

关系数据库模式信息的提取

许卓明 苏文萍

(河海大学计算机及信息工程学院 江苏 南京 210098)

摘要 鉴于关系数据库是当前 Web 的主要信息源,关系数据库模式信息的提取是实现当前 Web 向语义 Web 迁移的基础技术之一,介绍关系数据库模式信息提取原型工具 RDBSExtractor,分析与总结关系数据库的模式信息,给出基于 JDBC API 的 RDBSExtractor 设计思想和实现技术.案例研究表明,基于 JDBC API 的模式提取方法是可行的,RDBSExtractor 是有效的.

关键词 关系数据库;模式提取;语义 Web;CASE 工具

中图分类号 TP311.13 **文献标识码** A **文章编号** 1000-1980(2005)02-0202-05

数据库是各种数据密集型应用的主要信息源.近期一项研究^[1]表明,当前万维网上“deep Web”中从数据库产生的动态 Web 页数量是“surface Web”上静态 Web 页数量的 550 倍,而 Web 上联机数据库的类型主要是关系数据库.语义 Web^[2]是当前 Web 的一种扩充,Web 信息将被赋予机器可处理的语义,使自动工具(和人)能更有效地发现、处理、集成和复用 Web 内容和服务.通过定义公共的领域理论和具有明确语义的可共享词汇,Web 本体^[3]在创建机器可处理内容时发挥着重要作用.

数据库的结构和语义信息包含在数据库模式中,模式信息的提取是数据密集型 Web 站点向语义 Web 迁移^[4]、对“deep Web”进行语义标注^[5]、开发语义 Web 入口^①或构建新一代信息管理基础结构^[6]等前沿研究中的关键公共技术.作为当前研究课题(包括关系数据库模式向 OWL^②本体映射、动态 Web 页语义标注、语义 Web 入口设计与开发等)的一部分,本文着重介绍关系数据库模式信息的提取技术.基于 Java 2 平台,开发了一个关系数据库模式信息提取器原型 RDBSExtractor.案例研究表明:借助 JDBC API 实现关系数据库模式信息的提取,在技术上是可行的,RDBSExtractor 的设计是合理的,实现是有效的.

1 关系数据库的模式信息

一个关系数据库的设计主要包括关系模式(即基表结构)设计和完整性约束申明两部分^[7].基表结构定义了关系(表)的结构、属性(列)及其数据类型与长度等;完整性约束定义了语义施加在数据上的约束,包括关系层的全局约束、元组层的表约束以及属性层的列约束.这两部分信息构成了数据库模式的主要内容,并作为元数据存储于数据库的数据字典中.

当前,大多商业 RDBMS 产品(如 Oracle 9i,Microsoft SQL Server 2000,IBM DB2 UDB V8.1 等)均声称支持 SQL:1999 标准.对于完整性约束,SQL:1999 标准^[8]规定了一系列的构造子(如 PRIMARY KEY, CHECK, ASSERTION 等),但各商业 RDBMS 对 SQL:1999 的支持程度不一,且实现语法也不尽相同^[7,8].表 1 列出了 SQL:1999 标准和主流 RDBMS 对完整性约束的支持情况.其中“ Yes ”表示支持并完全遵循 SQL:1999 标准;“(Yes)”表示支持但语法与 SQL:1999 标准不同;“ No ”表示不支持.

收稿日期 2004-11-08

基金项目 江苏省基础研究计划(自然科学基金)重点资助项目(BK2003001)

作者简介 许卓明(1965—)男,江苏苏州人,教授,主要从事数据库、信息检索及语义 Web 技术的研究.

① STOLLBERG M, LAUSEN H, LARA R, et al. Towards semantic Web portals. In: BUSSLER C, DECKER S, SCHWABE D, et al, eds. Proc of the WWW2004 Workshop on Application Design, Development and Implementation Issues in the Semantic Web, CEUR Workshop (Online) Proceedings, 2004. Online: <http://CEUR-WS.org/Vol-105/>.

② HARMELEN F V, MCGUINNESS D L. OWL Web ontology language overview. W3C Recommendation. Online: <http://www.w3.org/TR/owl-features/> 2004-02-10/2004-11-05.

