

CHAPTER 30



XML

Practice Exercises

- 30.1** Write a query in XQuery on the XML representation in Figure 30.11 to find the total salary of all instructors in each department.

Answer:

The query is shown in ??

- 30.2** Write a query in XQuery on the XML representation in Figure 30.1 to compute the left outer join of department elements with course elements. (Hint: Use universal quantification.)

Answer: The query is shown in ??

- 30.3** 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 30.11.

Answer:

The query is shown in ??

```
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>
```

Figure 30.101 XQuery

```

<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>

```

Figure 30.102 XQuery

- 30.4** Give a relational schema to represent bibliographical information specified according to the DTD fragment shown below:

```

<!DOCTYPE bibliography [
  <!ELEMENT book (title, author+, year, publisher, place?)>
  <!ELEMENT article (title, author+, journal, year, number, volume, pages?)>
  <!ELEMENT author ( last_name, first_name) >
  <!ELEMENT title ( #PCDATA )>
  ... similar PCDATA declarations for year, publisher, place, journal, year,
    number, volume, pages, last_name and first_name
1 >

```

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>

```

Figure 30.103 XQuery

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 (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)
```

- 30.5** Show the tree representation of the XML data in Figure 30.1, and the representation of the tree using *nodes* and *child* relations described in Section 30.6.2.

Answer:

The answer is shown in ??.

- 30.6** Consider the following recursive DTD:

```
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 30.104 Relational representation of XML data as trees.

```

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 30.105 Continuation of ??.

```

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

```

- a. 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.
- b. Create a schema in XML Schema corresponding to this DTD.

Answer:

- a. 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.

- b. 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>
  <xs:complexType name="subpartinfoType">
    <xs:sequence>
      <xs:element name="part" type="partType"/>
      <xs:element name="quantity" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

