Journal of Digital Contents Society Vol. 5. No. 3. September. 2004(pp. 180-186)

XML과 관계 데이터베이스 자료 간의 매핑 설계

김길준*

요 약

XML은 전자상거래 분야 뿐만 아니라, 모든 학술분야와 산업분야에서 사용하고 있는 만큼, 표준화된 문서를 다루는 필수적인 기능을 가지고 있다. 또한 XML 자료를 관계형 데이터베이스 자료로 만들어 SQL 언어를 사용하여 자료 검색도 필요하다. XML과 관계 데이터베이스 간의 변환을 위해선 두 자료간의 매핑 관계가 필요하다. 그래서 본 논문에서는 문서 자체에서 항상 통일된 형식으로 문서를 작성하게 하고, 문서를 읽을 때마다 이런 통일성을 자동적으로 검사하게 하는 DTD(Document Type Definition)를 사용해서 XML과 데이터베이스 간의 매핑 관계를 연구하고, 그 결과 XML Data를 Realation Database table로 매핑하는 매핑 설계도를 연구하였다.

Mapping Design between XML and Table in Relation Database

Gil-Choon Kim*

Abstract

XML has an essential function of dealing with standardized document to be used in all academic areas and industrial areas as well as in e-commerce. The transformation of XML data into Relation Database table is also necessary for data search using SQL language. Mapping relation between XML and Table in Relation Database is required for transformation of XML. This article studies the mapping relation between XML and Relation Database using DTD which enables to check the unity automatically whenever document is read so that it studied mapping design for the transformation of XML data into Relation Database table.

Key words : DTD, mapping

1. 서 론

문서 중심 문서에 XML전용 데이터베이스를 사용하는 이유는 ① 문서 중심 문서는 비 구조적인 형태이지만 XML문서는 정규적인 문서를 갖는 관계형 데이터베이스에 저장하기는 너무나 변이가 많은 경우이다. 즉, 문서를 관계형 데이터베이스로 저장하기에는 테이블 수가 너무 많거나 null 값을 갖는 칼럼이 너무 많기 때문에 비효율적이므로 , XML 전용 데이터베이스에 저장함으로서 효율성을 높일 수 있다. ② XML 전용 데이터베이스에는 XML문서 전체가 한 곳에 저장되기 때문에 여러 테이블에 저장되는 관계형 데이터베이스에 비해서 검색 속도가 훨씬 빨라지게 된다.그러므로 XML 전용 데이터베이스는 관계형 데이터베이스에 비해서 검색 속도를 높일 수 있다. ③XML 질의어를 사용하는 문제가 있지만 이 부분에서는 대부분의 관계형 데이터베이스도 XML질의어를 지원하기 때문에 이 문제는 크게 중요하지 않다. XML 전용 데이터베이스를 사용할 때 문제점은 질의어를 통해서 나온 결과가 XML문서라는 것이다.따라서 질의어의 결과가 XML문서이므로 원하는 데이터를 얻기 위해서는 다시 SAX나 DOM를 이용해야 한다는 단점이 있다. 여기에서는 XML 데이터베이스를 관계형 데이터베이스 테이블로 매핑하는 매핑 설계도를 제시하고 있다.

2. XML과 데이터베이스와의 매핑

XML 문서와 데이터베이스 사이의 데이터 이동은 문서와 데이터베이스의 매핑을 이용하는 소프트웨어에 의해서 이루어지며, 이러한 매핑은 2가지 방법으로 분류

※ 제일저자(First Author) : 김길준
접수일 : 2004년 7 월 23 일, 완료일 : 2004년 8 월 18 일
* 성결대학교 공과대학 전자상거래학부 부교수
kimid@sungkyul.edu

할 수 있다.

♦템플릿 드리븐 매핑(templete-driven mapping)

♦모델 드리븐 매핑(model-driven mapping)

템플릿 드리븐 매핑은 명령어의 처리결과를 그 결과의 임의의 장소에 삽입할 수 있으며, 구조어 명령과 같은 프로그램 구조를 사용할 수 있다.

모델 드리븐 매핑에서 XML 문서의 데이터는 미리 정의된 모델에 따라서 XML 문서를 하나의 테이블이나 테이블의 집합으로 매핑된다. 모델 드리븐 매핑은 다시 2가지의 매핑 기법으로 분류된다.

