

医院信息系统中数据完整性约束机制的实现

广州市红十字会医院 张志强

摘要:任何一个数据库都会由于某些自然因素或人为因素而受到局部或全局的破坏, 这些因素有的是不可避免的, 数据完整性约束的目的就是保护数据库中数据的正确性和一致性。数据完整性由完整性规则定义, 关系数据库系统中可以有三类完整性约束: 实体完整性、参照完整性和用户定义的完整性。医院信息系统的特性决定确保数据的正确性、一致性、相容性是系统最重要也是最基本的要求之一, 作者在组织实施广州市红十字会医院信息系统的过程中, 结合系统固有的业务特点提出了针对现实系统的数据完整性约束机制, 框架中对系统的数据完整性约束做了四个层次的划分: 数据库管理系统 (SQL SERVER) 对数据的约束、开发程序 (delphi) 对实现的约束、后台作业和稽查程序对应用的约束、规章制度对用户的约束。并根据以上机制, 合理的采用现有的成熟技术实现了数据完整性约束, 取得了较好的实际效果, 有效的解决了实际工作中所遇到的问题。

关键词: 完整性约束 医院信息系统

1. 研究背景

任何一个数据库都会由于某些自然因素或人为因素而受到局部或全局的破坏, 这些因素可能是系统硬件设备故障或软件的错误, 也可能是应用程序编制上的错误和操作员的操作错误, 或者是数据迁移过程中出现的错误等等, 这些因素有的是不可避免的。因此, 如何及时发现并采取措施防止错误进一步蔓延并能及时恢复, 对于一个管理信息系统来说是非常重要的。

数据完整性是数据库系统的一个重要方面, 是指数据的正确性和一致性 (相容性), 它有两方面的含义: 一是数据取值的正确性, 即字段列的值必须符合数据的取值范围、类型和精度的规定; 二是相关数据的一致性, 即相关表的联系字段的列值匹配。实现完整性约束的主要目的是防止语义上不正确的数据进入数据库。为维护数据库的完整性, 数据库管理系统 (DBMS) 提供了一种比较完整的机制来检查数据库中的数据, 看其是否满足语义规定的条件。这些加在数据库数据之上的语义约束条件称为数据库完整约束条件, 它们作为数据库中模式的一部分。

医院自身的目标、任务和性质决定了医院信息系统是各类信息系统中最复杂的系统之一。根据数据流量、流向及处理过程, 将整个医院信息系统划分为以下五部分: 临床诊疗部分、药品管理部分、经济管理部分、综合管理与统计分析部分、外部接口部分。临床诊疗部分要求数据符合数据的取值范围, 数据在表与表之间的表示形式与语义内容一致, 数据在传输过程中不能丢失或变更, 否

则会导致严重的医疗差错，是我们完整性约束要处理的重点数据。药品信息贯穿整个医疗信息系统中，是医院信息系统的主线之一，药品的购销活动涉及管理、用户（医生、护士）、病人，直接影响诊疗效果。如药房得到的发药信息与原始的药品处方和药品医嘱不一致会导致严重后果，也是完整性约束的重点部分。经济数据直接关系到医院发展，是信息系统的基础，也是系统用户和系统的服务对象之一病人非常关心的部分，大部分医疗投诉来源费用数据不能合理透明的提供，此部分需要严格进行约束。管理与数据密不可分，管理所需的数据来源于多个途径，如医院就诊人次的计算，有些人挂号后不就诊或延后就诊，有些跨科室就诊、有些只是咨询，来源于挂号、医生接诊和收费的数据存在客观上的不一致，需要提供约束机制及解释途径。

在实际的应用过程中，数据对一致性、完整性和实时性的要求越来越高，实现的难度也越来越大。医院信息系统提供全院的经济运行状态、医疗质量状态、工作质量状态等等，以及获取各部门的信息反馈，从而使各部门的管理者进行计划决策、组织实施、协调控制。而这些信息的提供，要求数据来源可靠，数据间没有矛盾产生，数据能够真实的反映客观实际，也就决定了必须进行数据的完整性约束。目前在构建医院信息系统时，没有意识到数据完整性对整个系统架构的影响，对约束机制和约束的实现方法研究不足，没有在多个层面上用统一的技术手段实现一种机制，进行数据完整性和一致性约束，制约了系统的发展。

