文章编号:1671-4598(2019)01-0146-04 DOI:10.16526/j.cnki.11—4762/tp.2019.01.030 中图分类号:TP393 文献标识码:A

基于 Web 的用户自适应关系模型的设计及应用

陈 伟[1],陈 功[2],潘文杰[1],蔡 斌[1],熊永华[2]
(1. 贵州省烟草科学研究院,贵阳 550081;2. 中国地质大学 自动化学院,武汉 430074)

摘要:Web 开发过程中,固定的数据库关系模型设计往往难以满足数据库应用系统中多变的用户需求;本文提出一种用户自适应关系模型,利用主、从两个静态关系表来实现数据库动态表结构,主表用于存储用户自定义关系表信息,从表则用于存储所有用户自定义数据;利用该关系模型设计的数据库应用系统中,用户仅通过 Web 浏览器即可自定义数据类型和动态管理数据,从而在一定程度上实现了用户需求变化的自适应;以烟草业的数据共享平台为例,说明了模型的使用方法,实际应用结果表明该模型显著提高了数据库应用系统的用户适应性,改善了 Web 站点的性能。

关键词:Web;自适应;关系模型;数据库;设计及应用

Design and Application of User—Adaptive Relationship Model Based on Web

Chen Wei[1], Chen Gong[2], Pan Wenjie[1], Cai Bin[1], Xiong Yonghua[2]
(1. Guizhou Tobacco Science Institute, Guiyang 550081, China;
2. School of Automation, China University of Geosciences, Wuhan 430074, China)

Abstract: During the process of WEB development, the fixed database relational model often struggles to meet the changing needs of users in database application system (DAS). To achieve the dynamic structure of DB table, we present a user—adaptive relational model which adopts two static DB tables - master and slave. In this structure, the master table is used to store the user custom relational table information, while the slave to store all user—defined data. In the DAS which exploits this relational model, users can customize the data type and perform dynamic data management. Thus it, to some extent, realizes the adaptation of users' changing needs. We illustrate the use of this model with the Data Sharing Platform in tobacco industry. The practical results show that the user adaptability to DAS and the performance of WEB sites have obvious improvement.

Keywords: website; adaptive; relation model; database; design and application

0 引言

需求分析一直以来都是软件工程项目的重点和难点环节,如何做好用户需求分析,从而尽可能的避免应用系统投运后再返回修改功能需求,是工业界和学术界关注的重要问题。当前,Internet 技术的发展,使得基于 Web 的数据库应用项目数量激增,且遍布社会生活的各行各业,这使得在进行项目需求分析时,用户类别更加广泛,用户需求和数据类型则更加多样和易变,因此,提高数据库应用系统的用户适应性,进而减少用户需求变化对软件工程项目的不良影响,具有重要的研究意义和实际应用价值[1],在国内外都得到了广泛研究。

对于用户多变的数据需求或未知的数据结构,常见的处理方法有以下几类:第一类是把未知的数据结构保存为 xml 格式文件的形式[2-4];第二类是用键—值映射关系来实现静态数据库的列—属性映射关系[5-6];第三类是采用静态表结构描述动态表结构[7-8];第四类是 SaaS(Software as a service,软件即服务)模式下多租户技术[9-12]中的共享关系表方法[13]。前两类方法都存在编码量大、存取效率低、数据不易管理等缺点,第三类方法是比较高效的方法[14],但是逻辑处理比较复杂,难以应用于大型关系模式;第四类方法只能在一定范围内满足有同类数据需求的用户自适应性,无法满足大多数用户的数据需求,局限性较大。

本文结合实际工程项目中所出现的:非信息专业用户数据库知识欠缺,同时又希望能够自定义和管理自身数据,从而造成用户需求难以在需求分析阶段完全确定等问题,将上述第三类和第四类方法相结合,使用主、从两个相关联的静态关系表来描述动态表结构,主表用于存储用户自定义虚拟关系表信息;使用多租户技术实现从表的数据共享功能,用于存储所有用户自定义数据。给出了一个实际的工程用例,运行结果表明,利用该关系模型设计数据库,用户可以通过浏览器自定义并且管理虚拟关系表,从而有效提高数据库应用系统对用户需求变化的适应性,同时改善 Web 站点的性能。