♦테이블 기반 매핑(table-based mapping)

♦개체-관계 매핑(entity-relational mapping)

❐ 테이블 기반 매핑

테이블 기반 매핑은 XML문서와 관계형 데이터베이스 사이의 데이터를 이동시킬 때 많이 사용되며, XML 문서를 하나의 테이블이나 테이블들의 집합으로 매핑한다. 이 방법은 관계형 데이터베이스를 직렬화 할 때 효과적일 수 있다.

❐ 개체-관계 매핑

테이블 기반 매핑은 매우 제약적인 형태의 XML 문서에만 적용될 수 있다. 이 방법은 XML문서를 개체의 트리 형태로 보고, SQL3나 기존의 기술을 통해 개체-관계 매핑을 통해 Class들은 테이블로 매핑되고, 스칼라 속성들은 칼럼으로 매핑되면서 데이터베이스에 저장된다, 개체-관계 매핑은 두 단계로 이루어진다. 첫째 단계는 XML DTD가 개체 DTD로 매핑되는 단계이다. 두 번째 단계는 개체 DTD가 데이터베이스 DTD로 매핑되는 단계이다. DTD를 개체로 매핑하는 것은 원소의 타입과 데이터 타입을 인식하는 단계에서부터 시작된다. 이때 DTD내의 PCDATA를 갖는 원소와 속성은 모두 단순 타입으로 간주되고, 단순 타입은 모두 스칼라 타입(기본 타입형,string)으로 매핑된다. 자식 원소를 갖거나 속성을 갖는 원소들은 모두 복합 타입으로 간주된다. 복합 타입인 경우에는 내부에 구조적인 값을 가지고 있기 때문에 클래스로 매핑되며, 복합 타입의 내부에 있는 자식 원소들과 속성은 모두 클래스의 멤버 필드로 매핑된다. 이때 자식 원소와 속성은 모두 멤버 필드로 매핑되기 때문에 차이점이 없다. DTD를 이용해서 개체로 변환하는 경우에 이러한 데이터 타입으로 매핑되는 것은 사람의 간섭이 있어야만 가능하다. 그러나 XML DTD를 사용하는 경우에는 XML DTD의 데이터 타입이 직접 개체의 데이터 타입으로 변환되는 것이 가능하다. 개체-관계 매핑에서 두 번째 단계에서 클래스는 테이블로 매핑되고, 스칼라 타입은 칼럼으로 매핑된다. 또한 포인터와 레퍼런스 관계는 기본 키와 외래 키의 관계로 매핑한다. 이 관계에서 부모와 자식 원소는 1:1 관계인 경우에 기본 키는 어느 테이블에 있든지 관계없다. 만약 관계가 1:N 인 경우에는 원소의 부모 자식 관계에 상관없이 1 부분이 항상 기본 키를 가지고 있어야 한다. 매핑 단계에서 기본 키 칼럼이 생성된다. 만약 기본 키 칼럼이 생성되었다면, 기본 키의 값은 데이터베이스나 데이터 변환 소프트웨어에 의해서 자동적으로 생성되게 된다.

복잡한 XML문서는 여러 개의 테이블로 구성된 하나의 데이터베이스로 표현될 수 있지만 XML 문서의 물리적인 구조,문서 정보(DTD),주석,처리 명령어 등은 저장할 수 없다.테이블 기반 매핑은 미들웨어를 이용해서 XML문서와 데이터베이스 사이에 데이터를 이동시키는 경우에 많이 사용된다. 특히 웹 응용 프로그램에서 관계형 데이터베이스에 있는 정보를 XML 문서 형태로 변환할 때 많이 사용된다. 테이블 기반 매핑은 다음과 같은 장점을 가지고 있다.

♦ 장점

-간단하다. XML 문서 구조가 관계형 데이터베이스의 테이블에 쉽게 매핑 될 수 있고, 변환 소프트웨어를 쉽게 작성할 수 있다.

-변환 소프트 웨어가 빠르고, 확장성이 높다.

