Type>
        <Country>Japan</Country>
        <NumGuns>8</NumGuns>
        <Bore>14</Bore>
        <Displacement>32000</Displacement>
    </Classes>
    <Classes>
        <Class>North Carolina</Class>
        <Type>bb</Type>
        <Country>USA</Country>
        <NumGuns>9</NumGuns>
        <Bore>16</Bore>
        <Displacement>37000</Displacement>
    </Classes>
    <Classes>
        <Class>Renown</Class>
```

```
37          <Type>bc</Type>
38          <Country>Gt. Britain</Country>
39          <NumGuns>6</NumGuns>
40          <Bore>15</Bore>
41          <Displacement>32000</Displacement>
42      </Classes>
43      <Classes>
44          <Class>Revenge</Class>
45          <Type>bb</Type>
46          <Country>Gt. Britain</Country>
47          <NumGuns>8</NumGuns>
48          <Bore>15</Bore>
49          <Displacement>29000</Displacement>
50      </Classes>
51      <Classes>
52          <Class>Tennessee</Class>
53          <Type>bb</Type>
54          <Country>USA</Country>
55          <NumGuns>12</NumGuns>
56          <Bore>14</Bore>
57          <Displacement>32000</Displacement>
58      </Classes>
59      <Classes>
60          <Class>Yamato</Class>
61          <Type>bb</Type>
62          <Country>Japan</Country>
63          <NumGuns>9</NumGuns>
64          <Bore>18</Bore>
65          <Displacement>65000</Displacement>
66      </Classes>
67  </shipsData>
68
```

7.　　• DocRoot

```
1       <DocRoot docID="" rootElementID="" />
2
```

• SubElement

```
1       <Subelement parentID="" childID="" position="" />
2
```

• ElementAttribute

```
1       <ElementAttribute elementID="" name="" value="" />
2
```

• ElementValue

```
1       <ElementValue elementID="" name="" value="" />
2
```

Notes:

- Empty element
  - is an element that is complete by itself
  - rather than being composed of a start tag, data and an end tag, the empty element is a combined start and end tag.

    Example:

```
1        <phone>
2            <entry>
3                <name>Chip</name>
4                <extension number="3"/>
5            </entry>
6        </phone>
7
```

  - * Phone is not an empty element
  - * extension is an empty element

8. The following will be used as an example.


```
1    <? xml version = "1.0" encoding="utf-8" standalone = "yes">
2    <DocRoot docID="root" rootElementID="">
3        <SubElement parentID="root" childID="cf" position="1">
4            <ElementAttribute elementID="cf-starsIn" name="starsIn"
     value="sw"/>
```

```
 5              <ElementValue elementID="cf-name" name="name" value="Carrie
     Fisher"/>
 6              <SubElement parentID="cf" childID="cf-add_1" position="1">
 7                  <City elementID="cf-add_1-city" value="Hollywood">
 8                  <Street elementID="cf-add_1-street" value="Maple">
 9              </SubElement>
10              <SubElement parentID="cf" childID="cf-add_2" position="2">
11                  <ElementValue elementID="cf-add_2-city" value="Malibu">
12                  <ElementValue elementID="cf-add_1-street" value="Locust"
     >
13              </SubElement>
14          </SubElement>
15          <SubElement parentID="root" childID="mh" position="2">
16              <ElementAttribute elementID="mh-starsIn" name="starsIn"
     value="sw"/>
17              <ElementValue elementID="mh-name" value="Mark Hamill"/>
18              <SubElement parentID="mh" childID="mh-add_2" position="1">
19                  <ElementValue elementID="mh-add_2-city" value="Bollywood
     ">
20                  <ElementValue elementID="mh-add_2-street" value="Oak">
21              </SubElement>
22          </SubElement>
23          <SubElement parentID="root" childID="sw" position="3">
24              <ElementAttribute elementID="mh-starsOf_1" name="starsOf"
     value="cf"/>
25              <ElementAttribute elementID="mh-starsOf_2" name="starsOf"
     value="mh"/>
26              <ElementValue elementID="sw-title" value="Star Wars" />
27              <ElementValue elementID="sw-year" value="1977" />
28          </SubElement>
29      </DocRoot>
30
```

9. a)
```
 1     <? xml version = "1.0" encoding="utf-8" standalone = "yes">
 2     <StarMovieData>
 3         <Star starID="cf" starredIn="sw">
 4             <Name>Carrie Fisher</Name>
 5             <Address>
 6                 <Street>123 Maple St.<Street>
 7                 <City>Hollywood</City>
 8             </Address>
 9             <Address>
10                 <Street>5 Locust Ln.</Street>
11                 <City>Malibu</City>
12             </Address>
13         </Star>
14         <Star starID="mh" starredIn="sw">
15             <Name>Mark Hamill</Name>
16             <Address>
17                 <Street>456 Oak Rd.</Street>
18                 <City>Brentwood</City>
19             </Address>
20         </Star>
21         <Movie movieID="sw" starsOf="cf mh">
```

