

# A Research on Improved Table-Based Model-Driven Mapping Strategies

Yu Han<sup>1</sup>, Jinlai Lv<sup>2</sup>

<sup>1</sup>Shanxi Province Expressway Toll Collection Center, Taiyuan Shanxi

<sup>2</sup>College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan Shanxi

Email: 381415955@qq.com

Received: Apr. 1<sup>st</sup>, 2016; accepted: Apr. 19<sup>th</sup>, 2016; published: Apr. 22<sup>nd</sup>, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

To solve the problem of mapping database to XML during data exchange process, the currently widely used model-driven mapping method has been analyzed and studied. The table-based model-driven mapping method has been improved, and a combination of this method with XML Schema is proposed. And mapping rules have been developed. This improved mapping method can preserve the structural characteristics of relational database tables and constraints better than others. On the other hand, to solve the data inserting sequence problem caused by reference between data in different tables during mapping process, a table data inserting priority calculating algorithm has been proposed.

## Keywords

XML Schema, Model-Driven, Data Exchange, Table-Based Mapping

---

# 一种改进的基于表的模型驱动映射策略研究

韩 昱<sup>1</sup>, 吕进来<sup>2</sup>

<sup>1</sup>山西省高速公路收费管理结算中心, 山西 太原

<sup>2</sup>太原理工大学计算机科学与技术学院, 山西 太原

Email: 381415955@qq.com

收稿日期: 2016年4月1日; 录用日期: 2016年4月19日; 发布日期: 2016年4月22日

## 摘要

为解决数据交换过程中, 数据从数据库到XML的映射问题, 对目前广泛使用的模型驱动映射方法进行了分析研究。对基于表的模型驱动映射方法进行改进, 提出结合XML Schema的基于表的模型驱动映射方法, 并制定了映射规则。这种改进的映射方法很好地保留了关系数据库表的结构特性和约束条件。还对映射过程中因参照关系而需有序插入数据的现象, 提出了表插入优先计算方法。

## 关键词

XML Schema, 模型驱动, 数据交换, 基于表的映射

## 1. 引言

多年来, 在信息化建设的进程中, 由于人们对信息共享认识上的不足, 造成各个应用系统之间的数据信息难以共享, 形成彼此隔离的“信息孤岛”, 极大地阻碍了信息化进程。如何共享已有的数据信息, 已成为当今数据库研究领域的热点问题[1]。

为解决这一问题, 许多研究者先后提出了多种不同的用于解决数据共享问题的数据集成技术。总的来讲, 异构数据库集成所采用的技术有三种[2]: 数据库迁移和转换技术、多数据库系统技术及使用中间件技术。这三种方法各有优缺点, 一些研究者对第一种技术进行了研究, 并开发了相应的程序, 但其适用范围受到数据库系统版本及操作系统版本等因素的限制。到目前为止, 还没有商品化的多数据库系统, 即使在 CIMS(Computer Integrated Manufacturing Systems, 计算机集成制造系统)环境中实施也有一定的难度; 而大多数研究者倾向于使用中间件技术来完成数据的集成。

为了完成数据的交换, 需要采用统一的标准和规范, 生成一个可供异构数据库之间共享数据的数据源。如今, XML(eXtensible Markup Language, 可扩展标记语言)逐渐成为 Internet 上数据交换的标准模式[3], 同时, XML 的诸多优点使得采用 XML 做为中间数据模型来完成数据的跨平台交换[4]得到了广泛的研究。采用这种模型的关键点在于数据库的关系模式向 XML 模式转换的映射问题。关系模式映射为 XML 模式的主要方法有模板驱动映射和模型驱动映射, 国内外的许多学者针对这两种方法做了研究和实现。哈尔滨工程大学的曹虎[5]提出了基于 XML 中间件的异构数据库集成的中间件技术, 以此来完成异构数据库数据的整合集成。山东大学的蔡娜娜[6]提出了以 XML 文档为异构数据交换中介, 实现基于 XML 的数据交换平台。这些方法大都是基于 DTD 技术来保留、完善映射过程中的信息匹配问题, 但由于 DTD 本身的局限性, 它们在映射过程中都没有很好的解决语义丢失等问题。