关系数据库模型中的数据完整性约束理论上已经比较成熟，近年来，人们对数据完整性约束进行了大量的研究工作，在不同的侧面进行研究并应用于实际应用领域。对理论的阐述也比较多，关系模型和关系数据库上的完整性约束均有涉及。不同的数据库管理系统中数据完整性的实现是研究重点，如 Oracle、SQL Server、Note 等；有人提出了比较完整的开发工具和 XML 中完整性约束的实现方式。显然有人意识到了完整性约束对医院信息系统的影响，唐月华等针对医院信息系统表间的数据不一致进行了分析并查找出了原因，完整性约束的相关文献也列举了一些医院信息系统的实例来进行说明。没有发现有文章阐述医院信息系统中数据完整性约束整体机制的应用。

2. 医院信息系统完整性的实现方法

医院信息系统数据完整性控制策略在系统实施以前就必须进行充分讨论，制定完整性约束的整体框架，在系统的实施过程中，可根据实际情况对框架的具体内容进行调整。以用 BOLAND DELPHI 开发的构建在 SQLSERVER 上的应用系统为例，完整性约束主要体现在以下几个方面：

- (1) 通过 SQL SERVER 提供的完整性约束进行数据约束；
- (2) 通过 BOLAND DELPHI 提供的完整性约束进行实现约束；
- (3) 运用后台作业和稽查程序进行应用约束；

(4) 通过制定规章制度来进行用户约束

其中，由数据库管理系统提供的约束是整个机制中的核心。

2.1. 通过SQL SERVER提供的完整性约束策略进行控制

2.1.1 声明型数据完整性约束

在sql server 中，实现声明型数据完整性约束主要有以下几种方法：

(1)列的IDENTITY 属性说明为IDENTITY 的列由系统自动为其赋值，并保证其值在该表中的惟一性，每一个表中只允许有一个列定义为IDENTITY 的。这是系统主键的主要生成方式，这种方式降低了数据结构设计的难度。

(2) 用户定义的数据类型

用户定义的数据类型是根据基本数据类型进行定义的。它们提供了一种机制，可以将一个名称用于一个数据类型，这个名称更能清楚地说明该对象中保存的值的类型。当多个表的列中要存储同样类型的数据，且想确保这些列具有完全相同的数据类型时，可使用用户定义数据类型。

(3) 主键 (primary key) 约束

指定表中一列或数列作为数据表的主键，不允许有重复值，而且也不允许有空值。

(4) 惟一性 (unique) 约束

和primary key约束相类似，定义数据列值的惟一性约束，有一行可取null值，并且惟一性约束在一个表中可以使用多次，而主键值只可以在一个表中使用一次。在缺省的情况下，惟一约束产生非聚簇索引，而主键约束产生聚簇索引。

(5) 外键约束

外键约束又称参照完整性约束，用来限定本表外键码列值与相关表主键码字段列值的匹配，即保证相关数据的一致性。通过在同一个数据库的两个表中进行主键约束和外键约束来实现，参照的列和被参照的列的名字必须相同。在被参照表中，主键值被其它表所参照时，该行不能被删除，也不允许改变。在参照表中，不允许参照不存在的主键值。合理的引用约束在不影响系统性能的前提下，提高了系统的完整性。

(6) 缺省值 (DEFAULT CONSTRAINT) 约束

缺省值 (DEFAULT CONSTRAINT) 约束与列值相关联，而不是与数据类型相关联，可以为任何数据类型 (除timestamp和具有identity属性的列外) 定义缺省值约束。也可以创建称为缺省值的对象，并将其绑定到列或用户定义数据类型时，如果插入新数据时没有明确的提供值，缺省值便指定

一个值，并将其插入到对象所绑定的列中（或者，在用户定义数据类型的情况下，插入到所有列中）。

(7) 检查 (check) 约束

该约束限制输入到一系列或多列中的可能值，从而保证sql server数据库中数据的域完整性，在check约束中可以包含搜索条件，但是不能包含子查询，一个表中可以包含多个check约束，对于列也可以定义多个check约束。如例1中的记录状态字段，必须在指定的状态中变迁，不合理的状态不能获得合理的语义解释。