```
22          <Title>Star Wars</Title>
23          <Year>1977</Year>
24      </Movie>
25      <Movie movieID="esb" starsOf="cf mh">
26          <Title>The Empire Strikes Back</Title>
27          <Year>1980</Year>
28      </Movie>
29      <Movie movieID="roj" starsOf="cf mh">
30          <Title>Return of the Jedi</Title>
31          <Year>1983</Year>
32      </Movie>
33  </StarMovieData>
34
```

```
1   <? xml version = "1.0" encoding="utf-8" standalone = "yes">
2   <StarMovieData>
3       <Star starID="cf" starredIn="sw">
4           <Name>Carrie Fisher</Name>
5           <Address>
6               <Street>123 Maple St.<Street>
7               <City>Hollywood</City>
8           </Address>
9           <Address>
10              <Street>5 Locust Ln.</Street>
11              <City>Malibu</City>
12          </Address>
13      </Star>
14      <Star starID="hf" starredIn="sw fw">
15          <Name>Harrison Ford</Name>
16      </Star>
17
18      <Star starID="mh" starredIn="sw">
19          <Name>Mark Hamill</Name>
20          <Address>
21              <Street>456 Oak Rd.</Street>
22              <City>Brentwood</City>
23          </Address>
24      </Star>
25      <Movie movieID="sw" starsOf="cf mh hf">
26          <Title>Star Wars</Title>
27          <Year>1977</Year>
28      </Movie>
29      <Movie movieID="esb" starsOf="cf mh hf">
30          <Title>The Empire Strikes Back</Title>
31          <Year>1980</Year>
32      </Movie>
33      <Movie movieID="roj" starsOf="cf mh hf">
34          <Title>Return of the Jedi</Title>
35          <Year>1983</Year>
36      </Movie>
37      <Movie movieID="fw" starsOf="hf">
38          <Title>Firewall</Title>
39          <Year>2006</Year>
40      </Movie>
```

```
41        </StarMovieData>
42
```

b)
```
1     <? xml version = "1.0" encoding="utf-8" standalone = "yes">
2     <StarMovieData>
3         <Star starID="cf" starredIn="sw">
4             <Name>Carrie Fisher</Name>
5             <Address>
6                 <Street>123 Maple St.<Street>
7                 <City>Hollywood</City>
8             </Address>
9             <Address>
10                <Street>5 Locust Ln.</Street>
11                <City>Malibu</City>
12            </Address>
13        </Star>
14        <Star starID="hf" starredIn="sw fw">
15            <Name>Harrison Ford</Name>
16        </Star>
17
18        <Star starID="mh" starredIn="sw">
19            <Name>Mark Hamill</Name>
20            <Address>
21                <Street>456 Oak Rd.</Street>
22                <City>Brentwood</City>
23            </Address>
24        </Star>
25        <Movie movieID="sw" starsOf="cf mh hf">
26            <Title>Star Wars</Title>
27            <Year>1977</Year>
28        </Movie>
29        <Movie movieID="esb" starsOf="cf mh hf">
30            <Title>The Empire Strikes Back</Title>
31            <Year>1980</Year>
32        </Movie>
33        <Movie movieID="roj" starsOf="cf mh hf">
34            <Title>Return of the Jedi</Title>
35            <Year>1983</Year>
36        </Movie>
37        <Movie movieID="fw" starsOf="hf">
38            <Title>Firewall</Title>
39            <Year>2006</Year>
40        </Movie>
41        <Movie movieID="hhs" starsOf="cf">
42            <Title>Hannah and Her Sisters</Title>
43            <Year>1985</Year>
44        </Movie>
45    </StarMovieData>
46
```

d)
```
1     <? xml version = "1.0" encoding="utf-8" standalone = "yes">
2     <StarMovieData>
```

