CHAPTER 23

XML

Practice Exercises

23.1 Give an alternative representation of university information containing the same data as in Figure 23.1, but using attributes instead of subelements. Also give the DTD or XML Schema for this representation.

Answer:

a. The XML representation of data using attributes is shown in Figure 23.100.

b. The DTD for the bank is shown in Figure 23.101.

```
<university>
    <department dept_name="Comp. Sci." building="Taylor"
            budget="100000">
    </department>
    <department dept_name="Biology" building="Watson"
            budget="90000">
    </department>
    <course course_id="CS-101" title="Intro. to Computer Science"
            dept_name="Comp. Sci." credits="4">
    </course>
    <course course_id="BIO-301" title="Genetics"
            dept_name="Biology." credits="4">
    </course>
    <instructor IID="10101" name="Srinivasan"
            dept_name="Comp. Sci." salary="65000">
    </instructor>
    <instructor IID="83821" name="Brandt"
            dept_name="Comp. Sci" salary="92000">
    </instructor>
    <instructor IID="76766" name="Crick"
            dept_name="Biology" salary="72000">
    </instructor>
    <teaches IID="10101" course_id="CS-101">
    </teaches>
    <teaches IID="83821" course_id="CS-101">
    </teaches>
    <teaches IID="76766" course_id="BIO-301">
    </teaches>
</university>
```

Figure 23.100 XML representation.

23.2 Give the DTD or XML Schema for an XML representation of the following nested-relational schema:

Emp = (ename, ChildrenSet **setof***(Children), SkillsSet* **setof***(Skills))*
Children = (name, Birthday)
Birthday = (day, month, year)
Skills = (type, ExamsSet **setof***(Exams))*
Exams = (year, city)

Answer: Query:

```
<!DOCTYPE db [
    <!ELEMENT emp (ename, children*, skills*)>
    <!ELEMENT children (name, birthday)>
    <!ELEMENT birthday (day, month, year)>
    <!ELEMENT skills (type, exams+)>
    <!ELEMENT exams (year, city)>
    <!ELEMENT ename( #PCDATA )>
    <!ELEMENT name( #PCDATA )>
    <!ELEMENT day( #PCDATA )>
    <!ELEMENT month( #PCDATA )>
    <!ELEMENT year( #PCDATA )>
    <!ELEMENT type( #PCDATA )>
    <!ELEMENT city( #PCDATA )>
] >
```

```
<!DOCTYPE university [
    <!ELEMENT department >
    <!ATTLIST department
        dept_name ID #REQUIRED
        building CDATA #REQUIRED
        budget CDATA #REQUIRED >
    <!ELEMENT instructor >
    <!ATTLIST instructor
        IID ID #REQUIRED
        name CDATA #REQUIRED
        dept_name IDREF #REQUIRED >
        salary CDATA #REQUIRED >
    <!ELEMENT course >
    <!ATTLIST course
        course_id ID #REQUIRED
        title CDATA #REQUIRED
        dept_name IDREF #REQUIRED >
        credits CDATA #REQUIRED >
    <!ELEMENT teaches >
    <!ATTLIST teaches
        IID IDREF #REQUIRED >
        course_id IDREF #REQUIRED
] >
```

Figure 23.101 The DTD for the university.

23.3 Write a query in XPath on the schema of Practice Exercise 23.2 to list all skill types in *Emp*.
Answer: Code:

```
/db/emp/skills/type
```

23.4 Write a query in XQuery on the XML representation in Figure 23.11 to find the total salary of all instructors in each department.
Answer: Query:

```
for $b in distinct (/university/department/dept_name)
return
<dept-total>
        <dept_name> $b/text() </dept_name>
        let $s := sum (/university/instructor[dept_name=$b]/salary)
        return <total-salary> $s </total-salary>
</dept-total>
```

23.5 Write a query in XQuery on the XML representation in Figure 23.1 to compute the left outer join of department elements with course elements. (Hint: Use universal quantification.)
Answer: Query:

```
<lojoin>
for $d in /university/department,
        $c in /university/course
where $c/dept_name = $d/dept_name
return <dept-course> $d $c </dept-course>
|
for $d in /university/department,
where every $c in /university/course satisfies
(not ($c/dept_name =$d/dept_name))
return <dept-course > $c </dept-course >
</lojoin>
```

23.6 Write queries in XQuery to output course elements with associated instructor elements nested within the course elements, given the university information representation using ID and IDREFS in Figure 23.11.

Answer: The answer in XQuery is

```
<university-2>
    for $c in /university/course
    return
        <course>
            <course_id> $c/* </course_id>
            for $a in $c/id(@instructors)
            return $a
        </course>
</university-2>
```

23.7 Give a relational schema to represent bibliographical information specified according to the DTD fragment in Figure 23.16. The relational schema must keep track of the order of author elements. You can assume that only books and articles appear as top-level elements in XML documents.

Answer: Relation schema:

book (<u>bid</u>, title, year, publisher, place)
article (<u>artid</u>, title, journal, year, number, volume, pages)
book_author (<u>bid</u>, <u>first_name</u>,<u>last_name</u>, order)
article_author (<u>artid</u>, <u>first_name</u>,<u>last_name</u>, order)

23.8 Show the tree representation of the XML data in Figure 23.1, and the representation of the tree using *nodes* and *child* relations described in Section 23.6.2.

