特种数据库（XML基础知识）

单 位：重庆大学计算机学院

这是什么文档？

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="messagePrinter" class="hello.MessagePrinter">
        <constructor-arg>
            <ref bean="message" />
        </constructor-arg>
    </bean>
    <bean id="message" class="hello.Message">
    </bean>
</beans>
```

主要学习目标

- XML基本概念
- 理解XML的数据格式

思考问题

```
<person sex="female">
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

```
<person>
<sex>female</sex>
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

- 有什么区别？

一 XML引言

讨论1.如何通过XML标签描述数据和结构？

1. 什么是XML

返回

1)什么是XML，一种语言、数据结构、或什么?

2)比较XML与HTML的共同与不同之处?

XML是一种语言，更是一种适合灵活描述各种半结构化的数据和结构的好工具！在一应用程序与另一应用程序需通信(交换数据)时、或在整合数据时，XML都是一种特别有用的数据格式！

```
<html>
 <title> 主页链接其他网站或页面</title>
 <body background="http://www.jxstnu.cn/wskc/html/download/bk2.gif">
  <h1 align="center">选择你要进入的页面或网站</h1>
  <br>
    <a href="gif.htm" >
      <font size=5 face="黑体" color="green">有动画的网页</font>
    </a>
  <br>
    <a href="http://www.jxstnu.cn" >
     <font size=6  color="navy">我们的校园主页</font>
    </a>
  <br>
    <a href="http://cy.jxstnu.cn"  target="_blank">
     <font size=6  color="navy">新窗口打开春雨主页</font>
    </a>
  </body>
</html>
```

XML与HTML（共同之处）都是标记语言，(不同之处)但1）用途不同,HTML重在表示，XML重在数据交换(数据及结构灵活描述)；2）XML标签集不固定,应用可根据描述需要选择自己特有的标签集。

```
<bank>
  <account>
        <account_number> A-101
      </account_number>
        <branch_name> Downtown
      </branch_name>
        <balance> 500  </balance>
  </account>
    ……
  <customer>
    <customer_name> Johnson </customer_name
>
    <customer_street> Alma </customer_street >
    <customer_city> Palo Alto </customer_city >
  </customer>
    ……
  <depositor>
        <account_number> A-101
      </account_number>
        <customer_name> Johnson
      </customer_name>
  </depositor>
    ……
</bank>
```

```
<account>
    This account is seldom used any more.
    <account_number> A-102</account_number>
    <branch_name> Perryridge</branch_name>
    <balance>400 </balance>
</account>
```

3)什么是(tag)标签和元素，如何用于描述结构化数据？

2. XML的基本要素-元素

Every document must have a single top-level element!

元素是XML数据文档的基本结构，采用配对的自定义标识符(标签)来描述，且必须恰当地嵌套

甚至，可以插入文字说明！(见上例)

```
<university>
    <department>
        <dept_name> Comp. Sci. </dept_name>
        <building> Taylor </building>
        <budget> 100000 </budget>
    </department>
    <department>
        <dept_name> Biology </dept_name>
        <building> Watson </building>
        <budget> 90000 </budget>
    </department>
    <course>
        <course_id> CS-101 </course_id>
        <title> Intro. to Computer Science </title>
        <dept_name> Comp. Sci </dept_name>
        <credits> 4 </credits>
    </course>
    <course>
        <course_id> BIO-301 </course_id>
        <title> Genetics </title>
        <dept_name> Biology </dept_name>
        <credits> 4 </credits>
    </course>
```

后部见图23-2

图 23-1 (部分)大学信息的 XML 表示

返回

```
    <instructor>
        <IID> 10101 </IID>
        <name> Srinivasan </name>
        <dept_name> Comp. Sci. </dept_name>
        <salary> 65000 </salary>
    </instructor>
    <instructor>
        <IID> 83821 </IID>
        <name> Brandt </name>
        <dept_name> Comp. Sci. </dept_name>
        <salary> 92000 </salary>
    </instructor>
    <instructor>
        <IID> 76766 </IID>
        <name> Crick </name>
        <dept_name> Biology </dept_name>
        <salary> 72000 </salary>
    </instructor>
    <teaches>
        <IID> 10101 </IID>
        <course_id> CS-101 </course_id>
    </teaches>
    <teaches>
        <IID> 83821 </IID>
        <course_id> CS-101 </course_id>
    </teaches>
    <teaches>
        <IID> 76766 </IID>
        <course_id> BIO-301 </course_id>
    </teaches>
</university>
```

图 23-2 续图 23-1

3. XML的基本要素-属性

4)采用属性和子元素描述有何不同？

