

逻辑和物理模型同步变更规则研究

袁 满,王 敏,黄 刚,胡 庆

(东北石油大学 计算机与信息技术学院,黑龙江 大庆 163318)

摘 要:目前,绝大多数企业的逻辑模型和物理模型还处于一种初级的手工管理阶段,即使部分企业采用了电子化的管理,也仅局限于对逻辑模型的管理,企业很少将逻辑模型变更与物理模型变更进行同步考虑。文中通过对逻辑模型的各种变更情况进行梳理,制定了一套规则及算法并对规则和算法进行定义,保证了逻辑模型和物理模型的一致性,该规则的定义不仅方便企业对数据的管理,而且保证数据在逻辑和物理的一致性。最后将构建的规则运用到油田企业数据字典的管理中,取得了很好的应用效果。

关键词:逻辑模型;物理模型;同步变更;规则

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2012)01-0059-04

Research on Synchronization Change Rule for Logical and Physical Model Structure

YUAN Man, WANG Min, HUANG Gang, HU Qing

(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: At present, the logical model and physical model in most enterprises are still in a primitive manual management phase. Even if parts of enterprises adopt electronic management, which are only limited to logical model, they rarely take logical model changes into account with physical model changes. Through collating various logical model changes situations, a set of rules and algorithms are made and defined. The definition of the rules will not only facilitate the management of enterprise data, but also ensure data consistency in the logical and physical model. Meanwhile the data management of enterprises becomes convenient. At last the rules constructed are applied to oilfield enterprises data management, and achieved very good application effect.

Key words: logical model; physical model; synchronization change; rule

0 引 言

传统的模型管理都是针对底层数据结构的管理,很少有将逻辑和物理模型结合考虑的,对于二者的同步问题的考虑更是少之又少。因此,对于逻辑和物理模型结构同步变更规则将是文中研究的一个重要问题。山东财经大学聂培尧^[1]等人通过面向对象的方法提出如何将关系数据库进行操作以执行各种模型管理功能,并使用面向对象的方法对模型进行管理问题的探讨。兰州交通大学王奕^[2]等人提出了基于本体的模型管理,他将本体论的研究作为基础,提出一种决策本体的概念来解决模型管理上的一些问题。2003年美国加利福尼亚圣克鲁兹存储系统研究中心的Brandt^[3]等人提出了LH算法,该算法主要是通过使用

hash函数计算每次管理员对数据所进行的操做记录,在管理数据过程中通过权限访问机制进行分区管理^[4],提高了更改访问授权等操作的性能。在对企业中的数据进行管理的过程中,随着企业中数据量的不断增加,数据的不一致性也就随之而来。要想对数据进行统一管理,必须实现逻辑和物理模型的同步来保证数据的一致性。

1 几个定义与支撑元模型设计

1.1 几个概念的定义

定义1:规则(Rule),所谓的规则就是对逻辑模型进行各种变更操作所遵循的规律的描述。例如,对于一个关系模型来讲,当模型中的主键进行删除操作时,会引发一系列的相关联模型的同步变更,具体来讲如何变更,对这些规则采用元数据进行描述与存储。

定义2:主题(Topic),所谓的主题就是对逻辑模型进行分类的规则,例如,可以按模型所属的职能域分类,可按照具体的业务进行分类等。

收稿日期:2011-06-17;修回日期:2011-09-21

基金项目:黑龙江省教育基金(11541008)

作者简介:袁 满(1965-),男,教授,CCF会员,研究方向为企业信息集成、高级数据管理、网络计算等;王 敏(1985-),女,硕士研究生,研究方向为数据仓库技术。

