

Definición de Esquemas Relacionales en XML para Propósitos de Integración

Luis Fernando Espino Barrios
Instituto Tecnológico de Costa Rica
luisespino@yahoo.com
2009

Resumen: Como es bien sabido XML es un lenguaje de marcas por excelencia, su sintaxis tan simple permite la manipulación de metadatos de cualquier índole. Este artículo pretende establecer un archivo de intercambio que su estructura coadyuve a la expresión semántica de los esquemas de bases de datos relacionales, con el fin de establecer un medio integrador para hacer diferentes tipos de transformaciones, en especial, entre esquemas conceptuales/lógicos y físicos, y también entre esquemas relacionales y especificaciones formales.

1. Introducción

XML significa Lenguaje de Marcas Extensible, del inglés eXtensible Markup Language, es un conjunto de reglas para definir etiquetas semánticas que dividen un documento en partes e identifica las diferentes partes del documento[1]. XML está derivado de un lenguaje para estructurar documentos[2], conocido como lenguaje estándar generalizado de marcas, del inglés Standard Generalized Markup Language, SGML.

La razón por la cual se seleccionó XML para almacenar el esquema de la base de datos fue porque en sí, dicho esquema se compone de metadatos, lo cual, se acopla muy bien con XML, además de ser un formato portable. Se propone una definición estándar de esquemas relacionales en XML, la finalidad de definir estándares es algo muy común, y resulta necesario para realizar transformaciones entre los diferentes esquemas definidos, por ejemplo, de ANSI/SPARC[3] y también con especificaciones formales, para propósitos de integración.

2. Elementos

Debido a que se utiliza el concepto relacional, los elementos tendrán nombres de la teoría relacional, a continuación se mencionan a detalle.

2.1. Cabecera del XML

Como cualquier archivo XML necesita una cabecera, por defecto será:

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

2.2. Relación

El primer nivel del XML es lo que corresponde a una relación. El nombre del elemento a utilizar es *relation*, es lo que correspondería en un nivel físico a una base de datos, a manera de simplicidad un archivo contendrá solamente un esquema de base de datos, los atributos que debe utilizar son:

- *name*: nombre de la relación.
- *version*: versión de esta definición, por defecto es "1.0".

A continuación se muestra un ejemplo del elemento que representa la relación.

```
<relation name="relation_name" version="relation_version">  
</relation>
```

2.3. Entidades

Dentro de una relación se puede tener entidades y relaciones, para las entidades se utilizará el elemento *entities*:

```
<entities>  
</entities>
```

En *entities* se pueden definir cada entidad con el elemento *entity*, los atributos a utilizar son:

- *id*: identificador de la entidad.
- *name*: nombre de la entidad.
- *x1*: posición horizontal inicial.
- *y1*: posición vertical inicial.
- *x2*: posición horizontal final.
- *y2*: posición vertical final.

```
<entity id="entity_id" name="entity_name" x1="100" y1="100"  
x2="200" y2="200">  
</entity>
```

2.4. Atributos

De la misma manera que se definen las entidades, así mismo, se definen los atributos, el elemento contenedor es *attributes*:

Dentro del elemento *attributes* se define cada atributo con el elemento *attribute*, el cual, tiene los siguientes atributos:

- *id*: identificador del atributo.
- *name*: nombre del atributo.
- *type*: tipo de dato del atributo.
- *key*: define un atributo como llave, se representa con 0 si no es y con 1 si es llave.
- *null*: define un atributo nulo, se representa con 0 si no es nulo y con 1 si puede ser nulo.
- *increment*: si el atributo es de tipo entero puede definirse como auto incremental, se representa con 0 si no es y con 1 si lo es.

```
<attribute id="attribute_id" name="attribute_name" type="varchar"
key="0" null="0" increment="0"/>
```

2.5. Relaciones

Dentro del elemento principal *relation* se definen las diferentes relaciones (con el sentido de relationship) y se define con el elemento *relationships*:

```
<relationships>
</relationships>
```

Y dentro de las *relationships* se definen las diferentes relaciones entre entidades con el elemento *relationship* y los siguientes atributos:

- *id*: identificador de la relación.
- *id_parent*: identificador de la entidad que representa el padre o de quien se extrae información de referencia.
- *id_child*: identificador de la entidad que representa el hijo o quien extraerá información de su padre.
- *legend_parent*: leyenda que tiene la relación del lado del padre, puede ser opcional.
- *legend_child*: leyenda que tiene la relación del lado del hijo, puede ser opcional.
- *min_parent*: cardinalidad mínima del padre.
- *max_parent*: cardinalidad máxima del padre.
- *min_child*: cardinalidad mínima del hijo.
- *max_child*: cardinalidad máxima del hijo.

```
<relationship id="relationship_id" id_parent="entity_parent_id"
id_child="entity_child_id" legend_parent="legend"
legend_child="legend" min_parent="0" max_parent="1" min_child="0"
max_child="*" />
```

3. Esquema XML

El esquema XML es un lenguaje que describe la estructura y restricciones de los documentos XML, que mejora los problemas que tenía DTD o definición de tipo de documentos. En otro sentido representa los metadatos para un documento XML asociado o clases de documentos XML[4].

