

초보 개발자의 성장기

대학교 3-1/웹서

HW #2 (XML Schema 설계)

Launa 2025. 5. 1. 12:27

- 실습문제 및 주의사항: 강의자료 게시글 참조

도서에 관한 정보를 XML로 표현하고 관리하기 위해 Book Markup Language(BookML) v2.0을 정의하고자 한다. 1. XML Schema를 이용해서 다음 요구사항을 만족하는 BookML 스키마를 설계하시오. 2. 설계한 스키마를 따르는 유효한 BookML 문서의 예를 작성하시오. (아래 예제 문서 참조)

가. 기본 요구사항

| | |
|----|--|
| R1 | BookML 문서는 도서들의 목록과 도서 분류 체계로 구성된다. |
| R2 | 각 도서는 제목, 저자, 출판사, 출판일, 가격, 할인율, 페이지수, 평점, 분야(category)에 대한 정보를 갖고, 추가적으로 개정일, 표지그림, 책소개, 독자리뷰를 가질 수 있다. 번역서는 원제목과 역자 정보도 있다. 저자와 분야, 개정일은 여러 개가 존재할 수 있고, 독자리뷰는 최대 50개까지만 가능하다. |
| R3 | 도서의 식별자는 ISBN10 (xxxxxxxxx-x) 또는 ISBN13 (978(또는 979)-xxxxxxxxx-x) 형식을 사용하고, 도서마다 서로 다른 유일한 값을 갖는다. |
| R4 | 도서의 출판일과 개정일은 연, 월, 일로 구성된다. 연도는 1900~2025 사이의 값을 갖고, 월과 일은 각각 01~12, 01~31 사이의 값을 갖는다. |
| R5 | 가격은 정가와 판매가(할인가)로 구분되고 양수 값을 갖는다. 가격은 원, 달러, 엔으로 표시된다. 원화인 경우 생략 가능하다(즉, 통화가 생략될 경우 원화로 간주함). 할인율(%)은 0이상 80이하의 값을 갖는다. |
| R6 | 페이지수는 1이상 2000이하의 값을 갖고, 평점(rating)은 1~5 사이의 실수 값을 갖는다. |
| R7 | 도서가 속한 분야 정보는 별도로 정의되는 도서 분류 체계에 포함된 분야(들)에 대응되어야 한다 (즉, 도서 분류 체계에 존재하는 분야(들)만 이용 가능). |
| R8 | 도서의 표지그림은 외부의 이미지 파일을 지정해야 하며, 파일의 포맷(JPEG, GIF, PNG 등)과 파일을 읽고 출력할 수 있는 응용 프로그램(예: imageView.exe)을 나타내야 한다. |

| | |
|-----|--|
| R9 | 다운로드 받을 수 있는 전자책(eBook) 도서는 위 R2에 나열된 정보들 뿐만 아니라 파일형식과 파일크기, 이용기간이 추가된다. 이용기간은 제한이 없을 경우 생략 가능하다. |
| R10 | 전자책의 파일형식은 EPUB, PDF, StreamText, StreamImage 중 하나의 값을 갖는다. 파일크기는 실 수 값이고 단위가 있다(예: 985 KB, 1.45 MB 등). 이용기간은 일 수로 정의된다. |
| R11 | 새로 나온 신간 전자책(newEBook)은 이용기간이 최대 3주이고, 개정일은 존재할 수 없다. |
| R12 | 도서 분류 체계는 문학, 인문, 사회, 과학, 유아, 컴퓨터 등 다양한 분야를 포함할 수 있다. 각 분야는 이름뿐만 아니라 식별자를 갖고, 다시 세부 분야로 나뉘 수 있다(예: 문학은 소설, 시, 희곡 등으로 세분화될 수 있고, 소설은 다시 한국소설, 영미소설, 장르소설 등으로 더 세분화 가능). 각 분야의 이름과 분류 체계의 구성은 DTD 문서에서 미리 정의되지 않고 BookML 문서마다 사용자에 의해 자유롭게 정의될 수 있어야 한다. |

나. 기타 요구사항

1. BookML 스키마 문서(bookML.xsd)는 다음과 같이 작성함

BookML을 위한 target namespace를 선언함. elementFormDefault 속성은 "qualified"로 선언함
 도서의 ISBN, 가격, 통화, 할인율, 페이지수, 평점, 표지그림, 파일형식, 파일크기 등을 위한 타입을 각각 전역 타입(global type)으로 정의함

