Networks and Systems –Databases

Practical 2: XML
(model answers)

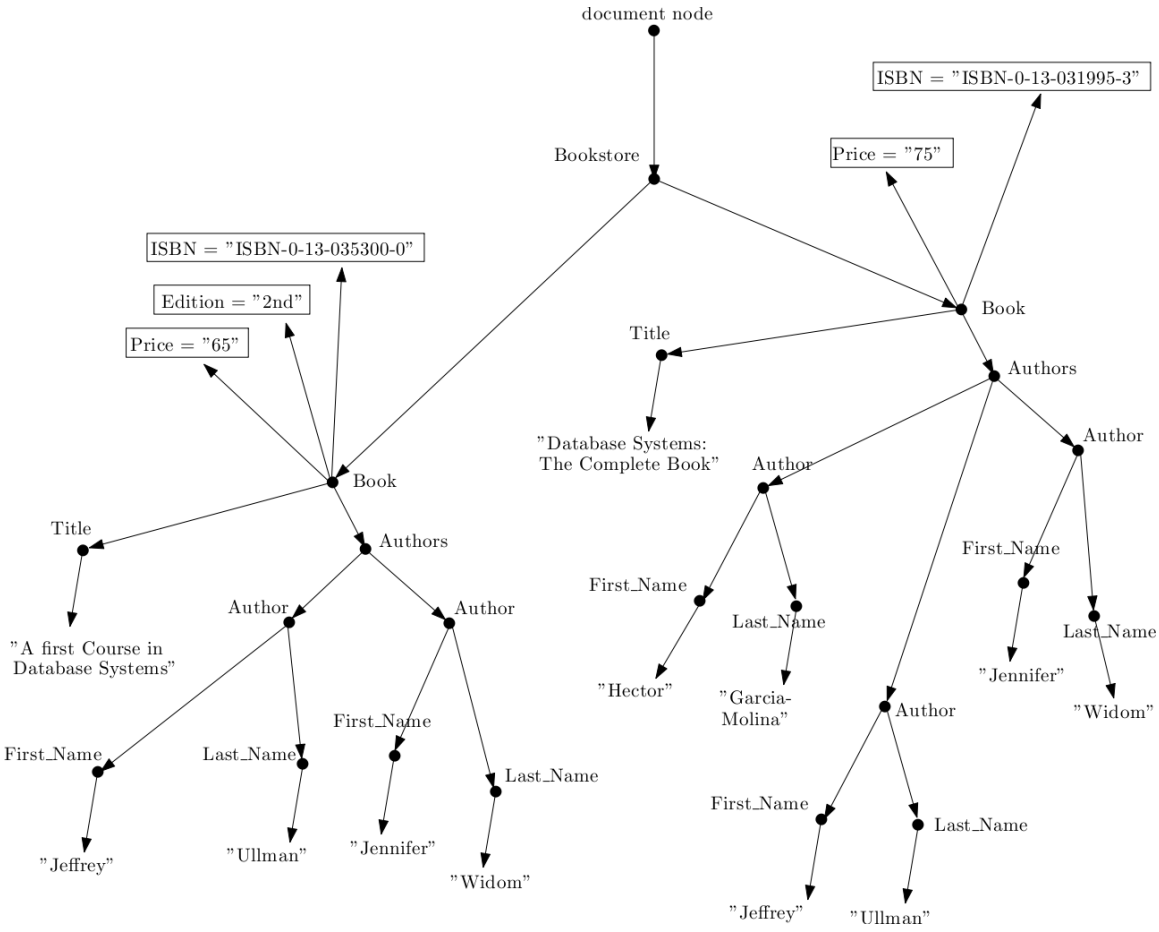
Consider the following XML document.

```
<?XML VERSION = "1.0" STANDALONE = "yes"?>

   <Bookstore>

      <Book ISBN="ISBN-0-13-035300-0" Price="65" Edition="2nd">
         <Title>A First Course in Database Systems</Title>
         <Authors>
            <Author>
               <First_Name>Jeffrey</First_Name>
               <Last_Name>Ullman</Last_Name>
            </Author>
            <Author>
               <First_Name>Jennifer</First_Name>
               <Last_Name>Widom</Last_Name>
            </Author>
         </Authors>
      </Book>

      <Book ISBN="ISBN-0-13-031995-3" Price="75">
         <Title>Database Systems: The Complete Book</Title>
         <Authors>
            <Author>
               <First_Name>Hector</First_Name>
               <Last_Name>Garcia-Molina</Last_Name>
            </Author>
            <Author>
               <First_Name>Jeffrey</First_Name>
               <Last_Name>Ullman</Last_Name>
            </Author>
            <Author>
               <First_Name>Jennifer</First_Name>
               <Last_Name>Widom</Last_Name>
            </Author>
         </Authors>
         <Remark>
         Great deal!
         </Remark>
      </Book>
   </Bookstore>
```