1 用户自适应关系模型

多租户技术中的共享关系表可以满足所有同类租户的数

收稿日期:2018-05-17; **修回日期**:2018-07-19。
基金项目:贵州省科技重大专项计划(黔科合重大专项字[2014]6015—1);国家自然科学基金项目(61202340);中国博士后基金博士后交流派出计划资助项目(20140011)。
作者简介:陈 伟(1973-),男,四川武胜人,博士,主要从事作物栽培与模型构建方向的研究。

据需求。在没有单独为每个租户建立独立关系表的情况下，通过一张共享关系表来存储所有同类租户数据，以租户 ID 来隔离不同租户之间的数据，同时每个租户拥有自己独立的关系表，但这些独立关系表在数据库中并不是物理存在的，而是一种虚拟或逻辑关系表。所有虚拟关系表实际上共用了共享关系表的关系模型，因此不需要再为每个虚拟关系表建立独立的关系模型。

本文利用了该方法通过通用关系模型，在不新建数据库关系表的情况下可以拥有任意多个虚拟关系表的特点，采用静态表结构表现动态表结构设的思想计出一种基于 Web 的用户自适应关系模型，如图 1 所示。

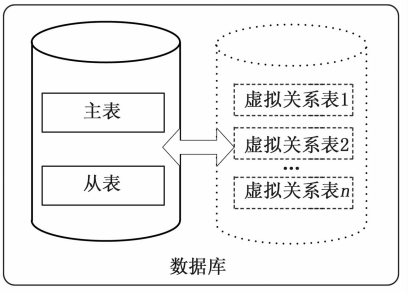

图 1　自适应关系模型的数据库关系表

该关系模型分为两部分，主关系模型和从关系模型，分别对应数据库中主、从两种静态关系表。主表用于存储用户自定义关系表的表名、字段名以及该表在从表中所使用的起止字段等结构信息；从表类似于多租户技术中的共享关系表，用于存储用户自定义数据，插入的记录通过表名来隔离、区分。主表中一条记录映射到用户新建虚拟关系表的结构信息，因此可对应从表中同一表名的多条记录。

当用户通过浏览器网页新建数据库关系表时，新建虚拟关系表的结构信息存储在主表中，当用户需要管理自定义虚拟关系表的数据时，后台逻辑就先通过主表查询到该虚拟关系表的结构信息，比如该表的字段名、该表在从表中所使用的起止字段等等，然后根据查询到的虚拟关系表结构信息来对从表进行相应操作。

1.1　自适应关系模型 ER 图

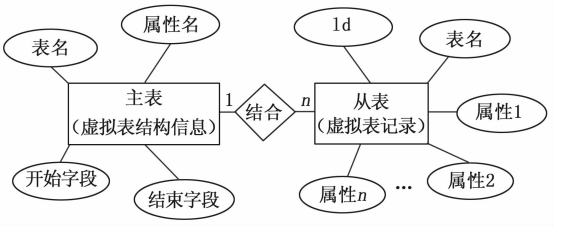

图 2　自适应关系模型 ER 图

1.2　主表设计

用户建立的所有虚拟关系表结构信息都存储在主表中，表名不允许重复，故设置表名 tableName 为主键，主表中还有如下几个字段：startColumn、endColumn、columnName。startColumn 字段表示该虚拟关系表的记录在从表中是从第几个字段开始，endColumn 字段表示该虚拟关系表的记录在从表中到第几个字段结束，columnName 字段表示该虚拟关系表的字段名，把用户新建的虚拟关系表的所有字段名用特殊符号“|”隔开，连接成字符串存储在 columnName 这个字段。主表的数据字典如表 1 所示。

表 1　主表 TableInfor 数据字典

字段名	数据类型	长度	允许空	主键	注释
tableName	varchar	100	not null	PK	虚拟表名
startColumn	int		not null		起始字段
endColumn	int		not null		终止字段
columnName	varchar	Max	not null		字段名

1.3　从表设计

从表设置一个 int 类型自增主键 id，用来表示每条记录的编号。由于从表中的记录通过表名来区分该条记录属于哪个虚拟关系表，用户只有在新建了虚拟关系表之后才能向新建的虚拟关系表中插入数据，因此设置主表的主键 tableName 为从表的外键，这样能够保证所有出现在从表中的记录的表名都能够在主表中找到。由于无法预知用户新建的虚拟关系表的字段数，所以从表的字段数应尽量大，这样才能够满足所有用户的需求。从表的其他字段就用来存储用户插入到虚拟表中的记录，其他字段的命名方式按照字段编号来命名，考虑到 Java 语言中一个对象最多只能拥有 255 个属性，这里我们把其他字段分别命名为 column1、column2 … column250。从表的数据字典如表 2。

