

# CSLR 51 : DBMS LAB-9

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1)

a. Demonstrating DTD:

**1. filename :** *bookstore.dtd*

```
<!ELEMENT bookstore (book+)>
<!ELEMENT book (title,author,category,isbn,publisher,edition,price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

**2. filename :** *bookstore.xml*

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
  <bookstore>
    <book>
      <title>XML Developer's Guide</title>
      <author>Gambardella, Matthew</author>
      <category>textbook</category>
      <isbn>111111</isbn>
      <publisher>wiley</publisher>
      <edition>second</edition>
      <price>20.00</price>
    </book>
```

```
<book>
  <title>Maeve Ascendant</title>
  <author>Corets, Eva</author>
  <category>textbook</category>
  <isbn>222222</isbn>
  <publisher>tata</publisher>
  <edition>second</edition>
  <price>60.00</price>
</book>
<book>
  <title>Midnight Rain</title>
  <author>Ralls, Kim</author>
  <category>fiction</category>
  <isbn>333333</isbn>
  <publisher>O'reilly</publisher>
  <edition>second</edition>
  <price>100.00</price>
</book>
</bookstore>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

-<bookstore>
  -<book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>111111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  -<book>
    <title>Maevae Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  -<book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'relilly</publisher>
    <edition>second</edition>
    <price>100.00</price>
  </book>
</bookstore>

```

## b. Demonstrating XSD:

### 1. **filename :** *bookstore.xml*

```

<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bookstore">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="book" minOccurs="1" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="book">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="title" minOccurs="1" maxOccurs="1" />
            <xs:element ref="author" minOccurs="1" maxOccurs="1" />
            <xs:element ref="category" minOccurs="1" maxOccurs="1" />
            <xs:element ref="isbn" minOccurs="1" maxOccurs="1" />
            <xs:element ref="publisher" minOccurs="1" maxOccurs="1" />
            <xs:element ref="edition" minOccurs="1" maxOccurs="1" />
            <xs:element ref="price" minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="title" type="xs:string" />
<xs:element name="author" type="xs:string" />
<xs:element name="category" type="xs:string" />
<xs:element name="isbn" type="xs:string" />
<xs:element name="publisher" type="xs:string" />
<xs:element name="edition" type="xs:string" />
<xs:element name="price" type="xs:string" />
</xs:schema>

```

## 2. filename : *bookstore.css*

```
bookstore {  
    color: orange;  
}  
book {  
    color: Red;  
}  
title {  
    color: blue;  
    font-weight: bold;  
    margin-left: 10pt;  
    display: block;  
}  
author {  
    color: red;  
    font-weight: bold;  
    margin-left: 10pt;  
}  
category {  
    color: grey;  
    font-weight: bold;  
    margin-left: 10pt;  
}  
isbn {  
    color: green;  
    font-weight: bold;  
    margin-left: 10pt;  
}  
edition {  
    color: red;  
    font-weight: bold;  
    margin-left: 10pt;  
}
```

```
publisher {  
  color: green;  
  font-weight: bold;  
  margin-left: 10pt;  
}
```

### 3. filename : *bookstore.xsl*

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/css" href="bookstore.css"?>  
<bookstore xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="bookstore.xsd">  
  <book>  
    <title>XML Developer's Guide</title>  
    <author>Gambardella, Matthew</author>  
    <category>textbook</category>  
    <isbn>111111</isbn>  
    <publisher>wiley</publisher>  
    <edition>second</edition>  
    <price>20.00</price>  
  </book>  
  <book>  
    <title>Maeve Ascendant</title>  
    <author>Corets, Eva</author>  
    <category>textbook</category>  
    <isbn>222222</isbn>  
    <publisher>tata</publisher>  
    <edition>second</edition>  
    <price>60.00</price>  
  </book>  
  <book>  
    <title>Midnight Rain</title>  
    <author>Ralls, Kim</author>
```



```

    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
    <price>100.00</price>
  </book>
</bookstore>

```



### c. Demonstrating XSL:

#### 1. **filename :** *bookstore.xsl*

```

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>bookstore</title>

```



```
</head>
<body>
  <table border="1">
    <tr>
      <th>title</th>
      <th>author</th>
      <th>category</th>
      <th>isbn</th>
      <th>publisher</th>
      <th>edition</th>
      <th>price</th>
    </tr>
    <xsl:for-each select="/bookstore/book">
      <tr>
        <td bgcolor="green">
          <xsl:value-of select="title" />
        </td>
        <td bgcolor="red">
          <xsl:value-of select="author" />
        </td>
        <td bgcolor="grey">
          <xsl:value-of select="category" />
        </td>
        <td bgcolor="cyan">
          <xsl:value-of select="isbn" />
        </td>
        <td bgcolor="yellow">
          <xsl:value-of select="publisher" />
        </td>
        <td bgcolor="silver">
          <xsl:value-of select="edition" />
        </td>
        <td bgcolor="blue">
```