- ISBN, 통화, 할인율, 페이지수, 평점 -> 값을 가지므로 Simple Type으로 정의

- 가격, 표지그림, 파일형식, 파일크기 -> 자식 엘리먼트나 속성을 가지므로 Complex Type으로 정의
 표지그림을 위한 Notation들을 선언하고 표지그림 타입에서 속성 정의 시 사용함

출판일과 개정일을 효율적으로 정의하기 위해 연, 월, 일에 대한 속성 그룹을 정의함. 연, 월, 일 속성 들은 타입은 지역적(local type)으로 정의함

출판일과 개정일 엘리먼트는 전역적(global element)으로 정의하고, 아래의 도서 타입들에서 참조함
 세 가지 도서 타입과 분야(category)에 관한 타입을 각각 전역 타입(global type)으로 정의함. 도서 타입들은 상속을 이용해서 정의함. 도서가 속한 분야는 도서의 자식 엘리먼트로 나타냄

- eBookType은 bookType을 확장하여 파일형식, 파일크기, 이용기간 엘리먼트들을 추가함

- newEBookType은 bookType과 eBookType의 모든 자식 엘리먼트들을 순서대로 재선언하되, 개정일의 최대 빈도 수를 제한하고, 이용기간에 대해 타입을 지역적(local type)으로 재정의(제한) 함

루트 엘리먼트(catalog)와 도서 목록(bookList), 분류 체계(bookCategory) 엘리먼트를 각각 전역적으로 정의하되, 이들의 타입은 지역적(local type)으로 정의함

<xsd:annotation>이나 주석을 이용하여 각 요구사항에 관련된 부분을 간략히 설명함

(선택사항) 도서의 식별자와 분야의 식별자에 대한 Key 제약조건들과, 도서가 속한 분야에 대한 참조 무결성(Keyref) 제약조건을 정의함

2. BookML 예제 문서에서는 BookML 언어에 대해 default namespace를 선언함. 도서 엘리먼트들은 타입의 다형성을 활용하여 표현함 (book 엘리먼트 및 xsi:type 속성 이용 -> 아래 예제 문서 참조)

3. 스키마 문서와 예제 문서에 대해 각각 유효성 검사를 실시하여 오류가 없음을 확인할 것

제출방법: BookML 스키마 파일과 BookML 예제 파일, 각 파일에 대한 유효성 검사 결과 화면(capture image) 들을 제출함 (과제 게시글 내용 참조)

참고: BookML 스키마를 따르는 예제 문서 (book-catalog.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE catalog [
```

```
  <!ENTITY favoriteAuthor SYSTEM "김연수"> <!-- 내부 일반 parsed entity 정의 -->
]>
```

```
<catalog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns="http://www.dongduk.ac.kr/bookML/2.0"
```

```
  xsi:schemaLocation="http://www.dongduk.ac.kr/bookML/2.0 bookML.xsd">
```

```
  <bookList>
```

```
    <book isbn="895468000-3"> <!-- 일반 책 -->
```

```
      <title>이토록 평범한 미래</title>
```

```
      <author>&favoriteAuthor;</author> <!-- 내부 일반 parsed entity 참조 -->
```

```
      <publisher>문학동네</publisher>
```

```
      <publishDate year="2022" month="10" day="07" />
```

```
<revisionDate year="2023" month="11" day="15" />
```

```
      <price currency="won">
```

```
        <listPrice>14000</listPrice>
```

```
        <salesPrice discountRate="10">12600</salesPrice>
```

```
      </price>
```

```
      <pages>276</pages>
```

```
<rating>4.8</rating>
```

```
      <category>c101-1-1</category> <!-- 문학/소설/한국소설 분야에 속함 -->
```

```
      <coverImage src="https://image.yes24.com/goods/113737429/XL" type="jpeg" />
```