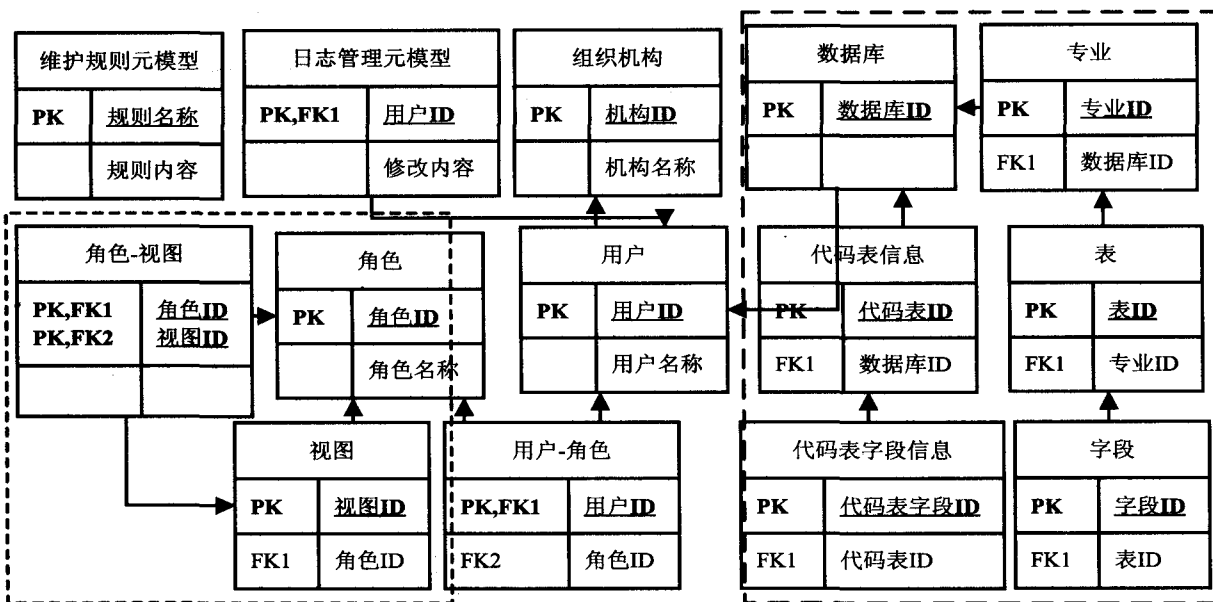


图 1 支撑元模型结构框架

定义 3: 数据源 (Data Source), 用来描述逻辑数据模型字典的来源信息^[5], 例如, 描述油田开发的字典可能来源于油田开发数据库, 也可能来源于 EPDM 等。

1.2 支撑元模型设计

逻辑模型与物理模型同步变更实质上是字典管理的核心问题, 逻辑模型的变更和物理模型相关联。支撑元模型的设计结果必须要将逻辑模型和物理模型结合起来。支撑元模型的数据组织是按数据源、主题以及数据字典进行组织, 构建了用于描述与存储数据源、主题、字典、用户角色以及同步变理规则系列元模型^[6]。根据需要设计出存储元模型、用户相关元模型、日志管理元模型和参照规则元模型, 如图 1 所示, 存储元模型描述逻辑模型中数据的底层结构, 涵盖了数据库、专业数据、代码表信息及代码表具体字段信息的存储等; 用户相关元模型是从职能域的角度对各用户、角色按照组织机构进行划分, 出于对数据安全性的考虑这里引入了权限控制, 不同层面的用户可以根据权限管理相关数据; 日志管理元模型则是对管理数据的过程中对数据进行操作的记录, 通过该模型可以方便管理人员根据记录信息进行高效的管理数据^[7,8]; 对于以上三种元模型的构建都要遵循规则元模型里设计的相关规则, 保证数据的规范性。

2 逻辑与物理模型同步变更规则

2.1 基本规则构建

为了更好地描述数据源、主题、字典和表及字段之间的这种逻辑关系, 根据信息资源规划理论将逻辑模型和物理模型的同步规则进行构建, 一是从名称、代码命名规则进行构建; 二是从表中主键同步、外键同步

