（1）为下面的xml文件写一个schema文档

<?xml version="1.0" encoding="GB2312"?>

<学生 学号="1">

<姓名>张三</姓名>

<性别>男</性别>

<年龄>20</年龄>

</学生>

<学生 学号="2">

<姓名>李四</姓名>

<性别>女</性别>

<年龄>19</年龄>

</学生>

<学生 学号="3">

<姓名>王二</姓名>

<性别>男</性别>

<年龄>21</年龄>

</学生>

<学生 学号="4">

<姓名>王二</姓名>

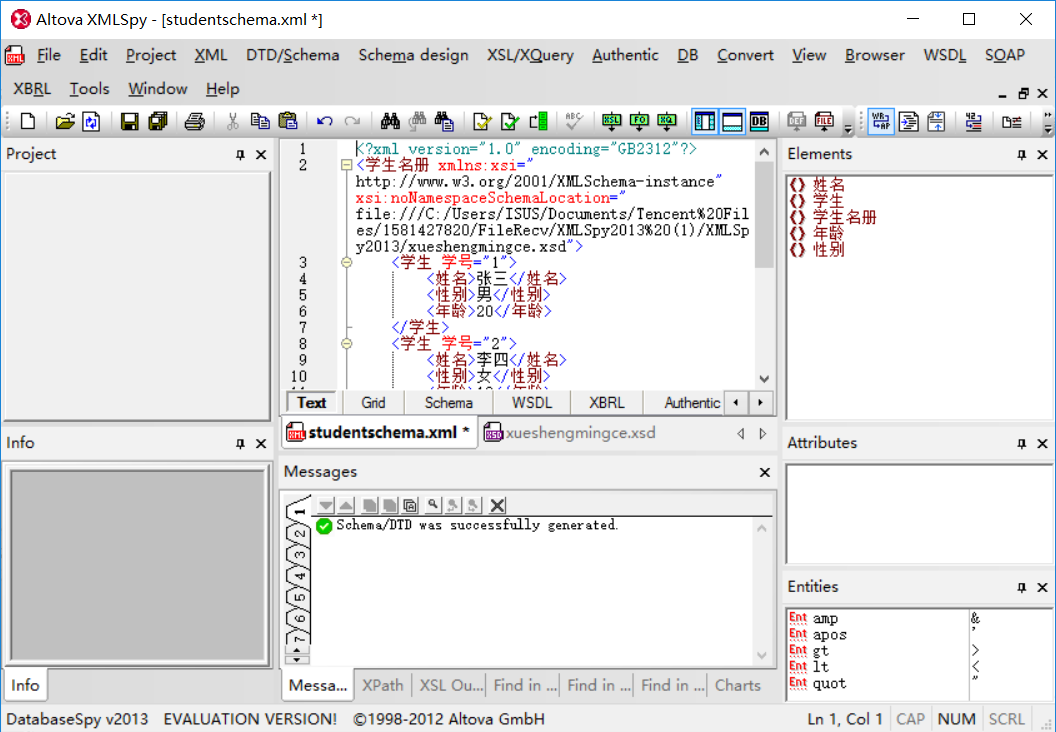
<性别>男</性别>

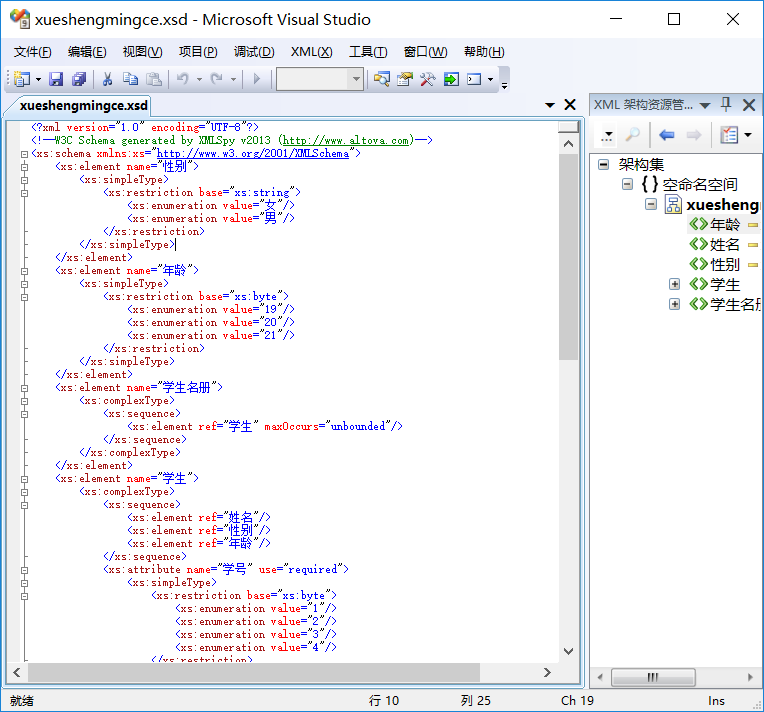
<年龄>21</年龄>

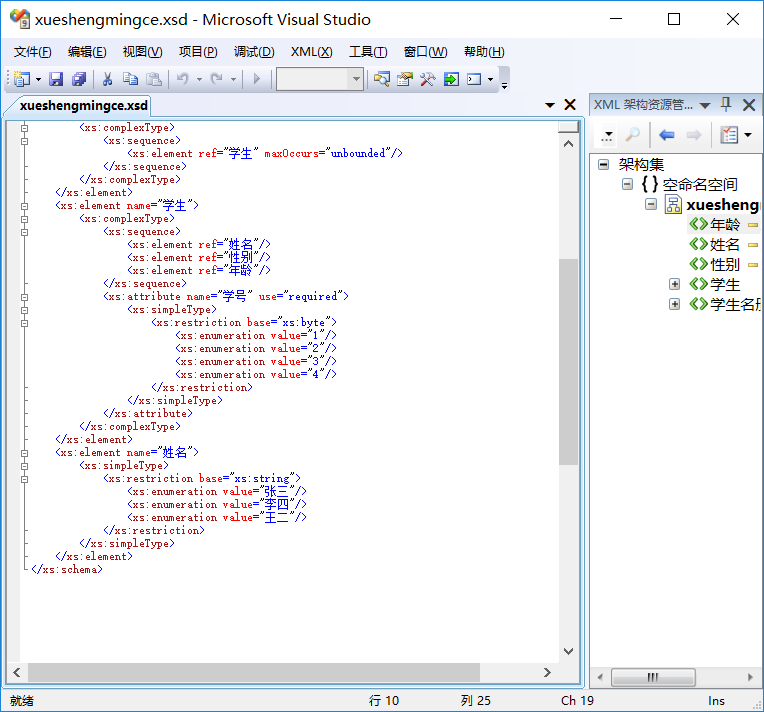
</学生>

</学生名册>

用XML spy进行校验







<?xml version="1.0" encoding="GB2312"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"

targetNamespace="http://www.hexinli.com"

xmlns="http://www.hexinli.com">

<xs:element name="学生名册">

<xs:complexType>

<xs:sequence>

<xs:element name="学生"type="我的类型"maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:complexType name="我的类型">

<xs:sequence>

<xs:element name="名字"type="xs:string"/>

</xs:sequence>

<xs:attribute name="学号"type="xs:int"/>

</xs:complexType>

</xs:schema>

<xs:element name="学生">

<xs:complexType>

<xs:sequence>

<xs:element ref="姓名"/>

<xs:element ref="性别"/>

<xs:element ref="年龄"/>

</xs:sequence>

<xs:attribute name="学号"use="required"typr="xs:int"/>

</xs:complexType>

</xs:element>

<xs:element name="姓名"type="xs:string"/>

<xs:element name="性别">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="女"/>

<xs:enumeration value="男"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="年龄"type="xs:int"/>