```
<course>
    This course is being offered for the first time in 2009.
    <course_id> BIO-399 </course_id>
    <title> Computational Biology </title>
    <dept_name> Biology </dept_name>
    <credits> 3 </credits>
</course>
```

图 23-4

数据的一些特殊特征，可以采用属性方式进行说明

```
<course course_id= "CS-101">
    <title> Intro. to Computer Science</title>
    <dept_name> Comp. Sci. </dept_name>
    <credits> 4 </credits>
</course>
```

图 23-7

文档结构角度看：

属性是标记的一部分，而子元素内容是基本文档内容的一部分。

数据表示角度看：

在数据表示的上下文中，差异是不清楚的，并且可能是混淆的
---相同的信息可以用两种方式表示
---建议：使用属性作为元素的标识符，使用子元素作为内容

```
<university xmlns:yale="http://www.yale.edu">
    ...
    <yale:course>
        <yale:course_id> CS-101 </yale:course_id>
        <yale:title> Intro. to Computer Science</yale:title>
        <yale:dept_name> Comp. Sci. </yale:dept_name>
        <yale:credits> 4 </yale:credits>
    </yale:course>
    ...
</university>
```

图 23-8　通过使用名字空间来指定唯一标签名

4. 名字空间(Namespaces)

5)什么是名字空间，有何作用？

- 问题
  - XML data has to be exchanged between organizations
  - Same tag name may have different meaning in different organizations, causing confusion on exchanged documents

标签名在不同单位的含义可能不同，必须进行消歧！

- 土办法
  Specifying a unique string as an element name avoids confusion消歧

各元素描述长而且大量重复

- XML解决方案
  - Better solution: use unique-name:element-name
  - Avoid using long unique names all over document by using XML Namespaces

```
<bank Xmlns:FB='http://www.FirstBank.com'>
    …
    <FB:branch>
        <FB:branchname>Downtown</FB:branchname>
        <FB:branchcity>  Brooklyn  </FB:branchcity>
    </FB:branch>
    …
</bank>
```

区分标识缩写申明

为本单位的标签名添加区分标识(前缀)

各元素描述简略！

二 XML数据文档的模式

返回

讨论2. 如何描述XML数据文档的模式(结构)?

1. 文档类型定义(DTD)

1)什么是DTD，涉及要素，及其作用?

元素　　子元素　　“|” 或者

```
<!DOCTYPE university [
    <!ELEMENT university ( (department|course|instructor|teaches)+)>
    <!ELEMENT department ( dept_name, building, budget)>
    <!ELEMENT course ( course_id, title, dept_name, credits)>
    <!ELEMENT instructor (IID, name, dept_name, salary)>
    <!ELEMENT teaches (IID, course_id)>
    <!ELEMENT dept_name( #PCDATA )>
    <!ELEMENT building( #PCDATA )>
    <!ELEMENT budget( #PCDATA )>
    <!ELEMENT course_id ( #PCDATA )>
    <!ELEMENT title ( #PCDATA )>
    <!ELEMENT credits( #PCDATA )>
    <!ELEMENT IID( #PCDATA )>
    <!ELEMENT name( #PCDATA )>
    <!ELEMENT salary( #PCDATA )>
]>
```

“+” 1个或多个

“*” 0个或多个

#PCDATA，即字符串

2)该DTD描述什么样结构的XML数据文档?

该DTD描述的XML数据文档为：

图23-1&图23-2

图 23-9　一个 DTD 示例

DTD中的属性和引用说明

3)DTD如何说明属性及引用?

```
<!DOCTYPE university-3 [
    <!ELEMENT university ( (department|course|instructor)+)>
    <!ELEMENT department ( building, budget )>
    <!ATTLIST department
        dept_name ID #REQUIRED >
    <!ELEMENT course (title, credits )>
    <!ATTLIST course
        course_id ID #REQUIRED
        dept_name IDREF #REQUIRED
        instructors IDREFS #IMPLIED >
    <!ELEMENT instructor ( name, salary )>
    <!ATTLIST instructor
        IID ID #REQUIRED
        dept_name IDREF #REQUIRED >
    ... declarations for title, credits, building,
        budget, name and salary ...
]>
```

department的属性说明(类型为ID)

对属性的引用说明可是单个或多引用(类型为IDREFS)

图 23-10 具有 ID 和 IDREFS 属性类型的 DTD

