CHAPTER 10

XML

Solutions to Practice Exercises

10.1 **a.** The XML representation of data using attributes is shown in Figure 10.1.
b. The DTD for the bank is shown in Figure 10.2.

10.2 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 )>
] >
```

10.3 Code:

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

```
<bank>
    <account account-number="A-101" branch-name="Downtown"
            balance="500">
    </account>
    <account account-number="A-102" branch-name="Perryridge"
            balance="400">
    </account>
    <account account-number="A-201" branch-name="Brighton"
            balance="900">
    </account>
    <customer customer-name="Johnson" customer-street="Alma"
            customer-city="Palo Alto">
    </customer>
    <customer customer-name="Hayes" customer-street="Main"
            customer-city="Harrison">
    </customer>
    <depositor account-number="A-101" customer-name="Johnson">
    </depositor>
    <depositor account-number="A-201" customer-name="Johnson">
    </depositor>
    <depositor account-number="A-102" customer-name="Hayes">
    </depositor>
</bank>
```

Figure 10.1 XML representation.

```
<!DOCTYPE bank [
    <!ELEMENT account >
    <!ATTLIST account
        account-number ID #REQUIRED
        branch-name CDATA #REQUIRED
        balance CDATA #REQUIRED >
    <!ELEMENT customer >
    <!ATTLIST customer
        customer-name ID #REQUIRED
        customer-street CDATA #REQUIRED
        customer-street CDATA #REQUIRED >
    <!ELEMENT depositor >
    <!ATTLIST depositor
        account-number IDREF #REQUIRED
        customer-name IDREF #REQUIRED >
] >
```

Figure 10.2 The DTD for the bank.

10.4 Query:

```
for $b in distinct (/bank/account/branch-name)
return
<branch-total>
      <branch-name> $b/text() </branch-name>
      let $s := sum (/bank/account[branch-name=$b]/balance)
      return <total-balance> $s </total-balance>
</branch-total>
```

10.5 Query:

```
      <lojoin>
      for $b in /bank/account,
            $c in /bank/customer,
            $d in /bank/depositor
      where $a/account-number = $d/account-number
            and $c/customer-name = $d/customer-name
      return <cust-acct> $c $a </cust-acct>
      |
      for $c in /bank/customer,
      where every $d in /bank/depositor satisfies
      (not ($c/customer-name=$d/customer-name))
      return <cust-acct> $c </cust-acct>
      </lojoin>
```

10.6 The answer in XQuery is

```
<bank-2>
      for $c in /bank/customer
      return
            <customer>
                  <customer-name> $c/* </customer-name>
                  for $a in $c/id(@accounts)
                  return $a
            </customer>
</bank-2>
```

10.7 Realtion schema:

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

10.8 The answer is shwn in Figure 10.3.

```
nodes(1,element,bank,–)
nodes(2,element,account,–)
nodes(3,element,account,–)
nodes(4,element,account,–)
nodes(5,element,customer,–)
nodes(6,element,customer,–)
nodes(7,element,depositor,–)
nodes(8,element,depositor,–)
nodes(9,element,depositor,–)
child(2,1) child(3,1) child(4,1)
child(5,1) child(6,1)
child(7,1) child(8,1) child(9,1)
nodes(10,element,account-number,A-101)
nodes(11,element,branch-name,Downtown)
nodes(12,element,balance,500)
child(10,2) child(11,2) child(12,2)
nodes(13,element,account-number,A-102)
nodes(14,element,branch-name,Perryridge)
nodes(15,element,balance,400)
child(13,3) child(14,3) child(15,3)
nodes(16,element,account-number,A-201)
nodes(17,element,branch-name,Brighton)
nodes(18,element,balance,900)
child(16,4) child(17,4) child(18,4)
nodes(19,element,customer-name,Johnson)
nodes(20,element,customer-street,Alma)
nodes(21,element,customer-city,Palo Alto)
child(19,5) child(20,5) child(21,5)
nodes(22,element,customer-name,Hayes)
nodes(23,element,customer-street,Main)
nodes(24,element,customer-city,Harrison)
child(22,6) child(23,6) child(24,6)
nodes(25,element,account-number,A-101)
nodes(26,element,customer-name,Johnson)
child(25,7) child(26,7)
nodes(27,element,account-number,A-201)
nodes(28,element,customer-name,Johnson)
child(27,8) child(28,8)
nodes(29,element,account-number,A-102)
nodes(30,element,customer-name,Hayes)
child(29,9) child(30,9)
```

Figure 10.3 Relational Representation of XML Data as Trees.

10.9 **a.** The answer is shown in Figure 10.4.

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. No answer

```
<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 10.4 Example Parts Data in XML.