Question 1.
Draw the directed tree structure of the above XML file (i.e. the Hierarchical Tree Model). Please ensure that you include in your tree all the necessary labels.

Hint: If you have not enough space, you can write the names of the nodes next to the nodes (instead of inside the nodes).

Answer:



Question 2.
Write a DTD for the teachers.xml XML file above.

Answer:

```
<!DOCTYPE Bookstore[
   <!ELEMENT Bookstore (Book | Magazine)*>
   <!ELEMENT Book (Title, Authors, Remark?)>
   <!ATTLIST Book ISBN CDATA #REQUIRED
             Price CDATA #REQUIRED
             Edition CDATA #IMPLIED>
   <!ELEMENT Magazine (Title)>
   <!ATTLIST Magazine Month CDATA #REQUIRED Year CDATA #REQUIRED>
   <!ELEMENT Title (#PCDATA)>
   <!ELEMENT Authors (Author+)>
   <!ELEMENT Remark (#PCDATA)>
   <!ELEMENT Author (First_Name, Last_Name)>
   <!ELEMENT First_Name (#PCDATA)>
   <!ELEMENT Last_Name (#PCDATA)>
]>
```

Question 3.
Write the result of the following XPath queries:

1. all book titles

Answer: //Book/Title

2. all book or magazine titles (although in the above example there is no magazine)

Answer: //[Book | Magazine]/Title

3. all ISBN numbers

Answer: //Book/@ISBN

4. all books costing < 70

Answer: //Book[@Price < 70]

5. all ISBN numbers of books costing < 70

Answer: //Book[@Price < 70]/@ISBN

6. all titles of books costing < 70 where "Ullman" is an author

Answer: //Book[@Price < 70] [//Author/Last_Name = “Ullman”]/title

7. all second authors anywhere

Answer: //Author[2]

8. all author last names anywhere

Answer: //Author/Last_name

9. for every book, the first author after Hector Garcia-Molina

Answer: //Author[First_Name = “Hector”] [Last_Name = “Garcia-Molina”]/following-sibling::author[position()=1]

Question 4.
Write the result of the following XQuery queries:

1. all titles of books costing < 70 where "Ullman" is an author

FOR $x **IN** doc(“books.xml”)/Book
WHERE $x/@Price < 70 **AND SOME** $y **IN** $x//Author/Last_Name
SATISFIES $y = “Ullman”
RETURN $x

2. all author Last_Name's of books or magazines with price < 70

FOR $x **IN** doc(“books.xml”)/Bookstore/[Book | Magazine]
WHERE $x/@Price < 70
RETURN $x//Author/Last_Name

3. the titles of the books with price more than the average price of all books

LET $x := avg(doc(“books.xml”)/Bookstore/Book/@Price)
FOR $y **IN** doc(“books.xml”)/Bookstore/Book
WHERE $y/@price > $x
RETURN $y/Title

4. titles and prices of all books, sorted by price

FOR $x **IN** distinct-values(doc(“books.xml")//Book)
ORDER BY $x/@Price
RETURN $x/Title, $x/@Price

5. all book titles where all remarks include "great"

(Hint: in XQuery, the command *contains(s1,s2)* returns TRUE when the string *s1* contains the string *s2*.)

FOR $x **IN** doc("books.xml")/Bookstore/Book
WHERE contains($x/Remark, "great")
RETURN $x/Title

6. all pairs of book titles, which have at least one author's last name in common

FOR $b1 **IN** doc("books.xml")/Book
$b2 **IN** doc("books.xml")/Book
WHERE $b1 != $b2 **AND** $b1//Author/Last_Name = $b2//Author/Last_Name
RETURN $b1/Title, $b2/Title