```
<university-3>
    <department dept_name="Comp. Sci.">
        <building> Taylor </building>
        <budget> 100000 </budget>
    </department>
    <department dept_name="Biology">
        <building> Watson </building>
        <budget> 90000 </budget>
    </department>
    <course course_id="CS-101" dept_name="Comp. Sci"
                                instructors="10101 83821">
        <title> Intro. to Computer Science </title>
        <credits> 4 </credits>
    </course>
    <course course_id="BIO-301" dept_name="Biology"
                                instructors="76766">
        <title> Genetics </title>
        <credits> 4 </credits>
    </course>
    <instructor IID="10101" dept_name="Comp. Sci.">
        <name> Srinivasan </name>
        <salary> 65000 </salary>
    </instructor>
    <instructor IID="83821" dept_name="Comp. Sci.">
        <name> Brandt </name>
        <salary> 72000 </salary>
    </instructor>
    <instructor IID="76766" dept_name="Biology">
        <name> Crick </name>
        <salary> 72000 </salary>
    </instructor>
</university-3>
```

图 23-11 具有 ID 和 IDREF 属性的 XML 数据

4)该模式定义了什么样的数据结构?

2. XML Schema(模式)-of Bank DTD

```
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema>
<xs:element name="bank" type="BankType"/>
<xs:element name="account">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="account_number" type="xs:string"/>
            <xs:element name="branch_name"    type="xs:string"/>
            <xs:element name="balance"        type="xs:decimal"/>
        </xs:squence>
    </xs:complexType>
</xs:element>
... definitions of customer （省略，详见书p.406）
<xs:element name="depositor">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="customer _name" type="xs:string"/>
            <xs:element name="account_number" type="xs:string"/>
        </xs:squence>
    </xs:complexType>
</xs:element>
<xs:complexType name="BankType">
    <xs:squence>
        <xs:element ref="account"   minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="customer"  minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="depositor" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>
```

定义根元素bank,其类型在后面说明

定义元素account，该元素为复杂类型，该元素有三个子元素的序列构成

定义子元素balence,其类型为小数

XML Schema 定义了很多内置类型，如 string、integer、decimal、date 和 boolean。

XML Schema 可以用 minOccurs 和 maxOccurs 来定义子元素出现的最少次数和最多次数

XML Schema对数据说明更加精细（优点详见P.562）

定义新的复杂类型BankType，该类型由多个子元素的序列构成

子元素可以是0或多个前面定义的account

Ref用于指明子子元素为前面定义的有效元素

5)XML Schema,及与DTD差异?

该XML模式定义的XML文件，结构同前面的bank文件，由0或多个account，customer，depositor子元素构成！

XML Schema（示例2）选讲/略讲

象DTD一样，XML Schema也用于描述XML文档的数据组织结构。

```
<xs: schema xmlns: xs = "http: //www. w3. org/2001/XMLSchema" >
<xs: element name = "university" type = "universityType" / >
<xs: element name = "department" >
    <xs: complexType >
        <xs: sequence >
            <xs: element name = "dept_ name" type = "xs: string"/ >
            <xs: element name = "building" type = "xs: string"/ >
            <xs: element name = "budget" type = "xs: decimal"/ >
        </xs: sequence >
    </xs: complexType >
</xs: element >
<xs: element name = "course" >
    <xs: complexType >
        <xs: sequence >
            <xs: element name = "course_ id" type = "xs: string"/ >
            <xs: element name = "title" type = "xs: string"/ >
            <xs: element name = "dept_ name" type = "xs: string"/ >
            <xs: element name = "credits" type = "xs: decimal"/ >
        </xs: sequence >
    </xs: complexType >
</xs: element >
<xs: element name = "instructor" >
    <xs: complexType >
        <xs: sequence >
            <xs: element name = "IID" type = "xs: string"/ >
            <xs: element name = "name" type = "xs: string"/ >
            <xs: element name = "dept_ name" type = "xs: string"/ >
            <xs: element name = "salary" type = "xs: decimal"/ >
        </xs: sequence >
    </xs: complexType >
</xs: element >
```

图23-9中DTD

图 23-9 中 DTD 的 XML Schema 版本

后部见图 23-13

图 23-12