本文通过对目前存在的关系模式向 XML 模式映射方法的分析研究, 提出了采用 XML 做为中间模型, 使用基于改进的表的模型驱动映射方法来完成模式的映射。主要特点是对基于表的模型驱动映射方法进行了改善, 用 XML Schema 技术取代 DTD 技术, 很好的解决了映射过程中的语义丢失等问题; 其次, 考虑到实际的应用存在着表插入的优先顺序问题, 为方便数据的操作, 提出了表插入优先级计算方法, 以简化 XML 数据向关系数据库插入的工作, 提高数据交换的效率。

## 2. 映射方法的选择

根据以往的一些学者们的研究, 异构数据库的数据集成涉及的主要问题有: 异构性(包括系统异构和数据异构)、完整性(包括数据完整性和约束完整性)、透明性(包括平台透明性、系统透明性、数据源透明

性)、针对不同数据源访问时的权限控制等[7] [8]。正如上述引言中所说的那样, 采用 XML 作为中间数据模型加上本文下面介绍的映射技术和方法很好的解决了这些问题。

常用的建立数据库与 XML 文档映射关系的方法主要有两种: 模板驱动和模型驱动。

## 2.1. 模板驱动

模板驱动映射方法的特点是不预先定义文档结构和数据库结构之间的映射关系, 而是将命令语句(如 select 查询语句)嵌入到 XML 模板中的指定位置, 然后让数据传输的中间件执行该模板, 获得执行结果并替换模板中对应的命令语句部分, 从而得到所要的 XML 文档。

## 2.2. 模型驱动

模型是对事物的一种抽象[9], 模型驱动就是对系统相关数据进行的抽象描述和规范的说明[10]。模型驱动映射方法的原理是在整个映射过程中以定义的模型为中心, 把模型语义定义为映射约束条件, 来驱动完成数据的映射[9]。采用基于 XML 的模型驱动映射的显著优点是简单易用, 这是因为它在映射转换前先定义数据库模式与 XML 文档模式之间的映射关系, 然后以此映射关系来完成数据库到 XML 文档之间的映射转换。通常这种模型驱动映射通过提前定义好的语义完成了需在映射过程中做的很多工作, 极大简化了异构数据库的整合工作。

模型驱动的映射有基于表的映射(Table Based Mapping)和基于对象关系型的映射(Object Relational Mapping) [3] [6]两种实现方法。这两种方法都是为数据建模, 非常适合以数据为中心的映射过程, 因而比较广泛地运用在实现数据库与 XML 文档之间的数据转换, 特别是在关系数据库与 XML 之间的数据转换中。更重要的一点是这两种方法都能实现数据库与 XML 文档的双向转换。现在, 这两种映射方法已经逐渐成为映射规范, 通过它们来定义 XML 视图, XML 查询语言(比如, XQuery 可以直接通过视图对非 XML 数据库进行查询)。

对象关系映射功能比较强大, 能够表示复杂的数据库模式, 但映射规则复杂, 转换效率较低。

基于表的映射则比较简单明了, 但转换出来的 XML 文档格式比较单一, 不能表示过于复杂的数据库模式, 主要问题表现在:

1. 数据转换过程中的类型保留问题;
2. 数据格式的验证问题;
3. 表中约束条件信息的保留问题;
4. 表间相互参照约束信息的保留问题;
5. 从 XML 中提取数据向数据库中插入时的先后顺序问题。

为了解决基于表的映射中存在的问题, 又有研究者提出了把基于表的映射和 DTD (Document Type Definition, 文档类型定义)相结合[6], 但由于 DTD 的语法结构独特[11] [12], 并且在约束和类型表达方面具有一定的缺陷, 因而没有完全解决上述问题。

XML 数据结构的另一种表示方式 XML Schema 具有丰富而强大的数据类型、基于命名空间的有效性验证以及可扩展、可伸缩等优点, 加上 XML Schema 中定义的空值、唯一及键的概念与关系数据库中的概念是相符的, 并且用 XML Schema 语言描述的 XML Schema 文档仍然是标准的 XML 文档, 可以用解析 XML 文档的方法来解析 XML Schema, 所以利用基于表的映射方法的简单、易用、高效的优点作为关系数据库向 XML 的映射方法, 同时采用 XML Schema 丰富强大的内建数据类型等优点来克服基于表的映射方法在表述关系数据库物理结构等方面的不足, 将基于表的映射方法与 XML Schema 结合, 由此产生的改进的基于表的映射方法将能很好的克服传统的基于表的映射方法存在的问题。