表 1 SQL :1999 标准和主流 RDBMS 对完整性约束的支持

Table 1 Integrity-constraints support in SQL :1999 standard and mainstream RDBMSs			SQL :1999 标准	Oracle 9i	IBM DB2 UDB V8.1	MS SQL Server 2000
完整性约束						
列约束(属性层)	NOT NULL(非空)		Yes	Yes	Yes	Yes
	DEFAULT(缺省值)		Yes	Yes	Yes	Yes
	UNIQUE(唯一值)		Yes	Yes	Yes	Yes
表约束(元组层)	CHECK(值检查)	基于属性	Yes	Yes	Yes	Yes
		基于元组	Yes	Yes	Yes	Yes
	PRIMARY KEY(主键)		Yes	Yes	Yes	Yes
	FOREIGN KEY (外键)	NO ACTION	Yes	(Yes)	Yes	(Yes)
		RESTRICT	Yes	No	Yes	No
		CASCADE	Yes	Yes	Yes	Yes
		SET NULL	Yes	Yes	Yes	No
		SET DEFAULT	Yes	No	No	No
		NO ACTION	Yes	(Yes)	Yes	(Yes)
		RESTRICT	Yes	No	Yes	No
		CASCADE	Yes	No	No	Yes
		SET NULL	Yes	No	No	No
		SET DEFAULT	Yes	No	No	No
	ASSERTION(断言)		Yes	No	No	No

2 模式提取的设计与实现

2.1 设计思想

为了实现关系数据库模式的提取 ,笔者设计、开发了一个关系数据库模式提取器 RDBSExtractor. 逻辑上 , RDBSExtractor 分为模式信息的提取和可视化输出两部分 ,彼此间以底层数据结构为联系 .“ 提取 ”部分把模式信息提取出来后存储于(内存中)一定的数据结构中 ;“ 可视化 ”部分则从数据结构中获取模式信息后 ,进行屏幕格式化输出 ,并与用户交互 .RDBSExtractor 主要设计思想如下 :

a. 模式信息的提取策略 .一般而言 ,数据库模式信息既可以直接从数据库的数据字典中提取 ,也可以通过分析 SQL 数据定义语言(DDL)语句来获得 .早期的数据库逆向工程工具往往采用 DDL 语句分析法实现数据库模式的提取 ,但是 ,DDL 语句分析法在此有以下缺陷 (a)需要进行文本处理和 DDL 语句语法分析 ,处理较为繁琐 (b)可能需要进行多个 DDL 脚本文件的合并处理 ,当有 ALTER TABLE 语句时 ,合并处理需要进一步的语义分析 (c)各 RDBMS 的 DDL 语法不一致 ,提取处理逻辑需针对各个特定 RDBMS 分别设计 .相反 ,数据字典中保存的模式信息反映的是数据库当前“ 最终 ”模式状态 ,只要利用能屏蔽各 RDBMS 异构性的中间层 API(如 ODBC API ,JDBC API)来设计提取处理逻辑 ,就可以实现直接、通用的模式提取 ,因此 ,RDBSExtractor 设计成通过读取数据字典来获取数据库模式信息 .

b. 模式信息输出组织与可视化策略 .将提取出的模式信息组织成数据库层(包括数据库名、DBMS 名称、版本、数据库的总表数等元数据)、关系层(包括关系的诸属性名以及各种表约束等)和属性层(包括属性 ID、数据类型、长度/小数位等元数据以及各种列约束等)3 个层次 .可视化时 ,把数据库名—关系名—属性名组成层次结构(树)来显示 ,层次树和数据库层模式信息作为 RDBSExtractor 的初始输出 ;当用户响应(即鼠标点击层次树中的“ 关系 ”或“ 属性 ”结点)时 ,关系层或属性层的模式信息采用表格的形式显示输出 .

2.2 实现技术

2.2.1 数据结构

RDBSExtractor 基于 Java 2 Platform ,Standard Edition ,version 1.4 (J2SE)实现 ,如图 1 所示 .按照上述模式信息的分层设计 ,分别以 DBMetaData ,RelationMetaData ,AttrMetaData 3 个 Java 类来存储数据库层、关系层和属性层的模式信息 ,组成模式信息的层次结构 .KeyMetaData ,FKMetaData ,CheckMetaData 3 个类分别存储键(分为主键和候补键)、外键、CHECK 约束的信息 .