规则进行构建。

(1) 现将数据源、主题和表的关系用矩阵定义如下:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{m1} & A_{m2} & A_{m3} & \cdots & A_{mn} \end{bmatrix}$$

$i = 1, 2, 3, \dots, n; j = 1, 2, 3, \dots, m; A$ 矩阵代表某个数据源中包含的所有主题;

$A_{11} = \{ B_{11} B_{12} B_{13} \cdots B_{1k} \}$, A_{11} 代表第一个主题下的 1 个子主题; k 代表该主题下的第 k 张表;

$B_{ap} = \{ c_x \}$, B_{ap} 代表第 a 主题下的第 p 张表; a 的取值从 1 到 m , p 的取值从 1 到 k ; x 表示表中列的个数; 设 c_1, c_2 为表 B_{11} 的主键, c_3 为表 B_{12} 的外键引自表 B_{11} 中的主键 c_2 ;

当 $a = 1, p = 5, x = 5$ 时, $B_{15} = \{ c_{11}, c_{12}, c_{13}, c_{14}, c_{15} \}$ 表示第一个主题中的第五张表中有 5 列。

(2) 名称、代码命名规则定义。

针对企业中数据之间的名称和代码冲突问题, 文中制定了数据源、主题、表及字段的名称、代码命名规则^[9], 用上面定义矩阵中的元素进行定义如下:

规则 1: 主题名称不重复规则: A_{1i_name} 同一级别内的 i 个主题名称不能重复; A_{1i_name} 与 A_{2i_name} ($i = 1, 2, \dots, n$) 不同级别的 i 个主题名称可以相同;

规则 2: 主题代码不重复规则: A_{1i_code} 同一级别内的 i 个主题代码不能重复; A_{1i_code} 与 A_{2i_code} ($i = 1, 2, \dots, n$) 不同级别的 i 个主题代码可以相同;

规则 3: 表名称不重复规则: B_{ij_name} 同一主题内

表的名称不能重复; B_{ij_name} 与 B_{2j_name} ($j = 1, 2, \dots, k$) 不同主题下的表的名称可以相同;

规则4:表代码不重复规则: B_{ij_code} 同一主题内表的代码不能重复; B_{ij_code} 与 B_{2j_code} ($k = 1, 2, \dots, k$) 不同主题下表的代码可以相同;

规则5:表中列的名称和代码遵循以上规则定义。

2.2 复杂规则构建

2.2.1 表删除规则算法描述

当删除一个表时,该表所包含的所有数据也要一并删除,其他表中引用该表中数据作外键的,要取消外键引用关系,其中外键若为随机码则删除外键,若为有实际意义的数据则不删除,但从删除表中复制的其他表中的数据要删除;录入或修改表基本信息时遵循表所在的主题中表的名称和代码不重复原则,修改表的代码时,同时要修改引用该表中数据作外键的数据的外键引用表代码。

表删除规则算法:

```
Begin
IF 删除表  $B_{ap}$ 
THEN
删除  $B_{ap}$  中的  $x$  条记录
取消表  $B_{ap+1}$  中的外键  $c_{x+1}$  且删除  $B_{ap+1}$  中从  $B_{ap}$  中复制来的记录
IF  $c_{x+1}$  为随机码
THEN 删除  $c_{x+1}$ 
ELSE 不删除  $c_{x+1}$ 
ELSE IF 录入或修改表  $B_{ap}$ 
THEN
 $B_{ap\_name}$ ,  $B_{ap\_code}$  与  $B_{ap+1\_name}$ ,  $B_{ap+1\_code}$  不能相同。其中  $B_{ap+1\_name}$  和  $B_{ap+1\_code}$  为字典中已存在的表。
ELSE IF 修改了  $B_{ap\_code}$ 
THEN
修改的  $c_{x+1}$  外键引用表代码和  $B_{ap+1\_code}$  中的从  $B_{ap\_code}$  中复制的数据的复制来源表代码
End
```

