Evidence

Module 05- Introducing the Markup Languages
Part II – Introducing the XML

[XML BASICS]

Evidence 1

An XML document is parsed by Internet Explorer and the Following output is displayed

```
<?xml version="1.0" ?>
- <!--
Character entities and Special characters example
 -->
-<examples>
        <example>Show & Tell Consulting ®</example>
        <example>©MIcrosof Corporation 2000</example>
        <example>7 is > 6</example>
        <example>This is a trainee's exam rsult</example>
</examples>
```

Write an XML document using notepad and make sure that the document can be parsed without error.

[DTD BASICS]

Evidence 2

You have the following XML file idbbisew.xml

```
<?xml version='1.0'?>
<scholarship>
        <runby>IDB-BISEW<runby>
        <objective>To improve skills of underprivileged Muslim youth in IT field<objective>
</scholarship>
```

Now declare two parsed **character entities** for the text phrases"IDB-BISEW" and "To improve skills of underprivileged Muslim youth in IT field".
Rewrite the XML document referencing the two entities in the element contents.
Use internal DTD subset. You do not need to declare elements or validate the document.

Evidence 3

You have the following sample XML file containing a IDB-BISEW scholarship course.

```
<?xml version='1.0'?>
<course course-id="ESAD-CS">
        <name>Enterprise System Analysis and Design usin Visusl C Sharp</name>
        <modules>
                <module moduleid="ESADCS-01">
                        <name>Programming Concepts</name>
                        <duration>80</duration>
                </module>
                <module moduleid="ESADCS-02">
                        <name>Designing and implementing databases</name>
                        <duration>60</duration>
                </module>
        </modules>
</course>
```

You have the following constraints to apply on the XML element and attributes

- course-id attribute must appear and its value must be ESAD-CS, ESAD-VB or NT
- module-id attribute is required and must unique in the documents
- modules must contain at least one module

Write a DTD file and validate the XML document.

Evidence 4

Consider the following facts

- A person's name consists of first-name, middle-name, and last-name in order as written
- First-name, last-name and middle-name – each one contains text data
- Some person has no middle-name

- ▪ First-name, last-name can appear only once

Tasks:
Write DTD code fragments for the above content model and specification.
Write sample XML content
Write DTD rules as internal subset in the XML document

Evidence 5

You have to create persons Document Type. Structure is described below
Persons

- ↘ persons contains zero one or more person elements
    - ↘ person element has an optional title attribute
        - ↘ title attribute may have “Mr” or “Mrs” as the value person element contains name element
        - ↘ name element contains text data

Tasks
Create an external DTD file
Write an example XML file with sample data and associate the external DTD reference
Validate the XML document

[SCRIPTING XML]

Evidence 6

You want to separate your HTML user interface from content. You stored data in XML in following structure

```
<?xml version=’1.0’>
<trainee id=”107898”>
        <name>Nazrul Islam</name>
        <email>nazrul@idb-bisew.org</email>
        <course>ESAD-CS</course>
        <round>07</round>
</trainee>
```

Tasks:
Create an HTML template
Show data on the HTML template using JavaScript as middle layer application

Evidence 7

You have the following XML data in book.xml

```
<?xml version="1.0"?>
<book>
        <name>SQL</name>
        <price>700.00</price>
        <author>
                <first-name>James</first-name>
                <last-name>Scott</last-name>
        </author>
</book>
```

Using Java Script show the xml data on an HTML page like below

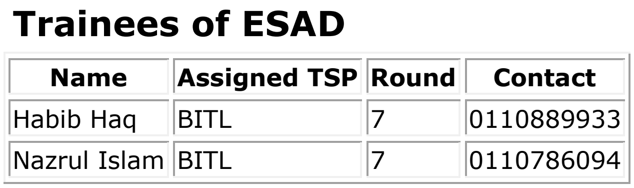
Book Information	
Name:	SQL
Author:	James Scott
Price:	700.00