<pre>class DBMetaData { String dbName //数据库名 String dbms //DBMS 类型 String dbmsVer //DBMS 版本 int numOfRelations //关系总个数 RelationMetaData[] allRelations //所有关系 } class RelationMetaData { String relationName //关系名 AttrMetaData[] attrInfo //所有属性 KeyMetaData[] keyInfo //所有键 FKMetaData[] fkeyInfo //所有外键 CheckMetaData[] checkTupleLevel ; //基于元组的 CHECK 约束 } class FKMetaData { String fkeyName //外键名 String[] attrNames //外键属性(组) String refedTable //外键引用的表名 String[] refedAttributes //引用的属性(组) String updateRule //更新规则 String deleteRule //删除规则 }</pre>	<pre>class AttrMetaData { int attrID //属性 ID String attrName //属性名 String attrType //数据类型 String attrMaxLen //最大长度 String attrDecimal //小数位 String attrDef //缺省值 boolean attrNotNull //非空值约束 CheckMetaData[] checkAttrLevel //基于属性的 CHECK 约束 } class KeyMetaData { String keyName //键名 String[] attrNames //属性(组) String keyType //键类型(主键或候补键) } class CheckMetaData { String checkType //CHECK 约束类型 String checkName //约束名 String definition //约束定义 }</pre>
---	--

图 1 数据结构定义

Fig.1 Data-structure definition

2.2.2 处理流程与逻辑

RDBSExtractor 中模式提取借助于内含在 J2SE 平台中的 JDBC API v3.0^①来实现.JDBC API 使用支持多种 RDBMS 驱动程序的驱动管理器,只需在驱动管理器中加载某种数据库驱动程序即可实现相应数据库的操作,因此屏蔽了各 RDBMS 间的差异.JDBC API 中的包 java.sql 提供了用 Java 语言访问和操纵 SQL 数据库的处理逻辑(接口和方法).RDBSExtractor 的处理流程如图 2 所示.

用 JDBC API 与数据库建立连接,首先需要加载相应数据库的 JDBC 驱动程序,然后调用 java.sql 包中 DriverManager 类 getConnection()方法,获得一个 Connection 对象.此 Connection 对象是 java.sql 包中接口 Connection

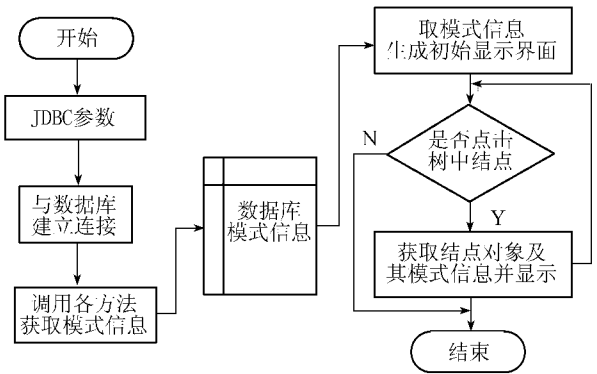


图 2 RDBSExtractor 处理流程

Fig.2 Flow chart of RDBSExtractor

表 2 模式信息提取使用的主要方法

Table 2 Main methods used for extraction of schema information

方法名	描 述
getDatabaseProductName()	获取 DBMS 的名称
getDatabaseProductVersion()	获取 DBMS 的版本号
getColumns()	获取指定关系的所有属性
getPrimaryKey()	获取指定关系中定义的主键
getImportedKey()	获取指定关系中定义的所有外键

的一个实现,表示与该数据库的一次连接.建立了连接之后,调用 Connection 对象的 getMetaData()方法,返回一个 DatabaseMetadata 对象.此 DatabaseMetadata 对象是 java.sql 包中接口 DatabaseMetadata 的一个实现.DatabaseMetadata 接口中提供了获取数据库模式信息的许多方法.RDBSExtractor 中提取模式信息时使用的 DatabaseMetadata 接口中的诸方法如表 2 所示.

为实现在数据库名—关系名—属性名组成的层次结构(树)中点击树结点即显示相应模式信息的功能,需要使用 Java 的事件监听机制.事件监听包括三部分处理逻辑:首先要在与被点击树结点对应的事件句柄类中定义实现事件反应器接口(TreeSelectionListener)的类,以监听树结点被点击的事件,然后定义一个事件处理

① Sun Microsystems, Inc. JDBC Technology. Online : <http://java.sun.com/products/jdbc/> 2002-05-09/2004-11-05.

类, 并将它的一个实例注册成与树对应的组件的响应器(用 `addActionListener()` 方法), 还要实现响应器接口的方法(即 `valueChanged()` 方法), 用于定义事件发生后进行的一些动作(如从被点击结点的对象中获得与该结点相关的信息)。

3 实 例

下面以一个实例来验证 `RDBSExtractor` 的有效性。用 Microsoft SQL Server 2000 创建一个学生选课情况的数据库 `stu_course`, 其中包括 `Student`, `Course` 和 `Enrol3` 个关系(基表)。数据库模式定义的 SQL DDL 语句见图 3。