### 3. 映射模型的提出

完成了异构数据库中的数据提取之后, 为了满足用户的需求, 要对其实现模式转换, 即从关系模式映射成 XML Schema 或者从 XML Schema 映射成关系模式, 才能使得异构数据在互联网上实现彼此通信和共享[13]。

将关系模式映射为 XML Schema 模式其本质就是用 Schema 来记录关系数据库中的表参照关系、字段类型信息以及表中或者表间的约束关系等信息。

以往的基于表的映射技术采用的是 DTD 辅助完成数据结构映射, 为了能够在 DTD 中对类型进行表述, 一种解决办法就是使用参数实体, 在 DTD 内部的参数实体中对 #PCDATA 的类型进行声明。例如:

```
<!ENTITY %float "#PCDATA">参数实体声明
<!ENTITY %integer "#PCDATA">参数实体声明
<!ELEMENT money(%integer;)>实体引用表明元素 money 类型为 integer
<!ELEMENT mount(%float;)>实体引用表明元素 mount 类型为 float
```

而 XMLSchema 模式语言却拥有很多内建的数据类型, 针对同样的问题, XML Schema 的声明如下:

```
<xs:element name="money" type="xs:integer"/>
<xs:element name="mount" type="xs:float"/>
```

从以上的比较可以看出, 采用 DTD 对 XML 进行约束存在着语法复杂等缺陷。本文借鉴以往的映射技术, 通过对 Schema 的学习和研究, 提出了从关系数据库(RDB)到 XML Schema 的映射模型 RTXS(RDB to XML Schema 模型, 简称 RTXS)。

首先, 将数据库中的表  $T = (T_1, T_2, \dots, T_n)$  ( $n$  为数据库中表的个数) 按照有相互参照关系的分为一个组, 得到组  $G_1, \dots, G_m$  ( $1 \leq m \leq n$ ), 其中  $G_i = (T_{i1}, T_{i2}, \dots, T_{is})$  ( $1 \leq i \leq m, 1 \leq i_1, \dots, i_s \leq n$ )。并将每一组中的表映射到一个 Schema 文件中, 确保有相互参照关系的表在同一 Schema 文件中, 以便对于 XML 文件进行参照方面的检验和 XML 数据向关系数据库数据转换时能够集中处理, 提高效率。

然后, 处理每一组中的表, 将每一表的字段都映射到 Schema 中。具体过程如下:

第一步, 将 Schema 中的根节点定义为 <database>;

第二步, 根节点 <database> 下定义为各个表 <table\_name> 节点, 每个表节点的名称定义时取表名称为节点名称, 这些节点相互不影响; Schema 中定义节点的语法为[14]:

```
<xsd:element name="Element Name"/>
```

经过上述两个步骤, 基本确定了 Schema 规范下的 XML 文档中的节点树, 如图 1 所示:

第三步, 将具体字段类型进行匹配映射, Schema 中定义元素类型的语法如下[15]:

```
<xsd:element name="Element Name" type="xsd:DataType"/>
```

第四步, 各种约束等的映射。这一步是关键映射, 也是 Schema 区别于 DTD 最主要的地方。具体如下[16]-[18]:

①定义主键(Primary Key):

```
<xsd:key name="PK_1"> <!--标识为主键, 且标识名为 PK_1-->
  <xsd:selector xpath="major"/> <!--表示为对应的主键在 major 表中-->
  <xsd:field xpath="@ID"/> <!--表示主键为 major 中的 ID 字段-->
</xsd:key>
```

## ②主键参照(键引用) (Key Reference):

```
<xsd:keyrefname="idref" refer="PK_1"> <!-- 外键引用, 引用名字定义为 PK_1 的主键-->
  <xsd:selector xpath="student"/> <!-- 外键在 student 表中-->
  <xsd:field xpath="@majorID"/> <!-- 表 student 表中的 majorID 参照 PK_1 中的-->
    <!-- 主键结合上面的例子, 就是参照 major 表中 ID 主键-->
</xsd:keyref>
```