♦단점

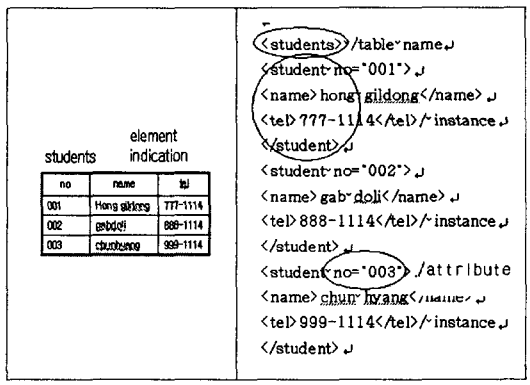
-간단한 XML 문서에만 적용될 수 있다.

-XML 문서의 물리적구조(entity 참조, CDATA 섹션 등), 문서의 정보(DTD), 주석, 처리 명령어 등을 저장할 수 없다.

2.1 단일 테이블 기반 매핑

XML문서를 하나의 테이블 혹은 테이블들의 집합으로 간주되며, 관계형 데이터베이스 테이블의 인스탄스를 XML문서의 요소와 속성으로 표현한다.

표 1. 테이블과 XML문서 매핑



❐속성 중심 매핑

속성 중심 구조는 요소로 표현되고, 내용은 속성으로 표현되어 구조와 내용을 분리할 수 있으며, 속성은 순서를 가지지 않으므로 복잡성을 감소 시킬 수 있다.

표 2. 속성 중심의 매핑 표현

no	name	tel
001	hong gildong	777-1114
001	gab doli	888-1114
003	chun hyang	999-1114

```
<students>
<student no="001" name="hong gildong" tel="777-1114"/>
<student no="001" name="gab doli" tel="888-1114"/>
<student no="003" name="chun hyang" tel="999-1114"/>
</students>
```

❑요소 중심 매핑

요소중심의 데이터 표현은 순서와 별로 중요하지 않으며, 또한 데이터 확장과 같은 변화에도 유연하게 대처할 수 있다.

표 3. 요소 중심 매핑 표현

no	name	tel
001	hong gildong	777-1114
001	gab doli	888-1114
003	chun hyang	999-1114

```
<students>
<student>
<no>001</no>
<name>hong gildong</name>
<tel>777-1114</tel>
</student>
<student>
<no>002</no>
<name>gab doli</name>
<tel>888-1114</tel>
</student>
<student>
<no>003</no>
<name>chun hyang</name>
<tel>999-1114</tel>
</student>
</students>
```

❑ DTD 를 정의한 경우 매핑

관계형 데이터베이스와 DTD를 매핑하여 XML문서로 Data를 검색할 수 있다.

표 4. DTD에 따른 XML문서 매핑 관계

students

no	name	tel
001	hong gildong	777-1114
002	gab doli	888-1114

```
<!ELEMENT students(student)>
<!ELEMENT student(no,name,tel)>
<!ELEMENT no (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT tel (#PCDATA)>
```

```
<students>
 <student>
 <no>001</no>
 <name>hong gildong</name>
 <tel>777-1114</tel>
</student>
 <student>
 <no>002</no>
 <name>gab doli</name>
 <tel>888-1114</tel>
</student>
</students>
```

students

no	name	tel
001	hong gildong	777-1114
002	gab doli	888-1114

```
<!ELEMENT students(student)>
<!ELEMENT student(no,name,tel)>
<!ELEMENT no (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT tel (#PCDATA)>
```

```
<students>
  <student>
<no>001</no>
<name>hong gildong</name>
<tel>777-1114</tel>
</student>
 <student>
<no>002</no>
<tel>888-1114</tel>
<name>gab doli</name>
</student>
</students>
```

이처럼 DTD를 정의한 XML문서와 관계형 데이타베이스를 매핑할 때는 순차성, 반복성, 데이터 타입 등에 주의해야 한다.