Answer: The answer is shown in Figure 23.102.

nodes(1,element,university,–)
nodes(2,element,department,–)
nodes(3,element,department,–)
nodes(4,element,course,–)
nodes(5,element,course,–)
nodes(6,element,instructor,–)
nodes(7,element,instructor,–)
nodes(8,element,instructor,–)
nodes(9,element,teaches,–)
nodes(10,element,teaches,–)
nodes(11,element,teaches,–)
child(2,1) child(3,1) child(4,1)
child(5,1) child(6,1)
child(7,1) child(8,1) child(9,1)

Continued in Figure 23.103

Figure 23.102 Relational Representation of XML Data as Trees.

23.9 Consider the following recursive DTD:

```
<!DOCTYPE parts [
    <!ELEMENT part (name, subpartinfo*)>
    <!ELEMENT subpartinfo (part, quantity)>
    <!ELEMENT name ( #PCDATA )>
    <!ELEMENT quantity ( #PCDATA )>
] >
```

a. Give a small example of data corresponding to this DTD.

b. Show how to map this DTD to a relational schema. You can assume that part names are unique; that is, wherever a part appears, its subpart structure will be the same.

c. Create a schema in XML Schema corresponding to this DTD.

Answer:

a. The answer is shown in Figure 23.104.

b. Show how to map this DTD to a relational schema.

part(partid,name)
subpartinfo(partid, subpartid, qty)

Attributes partid and subpartid of subpartinfo are foreign keys to part.

c. The XML Schema for the DTD is as follows:

```
<xs:schema xmlns:xs=“http://www.w3.org/2001/XMLSchema”>
<xs:element name=“parts” type=“partsType” />
<xs:complexType name=“partType”>
    <xs:sequence>
            <xs:element name=“name” type=“xs:string”/>
            <xs:element name=“subpartinfo” type=“subpartinfoType”
                minOccurs=“0” maxOccurs=“unbounded”/>
    </xs:sequence>
<xs:complexType name=“subpartinfoType”/>
    <xs:sequence>
            <xs:element name=“part” type=“partType”/>
            <xs:element name=“quantity” type=“xs:string”/>
    </xs:sequence>
</xs:schema>
```

child(10,1) child(11,1)
nodes(12,element,dept_name,Comp. Sci.)
nodes(13,element,building,Taylor)
nodes(14,element,budget,100000)
child(12,2) child(13,2) child(14,2)
nodes(15,element,dept_name,Biology)
nodes(16,element,building,Watson)
nodes(17,element,budget,90000)
child(15,3) child(16,3) child(17,3)
nodes(18,element,course_id,CS-101)
nodes(19,element,title,Intro. to Computer Science)
nodes(20,element,dept_name,Comp. Sci.)
nodes(21,element,credits,4)
child(18,4) child(19,4) child(20,4)child(21,4)
nodes(22,element,course_id,BIO-301)
nodes(23,element,title,Genetics)
nodes(24,element,dept_name,Biology)
nodes(25,element,credits,4)
child(22,5) child(23,5) child(24,5)child(25,5)
nodes(26,element,IID,10101)
nodes(27,element,name,Srinivasan)
nodes(28,element,dept_name,Comp. Sci.)
nodes(29,element,salary,65000)
child(26,6) child(27,6) child(28,6)child(29,6)
nodes(30,element,IID,83821)
nodes(31,element,name,Brandt)
nodes(32,element,dept_name,Comp. Sci.)
nodes(33,element,salary,92000)
child(30,7 child(31,7) child(32,7)child(33,7)
nodes(34,element,IID,76766)
nodes(35,element,dept_name,Biology)
nodes(36,element,salary,72000)
child(34,8) child(35,8) child(36,8)
nodes(37,element,IID,10101)
nodes(38,element,course_id,CS-101)
child(37,9) child(38,9)
nodes(39,element,IID,83821)
nodes(40,element,course_id,CS-101)
child(39,10) child(40,10)
nodes(41,element,IID,76766)
nodes(42,element,course_id,BIO-301)
child(41,11) child(42,11)

Figure 23.103 Continuation of Figure 23.102.

```
<parts>
    <part>
        <name> bicycle </name>
        <subpartinfo>
            <part>
                <name> wheel </name>
                <subpartinfo>
                    <part>
                        <name> rim </name>
                    </part>
                    <qty> 1 </qty>
                </subpartinfo>
                <subpartinfo>
                    <part>
                        <name> spokes </name>
                    </part>
                    <qty> 40 </qty>
                </subpartinfo>
                <subpartinfo>
                    <part>
                        <name> tire </name>
                    </part>
                    <qty> 1 </qty>
                </subpartinfo>
            </part>
            <qty> 2 </qty>
        </subpartinfo>
        <subpartinfo>
            <part>
                <name> brake </name>
            </part>
            <qty> 2 </qty>
        </subpartinfo>
        <subpartinfo>
            <part>
                <name> gear </name>
            </part>
            <qty> 3 </qty>
        </subpartinfo>
        <subpartinfo>
            <part>
                <name> frame </name>
            </part>
            <qty> 1 </qty>
        </subpartinfo>
    </part>
</parts>
```

Figure 23.104 Example Parts Data in XML.