## ③默认值(Default Value)、固定值(Fixed Value)约束定义:

```
<xsd:element name="color" type="xs:string" default="red"/> <!-- 默认值约束定义-->
<xsd:element name="color" type="xs:string" fixed="red"/> <!-- 固定值约束定义-->
```

## ④对于数据库中用户自定义的 check 约束, 当约束数据的大小范围时, 映射到 Schema 中如下:

```
<xsd:element name="age">
  <xsd:simpleType>
    <xsd:restriction base="xs:integer">
      <xsd:minInclusive value="0"/> <!-- 定义 age 的最小值为 0-->
      <xsd:maxInclusive value="120"/> <!-- 最大值为 120-->
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

⑤当约束为字符串的长度时, 可以通过在 Schema 中使用 length、maxLength 以及 minLength 限定来进行映射约束:

```
<xsd:element name="password">
  <xsd:simpleType>
    <xsd:restriction base="xs:string">
      <xsd:length value="8"/> <!-- 定义 password 的长度必须是 8 位-->
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

至此, 就完成了关系模式到 XML Schema 模式的映射。

完成了模式映射之后的 Schema 文件:

1. 可以对 XML 进行验证;
2. 当提取 XML 数据向数据库插入数据的时候, 对于数据类型的转换映射起到标准参照的作用;
3. 除了对 XML 的规范, 还使得 XML 中数据的意义表述出来, 便于 XML 数据的网络发布应用和信息共享。

## 4. XML 文件的生成

完成了模式的映射之后, 接下来的工作是将数据库中的数据根据基于表的映射方法映射生成 XML 文件。

在表映射到 XML 中时根节点是<database>, 根节点<database>下是<table>节点, <table>节点下才是具体的表格节点。由于一个 XML Schema 中映射了一组表格, 各表格相互之间会有参照关系, 这样映射生成的 XML 文件在将数据提取插入数据库的时候就会有先后顺序, 而在 Schema 文件和 XML 文件中都没有显式的描述插入的先后顺序信息。针对这一问题本文提出了表插入优先级算法 TIPC (Table Insert

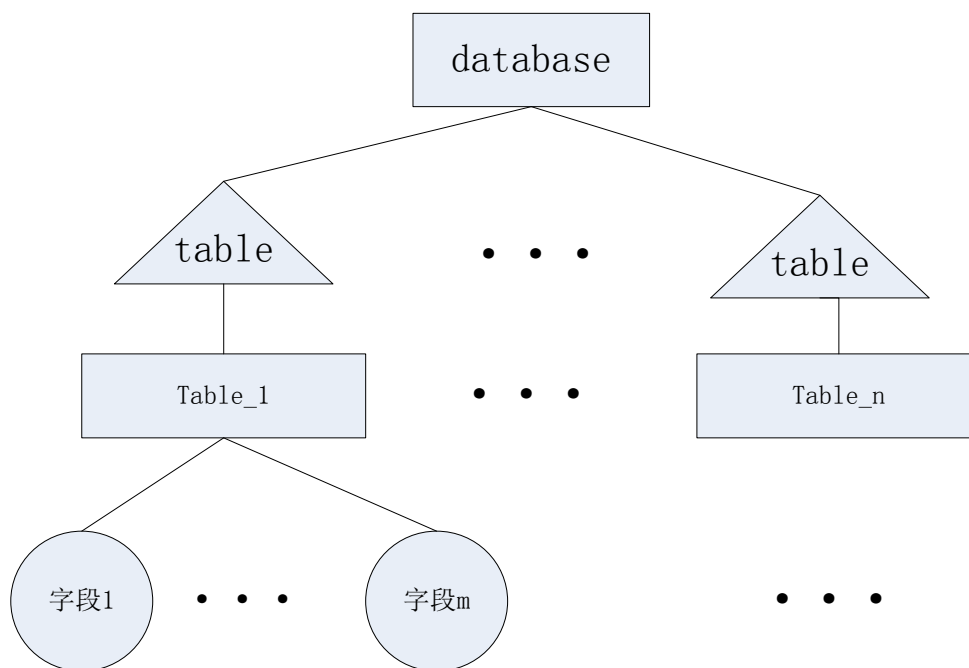


Figure 1. XML document tree structure diagram  
图 1. XML 文档树形结构图

Priority Compute, 简称 TIPC)。算法描述如下:

①初始情况下, 将各个表的优先级都置为 1, 设置各个表的访问标志为 FALSE, 并定义  $P(A)$  表示 A 表的优先级;

②遍历所有的表, 设当前访问的表为 A 表, 若 A 表无参照表, 继续执行步骤②; 若 A 表参照了 B 表, 顺序执行以下步骤(有几个参照表执行下列步骤几次):

i 比较 A、B 表的优先级,

若  $P(A) < P(B)$ , 执行步骤 ii;

若  $P(A) \geq P(B)$ , 将 B 的优先级置为 A 表优先级加 1, 即  $P(B) = P(A) + 1$ ;

若 B 的访问标志为 TRUE(表示 B 已经访问过), 访问 B 表, 执行步骤②;

若 B 的访问标志为 FALSE(表示 B 还没有访问过), 执行步骤 ii;

ii 设 A 表的下一个参照表为 B, 执行步骤 i; 若 A 表无其它参照表, 置 A 表的访问标志位 TRUE, 执行步骤③;

③判断是否所有表都已遍历, 若还有表没有遍历, 继续执行步骤②, 否则, 执行步骤④;

④结束。

整个算法可用如图 2 所示的框图表述。

TIPC 算法就是在具有参照关系的表之间建立一个被参照的表优先级大于参照表的优先级, 然后按照优先级从大到小的顺序提取各表中的数据生成 XML 文件, 以保证当把 XML 文件中的数据插入目的数据库的表中时, 不会发生参照约束错误。

## 5. 实验及结果

选择 Oracle 11g 为源数据库, MS SQL Server 2012 为目标数据库, 采用 JDBC 技术访问数据库, 使用 DOM 解析器来完成对 XML 文件的解析与生成。通过中间件技术及相应的应用程序完成对数据库的访

问操作以及对 XML 文件的操作。

实验所采用的数据库结构如图 3 所示。

数据库模式向 XML 模式映射的过程如下：

首先，从数据库中提取相应的信息，按照本文提出的映射规则生成描述表结构信息的 Schema 文件，如图 4 所示。

第二步，根据数据库中的表格参照关系利用 TIPC 算法计算表数据的插入先后顺序。

利用 TIPC 算法可得各表的插入优先级为  $P(vip) = 3$ ,  $P(incomeshare) = 3$ ,  $P(role) = 3$ ,  $P(rightsmanage) = 3$ ,  $P(except\_c\_o) = 3$ ,  $P(exceptstate) = 3$ ,  $P(goodssort) = 3$ ,  $P(goodsl\_stat) = 3$ ,  $P(truck\_stat) = 3$ ,  $P(range) = 3$ ,  $P(gl\_distance) = 3$ ,  $P(user1) = 2$ ,  $P(relationship\_5) = 2$ ,  $P(truckinfo) = 2$ ,  $P(con\_ordsheet) = 2$ ,  $P(station) = 2$ ,  $P(relationship\_7) = 1$ ,  $P(ordersheet) = 1$ ,  $P(distance) = 1$ 。

第三步，按照基于表的映射规则以及上述计算的表提取优先顺序将关系数据库数据提取出来生成 XML 数据文件，XML 数据文件(部分)如图 5 所示。

数据交换过程中，最主要问题包括数据类型、数据本身映射的准确性，以及映射效率。由于 XML Schema 所提供的数据类型远多于 DTD 所能提供的数据类型，所以在数据类型、数据本身映射的准确性上，本文所提出的 RTXS 模型的优点是明显的。为了比较两者的映射效率，针对 DTD 进行了同样的试验，结果发现利用 RTXS 模型完成本例表结构以及所用全部数据库数据，从源数据库到目标数据库迁移的时