```

                <xsl:value-of select="price" />
            </td>
        </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## 2. filename : *bookstore.xml*

```

<?xml-stylesheet type="text/xsl" href="bookstore.xsl"?>
<bookstore>
    <book>
        <title>XML Developer's Guide</title>
        <author>Gambardella, Matthew</author>
        <category>textbook</category>
        <isbn>111111</isbn>
        <publisher>wiley</publisher>
        <edition>second</edition>
        <price>20.00</price>
    </book>
    <book>
        <title>Maeve Ascendant</title>
        <author>Corets, Eva</author>
        <category>textbook</category>
        <isbn>222222</isbn>
        <publisher>tata</publisher>
        <edition>second</edition>
        <price>60.00</price>
    </book>
    <book>

```

```

<title>Midnight Rain</title>
<author>Ralls, Kim</author>
<category>fiction</category>
<isbn>333333</isbn>
<publisher>O'relilly</publisher>
<edition>second</edition>
<price>100.00</price>
</book>
</bookstore>

```

bookstore x +

File | D:/Documents/NIT%20Trichy/V-Sem/DBMS-LAB/Lab-9/c.%20Demonstrating%20XSL/bookstore.html

title	author	category	isbn	publisher	edition	price
XML Developer's Guide	Gambardella, Matthew	textbook	111111	wiley	second	20.00
Maeve Ascendant	Corets, Eva	textbook	222222	tata	second	60.00
Midnight Rain	Ralls, Kim	fiction	333333	O'relilly	second	100.00

## 2 ) Perform the following queries using XPath :

- (i) Find the title and price of non-fiction books with a price more than 50 USD

**Xpath query :** `/bookstore/book[category!="fiction" and price>50.00]/(title | isbn)`

The screenshot shows the XML Editor interface with the following components:

- XML Document (bookstore.xml):**

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>111111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maevae Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
  </book>
</bookstore>
```
- XPath/XQuery Builder:** The query `/bookstore/book[category!="fiction" and price>50.00]/(title | isbn)` is entered.
- Results:** A table showing the results of the query.

Description - 2 Items	XPath location	Resource	System ID	Location
Maevae Ascendant	/bookstore[1]/book[2]/title[1]	bookstore.xml	D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-9\1\1a. D...	14:17
222222	/bookstore[1]/book[2]/isbn[1]	bookstore.xml	D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-9\1\1a. D...	17:17

## (ii) Find average price of textbooks

**Xpath query:**  $\text{sum}(\text{bookstore/book}[\text{category}=\text{"textbook"}]/\text{price}/\text{number}(\text{text()})) \text{ div } \text{count}(\text{bookstore/book}[\text{category}=\text{"textbook"}]/\text{price})$

The screenshot shows the XML Editor interface with the following components:

- bookstore.xml**: XML document content.

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>11111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maevae Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
  </book>
</bookstore>
```
- XPath/XQuery Builder**: XPath 2.0 query editor.

Scope: Current File

Query:  $\text{sum}(\text{bookstore/book}[\text{category}=\text{"textbook"}]/\text{price}/\text{number}(\text{text()})) \text{ div } \text{count}(\text{bookstore/book}[\text{category}=\text{"textbook"}]/\text{price})$
- Results**: Table showing the result of the query.

Description - 1 item	XPath location	Resource	System ID	Location
40.0	XPath location - Not available	bookstore.xml	D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-9\1\...	

Text | Grid | Aut... XPath - bookstore.xml

D:\...\DBMS-LAB\Lab-9\1\1\... XPath - successful ( 0.0s ) U+003C 14 : 17 @ Days left for evaluation: 29 37 new message(s)

### (iii) Find the titles of textbooks on XML

**Xpath query:** `/bookstore/book[category="textbook" and contains(title, "XML")]/title/text()`

The screenshot displays the XML Editor interface with the file `bookstore.xml` open. The XML content is as follows:

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>111111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maevae Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
  </book>
</bookstore>
```

The XPath/XQuery Builder window shows the query: `/bookstore/book[category="textbook" and contains(title, "XML")]/title/text()`. The Results window displays the following table:

Description - 1 Item	XPath location	Resource	System ID
XML Developer's Guide	<code>/bookstore[1]/book[1]/title[1]/text()[1]</code>	bookstore.xml	D:\Documents\WIT Trichy\V-Sem\DBMS-LAB\Lab-9\1

The status bar at the bottom indicates: `XPath - successful ( 0.0s )`.

## 1) Perform the following queries using XQuery:

- (i) Create a new document which contain only the isbn and title of textbooks

### XQuery

```
<textbook>
{
  for $book in doc("bookstore.xml") // book
  where $book[category = "textbook"]
  return <textbook>{$book/title}{$book/isbn}</textbook>
}
</textbook>
```

The screenshot displays the XML Editor (Academic use only) and the XQuery Builder interface. The XML Editor shows the 'bookstore.xml' file with the following structure:

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>11111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maeve Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>22222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>33333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
    <price>100.00</price>
  </book>
</bookstore>
```

The XQuery Builder shows the following query:

```
<textbook>
{
  for $book in doc("bookstore.xml") // book
  where $book[category = "textbook"]
  return <textbook>{$book/title}{$book/isbn}</textbook>
}
</textbook>
```

The Results pane shows the output of the query:

```
1. textbook
1 <?xml version="1.0" encoding="UTF-8"?>
2 <textbook>
3   <textbook>
4     <title>XML Developer's Guide</title>
5     <isbn>11111</isbn>
6   </textbook>
7   <textbook>
8     <title>Maeve Ascendant</title>
9     <isbn>22222</isbn>
10  </textbook>
11 </textbook>
```

The status bar at the bottom indicates: D:\...DBMS-LAB\Lab-9\Lab-9.a. Demonstrating DTD(bookstore.xml) Validation successful. U+000A 18 : 44 Days left for evaluation: 28 37 new mess...