```
<xs:element name="teaches">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="IID" type="xs:string"/>
            <xs:element name="course_id" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="UniversityType">
    <xs:sequence>
        <xs:element ref="department" minOccurs="0"
                    maxOccurs="unbounded"/>
        <xs:element ref="course" minOccurs="0"
                    maxOccurs="unbounded"/>
        <xs:element ref="instructor" minOccurs="0"
                    maxOccurs="unbounded"/>
        <xs:element ref="teaches" minOccurs="0"
                    maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>
```

图 23-13　续图 23-12

三 XML数据文档的树模型

讨论3. XML树模型的作用，以及形成方式？

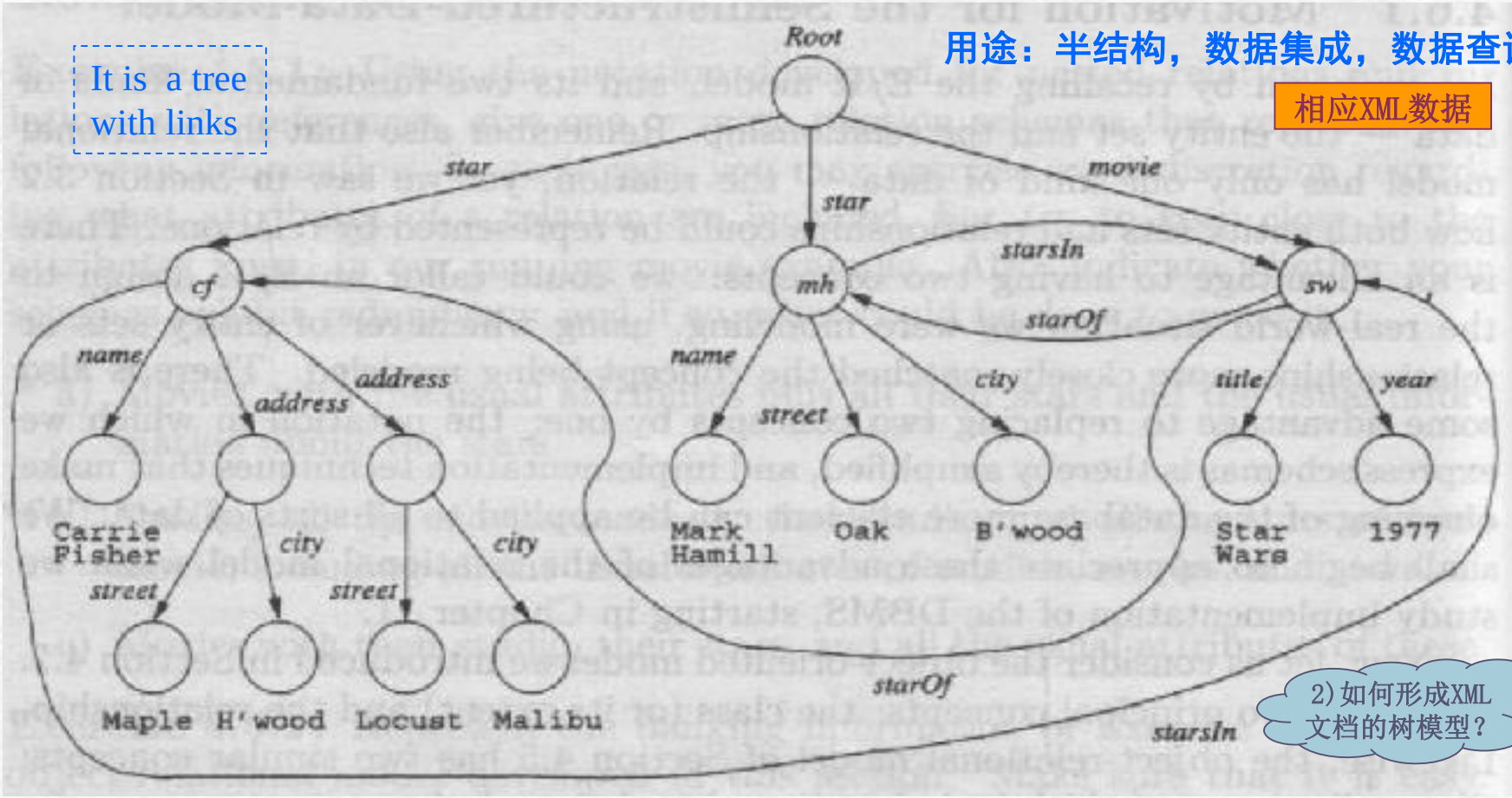
1. XML文档的树模型

1）XML文档树模型的主要用途？

用途：半结构，数据集成，数据查询

相应XML数据

It is a tree with links



2）如何形成XML文档的树模型？

XML文档可看做一棵带链接的树！

XML树模型的生成方式说明！

2. XML树模型的生成方式

2)如何形成XML文档的树模型？

- **Query and transformation 转换 languages are based on a tree model of XML data**

可以看着是一种半结构化数据！

- **An XML document is modeled as a tree, with nodes corresponding to elements and attributes**
  - Element nodes have child nodes, which can be attributes or subelements
  - Text in an element is modeled as a text-node child of the element
  - Children of a node are ordered according to their order in the XML document
  - Element and attribute nodes (except for the root node) have a single parent, which is an element node
  - The root node has a single child, which is the root element of the document
  - ID类属性和IDREF类属性之间的引用关系采用横向链接表示

XML Data

