```
-------------------------------------------------------------------
 * books.dtd *
-------------------------------------------------------------------

<!ELEMENT bib (book* )>
<!ELEMENT book (title, (author+ | editor+ ), publisher, price )>
<!ATTLIST book year CDATA #REQUIRED >
<!ELEMENT author (last, first )>
<!ELEMENT editor (last, first, affiliation )>
<!ELEMENT title (#PCDATA )>
<!ELEMENT last (#PCDATA )>
<!ELEMENT first (#PCDATA )>
<!ELEMENT affiliation (#PCDATA )>
<!ELEMENT publisher (#PCDATA )>
<!ELEMENT price (#PCDATA )>

-------------------------------------------------------------------
 * books.xml *
-------------------------------------------------------------------

<bib>

        <book year="1994">
                <title>TCP/IP Illustrated</title>
                <author>
                        <last>Stevens</last>
                        <first>W.</first>
                </author>
                <publisher>Addison-Wesley</publisher>
                <price>65.95</price>
        </book>

        <book year="1992">
                <title>Advanced Programming in the UNIX Environment</title>
                <author>
                        <last>Stevens</last>
                        <first>W.</first>
                </author>
                <publisher>Addison-Wesley</publisher>
                <price>65.95</price>
        </book>

        <book year="2000">
                <title>Data on the Web</title>
                <author>
                        <last>Abiteboul</last>
                        <first>Serge</first>
                </author>
                <author>
                        <last>Buneman</last>
                        <first>Peter</first>
                </author>
                <author>
                        <last>Suciu</last>
                        <first>Dan</first>
                </author>
                <publisher>Morgan Kaufmann Publishers</publisher>
                <price>65.95</price>
        </book>

        <book year="1999">
                <title>The Economics of Technology and Content for Digital TV</title>
                <editor>
                        <last>Gerbarg</last>
                        <first>Darcy</first>
                        <affiliation>CITI</affiliation>
                </editor>
                <publisher>Kluwer Academic Publishers</publisher>
                <price>129.95</price>
        </book>
</bib>


-------------------------------------------------------------------
```

```
 * Expression 1 * - Selects all book elements
-------------------------------------------------------------------


doc("books.xml")/bib/book


-------------------------------------------------------------------
 * Expression 2 * - Selects all book elements (regardless of level)
-------------------------------------------------------------------


doc("books.xml")//book


-------------------------------------------------------------------
 * Expression 3 * - Selects all authors with last = "Stevens"
-------------------------------------------------------------------


doc("books.xml")/bib/book/author[last="Stevens"]


-------------------------------------------------------------------
 * Expression 4 * - Returns the first author of each book
-------------------------------------------------------------------


doc("books.xml")/bib/book/author[1]


-------------------------------------------------------------------
 * Expression 5 * - Returns the year attribute of each book
-------------------------------------------------------------------


doc("books.xml")/bib/book/@year


-------------------------------------------------------------------
 * Expression 6 * - Wildcard - Returns all elements of book 1
-------------------------------------------------------------------


doc("books.xml")//book[1]/*


--------------------------------------------------------------------
 * Expression 7 * - Returns the title of all books published in 2000
--------------------------------------------------------------------


for $b in doc("books.xml")//book
where $b/@year = "2000"
return $b/title

//Result:

<title>Data on the Web</title>


--------------------------------------------------------------------
 * Expression 8 * - Returns title of each book + count of authors
--------------------------------------------------------------------


for $b in doc("books.xml")//book
let $c := $b/author
return <book>{ $b/title, <count>{ count($c) }</count>}</book>

//Result:

<book>
        <title>TCP/IP Illustrated</title>
        <count>1</count>
</book>
<book>
        <title>Advanced Programming in the UNIX Environment</title>
        <count>1</count>
</book>
<book>
        <title>Data on the Web</title>
        <count>3</count>
</book>
<book>
        <title>The Economics of Technology and Content for Digital TV</title>
        <count>0</count>
</book>


--------------------------------------------------------------------
 * Expression 9 * - Returns books whose prices are < $50.0
--------------------------------------------------------------------

for $b in doc("books.xml")//book
where $b/price < 50.00
return $b/title

//Result:

<title>Data on the Web</title>


--------------------------------------------------------------------
 * Expression 10 * - Returns title of books with > 2 authors
--------------------------------------------------------------------


for $b in doc("books.xml")//book
let $c := $b//author
where count($c) > 2