```
      <description>김연수 9년 만의 신작 소설집. 종말 이후의 사랑에 대한 여덟 편의 이야기. 작가
```

```
가
```

최근 2~3년간 집중적으로 단편 작업에 매진한 끝에 선보이는 소설집으로, '시간'을 인식하는 김연수의 변화된

```
시각을 확인할 수 있게 한다. ... </description>
```

```
<review>"우리가 달까지 갈 수는 없지만 갈 수 있다는 듯이 걸어갈 수는 있다. 달이 어디에 있
```

```
는지 찾을 수만 있다면. 마찬가지로 우리는 달까지 걸어가는 것처럼 살아갈 수 있다. 희망의 방향만 찾
```

```
을 수
```

있다면." (-<진주의 결말> 중에서) ... </review>

```
<review>...</review> ...
```

```
</book>
```

```
<book isbn="978-893497246-4" xsi:type="eBookType">  <!-- 전자책 -->
  <title>사피엔스</title>
  <originalTitle>Sapiens: A Brief History of Humankind & Earth</originalTitle>  <!--원제-->
  <author>유발 하라리</author>
  <translator>조현욱</translator>  <!-- 역자 -->
  <publisher>김영사</publisher>
  <publishDate year="2014" month="05" day="19" />
<price>  <!-- currency="won" -->
  <listPrice>12000</listPrice>
  <salesPrice discountRate="20">9600</salesPrice>
</price>
<pages>241</pages>
<rating>4.2</rating>
<category>c102-1</category>  <!-- 인문/일반교양, 역사/세계사 분야에 속함 -->
  <category>c103-2</category>
  <description>변방의 유인원 호모 사피엔스는 어떻게 세상의 지배자가 되었는가? 수렵채집
을 하
던 우리 조상들은 어떻게 한 곳에 모여 도시와 왕국을 건설하였는가? 인간은 왜 지구 역사상 가장 치명
적인
동물이 되었는가? 과학은 모든 종교의 미래인가? ... </description>
  <format value="EPUB" />  <!-- eBook 파일형식 -->
  <size unit="MB">26.60</size>  <!-- eBook 파일크기 -->
  <period>30</period>  <!-- eBook 이용기간 -->
</book>
...
<book isbn="979-119238935-6" xsi:type="newEBookType">  <!-- 신간 전자책 -->
  <title>마음의 기술: 부정적 감정을 지우는 효과적인 뇌 사용법</title>
<originalTitle>Devenez Votre Propre Psy</originalTitle>
  <author>안-엘렌 클레르</author>
  ...
  <publishDate year="2024" month="11" day="18" />
  <revisionDate year="2025" month="03" day="10" />  <!-- 오류: 개정일 존재 불가! -->
  ...
<category>c102-2</category>  <!-- 인문/심리학 분야에 속함 -->
  <format value="EPUB" />
  <size unit="MB">43.42</size>
  <period>25</period>  <!-- 오류: 이용기간은 3주 이내! -->
```