Evidence 8

You have the following XML data in trainees.xml

```
<?xml version="1.0"?>
<trainees>
        <trainee course="ESAD" round="7">
                <name>Habib Haq</name>
                <assignedto>BITL</assignedto>
                <contact>0110889933</contact>
        </trainee>
        <trainee batch="NT" round="7">
                <name>Amin Haq</name>
                <assignedto>DITL</assignedto>
                <contact>0110887766</contact>
        </trainee>
        <trainee course ="ESAD" round="7">
                <name>Nazrul Islam</name>
                <assignedto>BITL</assignedto>
                <contact>0110786094</contact>
        </trainee>
</trainees>
```

You have to show the information of **trainees in ESAD course** on an HTML file using **JavaScript**. The out will be as shown below



Trainees of ESAD

Name	Assigned TSP	Round	Contact
Habib Haq	BITL	7	0110889933
Nazrul Islam	BITL	7	0110786094

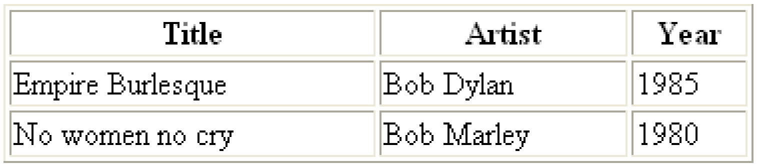
Evidence 9

You have the following XML in file catalog.xml. Content is as below

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
<CD>
<TITLE>No women no cry</TITLE>
<ARTIST>Bob Marley</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Sony</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1980</YEAR>
</CD>
</CATALOG>
```

Now you have to populate an HTML table using xmldso or XML Data Islands

A sample output is shown below



Title	Artist	Year
Empire Burlesque	Bob Dylan	1985
No women no cry	Bob Marley	1980

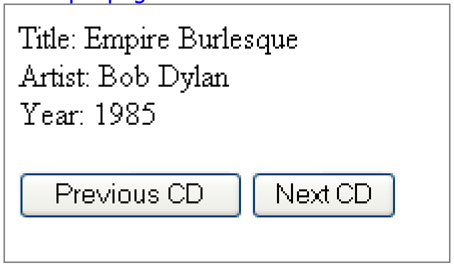
[XML DSO and XML Data ISLANDS]

Evidence 10

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
```

```
</CD>
<CD>
<TITLE>No women no cry</TITLE>
<ARTIST>Bob Marley</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Sony</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1980</YEAR>
</CD>
</CATALOG>
```

Now you have to show CD information on an HTML page.
You have to show one CD at a time and arrange navigation facilities so that users can move to next or previous CD.
A sample page is shown below



Title: Empire Burlesque
Artist: Bob Dylan
Year: 1985

[XSL BASICS]

Evidence 11

You have an xml file like below:

```
<?xml version=’1.0’>
<trainee id=”107898”>
            <name>Nazrul Islam</name>
            <email>nazrul@idb-bisew.org</email>
            <course>ESAD-CS</course>
            <round>07</round>
</trainee>
```

Create XSL rules to display XML data in HTML as below

Trainee Information	
Name	Nazrul Islam
ID	107898
Email	nazrul@idb-bisew.org
Course	ESAD-CS
Round	07

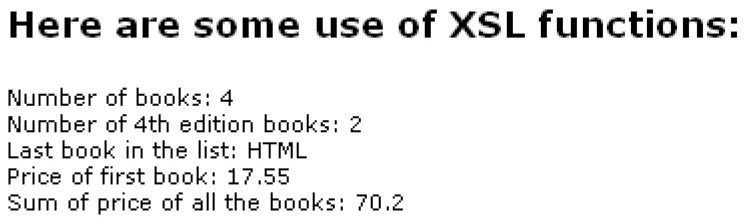
Evidence 12

Write the following XML file

```
<?xml version="1.0"?>
<books>
            <book isbn="10-90-887-909">
                        <name>SQL ins and outs</name>
                        <author>Clerk</author>
                        <listprice>24.99</listprice>
                        <price>17.55</price>
                        <edition>4</edition>
                        <publisher>QUE</publisher>
            </book>
            <book isbn="10-90-3467-819">
                        <name>OOSAD using UML</name>
                        <author>Bennet</author>
                        <listprice>24.99</listprice>
                        <price>17.55</price>
                        <edition>3</edition>
                        <publisher>Prentice Hall</publisher>
            </book>
            <book isbn="10-90-3467-819">
                        <name>XML</name>
```

```
        <author>Bennet</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>2</edition>
        <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819">
        <name>HTML</name>
        <author>Bennet</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>4</edition>
        <publisher>Prentice Hall</publisher>
    </book>
</books>
```