2.1.2 过程型数据完整性

主要指通过缺省、规则和触发器，或视图和存储过程实现。这一类完整性实现相对来说较复杂，但是具有更大的灵活性。主要用于实现域完整性与用户完整性。在数据库表的创建命令语句中定义约束的方法，虽然具有简单、方便、安全等特点，但它只对特定的表有效，不能应用到其它表，并且只能使用alter table 命令修改或删除约束，缺乏一定的灵活性。与此相反，在数据库中创建与表相对独立的规则、索引、和触发器对象，也能实现数据完整性，而且能实施更复杂、更完善的数据完整性约束。它的主要特点是：功能强、效率高、维护方便。

(1) 缺省 (default)

缺省是一种数据库对象，它与缺省值约束的作用类似，为insert 语句中未指定数据的列设置缺省值，缺省对象只适用受insert语句影响的行。

在应用缺省方法时如果该列对于该缺省不是足够大，系统自动舍弃超过该列长度的部分。同时缺省还需要符合该列的任何规则和所有MNDMO 约束。虽然缺省对象和缺省约束都可以为列设置缺省值，比较之下，使用缺省约束在一般情况下是一种较好的方法，因为约束的定义与表一起存储，当表被删除时缺省约束可以随表被自动删除，但是当需要为多列提供相同的缺省值时，使用缺省对象是一种更有效的方法。

(2) 规则 (rule)

规则是数据库对象之一，指定向某列（或与该规则绑定的用户定义数据类型的所有列）插入或更新数据时，限制输入新值的取值范围。一个规则可以是：值的清单或值的集合、值的范围、必须满足的单值条件、用like子句定义的编辑掩码等。当数据库中数据值被更新或插入时，就要检查新值是否遵循规则，如果不符合规则就拒绝执行更新或插入操作。规则类似于表定义中的校验约束，用来限定列的值域范围。但它不限于特定表，可以绑定到其它表的列或用户自定义的数据类型中使用。

(3) 定义索引

索引类似于表定义中的唯一约束,用来保证列值的唯一性。此外,它还能使用聚簇索引和非聚簇索引选项,来增强数据检索的性能。例如,对于住院病人收费明细帐表,分别建立了按执行日期和收费序号的非聚簇索引,建立了按照首页序号syxh的聚簇索引,首页序号标示一个病人的一次入院。大大提高了数据的查询速度,在未建立索引前,要在1000多万条记录中,查询满足条件的数据,大约需要5分钟左右,建立索引后只需几秒。

(4) 触发器(trigger)是特定类型的存储过程,作为数据修改(增删改)语句的一部分而在相应操作执行时自动执行,是维护数据库完整性最强大的工具,能在任何情况下去维护数据的完整性,应用程序不能绕过触发器,这也正是触发器和一般存储过程的一大区别。可以利用sql statements进行各种完整性检查。触发器是一种特殊的存储过程,它既不能带参数,也不能被直接调用,只能由系统自动激活。对于每个批命令或单一的数据操作语句,无论它影响一行还是多行数据,都只能被激活一次触发器。在触发器检测到错误时,它将撤消整个插入、修改或删除等操作。

触发器是一种实施复杂的完整性约束的有效方法,当对它所保护的数据进行修改时自动激活,防止对数据进行不正确、未授权或不一致的修改。类似于表定义中的参照完整性约束;触发器也可用来保证相关数据的一致性。它还能完成一些表定义参照完整性约束不能完成的任务:

(5) 存储过程(procedure)

是一个存储在数据库内的可执行的数据库对象,它可以接受参数、返回状态值和参数值,并且可以嵌套调用。本来存储过程是提供安全性考虑的,但是通过安全性提供了另一级的完整性,如果表的修改都是由存储过程来完成,便可以确保这些数据表的完整性和一致性,而不用担心手工操作可能产生的不一致的问题。存储过程一般与事务相联系,一个完整的处理过程能够在意外事件发生时保证数据的一致性,如药库药品出库到药房的过程中,要减少药库药品的库存,填写药库出库明细,同时,要增加药房药品的库存,填写药房入库明细,如果不是基于存储过程的事务处理,遇到网络中断或操作系统失去响应时,明显的会造成数据的不一致。