表 2　从表 TableData 数据字典

字段名	数据类型	长度	允许空	主键	注释
id	int			PK	id
tableName	varchar	100	not null	FK	虚拟表名
column1	varchar	200	null		字段 1
column2	varchar	200	null		字段 2
column3	varchar	200	null		字段 3
…	varchar	200	null		…
column250	varchar	200	null		字段 250

1.4　数据库物理模型

本方法中虚拟关系表实际上并不是真正的在数据库中去创建新关系表，而是通过数据库中已经存在的一个主表 TableInfor 来存储用户自定义关系表的结构信息，自定义关系表的结构信息包括：表名、起始字段、终止字段、字段名。把用户所要插入到自己创建的关系表的记录都存储在从表 TableData 中，如图 3 所示：在 TableData 表中设置 tableName 为外键，通过这个外键来约束从表 TableData 中记录的 tableName 字段一定是在主表 TableInfor 中存在的。主表中的 columnName 字段是用来存储用户创建的关系表的字段名，把用户定义的所有字段名通过“|”符号链接成一个字符串存储在表中。主表中的 startColumn 与 endColumn 字段是通过分割 columnName 后，根据得到虚拟关系表的字段数目计算而来。它们代表其对应虚拟关系表中的记录在从表 TableData 中所使用的是哪些字段，为简单起

见，默认设置用户新建虚拟关系表中除主键 id 外所有字段为 varchar 数据类型。

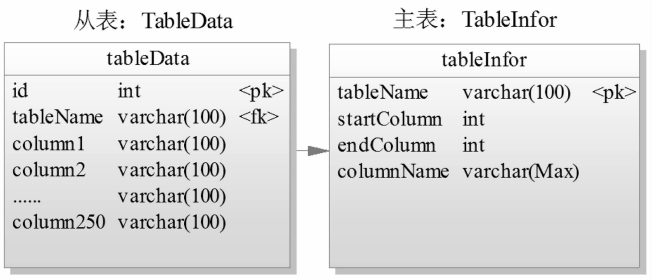

图 3 数据库物理模型

2 主、从表逻辑处理

2.1 主表逻辑处理

2.1.1 “新增”自定义虚拟关系表

通过 API（Application Programming Interface，应用程序接口）新增虚拟关系表，用户通过浏览器网页填写自定义虚拟关系表的结构信息确定后，后台通过 http 协议获取用户填写的关系表结构信息，并对用户填写的关系表结构信息进行校验，校验包括用户自定义虚拟关系表表名是否已经存在、自定义虚拟关系表字段名是否符合格式要求，如果校验失败，提示用户重新填写信息，如果校验成功，则向主表插入记录。

2.1.2 “删除”自定义虚拟关系表

通过 API 删除某自定义虚拟关系表，后台通过 http 协议获取到要删除的自定义虚拟关系表的表名，然后查看从表中是否存在该表名的记录，若存在则先删除中该表名的所有记录，然后再删除主表中该表名的记录。

2.1.3 “修改”自定义虚拟关系表

通过 API 修改某自定义虚拟关系表，后台查询主表中该表名的记录，在前台网页显示该表的结构信息，前台网页可以对该表结构信息进行修改后保存，由于主表中表名为从表的外键，简单起见不允许修改自定义虚拟关系表表名，然后跳转到后台，对修改后的表结构信息进行校验，校验成功后更新主表中该表名的记录，校验失败则返回页面提示用户校验失败原因，让用户重新修改。

2.1.4 “查询”自定义虚拟关系表

通过 API 查询自定义虚拟关系表，自定义虚拟关系表的信息查询实际是对主表的数据进行查询，根据查询条件生成相应的 sql 查询语句，执行 sql 查询语句，遍历查询结果，把查询结果赋值给主表对应的对象实例，然后通过 http 协议把实例对象传到前台网页显示。