(ii) Find the title and price of the book with isbn “222222”.

### XQuery :

```
for $book in doc("bookstore.xml") // book where $book[isbn="222222"]
return <book>{ $book/title, $book/price}</book>
```

The screenshot displays the XML Editor and XPath/XQuery Builder interface. The XML Editor on the left shows the 'bookstore.xml' file with the following content:

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>11111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maevae Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
    <price>100.00</price>
  </book>
</bookstore>
```

The XPath/XQuery Builder on the right shows the XQuery:

```
1 for $book in doc("bookstore.xml") // book where $book[isbn="222222"]
2 return <book>{ $book/title, $book/price}</book>
```

The Results pane at the bottom shows the output of the XQuery:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <book>
3   <title>Maevae Ascendant</title>
4   <price>60.00</price>
5 </book>
```

The status bar at the bottom indicates that the validation was successful.

(iii) Produce a list of non-fictions with their title and price, sorted by price

### XQuery :

```
<nonfiction-list>
{
  for $book in doc("bookstore.xml")//book, $title in $book/title, $price in $book/price
  where $book/category!="fiction" order by $price/text()
  return
    <nonfiction>{$title, $price}</nonfiction>
}
</nonfiction-list>
```

The screenshot displays the XML Editor and XPath/XQuery Builder interface. The XML Editor on the left shows the 'bookstore.xml' file with the following structure:

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>111111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maevae Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
    <price>100.00</price>
  </book>
</bookstore>
```

The XPath/XQuery Builder on the right shows the XQuery:

```
<nonfiction-list>
{
  for $book in doc("bookstore.xml")//book, $title in $book/title, $price in $book/price
  where $book/category!="fiction" order by $price/text()
  return
    <nonfiction>{$title, $price}</nonfiction>
}
</nonfiction-list>
```

The Results pane shows the output of the XQuery:

```
1. nonfiction-list
1 <?xml version="1.0" encoding="UTF-8"?>
2 <nonfiction-list>
3   <nonfiction>
4     <title>XML Developer's Guide</title>
5     <price>20.00</price>
6   </nonfiction>
7   <nonfiction>
8     <title>Maevae Ascendant</title>
9     <price>60.00</price>
10  </nonfiction>
11 </nonfiction-list>
```

The status bar at the bottom indicates: Validation successful, U+000A, 18 : 44, Days left for evaluation: 28, 37 new mess...

(iv) Find title of the textbook with highest price

**XQuery :**

```
<textbooks>
{ let $prices := doc("bookstore.xml")// book[category="textbook"]/price let $max := max($prices)
return
<max-price-textbook price="{ $max }">
{for $book in doc("bookstore.xml")// book where $book/price = $max return $book/title}
</max-price-textbook>
}
</textbooks>
```

The screenshot displays the XML Editor and XPath/XQuery Builder interface. The XML Editor on the left shows the 'bookstore.xml' file with the following structure:

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "bookstore.dtd">
<bookstore>
  <book>
    <title>XML Developer's Guide</title>
    <author>Gambardella, Matthew</author>
    <category>textbook</category>
    <isbn>111111</isbn>
    <publisher>wiley</publisher>
    <edition>second</edition>
    <price>20.00</price>
  </book>
  <book>
    <title>Maeve Ascendant</title>
    <author>Corets, Eva</author>
    <category>textbook</category>
    <isbn>222222</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>60.00</price>
  </book>
  <book>
    <title>Midnight Rain</title>
    <author>Ralls, Kim</author>
    <category>fiction</category>
    <isbn>333333</isbn>
    <publisher>O'reilly</publisher>
    <edition>second</edition>
    <price>100.00</price>
  </book>
</bookstore>
```

The XPath/XQuery Builder on the right shows the following XQuery:

```
<textbooks>
2 { let $prices := doc("bookstore.xml")// book[category="textbook"]/price let $max := max($prices)
3 return
4 <max-price-textbook price="{ $max }">
5 {for $book in doc("bookstore.xml")// book where $book/price = $max return $book/title}
6 </max-price-textbook>
7 }
8 </textbooks>
```

The Results pane at the bottom shows the output of the XQuery:

```
1. textbooks
1 <?xml version="1.0" encoding="UTF-8"?>
2 <textbooks>
3 <max-price-textbook price="60">
4 <title>Maeve Ascendant</title>
5 </max-price-textbook>
6 </textbooks>
```