(6) 视图(view):是查看存储在表中数据的一种逻辑方法,它是通过一个被存储的select语句来实现的。如果需要对多个表设置一些完整性约束,可以用带with check option从句的视图定义来实现。实现功能和触发器类似。

在医院信息系统中,数据结构复杂,数据量庞大,为了加快系统的响应速度,一般采取“当前表-历史表-年表”的三级结构,如门诊处方信息和明细信息一般在当前表中存在7天的数据,7天后的数据进入历史表,1年后数据进入年表,这样,如果把查询任务都交给应用系统来实现,程序相当复杂,而且容易出现关联错误,视图是一种合理的机制,在医院信息系统中有着广泛的应用意义。

2.2 客户端利用delphi开发的应用程序实现数据的完整性

2.2.1 利用delphi的控件来确保数据的正确性

在以DELPHI为开发工具编写的应用程序中,可使用数据窗口对象的列校验属性来检查输入数据的正确性;在窗口输入界面中,还可采用下拉列表、复选框、单选按钮等控件,实现固定值选取输入来确保数据的正确性;另外,应用程序本身也可对输入数据进行有效性检查来确保数据的完整性。

2.2.2 利用事务的特点来保证数据的一致性

在应用程序中使用事务可保证数据库的数据一致性,在多用户的环境下可保持数据完整性。如果在执行相关数据的一组修改操作时,硬件(如网络)发生故障,这个突发事件便破坏了数据的一致性;如果利用事务控件,将上述的这组操作作为一个事务来提交,则要么其中的所有操作都成功执行,要么一个操作也不执行,从而保证了对应数据的一致性。

当使用Delphi 创建一个数据库应用时,Delphi 为所有的数据库操作提供了事务控件。事务是一个数据库上的一组操作,只有当事务中的所有操作都成功后,这组操作对数据库中一个或多个表的所有修改才永久化。如果事务中有任一操作不成功,则整个事务将重作。一个完整的事务处理要集中在一个事件或一个操作单元中。例如:门诊划价收费中的付款处理,同一个收费单据号下的数据,经处理后要更改医生处方库处方状态、收费记录及收费明细表、药房发药表等;所有这些处理算是一个完整的事务,必须等所有的语句都执行成功后,才能对数据库进行提交;如果有一条语句未执行成功,就要撤销该事务的一切操作,退回到事务开始的状态,这样才能保证数据的一致。利用客户端应用程序实现数据完整性的方法,其主要特点是:交互性好、功能强,但编程量大、维护困难、可靠性差。

Delphi提供了两种事务的控制方式:隐式事务控制和显式事务控制。缺省情况下,Delphi 通过BDE (Borland Database Engine) 为数据库应用提供隐式的事务控制,当数据库应用程序使用隐式事务控制时,Delphi 分别为每一条准备写入数据库的记录自动建立一个事务控制,单独提交每一条记录写操作,如Post 、RecordAppend 等。隐式事务使用方便,不需任何附加程序代码。但这种方式只在最小程度保证数据的一致性(同一条记录的各记录项,要么同时修改成功,要么一项也不修改);另一方面,由于隐式事务是单记录提交,因而增加了网络传输量,降低了系统性能。

2.3 运用后台作业和稽查程序增加数据库的完整性约束

2.3.1. 运用后台作业进行库表间数据同步

作业是由 SQL Server 代理程序按顺序执行的一系列指定的操作。作业可以执行更广泛的活动，包括运行 Transact-SQL 脚本、命令行应用程序和 Microsoft® ActiveX® 脚本。可以创建作业来执行经常重复和可调度的任务，并且作业可产生警报以通知用户作业的状态。

在医院信息系统实例中，超过 10 个不同的应用在同同时运行，不同的应用之间的数据约束，我们采用了 Sql Server 提供的后合作业来同步不同系统之间的数据，如将已经发生的出院病人费用和首页信息同步到统计和病案系统中，将门诊费用信息按照统一的结账时间汇总后同步到财务系统，将医保系统的药品项目对应信息同步到医院系统等。