```
</book>
```

```
...
```

```
</bookList>
```

```
<bookCategory>
```

```
  <category cid="c101" name="문학">
```

```
    <category cid="c101-1" name="소설">
```

```
      <category cid="c101-1-1" name="한국소설"> ... </category>
```

```
      <category cid="c101-1-2" name="영미소설"> ... </category>
```

```
    ...
```

```
  </category>
```

```
...
```

```
</bookCategory>
```

```
</catalog>
```

- 제출 결과물:

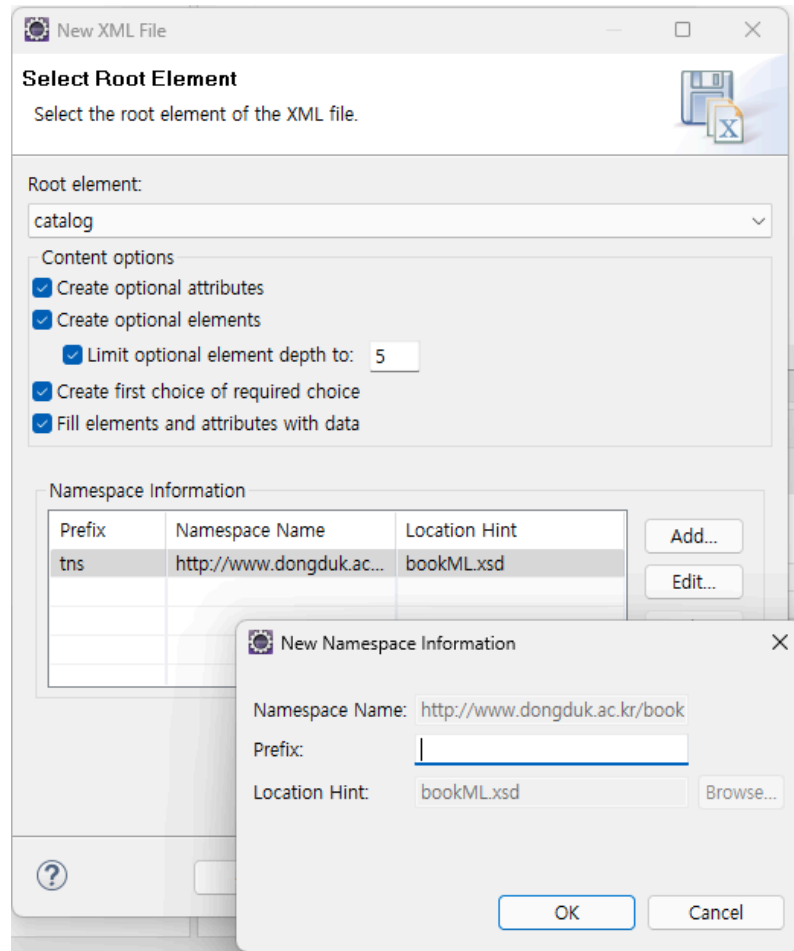
- XML Schema 문서(.xsd) 및 BookML 예제 문서(.xml) : 파일들을 첨부함
- XML Schema 문서 및 BookML 예제 문서에 대한 유효성 검사 결과 화면 : 각 문서의 내용과 유효성 검사 결과 팝업창이 함께 보이도록 화면을 캡처한 후 게시글 안에 직접 삽입함

실습 #2 부연 설명

1. XML Schema를 따르는 XML 파일 생성 방법

Eclipse에서 new > XML File 메뉴를 이용해서 XML 파일 생성 시,

- Create file using a DTD or XML Schema file 옵션을 선택하고, XML Schema 파일을 선택한 후
- 마지막 단계의 화면에서 아래 그림과 같이 Root element를 catalog로 선택하고
- 아래 쪽의 Namespace Information에서 tns 접두사를 마우스로 선택한 후 오른쪽의 Edit 버튼을 클릭하고 Prefix: 창의 값을 삭제하고 공란으로 만들면, XML 문서에서 target namespace가 default namespace로 선언되어 엘리먼트에 접두사를 생략하고 작성할 수 있음



2. 이름 참조 시 namespace prefix 이용

Schema 문서 내에서 정의되는 모든 이름(엘리먼트, 속성, 타입, 노테이션, 제약 등의 이름)을 다른 타입에서 이용하거나 참조할 때는 (즉, type, ref, value 속성의 값으로 사용할 때는) 반드시 그것이 속한 **target namespace**에 대한 **namespace prefix**(접두사)를 붙여야 합니다.

그렇지 않으면 XML Parser가 그 이름에 해당하는 것을 찾지 못해 오류가 발생합니다.

(target namespace를 default namespace로 지정한 경우에는 이름 앞에 접두사를 생략할 수 있으나, 대신 XSD의 엘리먼트나 타입 이름 앞에 xsd 접두사를 붙여야 합니다).

주의: 접두사는 이미 정의된 이름을 사용할 때 필요하며, 엘리먼트, 속성, 타입, 노테이션, 제약 등의 이름을 처음 정의할 때는 접두사를 붙이면 안 됨!

3. 재귀적인 구조를 갖는 엘리먼트 정의

<category>와 같이 재귀적(recursive)으로 중첩된 구조를 갖는(즉, <category>가 <category>들을 자식으로 갖는) 엘리먼트를 정의하려면 <category> 엘리먼트를 위한 타입을 global type으로 정의하되, 그 정의 안에서 <category>를 자식 엘리먼트로 포함시키고 그 자식 엘리먼트의 타입은 현재의 타입을 지정하면 됩니다.

예:

- XML Schema

```
<complexType name="myType">
  <sequence>
    <element name="myElement" type="tns:myType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<element name="myElement" type="tns:myType"/>
```

- XML data