2.2 다중 테이블 기반 매칭

관계형 데이터베이스는 개체를 연결해서 정보를 나타내며, 이를 XML문서로 매핑하면 중첩 표현을 사용해서 표현한다. 다음은 students table에서는 'no' field가 기본 키, course table 에서는 'sno' field가 외래 키일 경우이다.

students

no	name	tel
001	hong gildong	777-1114
002	chun hyang	888-1114

course

id	sno	subject
111	001	OS
222	002	DB
333	001	DS

ⓐ XML 기술 방법1

```
<students>
<student no="001" name="hong gildong" tel="777-1114">
  <course id="111" subject="OS"/>
  <course id="333" subject="DS"/>
</student>
<student no="002" name="chun hyang" tel="888-1114">
  <course id="222" subject="DB"/>
</student>
</students>
```

ⓑ XML기술 방법2

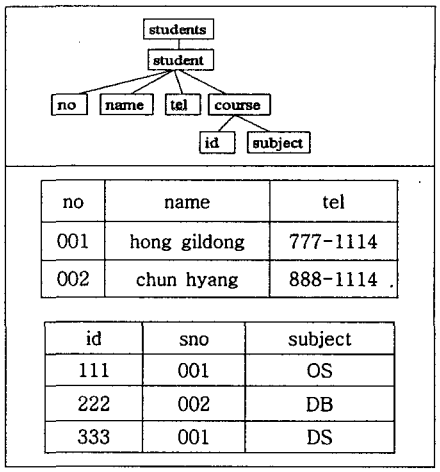
```
<students>
<student no="001">
 <name>hong gildong</name>
 <tel>777-1114</tel>
 <course>
    <subject id="111">OS</subject>
    <subject id="333">DS</subject>
  </course>
<students>
<student no="002">
 <name>chun hyang</name>
 <tel>888-1114</tel>
 <course>
    <subject id="222">DB</subject>
 </course>
</students>
```

3. XML과 개체-관계 매핑

독립적으로 존재하는 개체-관계(entity-relationship)는 데이터 베이스에서 개체와 개체간 또는 개체를 구성하는 속성간의 관계를 이용하여 필요한 정보를 검색하는 방법이다. 개체는 독립적으로 존재하는 기본적인 대상이다.

표 5. DTD에 의한 XML 문서와 테이블 간의 매핑

```
<students>
<student no="001">
 <name>hong gildong</name>
 <tel>777-1114</tel>
 <course>
 <subject id="111">OS</subject>
 <subject id="333">DS</subject>
</course>
</student>
<student no="002">
 <name>chun hyang</name>
 <tel>888-1114</tel>
 <course>
 <subject id="222">DB</subject>
</course>
</student>
</students>
```



no	name	tel
001	hong gildong	777-1114
002	chun hyang	888-1114

id	sno	subject
111	001	OS
222	002	DB
333	001	DS

판매자.고객.매출등을 말한다. 관계는 개체들 사이에 존재하는 연관성을 말한다. 개체-관계 매핑 방법은 XML문서를 트리로 표현하고, 해당 개체를 데이터베이스로 매핑한다. 첫째 단계는 XML의 DTD를 개체 Class로 매핑하고, 두 번째 단계는 Class가 테이블로, 기본 형은 칼럼으로 매핑한다.

❑ 순차 매핑

순차를 이용해서 기술되는 원소들은 클래스의 속성으로 매핑된다.

DTD를 고려한 테이블 매핑의 경우 순차적으로 표현한 요소는 클래스의 속성으로, 테이블의 각 속성에 따라 컬럼으로 매핑한다. 이때 primary 키와 reference 키를 고려해서 테이블 칼럼을 표현해야한다. 만약 속성을 갖거나 자식 원소가 있는 원소는 복합 타입이기 때문에 클래스로 매핑한다. 이처럼 여러개의 개체들로 구성된 경우에 테이블로 매핑될 때는 기본 키와 외래 키의 관계를 이용한다.

표 6. DTD,Class, Table 스키마 간의 순차 매핑 관계

XML	DTD	Class	Table

<S>
<X></X>
<Y></Y>
<Z></Z>
</S>
	<!ELEMENT S(X,Y,Z)>
	<!ELEMENT X(#PCDATA)>
	<!ELEMENT Y(#PCDATA)>
	<!ELEMENT Z(#PCDATA)>
		class S{
		string x;
		string y;
		string z;
		}
			create table S(
			X varchar(10) not null
			Y varchar(10) not null
			Z varchar(10) not null
			)

복합타입은 Class로 매핑하는데, 속성이나 자식 요소가 있는 요소 모두 복합요소이다. 또한 테이블로 매핑하면 primary 키와 foreign 키 관계를 표현할 수 있다.