2.3.2 运用后台稽查程序进行数据合法性检查

在系统运行过程中，我们发现有些涉及到完整性约束的问题不能够用上述的方法处理的，我们根据业务情况编制了一套稽查程序，每天将稽查结果提供给相关的业务部门进行处理。

如理论上住院病人费用信息来源于以下几个部分：（1）系统自动收取的床位费用；（2）医生录入的长期和临时医嘱；（3）护士录入的医嘱和材料费用；（4）医技辅助部门录入的检查、手术、会诊费用。这些费用信息涉及不同的分别记录的数据表，表与表之间有很强的逻辑关系，不合法的数据流动涉及到医疗安全，根据此情况我们设计了如下表数据之间的逻辑关系并用程序代码实现：

（1）长期医嘱和临时医嘱与药房发药表对照，看有没有医嘱没有发药，有没有发了药找不到对应医嘱信息，提供药剂科使用。

（2）临时医嘱与医技报告表对照，看有没有检查、检验医嘱没有报告，有没有有报告而找不到对应医嘱信息，提供相应检查科室使用。

（3）临时医嘱与手术表对照，看有没有手术医嘱而没有手术，或手术没有进行登记等，提供给医务科使用。

（4）药品发药表与费用信息、药品明细帐对照，看有没有漏收费或多收费情况发生。

（5） 医嘱和费用信息对照，看有没有漏收费或多收费情况发生。

（6） 医生医嘱和护士执行情况表对照，看有没有漏执行情况，提供给护理部使用。

以上只是部分示例，如果程序正常稳定，操作人员正常操作，这些问题出现的很少，但在实际过程中不可避免，如病人急症发作时的口头医嘱，再如系统正式运行后，在维护过程中，对表中的记录做了不合理的删除或修改等处理都会引发错误。根据业务流程和数据流转的关系，专门设计了一个稽查程序，找出非法流动的数据，改进程序和约束操作人员的非正常操作，在实际使用中取得了良好的效果。

2.4 通过制定规章制度来约束用户的行为

加深各级人员，包括计算机工程人员，尤其是管理人员，对系统构造、功能、子系统间关系的整体认识与理解，加强适应医院信息系统环境下新的工作模式，改变旧的习惯性工作与管理意识，使各级人员的工作与管理行为与医院信息系统得到充分的结合，使全体人员理解到个人的行为与数据质量密切相关，保证数据的质量是每一个人的工作职责。

建立适合医院信息系统的工作流程与工作制度。保证各个部门的工作在医院信息系统这个环境下，按照既定工作流程和工作规范进行。建立信息标准化与标准化管理，规范医院的管理流程，医院管理人员才能利用数据获取各类有用的管理信息，作出适应于医院实际情况的管理决策。加强对计算机中心和开发公司数据维护人员的管理，建立科学的数据维护制度与维护规范，严格数据维护行为，杜绝为解决应用问题通过后台进行数据的增、添、改操作，确保数据的安全性。

需要制定的规范大体分为四大类，即：工作流程、操作规范、岗位职责、工作制度四大类，使正确的人在正确的时间按照正确流程上完成自己应该完成的正确的事，在最外围杜绝数据的语义不合理性。

3. 数据完整性约束的实施策略

3.1 数据完整性约束的实现是一个综合的过程

进行数据完整性约束最原始的方法是将每一个完整性约束编入要访问数据库的每个应用程序中，这样需要在每个相关的数据库应用程序中，开发者都要对相同的完整性规则进行编码、测试和排错。若某一规则变动，所有相关的应用程序都得改动。应用程序须向 DBServer 请求大量的数据，以执行该应用程序内部的数据完整性规则检查，这将阻塞网络。因而这是一种耗时、低性能的方法。因此，数据完整性约束是一个综合的过程，需要在数据库、开发工具、应用程序和用户等每个环节进行约束与控制。

在前端可进行域完整性约束，并能定制业务规则，在后端几乎可以实现所有的完整性约束。Code 指出关系完整性约束条件必须用数据子语言定义并存储在数据字典中，而不是在应用程序中，当完整性约束改变时，只要修改数据字典即可，完整性的改变一般不会在逻辑上影响应用程序的活动。从这个意义上说，前端应用程序应当包含完整性检查和实施事务规则，但不能取代在数据库里的完整性约束定义。数据库中的完整性约束是整个完整性框架的核心。