```
 3          <Star starID="cf" starredIn="sw">
 4              <Name>Carrie Fisher</Name>
 5              <Address>
 6                  <Street>123 Maple St.<Street>
 7                  <City>Hollywood</City>
 8              </Address>
 9              <Address>
10                  <Street>5 Locust Ln.</Street>
11                  <City>Malibu</City>
12              </Address>
13          </Star>
14          <Star starID="hf" starredIn="sw fw">
15              <Name>Harrison Ford</Name>
16          </Star>
17          <Star starID="mh" starredIn="sw">
18              <Name>Mark Hamill</Name>
19              <Address>
20                  <Street>456 Oak Rd.</Street>
21                  <City>Brentwood</City>
22              </Address>
23          </Star>
24          <Star starID="md" starredIn="bi">
25              <Name>Matt Damon</Name>
26          </Star>
27          <Movie movieID="sw" starsOf="cf mh hf">
28              <Title>Star Wars</Title>
29              <Year>1977</Year>
30          </Movie>
31          <Movie movieID="esb" starsOf="cf mh hf">
32              <Title>The Empire Strikes Back</Title>
33              <Year>1980</Year>
34          </Movie>
35          <Movie movieID="roj" starsOf="cf mh hf">
36              <Title>Return of the Jedi</Title>
37              <Year>1983</Year>
38          </Movie>
39          <Movie movieID="fw" starsOf="hf">
40              <Title>Firewall</Title>
41              <Year>2006</Year>
42          </Movie>
43          <Movie movieID="hhs" starsOf="cf">
44              <Title>Hannah and Her Sisters</Title>
45              <Year>1985</Year>
46          </Movie>
47          <Movie movieID="bi" starsOf="md">
48              <Title>The Bourne Identity</Title>
49              <Year>2002</Year>
50          </Movie>
51      </StarMovieData>
52
```

```
10₁    <!DOCTYPE BankData [
 2          <!ELEMENT BankData (Accounts*, Customers*)>
 3          <!ELEMENT Accounts(acctNo, type, balance)>
```

```
 4              <!ATTLIST Accounts
 5                  acctNo ID #REQUIRED
 6                  acctIn IDREFS #IMPLIED
 7              >
 8          <!ELEMENT Customers(firsrName, lastName, idNo, account)>
 9              <!ATTLIST Accounts
10                  idNo ID #REQUIRED
11                  account IDREFS #IMPLIED
12              >
13          <!ELEMENT acctNo (#PCDATA)>
14          <!ELEMENT type (#PCDATA)>
15          <!ELEMENT balance (#PCDATA)>
16          <!ELEMENT firstName (#PCDATA)>
17          <!ELEMENT lastName (#PCDATA)>
18          <!ELEMENT idNo (#PCDATA)>
19          <!ELEMENT account (#PCDATA)>
20      ]>
21
```

Correct Solution:

```
 1      <!DOCTYPE BankData [
 2          <!ELEMENT BankData (Accounts*, Customers*)>
 3          <!ELEMENT Accounts(acctNo, type, balance)>
 4              <!ATTLIST Accounts
 5                  acctNo ID #REQUIRED
 6                  acctIn IDREFS #IMPLIED
 7              >
 8          <!ELEMENT Customers(firsrName, lastName, idNo, account)>
 9              <!ATTLIST Accounts
10                  idNo ID #REQUIRED
11                  account IDREFS #IMPLIED
12              >
13          <!ELEMENT acctNo (#PCDATA)>
14          <!ELEMENT type (#PCDATA)>
15          <!ELEMENT balance (#PCDATA)>
16          <!ELEMENT firstName (#PCDATA)>
17          <!ELEMENT lastName (#PCDATA)>
18          <!ELEMENT idNo (#PCDATA)>
19          <!ELEMENT account (#PCDATA)>
20      ]>
21
```

11.
```
 1      <!DOCTYPE TeamData [
 2          <!ELEMENT TeamData (Players*, Teams*, Fans*)>
 3          <!ELEMENT Players(FirstName, LastName, Team)>
 4              <!ATTLIST Players
 5                  playerId ID #REQUIRED
 6                  playerIn IDREFS #IMPLIED
 7              >
 8          <!ELEMENT FirstName (#PCDATA)>
```