综上所述，主表的“增、删、改、查”四种基本操作中除了删除操作以外，其他三种操作只需要直接对主表直接操作即可，由于从表外键约束的原因，删除操作必须联合从表进行相关检查后操作，如图 4 所示。

2.2 从表逻辑处理

2.2.1 “新增”自定义虚拟关系表中的记录

通过 API 新增某自定义虚拟关系表中记录时，先查询主表中该虚拟关系表的结构信息，把查询结果传到前台网页，在前台网页解析该虚拟关系表的结构信息（主要指字段名），在网页上显示该虚拟关系表的表结构，以供用户新增该虚拟关系表的记录，在用户新增记录提交后，后台通过 http 协议获取到用户新增的数据，并给从表对应的对象实例进行赋值，然后向从表插入记录。

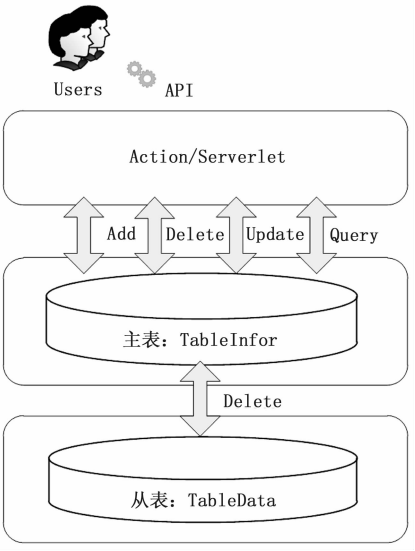

图 4 主表的基本操作

2.2.2 “删除”自定义虚拟关系表中的记录

通过 API 删除某自定义虚拟关系表中记录时，先查询主表中该虚拟关系表的结构信息，同时查询从表中该表名的记录，分别给主表与从表对应的实例对象赋值，传到前台网页，在网页上解析主表的结构信息，在网页上显示该虚拟关系表的表结构，并遍历从表实例对象，把遍历结果一一对应放在已经解析好的虚拟关系表中，选中记录删除后，后台通过 http 协议获取到需要删除的记录的 id，然后删除从表中对应的记录。

2.2.3 “修改”自定义虚拟关系表中的记录

通过 API 点击修改某自定义虚拟关系表中记录时，先查询主表中该虚拟关系表的结构信息，同时查询从表中该表名的记录，分别给主表与从表对应的实例对象赋值，传到前台网页，在网页上解析主表的结构信息，在网页上显示出该虚拟关系表的表结构，并遍历从表实例对象，把遍历结果一一对应放在已经解析好的虚拟关系表中，修改保存后，后台通过 http 协议获取到修改后的数据，并保存到数据库从表中。

2.2.4 “查询”自定义虚拟关系表中的记录

通过 API 点击查询某自定义虚拟关系表中记录时，先查询主表中该虚拟关系表的结构信息，同时查询从表中该表名的记录，分别给主表与从表对应的实例对象赋值，传到前台网页，在网页上解析主表的结构信息，在网页上显示出该虚拟关系表的表结构，并遍历从表实例对象，把遍历结果一一对应放在已经解析好的虚拟关系表中。

从表的“增、删、改、查”操作如图 5 所示。

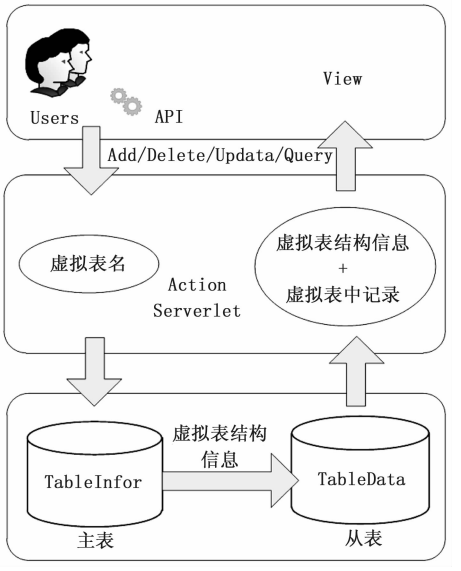

图 5　从表的基本操作

3　应用实例

3.1　用户需求

“烟草业数据共享平台”是一个基于 Web 的数据库应用系统，该系统按照统一的数据标准体系，提供统一的数据上传接口把烟叶生产各个环节的测试或试验数据进行集中的数据库存储与管理，并经过一定的关联性分析与数据挖掘，从而达到指导烟叶生产的目的。