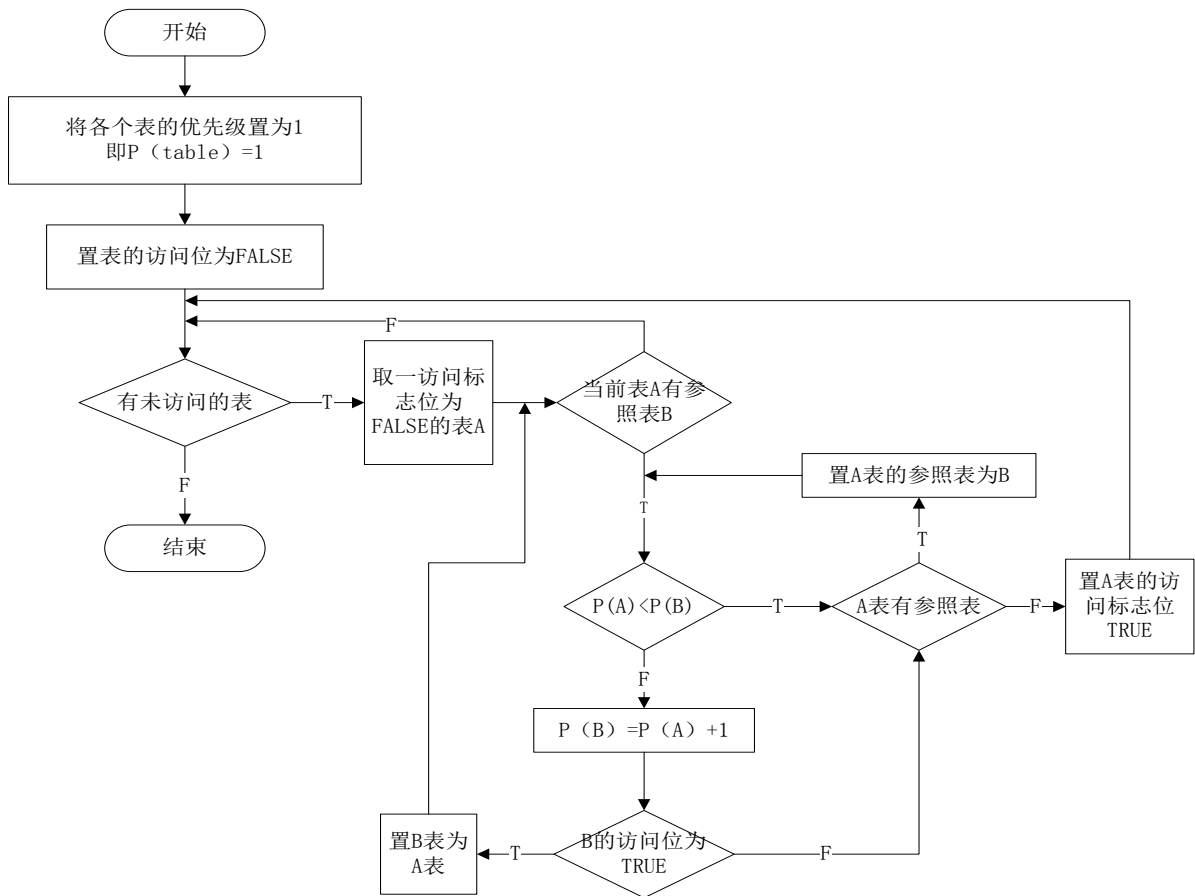


Figure 2. Block diagram of TIPC algorithm

图 2. TIPC 算法框图





```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.ws.org/2001/XMLSchema-instance">
  <xsd:element name="database">
    <xsd:complexType>
      <!--discription of Employee table-->
      <xsd:element name="user1">
        <xsd:complexType>
          <xsd:element name="row">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="user_id" type="xs:int">
                  <xsd:simpleType>
                    <xsd:restriction>
                      <xsd:lenth value=100001/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:complexType>
        </xsd:element>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

Figure 4. Describes the table structure information of the Schema file (part)  
图 4. 描述表结构信息的 Schema 文件(部分)