```
<myElement>
  <myElement>
    <myElement/>
  </myElement>
</myElement>
```

4. complex type의 제한(restriction) 시 주의사항

문제에서 eBookType을 제한하여 새로운 newEBookType을 정의할 때, 아래와 같은 오류가 발생하면 자식 엘리먼트들에 대한 재선언이 잘못된 것입니다.

- The particle of the type is not a valid restriction of the particle of the base.
- There is not a complete functional mapping between the particles.

이 경우 다음 사항들을 확인해 보기 바랍니다.

(1) 부모 그리고 조부모에 해당하는 타입들에 포함된 자식 엘리먼트들을 모두 순서대로 재선언 했는가?

- 주의: 엘리먼트들의 순서가 바뀌면 안되고, 조부모의 자식들이 먼저 오고, 그 뒤에 부모의 자식 엘리먼트들이 와야 함

(2) 부모(또는 조부모) 타입에 포함된 자식 엘리먼트들이 모두 **global type**을 이용하거나, **global element**를 참조(ref)해서 선언되어 있는가?

- 주의: 부모 또는 조부모 타입 정의 안에서 자식 엘리먼트의 타입이 local하게(즉, 엘리먼트 선언 내부에서) 정의되어 있으면, 새로운 자식 타입에서 **재선언할 수 없고 제한도 할 수 없음**
- local type은 name이 없기 때문에 base type으로 사용할 수 없고, 똑같이 재정의할 경우에도 동일한 구조의 타입인지 확인이 어렵기 때문임

(3) 부모(또는 조부모) 타입에 포함된 자식 엘리먼트를 재선언 하면서 타입을 제한할 때, base 속성 값으로 원래의 타입을 주었는가?

- 주의: 반드시 부모(또는 조부모) 타입에 포함된 **자식 엘리먼트의 타입**(즉, 자식 엘리먼트 선언에 사용된 global type)을 base로 지정하고 그 타입의 범위를 제한하는 새로운 타입을 local하게 (또는 global type으로) 정의해서 사용해야 함
- 예: 부모 타입에서 int 타입을 사용했는데 자식 타입에서는 integer 타입을 base로 사용하면 오류 발생 (int와 integer는 다른 타입)

(4) minOccurs, maxOccurs를 변경할 경우 빈도 수의 범위가 기존 범위 안에 포함되도록 좁혀져야 함

5. key, keyref 관련 참고 사항

- category에 대한 <key> 제약 정의

재귀적인 구조를 갖는 모든 <category>들에 대해 cid 속성을 키로 정의하고자 할 때, 만일 <key> 제약을 루트 엘리먼트인 <bookCatalog>에서 정의하면 <selector>의 XPath 식이 `"/bookCategory//category"`와 같이 되어야 합니다. (// 는 모든 자손 엘리먼트들을 의미함)
그러나 XML Schema에서는 XPath 식에서 `"/"`이 처음에만 나올 수 있고 중간에는 나올 수 없게 제한되어 있습니다.

따라서 <category>들에 대한 <key> 제약조건은 <bookCatalog>가 아니라 <bookCategory> 엘리먼트 선언 안에서 정의하기 바랍니다. 그러면 <selector>의 기준 엘리먼트가 <bookCategory>가 되어 xpath식을 "../category"와 같이 나타낼 수 있습니다.

```
<element name="bookCategory">
```

```
...
```

```
  <xsd:key name="...">
    <xsd:selector xpath="..." />
    <xsd:field xpath="..." />
  </xsd:key>
</xsd:element>
```

- book에 대한 <keyref> 제약

도서에 대해 그것이 속하는 여러 category들을 나타내고자 할 때, HW #1의 DTD에서는 book에 속한 categories 속성을 IDREFS 타입으로 정의했으나 HW #2에서는 <keyref>를 이용해야 합니다.

<keyref>는 하나의 값을 참조하는 것은 문제가 없으나, IDREFS와 같이 여러 개의 값을 갖는 리스트 타입의 속성이나 엘리먼트에 대해서는 그 값들이 올바른 참조 값들인지 검사하지 못합니다.

그래서 하나의 도서가 여러 category들의 key 값들을 참조하게 하려면 도서의 category 정보를 속성 대신 자식 엘리먼트로 표현하도록 정의해야 합니다.

즉, 문제에 포함된 예제 XML 문서와 같이 <book>이 <category> 엘리먼트들을 자식으로 갖는 아래와 같은 구조가 되도록 정의하고, 그 자식 엘리먼트들을 대상으로 <keyref> 제약을 정의하기 바랍니다.