2.2.2 主题删除规则算法描述

当删除数据源中的一个主题时,该主题所包含的所有表和表中的数据要一并删除;录入或修改主题基本信息时也要遵循与其所在的字典内的其他主题的名称和代码不重复原则。

主题删除规则算法:

```
Begin
IF 删除主题  $A_{1i}$ 
THEN
删除第一个主题下的第  $i$  张表 {  $B_{11}$   $B_{12}$ 
```

$B_{13} \dots B_{1k}$ }, $i = 1, 2, \dots, n$

删除表 B_{1i} 中的记录

ELSE IF 录入或修改专业 A_{1i}

THEN

A_{1i_name} , A_{1i_code} 与 A_{1d_name} , A_{1d_code} 不能相同;其中, $i = 1, 2, \dots, n$, $d = 1, 2, \dots, n$, $i \neq d$; A_{1d} 是数据源中已经存在的主题

End

2.2.3 数据源删除规则算法描述

当删除一个数据源时,它所包括的所有主题、表以及表中的数据也要一并删除;录入或修改数据源时要遵循与其他字典名称和代码不重复原则。

数据源删除规则算法:

```
Begin
IF 删除数据源  $A_0$ 
THEN
删除  $A_0$  里的主题
删除所选主题下的表中的记录
ELSE IF 录入或修改  $A_0$ 
THEN
所选数据源的名称和代码不能重复
End
```

2.2.4 主外键删除规则算法描述

根据数据在表中的约束类型将数据分为主键、外键、主键且外键、联合主键、普通数据等,对于不同类型的数据参照的规则不同,下面针对每种约束类型的数据进行详细的规则描述:

首先主外键的宽精度和数据类型同步规则:在计量单位相同的情况下外键和对应主键的宽度、精度应相同,在计量单位不同时,主外键的精度、宽度应保持一致;外键和对应主键的数据类型必须相同,外键的数据类型不允许随便修改。

(1) 删除主键规则描述:删除或取消主键时,取消对应外键的外键引用关系,外键若为随机码则删除外键,若为有实际意义的数据则不删除;修改主键,修改主键的名称和代码时遵循同一个表中名称代码不重复规则,并且在修改代码的同时同步修改该主键对应外键的外键引用代码,而外键的代码不需改变。

主键删除规则算法:

```
Begin
IF 删除  $c_2$ 
THEN 取消外键  $c_3$ 
IF  $c_3$  为随机码
THEN 删除  $c_3$ 
ELSE 不删除  $c_3$ 
ELSE IF 修改  $c_2\_code$ 
```

```

IF   $c_2\_code$  在  $B_{11}$  中不重复
THEN   $c_2\_code$  修改成功
THEN  修改  $c_3$  的外键引用数据代码,  $c_3\_code$  不
变

```

End

(2)删除外键规则描述:删除外键时可以直接删除,取消外键时,若该外键为随机码,则取消外键的同时删除该外键,若为有实际意义的数据,则只取消不删除;增加外键时,要添加新增外键的外键引用表代码和外键引用代码,构建出新的关联关系;修改外键名称和代码时遵循同一个表中名称和代码不重复规则。

外键删除规则算法:

Begin

IF 增加外键 c_3 ,且它引用 c_2

THEN 添加 c_3 的外键引用数据代码为 c_2_code ,
外键引用表代码为 B_{11}_code

ELSE IF 修改 c_3_code ,且 c_3_code 在表 B_{12} 中
不重复

THEN c_3_code 修改成功

End

既是主键又是外键数据的删除规则:对这种类型的数据进行删除时优先考虑主键删除规则,必要时兼顾主键和外键的删除规则。

根据制定的各种规则,利用元模型理论构建了用于描述和存储这些规则的规则元模型。

3 应用案例

3.1 规则流程

数据管理人员选择数据源,确定数据源再选择系统所需的主题数据,根据主题的组织形式选择具体主题中的逻辑模型,根据具体的规则算法从物理模型库选择和逻辑模型中相应的物理模型进行同步,最终将变更结果显示出来。这个过程是程序通过规则算法自动完成,而不需要管理人员手动的更改。保证逻辑和物理模型同步变更的一致性。