（2）研究数据交换领域的一两个用到XMLSchema的实例，查看常用的schema标签（如体育、地理信息系统），研究文件结构，写一篇小论文。

答：schema文档是对XML文档进行校验的文档。

一、在写schema文档时，必须定义一个根元素，且只有一个根元素。这个根元素属性用来描述文档的命空间以及文档的引用。这个根元素是<xs:schema>。

1、一般来讲，都会设置<http://www.w3.org/2001/XMLSchema>这个命名空间。如xmlns:xs=” <http://www.w3.org/2001/XMLSchema>”这句代码是在schema文档开头第二行定义的，意思是使用xs：作前缀的元素、属性、类型等名称是属于<http://www.w3.org/2001/XMLSchema>名空间的。

2、targetNamesapce=”htttp://www/w3schools.com”指示文档定义的元素。属性、类型等名称属于htttp://www/w3schools.com名空间。这个在文档前面也是需要定义的。名空间的名称是可以自己更改的，根据需要。

3、xmlns=” htttp://www/w3schools.com”是表示没有前缀的元素、属性、类型等名称是属于默认的名空间htttp://www/w3schools.com的。这个也是要有的。

4、elementFromDefault=”qualified”指示使用这个schema文档定义的XML文档所使用的元素必须在这个文档中定义且要使用名空间前缀。

二、在schema文档中定义的是文档中可以出现的元素、属性、元素可以包含的子元素、元素出现的顺序、元素出现的次数、元素与属性的数据类型、元素与属性的缺省值或固定值或值范围值列表以及哪些元素是空或可以包含text。

Schema支持命名空间和文档的引用，但命名空间一般使用URI作为名称。

三、schema文档所定义的元素有简单元素和复杂元素两种。

1.简单元素是指只包含text的XML元素，没有任何子元素或者属性。简单元素可以附加地定义其缺省值或固定值。只需限定元素的数据类型、缺省值或固定值时，定义的元素使用schema文档定义的基本类型；如果希望限定简单元素的取值范围等约束，应该使用简单类型定义并且在简单类型中嵌套值约束子元素。

2、复杂元素则是指包含属性或子元素的XML元素。复杂元素主要包含空元素、只包含属性与子元素的元素、只包含属性与text的元素、只包含属性与text与子元素的元素四中类型。

描述复杂元素可以采用复合结构和分离结构两种结构来描述。复合结构是直接将复杂类型定义为复杂元素的子元素；分离结构是将复杂类型定义在外部，再由复杂元素来引用的。

复合结构：<xs:element name=”sara”>

<xs:complexType>

…

</xs:complexType>

</xs:element>

分离结构：<xs:element name=”sara” type=”xs:string”/>

<xs:complexType name=”string”>

….