```
<bookCatalog>
  <bookList>
    <book ... >
      <title>... </title>
      <author>... </author>
      ...
      <category>c101</category>
      <category>c101-1</category>
      <category>c102-2</category>
      <category>c103</category>
    </book>
    ...
  </bookList>
  <bookCategory>
    <category cid="c101">
      <category cid="c101-1" />
      <category cid="c101-2" />
      <category cid="c101-2">
    </category>
    <category cid="c102">
      <category cid="c102-1" />
      <category cid="c102-2" />
      <category cid="c101" />
    </category>
    ...
```

```
</bookCategory>
</bookCatalog>
```

위와 같은 구조에서 bookList/book/catogory 들에 대해 <keyref> 제약조건을 정의하려면 <selector>에서 bookList/book/catogory들을 적용 대상으로 지정하고 <field>에서는 xpath="." (즉, category 엘리먼트 자체)로 지정하면 됩니다.

(주의: <field>의 xpath식은 반드시 하나의 속성이나 엘리먼트가 결과로 선택되는 식이어야 하므로, <selector>를 bookList/book으로 하고 <field>를 category로 지정할 수 없음)

이 제약은 위 3번에서 정의한 bookCategory/category들에 대한 <key> 제약을 참조(refer)해야 합니다.

- XPath식에서 namespace 접두사 사용

Schema 문서에서 elementFormDefault="qualified"로 설정한 경우 XML 문서에서는 모든 엘리먼트 이름 앞에 namespace 접두사를 붙여야 합니다.

default namespace를 이용해서 접두사를 생략할 경우 XML Parser가 내부적으로 namespace를 나타내는 접두사를 정의해서 모든 이름 앞에 붙이게 됩니다(qualification).

그리고 XML parser가 <key>나 <keyref> 등의 제약조건을 검사할 때 접두사가 붙어있는 이름 (Qualified Name(QName))을 검사하는 것이므로, Schema 문서에서 <key>나 <keyref>를 정의할 때 XPath 식 안에서도 엘리먼트 이름에 접두사를 붙여서 작성해야 합니다.

따라서, 위 4~5번 설명에서는 생략되어 있지만, <selector>나 <field>에서 xpath="tns:booklist/tns:book" 와 같이 정의해야 합니다.

내 코드는 처참해서..교수님코드

```

bookML.xsd
C:/Users/Changsup%20Park/eclipse-workspace-new/XMLExamples/GolfCountr...

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.dongduk.ac.kr/bookML/2.0"
  xmlns:tns="http://www.dongduk.ac.kr/bookML/2.0" elementFormDefault="qualified">

<!-- type definitions -->
  <simpleType name="ISBNType">
    <annotation>
      <documentation>book의 ID(ISBN) attribute에 대한 타입 정의</documentation>
    </annotation>
    <restriction base="string">
      <pattern value="(97(8|9)-)?#d{9}-#d"/>
    </restriction>
  </simpleType>

  <complexType name="priceType">
    <annotation>
      <documentation>도서 가격에 관한 데이터 타입 정의. currency 속성의 기본값은 "won"</documentation>
    </annotation>
    <sequence>
      <element name="listPrice" type="decimal"/>
      <element name="salesPrice">
        <complexType>
          <simpleContent>
            <extension base="decimal">
              <attribute name="discountRate" type="tns:discountRateType"/>
            </extension>
          </simpleContent>
        </complexType>
      </element>
    </sequence>
    <attribute name="currency" type="tns:currencyType" default="won"/>
  </complexType>

  <simpleType name="currencyType">
    <annotation>
      <documentation>가격의 통화(currency)를 위한 데이터 타입 정의</documentation>
    </annotation>
    <restriction base="string">
      <enumeration value="won"/>
      <enumeration value="dollar"/>
      <enumeration value="yen"/>
    </restriction>
  </simpleType>

  <simpleType name="discountRateType">
    <annotation>
      <documentation>책의 할인율을 나타냄</documentation>
    </annotation>
    <restriction base="integer">
      <minInclusive value="0"/>
      <maxInclusive value="80"/>
    </restriction>
  </simpleType>

  <simpleType name="pagesType">
    <annotation>
      <documentation>책의 페이지 수를 나타냄</documentation>
    </annotation>
    <restriction base="integer">
      <minInclusive value="1"/>
      <maxInclusive value="2000"/>
    </restriction>
  </simpleType>

  <simpleType name="ratingType">
    <annotation>
      <documentation>책의 평점을 나타냄</documentation>
    </annotation>
    <restriction base="double">
      <minInclusive value="1"/>
      <maxInclusive value="5"/>
    </restriction>
  </simpleType>

  <complexType name="fileFormatType">
    <annotation>
      <documentation>eBook의 파일 형식을 나타냄</documentation>
    </annotation>
    <attribute name="value">
      <simpleType>
        <restriction base="string">
          <enumeration value="EPUB"/>
          <enumeration value="PDF"/>
          <enumeration value="StreamText"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>

```

```

        <enumeration value="StreamImage" />
    </restriction>
</simpleType>
</attribute>
</complexType>