```
<?xml Version="1.0" encoding="gb2312"?>
<database>
  <user1>
    <row>
      <user_id>100001</user_id>
      <vip_id>20160102</vip_id>
      <user_name>zhang san</user_name>
      <gender>zea</gender>
      <user_pwd>zs690101</user_pwd>
      <email>zhangsan@163.com</email>
      <address>青年路3号</address>
      <postnum>030005</postnum>
      <mobile_tel>13903510988</mobile_tel>
      <immobile_tel>03517890421</immobile>
      <uvip_id>01</uvip_id>
      <mark> </mark>
    </row>
  </user1>
</database>
```

Figure 5. Describes the XML file for the database data (part)  
图 5. 描述数据库数据的 XML 文件(部分)

验研究证明这一方法很好地实现了数据从数据库到 XML 的映射, 与以往的方法相比较, 数据安全、交换效率都得到了提高, 为异构关系数据库系统实现数据交换提供了新的解决方案。

### 基金项目

山西省 2013 年科技攻关项目(20130321007-02)。

### 参考文献 (References)

- [1] 陈爱华. 基于 XML 和 Webservice 的异构数据库数据交换研究[J]. 软件工程师, 2011(5): 56-58.
- [2] 周长宝. 基于 XML 的制造业应用软件异构数据库集成技术研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨理工大学, 2012.
- [3] Xing, G.M., Xia, Z.H. and Ernest, A. (2007) Building Automatic Mapping between XML Documents Using Approximate Tree Matching. *Proceedings of the ACM Symposium on Applied Computing*, 525-526.
- [4] Della Penna, G., Di Marco, A., Intrigila, B., Melatti, I. and Pierantonio, A. (2014) Interoperability Mapping from XML Schemas to ER Diagrams. *Data and Knowledge Engineering*, **59**, 166-188.

<http://dx.doi.org/10.1016/j.datak.2005.08.002>

- [5] 曹虎. 基于 XML 的异构数据库集成中间件技术研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2009.
- [6] 綦娜娜. 基于 XML 的数据库转换技术研究[D]: [硕士学位论文]. 济南: 山东大学, 2011.
- [7] 吕晓妮. 基于 XML 的异构数据库集成技术与中间件开发[D]: [硕士学位论文]. 青岛: 山东科技大学, 2011.
- [8] 崔静, 翟玉梅. 基于 XML 技术的异构数据库集成研究[J]. 商丘职业技术学院学报, 2011, 10(2): 32-34.
- [9] 张亚军, 华一新, 赵军喜. 基于模型驱动的态势显示研究[J]. 测绘科学技术学报, 2010, 27(4): 289-293.
- [10] 姚全珠, 白敏, 黄蔚. 基于模型映射方法实现与关系数据的相互转换[J]. 软件工程师, 2014(6): 289-293.
- [11] 孙更新, 等. XML 编程与应用教程[M]. 第 2 版. 北京: 清华大学出版社, 2014.
- [12] 王珊, 萨师煊. 数据库系统概论[M]. 第 5 版. 北京: 高等教育出版, 2014.
- [13] 侯莹. 基于 XML 异构数据交换的研究与应用[D]: [硕士学位论文]. 大连: 大连理工大学, 2009.
- [14] Si, G.N., Zhou, Z.J., Li, N., Xu, J. and Yang, J.F. (2011) A Scheme for Evaluating XML Engine on RDBMS. *International Journal of Modern Education and Computer Science (IJMECS)*, **3**, 51-60.  
<http://dx.doi.org/10.5815/ijmeecs.2011.02.08>
- [15] Dietrich, S.W. and Chaudhari, M. (2013) The LINQ between XML and Databases: A Gentle Introduction. *Journal of Computing Sciences in Colleges*, 158-164.
- [16] Haw, S.-C., Lee, C.-S. and Mustapha, N. (2010) Bridging XML and Relational Databases: Mapping Choices and Performance Evaluation. *IETE Technical Review*, **27**, 308-317. <http://dx.doi.org/10.4103/0256-4602.62593>
- [17] Wang, F.S., Zaniolo, C. and Zhou, X. (2014) ArchIS: An XML-Based Approach to Transaction-Time Temporal Database Systems. *The VLDB Journal—The International Journal on Very Large Data Bases*, **17**, 1445-1463.
- [18] Wu, Q.L., Sun, Y.X., Yang, X.X., Liang, Y. and Geng, X. (2011) Study on XML-Based Heterogeneous Agriculture Database Sharing Platform. *IFIP Advances in Information and Communication Technology*, **346**, 1-7.  
[http://dx.doi.org/10.1007/978-3-642-18354-6\\_1](http://dx.doi.org/10.1007/978-3-642-18354-6_1)