```
<STARS-MOVIES>
    <STAR starId = "cf" starredIn = "sw, esb, rj">
        <NAME>Carrie Fisher</NAME>
        <ADDRESS><STREET>123 Maple St.</STREET>
            <CITY>Hollywood</CITY></ADDRESS>
        <ADDRESS><STREET>5 Locust Ln.</STREET>
            <CITY>Malibu</CITY></ADDRESS>
    </STAR>
    <STAR starId = "mh" starredIn = "sw, esb, rj">
        <NAME>Mark Hamill</NAME>
        <ADDRESS><STREET>456 Oak Rd.<STREET>
            <CITY>Brentwood</CITY></ADDRESS>
    </STAR>
    <MOVIE movieId = "sw" starsOf = "cf, mh">
        <TITLE>Star Wars</TITLE>
        <YEAR>1977</YEAR>
    </MOVIE>
    <MOVIE movieId = "esb" starsOf = "cf, mh">
        <TITLE>Empire Strikes Back</TITLE>
        <YEAR>1980</YEAR>
    </MOVIE>
    <MOVIE movieId = "rj" starsOf = "cf, mh">
        <TITLE>Return of the Jedi</TITLE>
        <YEAR>1983</YEAR>
    </MOVIE>
</STARS-MOVIES>
```

四 XPath语言

讨论4. 如何采用XPath工具查询XML数据？

XML数据文档上的XPath查询

1）Xpath查询数据时利用的关键特征是什么？

注：(university-3) (university-2)

‘XML树模型’和‘路径表达式’

```
/unicersity-3/instructor/name;          一组name元素
/university-3;                          根元素-即整个文档
/university-3/instructor;               一组instructor元素
```

[图4] XPath查询语句

一组值-学分大于4的课程名　　一组元素-讲授2门以上课程的教师

```
/university-3/instructor/name/text();                一组值-人名
/university-3/course[credits>=4]/@course_id;
/unibersity-2/instructor[count(. /teaches/course)>2];
/university-3/course/id(@dept_name);                 一组值-开课学院
/university-3/course[@dept_name="Comp.Sci"] |
      /university-3/course[@dept_name="Biology"];
/university-3//name;任意子层下的name元素  2个course元素-指定学院
doc("university.xml")/university/department？;
```

[图5] XPath查询语句　　Doc()返回文件的根

（路径表达式）

返回一组值(或XML元素)

2）Xpath提供哪些查询能力？

（读取文本内容）

（约束条件，访问属性值）

（聚集函数）

（id函数及引用）

（操作符| ---或）

3)各语句的查询结果查询？

（操作符// ---）

（内置函数doc）

```
<university-3>
    <department dept_name="Comp. Sci.">
        <building> Taylor </building>
        <budget> 100000 </budget>
    </department>
    <department dept_name="Biology">
        <building> Watson </building>
        <budget> 90000 </budget>
    </department>
    <course course_id="CS-101" dept_name="Comp. Sci"
                                    instructors="10101 83821">
        <title> Intro. to Computer Science </title>
        <credits> 4 </credits>
    </course>
    <course course_id="BIO-301" dept_name="Biology"
                                    instructors="76766">
        <title> Genetics </title>
        <credits> 4 </credits>
    </course>
    <instructor IID="10101" dept_name="Comp. Sci.">
        <name> Srinivasan </name>
        <salary> 65000 </salary>
    </instructor>
    <instructor IID="83821" dept_name="Comp. Sci.">
        <name> Brandt </name>
        <salary> 72000 </salary>
    </instructor>
    <instructor IID="76766" dept_name="Biology">
        <name> Crick </name>
        <salary> 72000 </salary>
    </instructor>
</university-3>
```

图 23-11　具有 ID 和 IDREF 属性的 XML 数据

返回

XML数据文档(大学信息-3)