```
 9          <!ELEMENT LastName (#PCDATA)>
10          <!ELEMENT Team (#PCDATA)>
11          <!ELEMENT Teams(Name)>
12              <!ATTLIST Teams
13                  teamId ID #REQUIRED
14                  playerOf IDREFS #IMPLIED
15                  fanOf IDREFS #IMPLIED
16              >
17          <!ELEMENT Name (#PCDATA)>
18          <!ELEMENT Fans(FirstName, LastName)>
19              <!ATTLIST Fans
20                  faId ID #REQUIRED
21                  fanIn IDREFS #IMPLIED
22              >
23      ]>
24
```

Notes:

- There is no need to write same tags twice. (e.g. FirstName, LastName)

```
12₁    <!DOCTYPE GenealogyData [
 2          <!ELEMENT TeamData (Players*, Teams*, Fans*)>
 3          <!ELEMENT Players(FirstName, LastName, Team)>
 4              <!ATTLIST Players
 5                  playerId ID #REQUIRED
 6                  playerIn IDREFS #IMPLIED
 7              >
 8          <!ELEMENT FirstName (#PCDATA)>
 9          <!ELEMENT LastName (#PCDATA)>
10          <!ELEMENT Team (#PCDATA)>
11          <!ELEMENT Teams(Name)>
12              <!ATTLIST Teams
13                  teamId ID #REQUIRED
14                  playerOf IDREFS #IMPLIED
15                  fanOf IDREFS #IMPLIED
16              >
17          <!ELEMENT Name (#PCDATA)>
18          <!ELEMENT Fans(FirstName, LastName)>
19              <!ATTLIST Fans
20                  faId ID #REQUIRED
21                  fanIn IDREFS #IMPLIED
22              >
23      ]>
24
```

```
13₁    <!DOCTYPE GenealogyData [
 2          <!ELEMENT GenealogyData (Person)>
 3          <!ELEMENT Person(FirstName, LastName, Age)>
 4              <!ATTLIST Person
 5                  personId ID #REQUIRED
 6                  spouse IDREFS #IMPLIED
```

```
7              husband IDREFS #IMPLIED
8              child IDREFS #IMPLIED
9              brother IDREFS #IMPLIED
10       >
11   <!ELEMENT FirstName (#PCDATA)>
12   <!ELEMENT LastName (#PCDATA)>
13 ]>
14
```

Notes:

- PCDATA
  - Is text that will be parsed by a parser.
  - Tags inside the text will be treated as markup and entities will be expanded.
- CDATA
  - Is text that will not be parsed by a parser.
  - Tags inside the text will not be treated as markup and entities will not be expanded
  - e.g. attributes

```
1    <Movie title="Star Wars" year="1977" genre="sciFi">
2
3    <!ELEMENT Movie EMPTY>
4        <!ATTLIST Movie
5            title CDATA #REQUIRED
6            year  CDATA #REQUIRED
7            genre (comedy | drama | sciFi | teen) #IMPLIED
8        >
9
10
```

```
14  1  <!DOCTYPE ShipBattleData [
2
3    ]>
4
```

15. **Notes:**

- XML Schema
  - is an alternative way to provide a schema for XML documents
  - is more powerful than DTD
    * allows to declare types such as integer or float
    * gives ability to declare keys and foreign keys

- The Form of an XML Schema
    - Each XML-Schema document has the form

```
1    <? xml version = "1.0" encoding = "utf-8" ?>
2
3    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4        ...
5    </xs:schema>
6
```

- Elements
- **Syntax:**

```
1    <xs:element name = element name type = element type>
2        constraints and / or structure information
3    </xs:element>
4
```

Example:

```
1    <xs:element name = "title" type = "xs:string" />
2    <xs:element name = "Year" type = "xs:integerg" />
3
4
```

- Complex Types
    - complex type element is an XML element that contains other elements and/or attributes. [1]

References:

1. w3School : Complex Type, link