<complexType name="fileSizeType">
    <annotation>
        <documentation>eBook의 파일 크기를 나타냄</documentation>
    </annotation>
    <simpleContent>
        <extension base="decimal">
            <attribute name="unit" type="string" />
        </extension>
    </simpleContent>
</complexType>

<!-- notation definitions -->

<!-- 표지그림을 위한 Notation 선언 -->
<notation name="bmp" public="img/bmp" system="photoshop.exe" />
<notation name="gif" public="img/gif" system="photoshop.exe" />
<notation name="jpeg" public="img/jpeg" system="photoshop.exe" />

<!-- NOTATION을 기반으로 표지그림 타입 정의 -->
<complexType name="bookCoverType">
    <attribute name="src" type="anyURI" />
    <attribute name="type">
        <simpleType>
            <restriction base="NOTATION">
                <enumeration value="tns:bmp" />
                <enumeration value="tns:gif" />
                <enumeration value="tns:jpeg" />
            </restriction>
        </simpleType>
    </attribute>
</complexType>

<!-- attribute group definitions -->

<attributeGroup name="dateAttrGroup">
    <annotation>
        <documentation>출판일/개정일을 위한 속성 그룹 정의</documentation>
    </annotation>
    <attribute name="year" use="required">
        <simpleType>
            <restriction base="gYear">
                <minInclusive value="1900" />
                <maxInclusive value="2025" />
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="month" use="required">
        <simpleType>
            <annotation>
                <documentation>주의: base type으로 gMonth를 이용하면 XML 문서에서 '--01' 형식의 값을 사용해야 함</documentation>
            </annotation>
            <restriction base="string">
                <pattern value="0[1-9]|1[0-2]" />
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="day" use="required">
        <simpleType>
            <annotation>
                <documentation>주의: base type으로 gDay를 이용하면 XML 문서에서 '---01' 형식의 값을 사용해야 함</documentation>
            </annotation>
            <restriction base="string">
                <pattern value="0[1-9]|1[1-2][0-9]|3[01]" />
            </restriction>
        </simpleType>
    </attribute>
</attributeGroup>

<!-- global element declarations -->

<element name="publishDate"> <!-- 출판일 -->
    <complexType>
        <attributeGroup ref="tns:dateAttrGroup" />
    </complexType>
</element>

<element name="revisionDate"> <!-- 개정일 -->
    <complexType>
        <attributeGroup ref="tns:dateAttrGroup" />
    </complexType>
</element>

```

<!-- 주의: 아래와 같이 date를 위한 global type을 정의해서 사용해도 되나 이 경우 속성 그룹을 이용하는 것은 의미가 없음 -->

```
<!--
<complexType name="dateType">
  <attributeGroup ref="tns:dateAttrGroup"/>
</complexType>
<element name="publishDate" type="tns:dateType"/>
<element name="revisionDate" type="tns:dateType"/>
```

-->

```
<complexType name="bookType">
  <annotation>
    <documentation>일반 책에 대한 데이터 타입 정의</documentation>
  </annotation>
  <sequence>
    <element name="title" type="string"/>
    <element name="originalTitle" type="string" minOccurs="0"/>
    <element name="author" type="string" minOccurs="1" maxOccurs="unbounded"/>
    <element name="translator" type="string" minOccurs="0"/>
    <element name="publisher" type="string"/>
    <element ref="tns:publishDate"/>
    <element ref="tns:revisionDate" minOccurs="0" maxOccurs="unbounded"/> <!-- 전역 엘리먼트를 참조 -->
    <element name="price" type="tns:priceType"/>
    <element name="pages" type="tns:pagesType"/>
    <element name="rating" type="tns:ratingType"/>
    <element name="category" type="string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="coverImage" type="tns:bookCoverType" minOccurs="0"/>
    <element name="description" type="string" minOccurs="0"/>
    <element name="review" type="string" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="isbn" type="tns:ISBNType" use="required"/>
</complexType>
```