```
<university-2>
    <instructor>
        <ID> 10101 </ID>
        <name> Srinivasan </name>
        <dept_name> Comp. Sci. </dept_name>
        <salary> 65000 </salary>
        <teaches>
            <course>
                <course_id> CS-101 </course_id>
                <title> Intro. to Computer Science </title>
                <dept_name> Comp. Sci. </dept_name>
                <credits> 4 </credits>
            </course>
        </teaches>
    </instructor>

    <instructor>
        <ID> 83821 </ID>
        <name> Brandt </name>
        <dept_name> Comp. Sci. </dept_name>
        <salary> 92000 </salary>
        <teaches>
            <course>
                <course_id> CS-101 </course_id>
                <title> Intro. to Computer Science </title>
                <dept_name> Comp. Sci. </dept_name>
                <credits> 4 </credits>
            </course>
        </teaches>
    </instructor>
</university-2>
```

图 23-6　嵌套 XML

返回

XML数据文档
(大学信息-2)

五 XQuery语言

XML数据文档上的XQuery查询语句

讨论5. 如何采用Xquery工具查询XML数据？

1）FLWOR表达式及其使用方法？

注：(university) (university-3)

```
For  $ x in /university-3/course                    (指定搜索范围)
Let  $ courseId := $ x/@course_id                   (设置临时变量，以简化表达式)
Where  $ x/credits>3                                (指定约束条件)
Return <course_id>{ $ courseid}</course_id>;        (构造查询返回结果)
```

[图6] FLW表达式查询

⇨ 学分大于3的课程编码，并构造成XML新元素

2）XQuery的查询能力？

```
For  $ c in /university/course,
     $ i in /university/instructor,
     $ t in /university/ teaches
Where  $ c/curse_id =  $ t/course_id
                and  $ t/IID =  $ i/IID          (指定元素间连接条件)
Return <course_instructor>{ $ c  $ i }</course_instructor>;
```

[图7] 含连接条件的查询

⇨ 课程(元素)与承担课程的教师(元素)，并构造成XML新元素

3)比较XQuery，XPath和SQL的异同？

XQuery：XPath和SQL融合

```
<university-1> {
For  $ d in /university/department
Order by  $ d/depat_name          (department新元素将按系名排序)
Return
   <department>
      {  $ d/* }                              (系基本信息元素)
      {For  $ c in /university/course[dept_name =  $ d/dept_name]
       Order by  $ c/course_id      (course新元素将按课程编号排序)
       Return <course>{  $ c/* }</course>} (课程元素)
   </department>
</university-1>
```

[图9] 含结果排序的查询

⇨ 根据系名(包括系基本信息元素)排列的课程(元素)，并构造成大学信息XML文档：(university-1)

To end

返回

XML数据文档
(大学信息-1)

```
<university-1>
    <department>
        <dept_name> Comp. Sci. </dept_name>
        <building> Taylor </building>
        <budget> 100000 </budget>
        <course>
            <course_id> CS-101 </course_id>
            <title> Intro. to Computer Science </title>
            <credits> 4 </credits>
        </course>
        <course>
            <course_id> CS-347 </course_id>
            <title> Database System Concepts </title>
            <credits> 3 </credits>
        </course>
    </department>
    <department>
        <dept_name> Biology </dept_name>
        <building> Watson </building>
        <budget> 90000 </budget>
        <course>
            <course_id> BIO-301 </course_id>
            <title> Genetics </title>
            <credits> 4 </credits>
        </course>
    </department>
    <instructor>
        <IID> 10101 </IID>
        <name> Srinivasan </name>
        <dept_name> Comp. Sci. </dept_name>
        <salary> 65000. </salary>
        <course_id> CS-101 </course_id>
    </instructor>
</university-1>
```

图 23-5　大学信息

课堂思考小问题

- XML数据如何存放？
- 如何访问数据？

课堂小结和作业安排

- 基本知识：
  - XML的基本概念
  - XML数据文档
- 扩展学习：
  - XML数据的存储和网络传输

- 作业

第章习题：*23.2，*23.3