</xs:complexType>

3、复合结构以及分离结构也同样适用于属性、简单元素的描述

四、schema的数据类型包括基本数据类型、简单数据类型和复杂数据类型。简单数据类型是对数据取值进行约束，是对基本数据类型的扩展；而复杂数据类型则是一个复杂的数据结构，通常包含了属性或子元素定义。

简单类型如：<xs:simpleType name=”XSDSimpleType”>

<xs:restriction base=”dateTime”>

<xs:enumeration value=”value1”/>

<xs:enumeration value=”value2”/>

<xs:enumeration value=”value3”/>

</xs:restriction>

</xs:simpleType>

五、约束是指对简单类型的元素或属性施加的值约束条件，主要包含范围约束、长度约束、枚举值、模式匹配、空白处理等。

1、范围约束：主要施加到数值型、日期型等类型，限制取值的范围。

如：<xs:restriction base=”xs:integer”>

<xs:minInclusive value=”0”/>

<xs:maxInclusive=”100”/>

</xs:restriction>

fractionDigits设置最大小数位数；totalDigits指定精确的数值位数；maxExclusive设置数值型的最大值；maxInclusive设置数值型的最大值；minExclusive设置数值型的最小值；minIxclusive设置数值型的最小值。

2、长度约束：施加到任何类型上限制值的长度。包括<xs:length>固定长度、<xs:maxLength>最大长度、<xs:minLength>最小长度。

3、枚举值约束：限制元素或属性的值只能在给定的列表中取值，并使用base属性指示值的数据类型。

4、模式匹配约束：限制元素或属性的值应满足给定的模式要求，并使用base属性指示值的数据类型。

如：<xs:restriction bas”xs:string”>

<xs:pattern value=”[a-z]”/>

</xs:restriction>

[a-z]a-z间的单个字符；

[A-Z][A-Z]两个字符，每个字符都在A-Z之间；

[a-zA-Z]a-z或A-Z间的单个字符；

[abc]单个字符a或b或c；

([a-z])\*零个或任意个字符a-z；

([a-z][A-Z])+一个或任意个字符对，每对字符大小写间隔出现；

“a”|”b”值只能是a或b,与枚举类似；

[a-zA-Z0-9]{8}精确的八个a-z或A-Z或0-9间的字符。

5、空白处理约束：限制XML文档解析器如何处理值得组成空白字符，使用<xs:whitespace>限制元素指示。

如：<xs:restriction base=”xs:string”>

<xs:whitespace value=”replace”/>

</xs:restriction>

Value的可能的取值包括：replace删除内容中的全部空白；presever保留内容中的全部空白；collapse删除前导、后导空白，替换内容中的一个或多个空白为一个空格。

六、属性本身总定义为简单类型的。属性可以指定缺省值、固定值、强制项、约束等属性。

如：<xs:attribute name=”ordered” type=”orderIdType”/>

<xs:simpleType name=”orderIdType”>

<xs:restriction base=”string”>

<xs:pattern value=”[A-Z]”d{8}”/>

</xs:restriction>

</xs:simpleType>

七、复杂类型指令：控制子元素如何在复杂元素中使用，包括三类指令：顺序、次数、分组。

1、顺序：控制子元素是否按顺序出现，包含三种类型。

（1）<xs:all>指示这些子元素最多出现一次，且顺序无关（minOccurs=0,maxOccurs=1）。例：<xs:complexType name=”fullpersoninfo”>

<xs:all>

<xs:element name=”address”type=”xs:string”/>

<xs:element name=”city”type=”xs:string”/>

<xs:element name=”country”type=”xs:string”/>

</xs:all>

</xs:complexType>

（2）<xs:sequecs>指示这些元素必须按顺序出现。

例：<xs:complexType name=”fullpersoninfo”>

<xs:sequecs>

<xs:element name=”address”type=”xs:string”/>

<xs:element name=”city”type=”xs:string”/>

<xs:element name=”country”type=”xs:string”/>

</xs:sequecs>

</xs:complexType>

（3）<xs:choice>指示这些元素中只能出现其一。

例：<xs:complexType name=”fullpersoninfo”>

<xs:choice>

<xs:element name=”address”type=”xs:string”/>

<xs:element name=”city”type=”xs:string”/>

<xs:element name=”country”type=”xs:string”/>

</xs:choice>

</xs:complexType>

2、次数：作为元素的属性，控制元素可以出现的次数，包含minOccurs和maxOccurs。

3、组：指示一个元素或属性集合。

对于schema文档，在生活中应用的实例很多，用处也是非常大，了解schema文档的结构对于创建和构造一个schema文档至关重要。