```
<complexType name="eBookType">
  <annotation>
    <documentation>전자책에 대한 데이터 타입 정의(bookType을 확장)</documentation>
  </annotation>
  <complexContent>
    <extension base="tns:bookType">
      <sequence>
        <element name="format" type="tns:fileFormatType"/>
        <element name="size" type="tns:fileSizeType"/>
        <element name="period" type="positiveInteger"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<complexType name="newEBookType">
  <annotation>
    <documentation>신간 전자책에 대한 데이터 타입 정의(eBookType을 제한)</documentation>
  </annotation>
  <complexContent>
    <restriction base="tns:eBookType">
      <!-- 주의: bookType과 eBookType의 모든 자식 엘리먼트를 재선언해야 함(순서도 동일해야 함) -->
      <sequence>
        <element name="title" type="string"/>
        <element name="originalTitle" type="string" minOccurs="0"/>
        <element name="author" type="string"
          minOccurs="1" maxOccurs="unbounded"/>
        <element name="translator" type="string" minOccurs="0"/>
        <element name="publisher" type="string"/>
        <element ref="tns:publishDate"/>
        <element ref="tns:revisionDate" minOccurs="0"
          maxOccurs="0"/> <!-- 최대 출현 가능 횟수를 0으로 제한 -->

```

>

```

        <element name="price" type="tns:priceType"/>
        <element name="pages" type="tns:pagesType"/>
        <element name="rating" type="tns:ratingType"/>
        <element name="category" type="string" minOccurs="0" maxOccurs="unbounded"/>
        <element name="coverImage" type="tns:bookCoverType" minOccurs="0"/>
        <element name="description" type="string" minOccurs="0"/>
        <element name="review" type="string" minOccurs="0" maxOccurs="unbounded"/>
        <element name="format" type="tns:fileFormatType"/>
        <element name="size" type="tns:fileSizeType"/>
        <element name="period">

```

```

          <simpleType> <!-- period 엘리먼트의 타입 제한 -->

```

```

            <restriction base="positiveInteger"> <!-- 주의: base 타입으로 반드시 기존 타입을

```

사용 -->

```

              <maxInclusive value="21"/> <!-- 이용기간을 최대 21로 제한 -->

```

```

            </restriction>

```

```

          </simpleType>

```

```

        </element>

```

```

      </sequence>

```

```

    </restriction>

```

```

  </complexContent>

```

```
</complexType>
```

```
<complexType name="categoryType">
```

```
  <sequence>
```

```
    <element name="category" type="tns:categoryType">
```

```

    <element name="category" type="tns:categoryType"
      minOccurs="0" maxOccurs="unbounded" />
  </sequence>
  <attribute name="cid" type="string" />
  <attribute name="name" type="string" />
</complexType>

<element name="bookList">
  <complexType>
    <sequence>
      <element name="book" type="tns:bookType"
        minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>

<element name="bookCategory">
  <complexType>
    <sequence>
      <element name="category" type="tns:categoryType"
        minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <key name="caKey">
    <annotation>
      <documentation>./category에 대한 key 제약 선언 (key는 cid 속성)</documentation>
    </annotation>
    <selector xpath="."/></selector>
    <field xpath="@cid"></field>
  </key>
</element>

<element name="catalog">
  <annotation>
    <documentation>루트 엘리먼트 선언</documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="tns:bookList" /> <!-- global element 참조 -->
      <element ref="tns:bookCategory" /> <!-- global element 참조 -->
    </sequence>
  </complexType>
  <key name="bookKey">
    <annotation>
      <documentation>./bookList/book에 대한 key 제약 선언 (key는 isbn 속성)</documentation>
    </annotation>
    <selector xpath="./tns:bookList/tns:book" />
    <field xpath="@isbn" />
  </key>
  <keyref name="caRef" refer="tns:caKey">
    <annotation>
      <documentation>
        ./bookList/book/category에 대한 keyref 제약 선언 (bookCategory 엘리먼트에 선언된 tns:caKey 제약 참조)
        즉 ./bookList/book/category/. 의 값이 bookCategory//category/@cid 값을 참조
      </documentation>
    </annotation>
    <selector xpath="./tns:bookList/tns:book/tns:category" />
    <field xpath="."/>
  </keyref>
</element>
</schema>

```

공감

대학교 3-1 > 웹서' 카테고리의 다른 글

웹서 기말고사 대비 (0)

2025.06.26

HW #4 (HTML DOM, jQuery Programming) (0)

2025.05.01

HW #3 (DOM Programming) (0)

2025.05.01

'대학교 3-1/웹서' Related Articles

NO IMAGE

NO IMAGE

NO IMAGE

웹서 기말고사 대비

HW #4 (HTML DOM,
jQuery Programming)

HW #3 (DOM
Programming)

초보 개발자의 성장기
Launa 님의 블로그입니다.

댓글 0

Launa

내용을 입력하세요.

등록