在“烟草业数据共享平台”项目开发需求调研阶段，有着这样一种用户需求：研究人员为了找出彰显烟叶风格特色相关因素，会对各个地区的烟叶样品采摘回来进行化学成分分析、代谢物质分析、评析指标分析等几十种甚至上百种测试或试验分析。然而像化学成分分析与物质代谢分析是无法预先知道化学成分与代谢物质的，有时会为了单独分析某种因素而临时增加一些试验，这就需要提供用户在浏览器网页上可以随时新建并修改数据库关系表的功能。

传统 Web 开发关系模型设计方式显然难以满足这样不确定的用户数据需求，然而本文提出的自适应关系模型能够提供用户通过浏览器网页自定义数据库关系表，解决了不明确的用户需求给项目开发带来的困扰。

3.2　开发环境

项目在 Windows 7 平台上进行开发，开发中使用 Eclipse 作为集成开发工具，Tomcat 7 作为 Web 服务器，SQL Server 2008 数据库。

3.3　应用实例及结果分析

下面通过烟“草业数据共享平台”项目中自适应关系模型的应用实例来对自适应关系模型进行分析说明，表 3、4 分别为主表 TableInfor 与从表 TableData 中的存储数据实例。

由主表实例可以看出，自定义虚拟关系表“代谢物质”的起始字段为 1，终止字段为 2，字段名为“草酸（mg/g）| 烟碱（mg/g）”，意思是从表中 tableName 为“代谢物质”的记录中 column1 的值即为虚拟关系表“代谢物质”中

表 3　主表 TableInfor 实例

tableName	startColumn	endColumn	columnName
代谢物质	1	2	草酸（mg/g）| 烟碱（mg/g）
化学成分	1	3	钾（mg/g）| 氯（mg/g）| 磷（mg/g）
感官质量	1	3	香气量 | 杂气 | 甜度
风格特征	1	4	清香 | 木香 | 果香 | 正甜香
…	…	…	…

表 4　从表 TableData 实例

id	tableName	column1	column2	column3	…	column250
1	代谢物质	3.24	2.87	NULL	NULL	NULL
2	代谢物质	3.15	2.56	NULL	NULL	NULL
3	化学成分	0.15	0.65	0.34	NULL	NULL
4	化学成分	0.24	0.53	0.28	NULL	NULL
5	感官质量	6.07	5.71	5.93	NULL	NULL
6	感官质量	7.26	5.67	5.84	NULL	NULL

“草酸（mg/g）”字段的值，column2 的值即为虚拟关系表“代谢物质”中“烟碱（mg/g）”字段的值；自定义虚拟关系表“化学成分”的起始字段为 1，终止字段为 3，字段名为“钾（mg/g）| 氯（mg/g）| 磷（mg/g）”，意思是从表中 tableName 为“化学成分”的记录中 column1 的值即为虚拟关系表“化学成分”中“钾（mg/g）”字段的值，column2 的值即为虚拟关系表“化学成分”中“氯（mg/g）”字段的值，column3 的值即为虚拟关系表“化学成分”中“磷（mg/g）”字段的值；自定义虚拟关系表“感官评价”的起始字段为 1，终止字段为 3，字段名为“香气量 | 杂气 | 甜度”，意思是从表中 tableName 为“感官评价”的记录中 column1 的值即为虚拟表关系“感官评价”中“香气量”字段的值，column2 的值即为虚拟关系表“感官评价”中“杂气”字段的值，column3 的值即为虚拟关系表“感官评价”中“甜度”字段的值。

从以上实例可以看出用户自定义虚拟关系表的表结构信息都存储在表 3 中，而对应的记录存储在表 4 中，同时，用户可以随意建立满足自己需求的虚拟关系表。

4　结论

数据库应用系统的用户往往不具有信息专业的相关知识和技能，系统开发过程中用户的需求也经常发生变化，从而严重影响了软件工程项目的进度和质量。针对这一问题，本文提出了一种基于 web 的用户自适应关系模型，该方法通过特殊的关系模型设计，以及建立虚拟关系表的思想，使用静态表结构来表现动态表结构，实现了用户可根据自身需要在网页上新建以及动态管理数据表及其数据的

（下转第 155 页）