La definición anterior de los elementos se pueden expresar a través de un esquema XML de la siguiente forma:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wmh="http://www.wmhelp.com/2003/eGenerator"
elementFormDefault="qualified">
  <xs:element name="relation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="entities"/>
        <xs:element ref="relationships"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string"/>
      <xs:attribute name="version" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="entities">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="entity" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="entity">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="attributes"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="x1" type="xs:string" use="required"/>
      <xs:attribute name="y1" type="xs:string" use="required"/>
      <xs:attribute name="x2" type="xs:string" use="required"/>
      <xs:attribute name="y2" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="attributes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="attribute" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

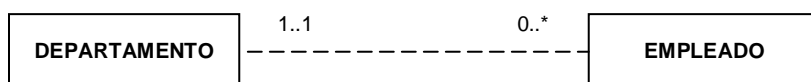
```

    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="attribute">
  <xs:complexType>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="type" type="xs:string" use="required"/>
    <xs:attribute name="key" type="xs:string" use="required"/>
    <xs:attribute name="null" type="xs:string" use="required"/>
    <xs:attribute name="increment" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="relationships">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="relationship"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="relationship">
  <xs:complexType>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="id_parent" type="xs:string" use="required"/>
    <xs:attribute name="id_child" type="xs:string" use="required"/>
    <xs:attribute name="legend_parent" type="xs:string"
use="required"/>
    <xs:attribute name="legend_child" type="xs:string"
use="required"/>
    <xs:attribute name="min_parent" type="xs:string" use="required"/>
    <xs:attribute name="max_parent" type="xs:string" use="required"/>
    <xs:attribute name="min_child" type="xs:string" use="required"/>
    <xs:attribute name="max_child" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

4. Ejemplo

A manera de ejemplo, y para ayudar a la comprensión se toma el siguiente diagrama simple basado en la notación XFER[5]:



El anterior diagrama conceptualiza la relación que existe entre empleados y departamento, es decir, un empleado pertenece a un y solo un departamento, y que un departamento tiene cero o muchos empleados asignados.

La cardinalidad 1..1 del departamento representa que todo empleado pertenece a un departamento, no hay nulos en la llave foránea de la entidad empleado, es decir, que cuando un empleado ingresa a la empresa tendrá asignado siempre un departamento específico.

Mientras que la cardinalidad 0..* del empleado representa que un departamento puede no tener empleados asignados, por ejemplo, si los empleados asignados renuncian y no hay reemplazos aún.

La línea de asociación discontinua indica que las entidades tienen identificadores independientes, es decir, cada entidad cuenta con su propia identificación y sus identificadores no dependen mutuamente, por lo que ambas entidades tienen llaves primarias diferentes.

El archivo de intercambio o la definición en XML del anterior diagrama es:

```
<?xml version="1.0" encoding="utf-8"?>
<relation name="database" version="1.0">
  <entities>
    <entity id="e1" name="empleado" x1="100" y1="100" x2="200"
      y2="200">
      <attributes>
        <attribute id="a1" name="id_empleado" type="integer" key="1"
          null="0" increment="1" />
        <attribute id="a2" name="nombre" type="varchar(32)" key="0"
          null="0" increment="0" />
      </attributes>
    </entity>
    <entity id="e2" name="depto" x1="300" y1="100" x2="400" y2="200">
      <attributes>
        <attribute id="a3" name="id_depto" type="integer" key="1"
          null="0" increment="0" />
        <attribute id="a4" name="nombre" type="varchar(32)" key="0"
          null="0" increment="0" />
      </attributes>
    </entity>
  </entities>
  <relationships>
    <relationship id="r1" id_parent="e2" id_child="e1"
      legend_parent="tiene varios" legend_child="pertenece a un"
      min_parent="1" max_parent="1" min_child="0" max_child="*" />
  </relationships>
</relation>
```

5. Referencias

- [1] E. Harold, *XML 1.1 Bible*, 3rd ed. United States of America: Wiley Publishing, Inc., 2004.
- [2] A. Silberschatz, H. Korth, and S. Sudarshan, *Fundamentos de Bases de Datos*, 5th ed. España: McGraw-Hill, 2006.
- [3] C. Bachman, "Summary of current work ANSI/X3/SPARC/study group: database systems," in *ACM SIGMOD Record*, vol. 6, United States of America, 1974, pp. 16-39.
- [4] C. Campbell, A. Eisenberg, and J. Melton, "XML Schema," in *SIGMOD Record*, Vol. 32, No. 2, 2003, pp. 86-101.
- [5] L. Espino, "Notación XFER: Transferencia de Semántica, Restricciones y Normalización a los diagramas Entidad-Relación," Instituto Tecnológico de Costa Rica, 2009.