C요소는 자식 요소를 갖는 복합요소 이므로, 클래스로 매핑하고, 클래스는 다시 테이블로 매핑 한다. 요소 A의 속성 F는 클래스 A의 멤버 변수로 매핑했고, 테이블 A 에 칼럼으로 정의해서 매핑했다. 또한 DTD

내의 PCDATA를 갖는 요소와 속성은 모두 단순 타입으로서, 단순 타입은 모두 기본 타입으로 매핑한다.

표 7. 순차 매핑 관계

XML	DTD	Class	Table

<A F="data">
<B></B>
<C>
<D></D>
<E></E>
</C>
</A>
	<!ELEMENT A(B,C)>
	<!ELEMENT B(#PCDATA)>
	<!ATTLIST A
	F CDATA #REQUIRED>
	<!ELEMENT C(D,E)>
	<!ELEMENT D(#PCDATA)>
	<!ELEMENT E(#PCDATA)>
		class A{
		string b;
		string f;
		C c;}
		
		class C{
		string d;
		string e;}
			class table A(
			B varchar(10),
			F varchar(10),
			C_fk varchar(10))
			
			class table C(
			C_pk varchar(10),
			D varchar(10),
			E varchar(10))

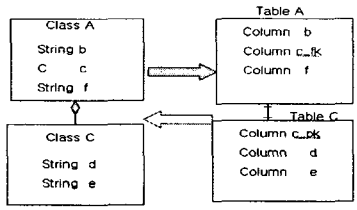


그림 1. 순차 내용 모델 매핑

❑선택 매핑

선택적 경우에도 클래스의 속성으로 매핑되고, 클래스 속성은 다시 테이블의 칼럼, 기본 키, 외래 키로 매핑된다. 단 선택적인 원소는 클래스의 속성과 테이블의 칼럼으로 매핑될 때 null 값을 가질 수 있다.

표 8. 선택 매핑 관계

XML	DTD	Class	Table

<A>
<B></B>
</A>
 	
<A>
<C>
<D></D>
<E></E>
</C>
</A
	<!ELEMENT A(B|C)>
	<!ELEMENT B(#PCDATA)>
	<!ATTLIST A
	F CDATA #IMPLIED>
	<!ELEMENT C(D,E)>
	<!ELEMENT D(#PCDATA)>
	<!ELEMENT E(#PCDATA)>
		class A{
		string b=null;
		string f=null;
		C c=null;}
		
		class C{
		string d;
		string e; }
			class table A(
			B varchar(10) null,
			F varchar(10) null,
			C_fk varchar(10) null)
			
			class table C(
			C_pk varchar(10) null,
			D varchar(10) null,
			E varchar(10) null)

Class에서 null 값을 지정한다고 해서 항상 null 값을 갖는 것은 아니다. 데이터가 지정하지 않을 때만 null값을 사용한다. 또한 #IMPLIED는 속성이 사용할 수도 있고, 사용하지 않을 수도 있음을 정의한다.

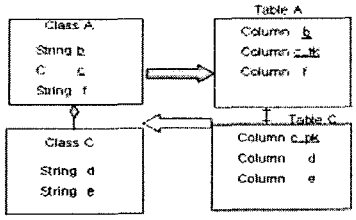


그림2. 선택 내용 모델 매핑

❑ 반복하는 자식 요소 매핑

부모 요소에서 여러 번 나타나는 자식 요소는 클래스에서 여러 값을 가질 수 있는 배열 속성으로 매핑하며, 다시 테이블의 여러 칼럼으로 매핑하거나 다른 외부 테이블로 매핑된다. 이러한 테이블을 property table 이라고 한다. 반복되는 횟수를 미리 알 수 있는 경우에는 테이블에서 여러개의 칼럼으로 표현될 수 있다.