3.2. 完整性约束与系统性能的关系

要考虑数据完整性控制和系统性能的关系。要有一个度，控制的太完善，每一个控制的执行都

要搜索其对应的基本表,这样如果过多地使用控制,明显的会影响后台数据库的性能。当应用系统的数据相对稳定、数据更新较少时,应尽量少用约束策略。如事务的使用就是如此,一个事务所包括的操作越多,在多用户环境下,同时存取该数据库的用户就越多;存取同一数据库、同时处于活动状态的事务越多,当该事务提交对数据库的修改时,与其他事务发生存取冲突的可能性就越大;但另一方面,事务越长,越能减小网上的数据传输量,从而提高系统的运行效率。

3. 3. 系统完整性约束机制的实现策略:

在我们对医院信息系统进行完整性约束的过程中,根据实施数据完整性的方法的各个特点和实际工作中遇到的问题,采用了如下策略:

(1) 用应用程序约束窗口的输入数据

对于通过窗口操作输入的数据,一般采用客户端应用程序来确保数据的完整性。这样,一方面可把非法数据在提交到数据库之前就拒之门外,另一方面可使用户及时得到操作反馈信息,做出正确的选择。

(2) 用SQL Server提供的表定义约束来规范数据

对于数据一致性的维护,宜用服务器端数据库的表定义的约束来实现。这样可降低客户端应用程序的开发量,提高应用系统的运行效率、可维护性及可靠性。数据类型尽量采用用户定义的数据类型,它比标准的数据类型更准确地限定了数据输入的范围。统一数据库中字段名称,并加强数据字典注释工作,以便新的技术人员能够准确把握数据的约束方式。

(3) 严格控制其他数据库转入的数据

对于通过其它途径(如其它数据库)转入的数据,一般需用服务器端数据库管理系统来实现数据的完整性。在转入时做好清洗过滤工作。

(4) 构建主动数据库系统

主动数据库系统是指由数据库服务器监控数据库状态和操作,在状态改变时,或在对数据库操作时,根据不同的条件实时地触发响应,这些响应可以自动的控制数据库的状态,可以根据权限和业务规则允许或禁止操作的进行,可以根据状态的变化产生用户维护应用系统需要的信息等。这一系列的过程遵循“事件-条件-动作”规则,存储在数据库中,与用户和应用无关,可以被应用程序共享,由服务器进行优化。

(5) 灵活应用触发器进行数据的完整性约束。

触发器的应用范围很广,它不仅能快速、方便地实现数据的一致性、完整性和实时性控制功能;

另外,通过设计相应的代码还能实现对数据进行错误的检查和纠正的功能以及实现实时计算和统计等功能。

触发器可以基于数据库的值控制用户操作数据库的某种权利。如基于时间限制用户的操作,基于数据库中的数据限制用户的操作等。也可以实现复杂的非标准的数据库相关完整性规则,触发器可产生比定义表时定义的约束更为复杂的约束。与约束不同,触发器可以引用记录、属性等数据库对象。可对数据库中相关的表进行级联更新。例如,在HIS的人员表中删除一行,触发器可导致相应删除在其他数据库表中与之相对应的行。

(6) 数据安全性和审计

数据库的安全性似乎与完整性约束无关,但在实际的操作过程中,把数据安全作为约束的一部分加入到整个约束控制策略中,控制数据增加、修改和删除的合法性,能够取得意想不到的效果。跟踪用户对数据库的操作,审计用户操作数据库的语句,把用户对数据库的更新写入审计表,并把这个政策在一定范围内公开,是一个很好的策略。

(7) 稽查程序是完整性约束的保证

完整性约束规则是系统的业务系统规则的体现,直接采用业务规则对数据进行完整性的核对工作,是发现系统数据不一致现象的良好方法。发现数据不一致后要积极查找原因,属于程序问题的,修改程序,也可用完整性约束的其他实现方法来进行替代;属于用户操作问题或者系统流程本身所固有的缺陷,要及时修改规章制度。稽查程序属于完整性约束环节的事后补救措施,是发现系统设计和代码实现问题的有力的辅助工具。