Now create a XSL file and write appropriate XSLT rules so that a XSLT processor generate following HTML output



Here are some use of XSL functions:

Number of books: 4
Number of 4th edition books: 2
Last book in the list: HTML
Price of first book: 17.55
Sum of price of all the books: 70.2

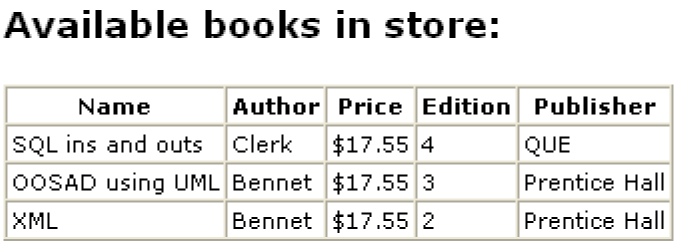
Evidence 13

Write the following XML file below

```
<?xml version="1.0"?>
<books>
    <book isbn="10-90-887-909" available="yes">
        <name>SQL ins and outs</name>
        <author>Clerk</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>4</edition>
        <publisher>QUE</publisher>
    </book>
    <book isbn="10-90-3467-819" available="no">
        <name>Programming practice</name>
        <author>Dave Hart</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>4</edition>
        <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819" available="yes">
        <name>OOSAD using UML</name>
        <author>Bennet</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>3</edition>
        <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819" available="yes">
        <name>XML</name>
        <author>Bennet</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>2</edition>
        <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819" available="no">
        <name>HTML</name>
        <author>Bennet</author>
        <listprice>24.99</listprice>
```

```
                    <price>17.55</price>
                    <edition>4</edition>
                    <publisher>Prentice Hall</publisher>
          </book>
</books>
```

Now create a XSL file and write appropriate XSLT rules so that a XSLT processor generate following HTML output (look carefully it only shows books those are available)



Available books in store:

Name	Author	Price	Edition	Publisher
SQL ins and outs	Clerk	$17.55	4	QUE
OOSAD using UML	Bennet	$17.55	3	Prentice Hall
XML	Bennet	$17.55	2	Prentice Hall

[XSD BASICS]

Evidence 14

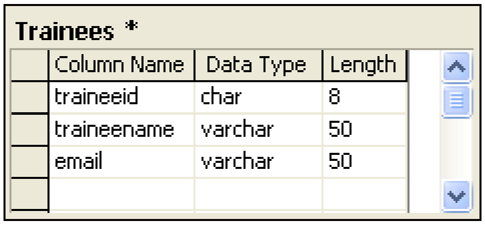
Write the XML File below

```
<?xml version="1.0"?>
<trainee id="109878">
                    <name>Al Amin</name>
                    <course>ESAD</course>
                    <round>5</round>
</trainee>
```

Now create XSD schema for the XML document in an external file.
id attribute in trainee element should always be present.
Attach the XSD file in the xml file so that a validating parser can validate it

Evidence 15

Looks at the table definition



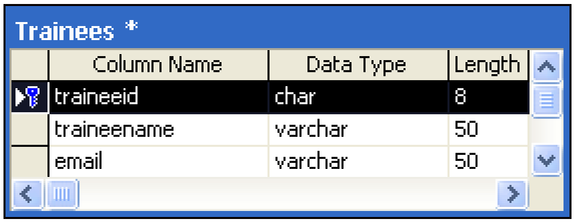
Trainees *

	Column Name	Data Type	Length
	traineeid	char	8
	traineename	varchar	50
	email	varchar	50

You have to convert the above definition into XSD schema
Create your own types to apply constrains on the data
Create a sample XML file

Evidence 16

Looks at the table definition



Trainees *

	Column Name	Data Type	Length
▶🔑	traineeid	char	8
	traineename	varchar	50
	email	varchar	50

You have to convert the above definition into XSD schema
Create your own types to apply constrains on the data
Create a sample XML file
[You have to apply primary-key constraint]