표 9. 한정해서 반복하는 자식 요소 매핑 관계

XML	DTD	Class	Table

<A>
<B></B>
<B></B>
<B></B>
<C></C>
</A>
	<!ELEMENT A(B,B,B,C)>
	<!ELEMENT B(#PCDATA)>
	<!ELEMENT C(#PCDATA)>
		class A{
		string[] b;
		string c;
		}
			create table A(
			B1 varchar(10),
			B2 varchar(10),
			B3 varchar(10),
			C varchar(10))

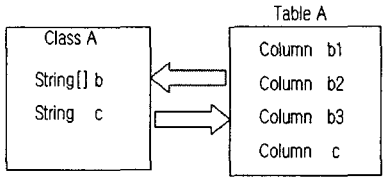


그림 3. 횟수가 지정된 반복되는 자식 원소들

DTD의 내용 모델에서 '+'나 '*' 연산자를 사용하는 경우에는 반복 횟수를 미리 알 수 없다. 이러한 경우에는 속성 테이블을 이용해서 매핑해야 하는데, 이때 테이블 A에는 기본 키가 있고, 속성 테이블은 테이블 A에 대한 외래 키가 있어야 한다.

표 10. 무한정 반복하는 자식 요소 매핑 관계

XML	DTD	Class	Table

<A>
<B></B>
<C></C>
</A>
	<!ELEMENT A(B+,C)>
	<!ELEMENT B(#PCDATA)>
	<!ELEMENT C(#PCDATA)>
		class A{
		string[] b;
		string c;
		}
			create table A(
			A_pk varchar(10),
			C varchar(10))
			
			create table B(
			A_fk varchar(10),
			B varchar(10))

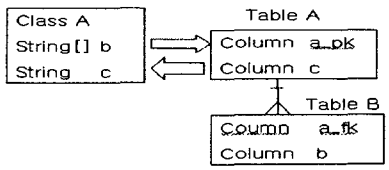


그림 4. 반복횟수를 알수 없는 경우

❑ 옵션 원소 매핑

부모요소에서 옵션 자식 요소를 갖는 경우에 클래스에서 자식요소는 null 값을 가질 수 있는 속성으로

표현되고, 다시 null 값을 가질 수 있는 칼럼으로 매핑한다.

표 11. 선택 요소 매핑 관계

XML	DTD	Class	Table

<A>
<B></B>
<C></C>
</A>
	<!ELEMENT A(B?,C*)>
	<!ELEMENT B(#PCDATA)>
	<!ELEMENT C(#PCDATA)>
		class A{
		string b=null;
		string[] c;
		}
			create table A(
			A_pk varchar(10),
			B varchar(10) null)
			
			create table C(
			A_fk varchar(10),
			C varchar(10))

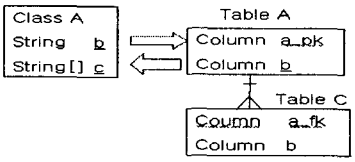


그림 5. 옵션 원소 매핑

❑혼합 내용 매핑

혼합한 내용은 *가 사용된 선택 그룹으로 간주할 수 있다. 따라서 혼합된 내용을 알 수 없고, null값을 가질 수 있는 클래스로 매핑하고, 데이터베이스에서는 테이블과 속성 테이블들로 매핑할 수 있다.

다음은 A요소 내에 혼합한 내용이 0개 이상 발생할 수 있으므로 A요소의 데이터를 저장할 멤버 변수 pd를 정의했으며, 테이블에서도 마찬가지로 pd 멤버 변수에 매칭할 수 있는 pd 테이블을 정의해서 생성했다.

표 12. 혼합 내용 요소 매핑 관계

XML	DTD	Class	Table

<A>
<B></B>
<C></C>
</A>
	<!ELEMENT A
	(#PCDATA|B|C)*>
	<!ELEMENT B(#PCDATA)>
	<!ELEMENT C(#PCDATA)>
		class A{
		string [] pd=null;
		string [] b=null;
		string [] c=null; }
			create table A(
			A_pk varchar(10))
			create table B(
			A_pk varchar(10)
			B varchar(10) null)
			create table C(
			A_fk varchar(10),
			C varchar(10) null)
			create table pd(
			A_pk varchar(10)
			pd varchar(10) null)

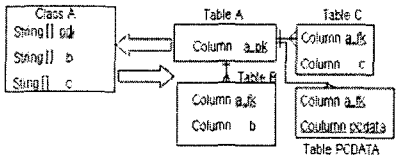


그림 6. 혼합된 매핑

❑속성 매핑

XML 문서의 속성은 스칼라 타입으로 매핑하며, 속성들은 단일 값을 사용하는 것과 여러 개 값을 갖는 것으로 분류할 수 있다. ID는 XML 문서에서 유일하게 값을 갖는 속성으로서, IDREF와 IDREFS는 ID 속성의 레퍼런스 값을 갖으며,ID는 primary 키로 매핑한다. 그리고 IDREF(S)는 foreign 키로 매핑된다.