用 `RDBSExtractor` 对这一数据库进行模式提取和可视化, 运行结果如图 4 所示。图 4(a)中, 左边的树型结构用于显示 `stu_course` 数据库中定义的各关系和属性的层次关系, 右边的面板用于显示数据库元数据和约束。鼠标点击树中表示 `Enrol` 关系的结点, 右边面板就显示 `Enrol` 关系中定义的元数据和相应表约束。点击表示 `Student` 关系中 `sex` 属性的结点时, 就显示该属性的元数据和相应列约束, 如图 4(b)所示。

```
CREATE DATABASE stu_course ;
CREATE TABLE Student (
    stuNo char(7) PRIMARY KEY ,
    stuName char(12) UNIQUE ,
    sex char(6) CHECK([sex] = 'Male ' or [sex]
        = 'Female '),
    CONSTRAINT CK_Stu CHECK([stuNo] LIKE 2002% AND
        [stuName] IS NOT NULL )
);
CREATE TABLE Course(
    csID char(6) PRIMARY KEY ,
    csName char(20) NOT NULL ,
    credit INT DEFAULT(2),
);
CREATE TABLE Enrol(
    stuNo char(7),
    csID char(6) REFERENCES Course( csID ),
    grade INT ,
    CONSTRAINT PK_Enrol PRIMARY KEY( stuNo ,csID ),
    CONSTRAINT enrolOf FOREIGN KEY( stuNo )
        REFERENCES Stude( stuNo ) ON DELETE CASCADE
);
```

图 3 数据库模式定义 SQL DDL 语句
Fig.3 SQL DDL statements for
database-schema definition



图 4 数据库结构及模式信息
Fig.4 Schema information for database structure

4 结 语

本文介绍了关系数据库模式信息提取器 `RDBSExtractor` 的设计与实现技术。`RDBSExtractor` 基于 Java 2 平台, 借助 JDBC API 进行关系数据库模式信息的提取, 屏蔽了各种 RDBMS 的差异性。由于 Java 是当前数据密集型 Web 站点、“deep Web”以及各种 Web 应用的常用开发工具, 因此, `RDBSExtractor` 具有通用性和可移植性强的特点。

数据库模式中包含了数据库的结构和语义信息, 关系数据库模式信息的提取是实现当前 Web 向语义 Web 迁移的基础技术之一。`RDBSExtractor` 可以集成到语义标注“deep Web”或构造语义 Web 入口的 CASE 工具中。

参考文献：

- [1] CHANG K C ,HE B ,LI C ,et al. Structured Databases on the Web :observations and implications[J]. SIGMOD Record ,2004 ,33(3) :61—70.
- [2] BERNERS-LEE T ,HENDLER J ,LASSILA O. The semantic Web[J]. Scientific American ,2001 ,284(5) :34—43.
- [3] HENDLER J. Ontologies on the semantic Web[J]. IEEE Intelligent Systems ,2002 ,17(2) :73—74.
- [4] STOJANOVIC L ,STOJANOVIC N ,VOLZ R. Migrating data-intensive Web sites into the semantic Web[A]. In :Proc of the 17th ACM Symposium on Applied Computing[C]. New York :ACM Press ,2002. 1100—1107.
- [5] HANDSCHUH S ,STAAB S ,VOLZ R. Annotation for the deep Web[J]. IEEE Intelligent Systems ,2003 ,18(5) :42—48.
- [6] MORENO N ,NAVAS I ,ALDANA J F. Putting the semantic Web to work with DB technology[J]. IEEE Data Engineering Bulletin ,2003 ,26(4) :49—54.
- [7] SILBERSCHATZ A ,KORTH H F ,SUDARSHAN S. Database system concepts[M]. Fourth Edition. New York :McGraw-Hill Companies ,2002. 921—1006.
- [8] TÜRKER C ,GERTZ M. Semantic integrity support in SQL :1999 and commercial (Object-)relational Database management systems[J]. The VLDB Journal ,2001 ,10(4) :241—269.

Extraction of relational database schema information

XU Zhuo-ming , SU Wen-ping

(College of Computer and Information Engineering , Hohai Univ. , Nanjing 210098 , China)

Abstract :Relational databases are the main information sources of the current Web , and the extraction of schema information from relational databases is one of the basic techniques for transferring the current Web into the Semantic Web. In this paper , a prototype tool RDBSExtractor , as well as its design ideas and implementation techniques based on JDBC API , is introduced , and the schema information of relational databases is analyzed. Case studies show that the JDBC API based schema extraction method is feasible , and that RDBSExtractor is effective.

Key words :relational database ; schema extraction ; Semantic Web ; CASE tool