return $b/title

//Result:

<title>Data on the Web</title>


--------------------------------------------------------------------
 * Expression 11 * - Returns title of books with > 2 authors
--------------------------------------------------------------------


for $t in doc("books.xml")//title
order by $t
return $t

//Result:

<title>Advanced Programming in the Unix Environment</title>
<title>Data on the Web</title>
<title>TCP/IP Illustrated</title>
<title>The Economics of Technology and Content for Digital TV</title>


---------------------------------------------------------------------------
 * Expression 12 * - Returns title of books sorted by name of first author
---------------------------------------------------------------------------


let $b := doc("books.xml")//book
for $t in distinct-values($b/title)
let $a1 := $b[title=$t]/author[1]
order by $a1/last, $a1/first
return $b/title

//Result:
```

```
<title>The Economics of Technology and Content for Digital TV<title>
<title>Data on the Web</title>
<title>Advanced Programming in the UNIX Environment</title>
<title>TCP/IP Illustrated</title>


---------------------------------------------------------------------
 * Expression 13 * - Returns new element of authors name (first,last)
---------------------------------------------------------------------


for $a in doc("books.xml")//author
return
<author>{ string($a/first), " ", string($a/last) }</author>

//Result:

<author>W. Stevens</author>
<author>W. Stevens</author>
<author>Serge Abiteboul</author>
<author>Peter Buneman</author>
<author>Dan Suciu</author>


-------------------------------------------------------------------------------
 * Expression 14 * - Returns the last names of each author (ignoring dupicates)
-------------------------------------------------------------------------------


for $l in distinct-values(doc("books.xml")//author/last)
return <last>{ $l }</last>

//Result:

<last>Stevens</last>
<last>Abiteboul</last>
<last>Buneman</last>
<last>Suciu</last>


-------------------------------------------------------------------------------
 * Expression 15 * - Compares and returns elements across 2 documents
-------------------------------------------------------------------------------


for $t in doc("books.xml")//title,
$e in doc("reviews.xml")//entry
where $t = $e/title
return <review>{ $t, $e/remarks }</review>

//Result:

<review>
<title> TCP/IP Illustrated</title>
<remarks>Excellent technical content. Not much plot.</remarks>
</review>


-------------------------------------------------------------------------------
 * Expression 16 * - Nesting queries within elements
-------------------------------------------------------------------------------


<listings>
{
for $p in distinct-values(doc("books.xml")//publisher)
order by $p
return
<result>
{ $p }
{
for $b in doc("books.xml")/bib/book
where $b/publisher = $p
order by $b/title
return $b/title
}
</result>
}
</listings>


//Result:

<listings>
<result>
<publisher>Addison-Wesley</publisher>
<title>Advanced Programming in the Unix Environment</title>
<title>TCP/IP Illustrated</title>
</result>
<result>
<publisher>Kluwer Academic Publishers</publisher>
<title>The Economics of Technology and Content for Digital TV</title>
</result>
<result>
<publisher>Morgan Kaufmann Publishers</publisher>
<title>Data on the Web</title>
</result>
</listings>


--------------------------------------------------------------------------------
 * Expression 17 * - Using quantifiers
--------------------------------------------------------------------------------


for $b in doc("books.xml")//book
where every $a in $b/author
satisfies ($a/last="Stevens" and $a/first="W.")
return $b/title

//Result:

<title>TCP/IP Illustrated</title>
<title>Advanced Programming in the Unix Environment</title>
<title>The Economics of Technology and Content for Digital TV</title>


--------------------------------------------------------------------------------
 * Expression 18 * - Using quantifiers, again
--------------------------------------------------------------------------------


for $b in doc("books.xml")//book
where $b/author/first = "Serge"
and $b/author/last = "Suciu"
return $b

//Result:

<book year = "2000">
<title>Data on the Web</title>
<author>
<last>Abiteboul</last>
<first>Serge</first>
</author>
<author>
<last>Buneman</last>
<first>Peter</first>
</author>
<author>
<last>Suciu</last>
<first>Dan</first>
</author>
<publisher>Morgan Kaufmann Publishers</publisher>
<price>39.95</price>
</book>
```

```
-------------------------------------------------------------------------------
 * Expression 19 * - Average function
-------------------------------------------------------------------------------


let $b := doc("books.xml")//book
let $avg := average( $b//price )
return $b[price > $avg]

//Result:

<book year = "1999">
<title>The Economics of Technology and Content for
Digital TV</title>
<editor>
<last>Gerbarg</last>
<first>Darcy</first>
<affiliation>CITI</affiliation>
</editor>
<publisher>Kluwer Academic Publishers</publisher>
<price>129.95</price>
</book>

//Other familiar functions in XQuery include:

        - min()
        - max()
        - count()
        - sum()
        - avg()
        - round()
        - floor()
        - ceiling()

        -       concat()
        - string-length()
        - starts-with()
        - ends-with()
        - substring()
        -       upper-case()
        -lower-case();
```