다음은 A,D테이블이 1:N 의 관계를 보이고 있다.

표 13. 속성 매핑 관계

XML	DTD	Class	Table
<A > <B></B> <C></C> </A>	<!ELEMENT A(B,C)> <!ATTLIST A D IDREFS #IMPLIED> <!ELEMENT B(#PCDATA)> <!ELEMENT C(#PCDATA)>	class A{ string b; string c; string[] d; }	create table A(A_pk varchar(10), B varchar(10) null) C varchar(10)) create table D(A_fk varchar(10), D varchar(10))

entity와 entities는 외부의 기계어 데이터를 XML문서로 포함시키기 위해서 사용될 수 있다.

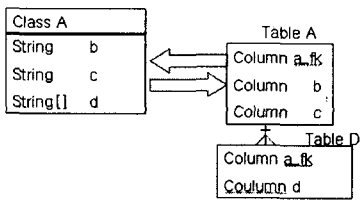


그림 7. 속성 매핑

4. 결 론

문서 중심 문서를 XML전용 데이터베이스로 사용하는 이유가 여러 가지가 있다. 저장의 비효율적, 검색 속도 저하등 이 그 것이다. 이런 문서의 전송시에는 표준화된 형식으로 전송해야 하는데 이를 위해 DTD나 Schema 와 같은 구조를 제시하고, 그에 따른 XML Data를 전송한다. 또한 전송된 XML 자료를 기존 관계형 데이터베이스 자료로 변환하여 SQL문을 이용하여 처리하거나, 기존의 *Relation Database* 자료를 XML로 전송하기 위한 변환 등이 필요하다. 이를 위해 XML과 관계형 데이터베이스 table간의 매핑 관계의 설계가 필요하다. 대분의 XML을 관계형 데이터베이스로 사용되며, XML문서는 객체들의 트리로 표현된다.또한 속성, 내용, 혼합된 요소들은 모두 Class로 표현되고, PCDATA, 공백 ,속성들은 스칼라 속성으로 매핑한다. SQL3나 기존의 기술을 통해 객체-관계 매핑을 통해 Class들은 테이블로 매핑되고, 스칼라 속성들은 칼럼으로 매핑되면서 데이터베이스에 저장된다. 데이터 중심 문서는 매핑이 좋은 반면, 문서 중심 문서는 매핑에 어려움이 많다. 본 논문에서는 위와 같이 두 자료간의 변

환과정을 통해 자료간 매핑하는 Mapping 설계도를 작성하였으며, 이로 인해 XML 자료는 안전성을 위해 데이터베이스 화를 필요로 하고, 자료 전송이나 교환을 위해선 데이타베이스 자료를 XML자료 만드는 것을 필요할 때 이러한 매칭 원리에 의하여 두 자료간의 변환이 이루워진다. 앞으로 자유로운 문서 형식의 콘텐츠를 정규화 하는 연구가 필요하다.

참 고 문 헌

[1] XML.SGML모임,표준 XML,대청미디어,2001
[2] 김영철외2, XML DTD 기반의 구문지향 문서 작성기,컴퓨터 교육학회 논문지,vol.7,NO.4, JULY 2004,PP67
[3] 구덕회외1,멀티미디어 자료의 교육적 활용을 위한 메타데이터 요소 정의 및 XML DTD 설계, 컴퓨터교육학회논문지,vol.7,NO.4,JULY 2004,pp131
[4] 서보원외1,XML입문,대림,2002.11
[5] 홍성용외1,XML원리와 응용,한빛미디어,2003,11
[6] 이하영,XML,가메출판사,2003.12
[7] 송정길,XML 프로그래밍,생능출판사,2003
[8] 박재성,XML실전 프로그래밍,가메출판사,2003.6
[9] 최종명외2,XML + Java,홍릉과학 출판사,2003.3

김 길 준

숭실대학교 공과대학 전자계산 전공 (석.박사)
현, 성결대학교 공과대학 전자상거래학부 부교수