3.2 具体实现

该系统主要包括三部分:树形导航菜单展示、逻辑字典展示区和物理表展示区。其中树形导航菜单是将逻辑数据字典按主题进行组织,分为公共数据、开发井业务数据、探井压裂业务数据和生产管理数据,以数据中心字典为根目录进行实际数据的表示,该部分涵盖了石油领域 7 个专业的业务数据^[10,11]。针对每个专业又分成了若干张表,当对表进行修改时要遵循上面定义的若干规则,从而使逻辑模型和物理模型保持同步。该部分是针对以上设计的元模型框架和规则的定义,采用 B/S 模式^[12],运用到实际项目中,实现了逻辑模

型和物理模型同步变更。底层数据是企业实际的业务生产数据,该系统的实现为油田企业的数据管理提供便利,更重要的是该模型的制定填补了对当前在数据字典同步管理的空白。

4 结束语

在油田领域中,石油生产数据是油田企业的重要资产,如何有效地管理这些重要数据、保证数据的一致性问题就成了目前管理层的重要课题。文中就是针对此问题进行展开研究,提出了一些详细的规则算法,该规则算法结合元数据管理策略并在前人的思想上提出了一种新方法、新思路,并将其运用到实际油田企业管理数据系统中,并取得了很好的应用效果,尽管这些规则是基于油田领域提出的,但是由于油田企业的数据库采用关系数据库,所以所提出的方法同样可以用于解决其它领域的逻辑模型与物理模型的同步问题。

参考文献:

- [1] 聂培尧. 面向对象的模型管理问题[J]. 计算机科学, 1993, 20(6): 71-74.
- [2] 王 奕. 基于本体的模型管理研究[J]. 计算机与信息技术, 2009(11): 201-205.
- [3] Brandt S A, Miller E L, Longd D E, et al. Efficient metadata management in large distributed file systems[C]//Proc of the 20th IEEE/11th NASA Goddard Conference on Mass Storage System and Technologies. San Diego: [s. n.], 2003: 290-298.
- [4] McBride B. Jena: Implementing the RDF Model and Syntax Specification[R]. [s. l.]: Hewlett Packard Laboratories, 2000.
- [5] 秦 晓. 元数据字典及其实现[J]. 计算机学报, 1994, 17(2): 81-86.
- [6] 胡长军, 全兆岐, 宫发明. EPICENTRE 数据模型应用研究[J]. 石油大学学报, 2000, 24(5): 88-91.
- [7] 吴 婷, 鞠时光, 蔡 涛. 基于 DBMS 的元数据管理策略[J]. 计算机应用研究, 2010, 27(4): 23-25.
- [8] 王亚宁, 程继红. 基于本体的军用元数据模型研究[J]. 计算机技术与发展, 2011, 21(4): 20-24.
- [9] Watkins A, Timmis J. Exploiting parallelism inherent in AIRS, an artificial immune classifier[C]//The 3rd International Conference on Artificial Immune System. Berlin, Heidelberg: Springer-berlag, 2004: 427-438.
- [10] 李子杰, 郑 诚. 流数据和传统数据存储及管理方法比较研究[J]. 计算机技术与发展, 2009, 19(4): 20-21.
- [11] 曹蓟光, 王申康. 元数据管理策略的比较研究[J]. 计算机应用, 2001, 21(2): 1-5.
- [12] 曾庆猛, 王冬梅. 中石油 A1 系统 PCEDM 数据模型基本实体初步研究[J]. 勘探地球物理进展, 2008, 31(6): 455-467.