(8) 用户是完整性约束的重要环节

在数据完整性约束体系中,最容易遗漏的是用户的因素。用户是系统的使用者,作为系统的一部分存在。应用程序和数据库后台控制在理论上能够完善的做到完整性控制,在现实中,也要有一套完善可行的规章制度来进行约束。如处方中药品的开立,由于人为的原因,将药品用量录入错误或药品规格选择错误,或者选择了一个错误的用法,这些语义上的错误只能通过规章制度来控制。

4 结论

医院信息系统是一个以数据库为核心,以网络为技术支撑环境,具有一定规模的计算机化的系统,在系统内部按一定原则划分若干子系统,各子系统、分系统之间互有接口,有些数据及结构会较多地受到人工干预和社会因素的影响。医院对信息系统的依赖越多,数据错误或不准确给医院信息系统带来的影响越大。在医院信息系统建立前和实施过程中,进行数据完整性约束有现实意义。

数据完整性是关系数据库的主要性质之一，完整性约束和关系数据结构、关系操作集合在一起构成了关系模型的三大组成部分，完整性保护是目前数据库主要研究课题之一。实体完整性规则、参照完整性规则以及用户定义完整性规则是三类关系模型的数据完整性约束规则，对维护关系数据库中数据的完整性有着重要的意义。

在系统建设过程中，用SQL Server和Delphi提供的完整性约束和辅助约束机制，较好地实现了数据库端的完整性约束机制，为应用程序的开发提供了良好的环境，能够很好地实现用户对完整性的要求。数据完整性约束机制的实现，是一个综合的过程，必须全面考虑，充分利用各种完整性约束的优点并将它们结合起来使用。在该机制的实现过程中，不能够忽略对终端用户的约束和控制。

我们利用SQL Server和Delphi提供的完整性约束机制，并结合医院信息系统固有的业务特点提出的辅助约束机制，作用于医院信息系统的筹建和系统实施过程中，很好的解决了系统中存在的条块分割的现象，消灭了信息孤岛，加强了整个医院中多个业务系统如HIS、RIS、LIS、财务管理、统计病案管理、办公自动化系统、仓库物资管理、图书管理等系统中数据的共享使用。解决了系统中数据结构设计的问题，基本消除了系统中合理的冗余数据的不一致性，合理一致的冗余数据提高了查询速度，使整个系统性能有了明显的提高。对来源的不一致的数据进行约束，使用户乐于使用系统取得相关数据。通过详细的完整性控制框架说明，使参与系统开发、实施和应用人员明白数据的来龙去脉，大大减少了因系统修改引起的数据的不一致和数据错误。

在我们的系统建设过程中，数据完整性约束框架的应用是成功的，对其他医院建立类似系统有指导意义。提出的用逻辑程序来进行数据完整性约束应该是未来的发展方向。

参考文献

- [1] 郭云怡, 张永林. 管理信息系统的数据完整性约束及其实现方法. 计算机应用研究, 2001, 6: 55-57.
- [2] 李孜等. 医院信息系统评价复杂性研究. 中国医院管理, 2005, 25 (4): 45-47.
- [3] 程红梅. 关系模型的数据库结构与完整性约束. 电脑知识与技术, 2003, 26: 54-58.
- [4] 金可音, 陈松乔. 关系数据库的完整性保护模型. 电脑与信息技术, 1997, 3: 10-13.
- [5] 壮志剑. 数据库数据完整性保证策略. 微机发展, 2002, 5: 28-30.
- [6] 黄文平, 高茂庭. Microsoft SQL Server数据完整性实现策略. 计算机时代, 2002, 12: 4-6.
- [7] 刘敏贤, 汤娟. SQL Server 数据库应用系统中数据完整性的设计与实施. 微机发展, 2002, 4: 50-53.
- [8] 许中卫, 周勇. 在Delphi中用事务保持数据库一致性和完整性. 微机发展, 2000, 6: 36-38.
- [9] 吴生武. 触发器在数据库完整性约束方面的应用. 计算机与现代化, 2005, 5: 67-69.
- [10] 唐月华, 付金海, 肖翠娥. HIS中表间数据不一致性分析. 医学信息, 2003, 7: 359-361.